DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COURSE FILE

NAME OF THE STAFF : Mr. RAGHAVENDRACHAR.S

SUBJECT CODE/NAME : 17CS73 / MACHINE LEARNING

YEAR/SEMESTER/SEC : IV/VII/A

ACADEMIC YEAR : (Sept 2020 – Jan 2021)(ODD)

BRANCH : CSE

S.Ragh

COURSE IN-CHARGE

HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Vision of the Institute

To impart quality technical education with ethical values, employable skills and research to achieve excellence

Mission of the Institute

- To attract and retain highly qualified, experienced & committed faculty.
- To create relevant infrastructure.
- Network with industry & premier institutions to encourage emergence of new ideas by providing research & development facilities to strive for academic excellence.
- To inculcate the professional & ethical values among young students with employable skills & knowledge acquired to transform the society.

Vision of the Department

To create competent professionals in Computer Science and Engineering with adequate skills to drive the IT industry

Mission of the Department

- Impart sound technical knowledge and quest for continuous learning.
- To equip students to furnish Computer Applications for the society through experiential learning and research with professional ethics.
- Encourage team work through inter-disciplinary project and evolve as leaders with social concerns.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru - 560 109

K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Program Educational Objectives

PEO1: Excel in professional career by acquiring knowledge in cutting edge technology and contribute to the society as an excellent employee or as an entrepreneur in the field of Computer Science & Engineering.

PEO2: Continuously enhance their knowledge on par with the development in IT industry and pursue higher studies in Computer Science & Engineering.

PEO3: Exhibit professionalism, cultural awareness, team work, ethics, and effective communication skills with their knowledge in solving social and environmental problems by applying computer technology.

Program Specific Outcomes (PSO)

PSO1: Ability to understand, analyze problems and implement solutions in programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.

PSO2: Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru - 560 109

K.S. INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Program Outcomes

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109

K. S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course: Machine Learning		
Type: Core	**Course Code: 17CS73**	**Academic Year : 2020 - 2021**
Faculty: Mr. Raghavendrachar S		**Sem / Section : VII / A**

No of Hours

Theory (Lecture Class)	Practical/Field Work/Allied Activities	Total/Week	Total teaching hours
4	0	4	50

Marks

Internal Assessment	Examination	Total	Credits
40	60	100	4

Aim/Objectives of the Course

Machine Learning is an important subject which provides exposure to new technology which is demanded by the industry. This course lays down the following objectives –

1. Recall the problems for machine learning and select the either supervised, unsupervised or reinforcement learning.
2. Understand theory of probability and statistics related to machine learning.
3. Illustrate concept learning, ANN, Bayes classifier, K nearest neighbour.

Course Learning Outcomes

After completing the course, the students will be able to

CO#	COURSE OUTCOMES	K - LEVEL
17CS73.1	**Make use of** concept learning to implement find s and candidate elimination algorithm.	Applying (K3)
17CS73.2	**Construct** decision tree using appropriate algorithms.	Applying (K3)
17CS73.3	**Choose** appropriate algorithms to implement artificial neural networks.	Applying (K3)
17CS73.4	**Identify** Bayes Classifier for solving problems.	Applying (K3)
17CS73.5	**Determine** Instance based and Reinforcement Learning Techniques.	Applying (K3)

Syllabus Content

Module – 1 **Introduction:** Well posed learning problems, Designing a Learning system, Perspective and Issues in Machine Learning. **Concept Learning:** Concept learning task, Concept learning as search, Find-S algorithm, Version space, Candidate Elimination algorithm, Inductive Bias. **LO:** At the end of this session the student will be able to 1. Define machine learning and discuss various issues in machine learning 2. List and explain the different design issues and approaches in machine learning 3. Illustrate with suitable example of candidate elimination algorithm	**CO1** 10 hrs PO1- 3 PO2- 2 PO3 - 2 PO5-2 PO10 - 1 PO12 - 1

4. Describe the FIND-S algorithm and apply it on the given dataset.	PSO1- 3 PSO2- 2
Module – 2 **Decision Tree Learning:** Decision tree representation, Appropriate problems for decision tree learning, Basic decision tree learning algorithm, hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning. **LO**: At the end of this session the student will be able to 1. What is decision tree learning? Illustrate ID3 algorithm for simplified version with Boolean valued function. 2. Explain the concept of entropy and information gain and apply the concepts to solve the problems on the given dataset.	**CO2** 10 hrs. PO1- 3 PO2- 2 PO3 - 2 PO5-2 PO10 - 1 PO12 - 1 PSO1- 3 PSO2- 2
Module – 3 **Artificial Neural Networks:** Introduction, Neural Network representation, Appropriate problems, Perceptron, Back propagation algorithm. **LO**: At the end of this session the student will be able to 1. What are artificial neural networks (ANN)? Discuss different characteristics that are appropriate for ANN problems. 2. What is perception? Discuss how a single perception can be used to represent the Boolean functions such as AND and OR. 3. Explain the differentiable sigmoid threshold unit. 4. Explain BackPropagation algorithm. Why it is not likely to be trapped in local minima	**CO3** 8 hrs PO1- 3 PO2- 2 PO3 - 2 PO5-2 PO10 - 1 PO12 - 1 PSO1- 3 PSO2- 2
Module – 4 **Bayesian Learning:** Introduction, Bayes theorem, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting probabilities, MDL principle, Naive Bayes classifier, Bayesian belief networks, EM algorithm. **LO**: At the end of this session the student will be able to 1. Explain Naïve bayes classifier 2. Explain mistake bound model for learning and apply it to FIND-S algorithm 3. What is Bayesian learning? Discuss the features of Bayesian learning method	**CO4** 10hrs PO1- 3 PO2- 2 PO3 - 2 PO5-2 PO10 - 1 PO12 - 1 PSO1- 3 PSO2- 2
Module – 5 **Evaluating Hypothesis:** Motivation, Estimating hypothesis accuracy, Basics of sampling theorem, General approach for deriving confidence intervals, Difference in error of two hypotheses, Comparing learning algorithms. **Instance Based Learning:** Introduction, k-nearest neighbor learning, locally weighted regression, radial basis function, cased-based reasoning, **Reinforcement Learning:** Introduction, Learning Task, Q Learning **LO**: At the end of this session the student will be able to 1. Describe KNN algorithm for continuous valued target functions. Discuss one	**CO5** 12 hrs PO1- 3 PO2- 2 PO3 - 2 PO5-2 PO10 - 1 PO12 - 1

major drawback of this algorithm and how it can be corrected. 2. What is reinforcement learning? 3. What is Q function and Q learning?	PSO1- 3 PSO2- 2

Text Books

1. Tom M. Mitchell, **Machine Learning,** India Edition 2013, McGraw Hill Education.

Reference Books (specify minimum two foreign authors text books)

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, **The Elements of Statistical Learning**, 2nd edition, Springer series in statistics.
2. Ethem Alpaydin, **Introduction to machine learning**, second edition, MIT press.

Useful Websites

1. https://nptel.ac.in/courses/106105152/
2. https://www.coursera.org/learn/machine-learning
3. https://www.slideshare.net/ColleenFarrelly/machine-learning-by-analogy-59094152

Useful Journals

- IEEE Transactions on Evolutionary Computation
- https://www.journals.elsevier.com/knowledge-based-systems
- https://www.journals.elsevier.com/neural-networks
- http://jmlr.csail.mit.edu/

Teaching and Learning Methods

1. Lecture class: 50 hrs
2. Revision: 2hrs

Assessment

Type of test/examination: Written examination

Continuous Internal Evaluation(CIE) : 40 marks (Average of three tests will be considered)

Semester End Exam (SEE): 100 marks (students have to answer 5 full questions)

Test duration: 1 :30 hrs

Examination duration: 3 hrs

CO to PO Mapping

PO1: Science and engineering Knowledge **PO2:** Problem Analysis **PO3:** Design & Development **PO4:**Investigations of Complex Problems **PO5:** Modern Tool Usage **PO6:** Engineer & Society	**PO7:**Environment and Society **PO8:**Ethics **PO9:**Individual & Team Work **PO10:** Communication **PO11:**Project Mngmt & Finance **PO12:**Life long Learning

PSO1: Ability to understand, analyze problems and implement solutions in programming languages, as well to apply concepts in core areas of Computer Science in association with professional bodies and clubs.

PSO2: Ability to use computational skills and apply software knowledge to develop effective solutions and data to address real world challenges.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
17CS73.1	3	2	2	-	2	-	-	-	-	1	-	1
17CS73.2	3	2	2	-	2	-	-	-	-	1	-	1
17CS73.3	3	2	2	-	2	-	-	-	-	1	-	1
17CS73.4	3	2	2	-	2	-	-	-	-	1	-	1
17CS73.5	3	2	2	-	2	-	-	-	-	1	-	1
17CS73	3	2	2	-	2	-	-	-	-	1	-	1

CO	PSO1	PSO2
17CS73.1	3	2
17CS73.2	3	2
17CS73.3	3	3
17CS73.4	3	2
17CS73.5	3	2
17CS73	3	2

3	Substantial (High) Correlation
2	Moderate (Medium) Correlation
1	Slight (Low) Correlation
-	No correlation.

Course In charge **Module Coordinator** **HOD**

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K. S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CALENDAR OF EVENTS: ODD SEMESTER (2020-2021)

SESSION: SEP 2020 - JAN 2021

Week No.	Month	Day						Days	Activities	Department Activities
		Mon	Tue	Wed	Thu	Fri	Sat			
1	SEP		1*	2	3	4	5	5	1*-Commencement of Higher Semester	
2	SEP	7	8	9	10	11	12	6		
3	SEP	14	15	16		18	19	5	17- Mahalaya	
4	SEP	21	22	23	24	25	26TA	6		
5	SEP / OCT	28 T1	29 T1	30 T1	1		3	5	2- Mahatma Gandhi Jayanthi	
6	OCT	5	6BV	7ASD	8	9	10	6	5-10 First Feed Back	
7	OCT	12	13	14	15	16		5		
8	OCT	19	20	21	22	23	24	6	24 - Monday Time Table	
9	OCT		27	28	29			3	26- Vijayadashami 30- Eid-Milad 31- Maharishi Valmiki Jayanti	Project Zeroth Review Presentation
10	NOV	2	3	4	5	6	7 TA	6	7 - Wednesday Time Table	
11	NOV	9	10	11	12	13		5		13-11-2020 Webinar on An Insight into Web Application Development
12	NOV		17 T2	18 T2	19 T2	20	21	5	16 - Balipadyami Deepavalli 18 - 21 Second Feed Back 21 - Friday Time Table	
13	NOV	23	24BV	25ASD	26	27		5		
14	NOV /DEC	30	1	2		4	5	5	3- Kanakadasa Jayanti 5 - Monday Time Table	Project Zeroth Phase Re-Presentation
15	DEC	7	8	9	10	11		5		
16	DEC	14	15	16	17	18	19	6	19- Monday Time	
17	DEC	21	22	23	24			4	25-Christmas	22-12-2020 & 23-12-2020 34th CSI Karnataka State Student Convention
18	DEC/ JAN	28	29	30	31	1 TA	2	6	2Thursday Time Table	Project Phase - 1 Review Presentation
19	JAN	4 DH	5 DH	6 DH	7 DH	8		5		
20	JAN	11 T3	12 T3	13 T3		15	16 *	5	14- Makara sankaranthi	
Total No of Working Days : 106										

Total Number of working days (Excluding holidays and Tests)=87

H	Holiday
BV	Blue Book
T1,T2, T3	Tests 1,2, 3
ASD	Attendance &
DH	Declared Holiday
LT	Lab Test
TA	Test attendance

Monday	16
Tuesday	16
Wednesday	16
Thursday	16
Friday	17
Saturday	6
Total	**87**

24/8/2020

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K. S INSTITUTE OF TECHNOLOGY, BENGALURU-560109

TENTATIVE CALENDAR OF EVENTS: ODD SEMESTER (2020-2021)

SESSION: SEP 2020 - JAN 2021

Week No.	Month	Day						Days	Activities
		Mon	Tue	Wed	Thu	Fri	Sat		
1	SEP		1*	2	3	4	5	5	1*-Commencement of Higher Semester
2	SEP	7	8	9	10	11	12	6	
3	SEP	14	15	16	17 H	18	19	5	17- Mahalaya Amavasya
4	SEP	21	22	23	24	25	26TA	6	
5	SEP / OCT	28 T1	29 T1	30 T1	1	2 H	3	5	2- Mahatma Gandhi Jayanthi
6	OCT	5	6BV	7ASD	8	9	10	6	5-10 First Feed Back
7	OCT	12	13	14	15	16	17 DH	5	
8	OCT	19	20	21	22	23	24	6	24 - Monday Time Table
9	OCT	26 H	27	28	29	30 H	31 H	3	26- Vijayadashami 30- Eid-Milad 31- Maharishi Valmiki Jayanti
10	NOV	2	3	4	5	6	7 TA	6	7 - Wednesday Time Table
11	NOV	9	10	11	12	13	14 DH	5	
12	NOV	16 H	17 T2	18 T2	19 T2	20	21	5	16 - Balipadyami Deepavalli 18 - 21 Second Feed Back 21 - Friday Time Table
13	NOV	23	24BV	25ASD	26	27	28 DH	5	
14	NOV /DEC	30	1	2	3 H	4	5	5	3- Kanakadasa Jayanti 5 - Monday Time Table
15	DEC	7	8	9	10	11	12 DH	5	
16	DEC	14	15	16	17	18	19	6	19- Monday Time Table
17	DEC	21	22	23	24	25H	26DH	4	25-Christmas
18	DEC/ JAN	28	29	30	31	1 TA	2	6	2Thursday Time Table
19	JAN	4 LT	5 LT	6 LT	7 LT	8	9DH	5	
20	JAN	11 T3	12 T3	13 T3	14H	15	16 *	5	14- Makara sankaranthi 16* -Last Working Day
Total No of Working Days : 106									

Total Number of working days (Excluding holidays and Tests)=87

H	Holiday
BV	Blue Book Verification
T1, T2, T3	Tests 1, 2, 3
ASD	Attendance & Sessional Display
DH	Declared Holiday
LT	Lab Test
TA	Test attendance

Monday	16
Tuesday	16
Wednesday	16
Thursday	16
Friday	17
Saturday	6
Total	**87**

PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109.

KSIT BANGLORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

STUDENTS DETAILS

Year/Semester/Section: IV/VII/A

SL. No	UI Number	Students name	Hostel/ day scholar	Total Arrears	Gender	Mail ID	Student Phone Number	Parents Phone Number
1.	1KS17CS001	AAFREEN HUSSAIN	Day Scholar	Nil	Female	aafreenhussain1999@gmail.com	9110670119	9900269695
2.	1KS17CS002	ABHISHEK GOWDA.M.V	Day Scholar	Nil	Male	abhishek.mvg@gmail.com	6361599901	8217357488
3.	1KS17CS003	AKSHATHA RAMESH	Day Scholar	Nil	Female	akshatharamesh@yahoo.com	8884029225	9916928620
4.	1KS17CS004	AKSHITHA.B.S	Day Scholar	Nil	Female	akshithabsyadav@gmail.com	9845713215	6363320784
5.	1KS17CS005	AMOGH.R	Day Scholar	1	Male	amoghpavan5363@gmail.com	9972875490	9449426028
6.	1KS17CS006	AMOGHA MANJUNATHA.K	Day Scholar	Nil	Female	amoghamanju@gmail.com	8147601563	8050837796
7.	1KS17CS007	AMRUTHA.V.DESHPANDE	Day Scholar	1	Female	amruthadeshpande292@gmail.com	9739953140	9686422133
8.	1KS17CS008	ANOOP.P.S	Day Scholar	Nil	Male	anoop.purohith.04@gmail.com	9742853412	9845048630
9.	1KS17CS010	ANUSHA.A.G	Day Scholar	Nil	Female	anushashrike1999@gmail.com	9036169586	9880890842
10.	1KS17CS011	ANUSHREE.J	Day Scholar	Nil	Female	anushreej.15@gmail.com	7619252293	9110682135
11.	1KS17CS013	ASHISH.K.AMAR	Day Scholar	Nil	Male	ashishamar1999@gmail.com	9483952634	9480633453
12.	1KS17CS014	LAKSHMI PRASANNA.B	Day Scholar	Nil	Female	bodireddylakshmiprasanna@gmail.com	8310858767	9740981666
13.	1KS17CS016	BHAVESH BHANSALI	Day Scholar	0	Male	bbhansali18.bb@gmail.com	9341908369	9379449171
14.	1KS17CS017	CHAITRA	Day Scholar	Nil	Female	chaitrasharanayya@gmail.com	7019432308	9741660853
15.	1KS17CS018	CHANDANA.B.R	Day Scholar	Nil	Female	chandanaramesh107@gmail.com	9110867885	9902739361

SL. No	UI Number	Students name	Hostel/ day scholar	Total Arrears	Gender	Mail ID	Student Phone Number	Parents Phone Number
16.	1KS17CS019	CHENNAKESHAVA.N.T	Day Scholar	Nil	Male	chennakeshavant@gmail.com	8296886785	8123999118
17.	1KS17CS020	DARSHAN.S	Day Scholar	2	Male	darshan.s.889@gmail.com	9945499929	9845030835
18.	1KS17CS021	DEEKSHITHA.R	Day Scholar	Nil	Female	manupatideekshu@gmail.com	9845558728	9448880435
19.	1KS17CS022	DEEPIKA.S.H	Day Scholar	1	Female	deepikahande06@gmail.com	9986457979	8722297778
20.	1KS17CS023	DIVYA YASHASWI KANNEY	Day Scholar	Nil	Female	disney.kstew@gmail.com	9663691519	9010301099
21.	1KS17CS024	GANESH.G.B	Hostler	2	Male	iganeshgb@gmail.com	9449088564	9448028648
22.	1KS17CS025	GANESH MAUDGHALYA.H.G	Day Scholar	3	Male	ganeshmaudghalya@gmail.com	8792894800	9449446341
23.	1KS17CS026	GAUTHAM.C.R	Day Scholar	Nil	Male	gauthamcr82@gmail.com	6360618870	9060920050
24.	1KS17CS027	H.PRIYANKA	Day Scholar	Nil	Female	hannipriyanka@gmail.com	6364361270	9845537616
25.	1KS17CS028	HANUMESH.V.T	Day Scholar	2	Male	hanumeshvt99@gmail.com	8073665310	8722302210
26.	1KS17CS029	HARSHITHA.V	Day Scholar	Nil	Female	harshithasrirang@gmail.com	9036742864	9740285937
27.	1KS17CS030	INDRASENA KALYANAM	Day Scholar	Nil	Female	kindrasena8@gmail.com	8892186871	8867349693
28.	1KS17CS032	KARAN RAGHUNATH	Day Scholar	0	Male	karanraghunath@gmail.com	8861302491	9845106993
29.	1KS17CS033	KARTHIK.T.C	Day Scholar	Nil	Male	dontmailme1999@gmail.com	8497085624	9343307902
30.	1KS17CS034	KAVITHA.S	Day Scholar	0	Female	pavan.19346s@gmail.com	7019557683	9980019346
31.	1KS17CS035	KEERTHI.N	Day Scholar	Nil	Female	keerthin321@gmail.com	8884129614	8105142466
32.	1KS17CS036	KRITHIKA JAGANNATH	Day Scholar	Nil	Female	krithika1089@gmail.com	8197270285	9845265432
33.	1KS17CS037	LAVANYA.V	Day Scholar	Nil	Female	lavanya08.vasanth@gmail.com	8123406158	9448383988
34.	1KS17CS038	LOKESH.B.M	Day Scholar	Nil	Male	lokeshbm2021@gmail.com	7019152937	9448490890

SL. No	UI Number	Students name	Hostel/ day scholar	Total Arrears	Gender	Mail ID	Student Phone Number	Parents Phone Number
35.	1KS17CS040	MANJUNATH.A	Day Scholar	Nil	Male	manjunaidu888@gmail.com	9880935251	9880237731
36.	1KS17CS041	MEGHANA.C.V	Day Scholar	Nil	Female	meghana2832@gmail.com	8546840404	9448295109
37.	1KS17CS042	MEGHANA.G	Day Scholar	Nil	Female	meghanagururaj99@gmail.com	9148313574	9379574129
38.	1KS17CS043	MEGHANA.G.R	Day Scholar	Nil	Female	reddymeghana9931@gmail.com	9620776358	7259958737
39.	1KS17CS044	MOUNIKA.M.K.L	Day Scholar	Nil	Female	mounika.marrey@gmail.com	9741902130	6375076447
40.	1KS17CS045	NEHA.K	Day Scholar	Nil	Female	nehakunapalli623@gmail.com	9902310379	9483965499
41.	1KS17CS046	NIKHIL SUBRAMANYA.K	Day Scholar	Nil	Male	nikhilsubramanya@yahoo.com	8073673068	9740485889
42.	1KS17CS047	NIKITHA KATARI	Day Scholar	Nil	Female	nikita.katari@gmail.com	6361528821	9980592779
43.	1KS17CS048	NISCHITHA.C	Hostler	Nil	Female	nishvnkma@gmail.com	8152014334	9945983663
44.	1KS17CS049	NITISH KUMAR GUPTA	Hostler	Nil	Male	kkumarnitish61@gmail.com	7352374098	9934666725
45.	1KS17CS050	NYDILE.G.R	Hostler	Nil	Female	grnydilegowda123@gmail.com	9110842902	9449414344
46.	1KS17CS051	P.KISHORE	Day Scholar	Nil	Male	kishorep.shrivatsa@gmail.com	9845611328	9900566370
47.	1KS17CS102	SHRIRAKSHA.S.KANAGO	Day Scholar	Nil	Female	shriraksha2000@gmail.com	9980190128	8073330382
48.	1KS18CS401	KRUTHIKA.B.M	Day Scholar	8	Female	kruthikruthikabm@gmail.com	7353229125	9900320279
49.	1KS16CS042	MEGHANA.H.S	Day Scholar	2	Female	meghana.hs12@gmail.com	8971711921	9845259148
50.	1KS15CS050	LAXMI K V	Day Scholar	0	Female	lakshmikanda1996@gmail.com	9663591316	9448064857
51.	1KS16CS090	SHASHANK KAVUR	Day Scholar	7	Male	shashankavoor05@gmailcom	9008198140	8050272705

Signature of the class Coordinator

Signature of the HOD

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIVIDUAL ONLINE TIME TABLE FOR THE YEAR 2020-21 (ODD SEMESTER)

W.E.F: 01-09-2020

NAME OF THE FACULTY: Mr. RAGHAVENDRACHAR.S **DESIGNATION: ASST. PROF.**

PERIOD	1	2		3	4		5	6
TIME DAY	9:00 AM-10:00 AM	10:00 AM-11.00 AM	11:00 AM-11.30 AM	11:30 AM-12:30 PM	12:30 PM-1.30 PM	1:30 PM-02:00 PM	02:00 PM - 03:00 PM	03:00 PM-04:00 PM
MON			TEA BREAK	ML(A)		LUNCH BREAK	ML LAB(A1,A2&A3)	
TUE							ML LAB (B1,B2&B3)	
WED	ML(A)						PROJECT PHASE 1 + SEMINAR	
THUR								
FRI		ML(A)					PROJECT PHASE 1 + SEMINAR	
SAT				ML(A)				

TIME TABLE COORDINATOR

HOD

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VII SEMESTER ONLINE CLASS TIME TABLE FOR THE YEAR 2020-21 (ODD SEMESTER)(TENTATIVE)

W.E.F: 01-09-2020 **CLASS TEACHER: Mrs. Sougandhika Narayan**

SEC: 'A'

PERIOD	1	2	11:00 AM-11.30 AM	3	4			6
TIME DAY	9:00 AM-10:00 AM	10:00 AM-11.00 AM		11:30 AM-12:30 PM	12:30 PM-1.30 PM	1:30 PM-2:00 PM	2:00 PM-3:00 PM	03:00 PM- 04:00 PM
MON	INS	WT	BREAK	ML	ACA	LUNCH BREAK	ML LAB	
TUE	ACA	WT		SAN	INS		WT LAB	
WED	ML	SAN		INS	ACA		PROJECT PHASE 1 + SEMINAR	
THUR	PLACEMENT ACTIVITIES			PLACEMENT ACTIVITIES			TUTORIAL / PEDAGOGY ACTIVITIES	
FRI	SAN	ML		WT	ACA		PROJECT PHASE 1 + SEMINAR	
SAT	WT	SAN		ML	INS		TUTORIAL / PEDAGOGY ACTIVITIES	

Subject Code	Subject Name	Faculty Name
17CS71	WEB TECHNOLOGY AND ITS APPLICATION	Mr. Harshavardhan J R
17CS72	ADVANCED COMPUTER ARCHITECHTURE	Mr.Aditya Pai H
17CS73	MACHINE LEARNING	Mr. Raghavendrachar S
17CS743	INFORMATION NETWORK SECURITY **(ELECTIVE)**	Mrs. Sougandhika Narayan
17CS754	STORAGE AREA NETWORK **(ELECTIVE)**	Mr. Roopesh Kumar B N
17CSL76	MACHINE LEARNING LABORATORY	Mr. Raghavendrachar S & Dr. Rekha B Venkatapur
17CSL77	WEB TECHNOLOGY LABORATORY WITH MINI PROJECT	Mr. Aditya Pai H & Mr. Roopesh Kumar B N
17CSP78	PROJECT PHASE 1 + PROJECT WORK SEMINAR	Dr. Rekha B Venkatapur, Mr.Venkata Rao, Mrs. Vaneeta M,Mr.Raghavendrachar S , Mr. Aditya Pai H, Mrs. Sneha Karamadi

RN 25/8/2020
TIME TABLE INCHARGE

Narayan 25/8/2020
HOD
Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Tc
Bengaluru -560 10

25/8/2020
PRINCIPAL
PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109.

MACHINE LEARNING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)
SEMESTER – VII

Subject Code	17CS73	IA Marks	40
Number of Lecture Hours/Week	03	Exam Marks	60
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Module – 1	**Teaching Hours**
Introduction: Well posed learning problems, Designing a Learning system, Perspective and Issues in Machine Learning. **Concept Learning:** Concept learning task, Concept learning as search, Find-S algorithm, Version space, Candidate Elimination algorithm, Inductive Bias. **Text Book1, Sections: 1.1 – 1.3, 2.1-2.5, 2.7**	**10 Hours**
Module – 2	
Decision Tree Learning: Decision tree representation, Appropriate problems for decision tree learning, Basic decision tree learning algorithm, hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning. **Text Book1, Sections: 3.1-3.7**	**10 Hours**
Module – 3	
Artificial Neural Networks: Introduction, Neural Network representation, Appropriate problems, Perceptrons, Backpropagation algorithm. **Text book 1, Sections: 4.1 – 4.6**	**08 Hours**
Module – 4	
Bayesian Learning: Introduction, Bayes theorem, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting probabilities, MDL principle, Naive Bayes classifier, Bayesian belief networks, EM algorithm **Text book 1, Sections: 6.1 – 6.6, 6.9, 6.11, 6.12**	**10 Hours**
Module – 5	
Evaluating Hypothesis: Motivation, Estimating hypothesis accuracy, Basics of sampling theorem, General approach for deriving confidence intervals, Difference in error of two hypothesis, Comparing learning algorithms. **Instance Based Learning:** Introduction, k-nearest neighbor learning, locally weighted regression, radial basis function, cased-based reasoning, **Reinforcement Learning:** Introduction, Learning Task, Q Learning **Text book 1, Sections: 5.1-5.6, 8.1-8.5, 13.1-13.3**	**12 Hours**

Course Outcomes: After studying this course, students will be able to

- Recall the problems for machine learning. And select the either supervised, unsupersvised or reinforcement learning.
- Understand theory of probability and statistics related to machine learning
- Illustrate concept learning, ANN, Bayes classifier, k nearest neighbor, Q,

Question paper pattern:

The question paper will have ten questions.

There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Education.

Reference Books:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, h The Elements of Statistical Learning, 2nd edition, springer series in statistics.
2. Ethem Alpaydın, Introduction to machine learning, second edition, MIT press.

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

K S INSTITUTE OF TECHNOLOGY, BENGALURU
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NAME OF THE STAFF : Mr. RAGHAVENDRACHAR.S
SUBJECT CODE/NAME : 17CS73/ MACHINE LEARNING
SEMESTER/SEC/YEAR : VII / A / IV
ACADEMIC YEAR : 2020-2021 [ODD SEMESTER]

Sl. No.	Topic to be covered	Mode of Delivery	Teaching Aid	No. of Periods	Cumulative No. of Periods	Proposed Date
MODULE 1: Introduction, Concept Learning						
1	Introduction: Well posed learning problems,	L+D	Microsoft Teams	1	1	02-09-2020
2	Designing a Learning system,	L+D	Microsoft Teams	1	2	04-09-2020
3	Perspective, Issues in Machine Learning.	L+D	Microsoft Teams	1	3	05-09-2020
4	Concept Learning: Concept learning task,	L+D	Microsoft Teams	1	4	07-09-2020
5	Concept learning as search,	L+D	Microsoft Teams	1	5	09-09-2020
6	Find-S algorithm,	L+D	Microsoft Teams	1	6	11-09-2020
7	Example on Find-S algorithm	L+D	Microsoft Teams	1	7	12-09-2020
8	Version space,	L+D	Microsoft Teams	1	8	14-09-2020
9	Candidate Elimination algorithm and its examples	L+D	Microsoft Teams	1	9	16-09-2020
10	Inductive Bias	L+D	Microsoft Teams	1	10	18-09-2020

MODULE 2: Decision Tree Learning						
11	Decision tree representation	L+D	Microsoft Teams	1	11	19-09-2020
12	Appropriate problems for decision tree learning	L+D	Microsoft Teams	1	12	21-09-2020
13	Basic decision tree learning algorithm	L+D	Microsoft Teams	1	13	23-09-2020
14	Example on ID3 algorithm	L+D	Microsoft Teams	1	14	25-09-2020
15	Example on ID3 algorithm	L+D	Microsoft Teams	1	15	26-09-2020
16	**First Test**					29-09-2020
17	hypothesis space search in decision tree learning	L+D	Microsoft Teams	1	16	03-10-2020
18	hypothesis space search in decision tree learning	L+D	Microsoft Teams	1	17	05-10-2020
19	Inductive bias in decision tree learning	L+D	Microsoft Teams	1	18	07-10-2020
20	Issues in decision tree learning.	L+D	Microsoft Teams	1	19	09-10-2020
21	**Revision**	L+D	Microsoft Teams	1	20	10-10-2020
MODULE 3: Artificial Neural Networks						
22	Introduction	L+D	Microsoft Teams	1	21	12-10-2020
23	Neural Network representation	L+D	Microsoft Teams	1	22	14-10-2020
24	Neural Network representation	L+D	Microsoft Teams	1	23	16-10-2020
25	Appropriate problems	L+D	Microsoft Teams	1	24	19-10-2020
26	Appropriate problems	L+D	Microsoft	1	25	21-10-2020

			Teams			
27	Perceptron	L+D	Microsoft Teams	1	26	23-10-2020
28	Perceptron	L+D	Microsoft Teams	1	27	24-10-2020
29	Back propagation algorithm	L+D	Microsoft Teams	1	28	28-10-2020
30	Back propagation algorithm	L+D	Microsoft Teams	1	29	02-11-2020
31	Revision	L+D	Microsoft Teams	1	30	04-11-2020
MODULE 4: Bayesian Learning						
32	Introduction,	L+D	Microsoft Teams	1	31	06-11-2020
33	Bayes theorem	L+D	Microsoft Teams	1	32	07-11-2020
34	**Second Test**					10-11-2020
35	Bayes theorem	L+D	Microsoft Teams	1	33	13-11-2020
36	Concept Learning	L+D	Microsoft Teams	1	34	18-11-2020
37	ML and LS error hypothesis	L+D	Microsoft Teams	1	35	20-11-2020
38	ML for predicting probabilities	L+D	Microsoft Teams	1	36	21-11-2020
39	MDL principle	L+D	Microsoft Teams	1	37	23-11-2020
40	Naive Bayes classifier	L+D	Microsoft Teams	1	38	25-11-2020
41	Bayesian belief networks	L+D	Microsoft Teams	1	39	27-11-2020
42	EM algorithm	L+D	Microsoft Teams	1	40	30-11-2020

MODULE 5: Evaluating Hypothesis, Instance Based Learning, Reinforcement Learning						
43	Motivation	L+D	Microsoft Teams	1	41	02-12-2020
44	Estimating hypothesis accuracy	L+D	Microsoft Teams	1	42	04-12-2020
45	Basics of sampling theorem	L+D	Microsoft Teams	1	43	04-12-2020
46	General approach for deriving confidence intervals	L+D	Microsoft Teams	1	44	04-12-2020
47	Difference in error of two hypotheses	L+D	Microsoft Teams	1	45	05-12-2020
48	Comparing learning algorithms.	L+D	Microsoft Teams	1	46	05-12-2020
49	Instance Based Learning: Introduction	L+D	Microsoft Teams	1	47	05-12-2020
50	k-nearest neighbor learning	L+D	Microsoft Teams	1	48	07-12-2020
51	Learning Task,	L+D	Microsoft Teams	1	52	07-12-2020
52	Q Learning	L+D	Microsoft Teams	1	53	07-12-2020
53	**Third Test**					15-12-2020

Text Books

1. Tom M. Mitchell, **Machine Learning,** India Edition 2013, McGraw Hill Education.

Reference Books (specify minimum two foreign authors text books)

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, **The Elements of Statistical Learning**, 2nd edition, Springer series in statistics.
2. Ethem Alpaydin, **Introduction to machine learning**, second edition, MIT press.

Useful Websites

1. https://nptel.ac.in/courses/106105152/
2. https://www.coursera.org/learn/machine-learning
3. https://www.slideshare.net/ColleenFarrelly/machine-learning-by-analogy-59094152

Signature of Course in charge

Signature of Module Coordinator

Signature of H.O.D

Head of the Department
Dept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

KSIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ASSIGNMENT QUESTIONS

Academic Year	**2020-2021 [Odd Semester]**		
Batch	**2017-2021**		
Year/Semester/section	**IV/VII/A**		
Subject Code-Title	**17CS73- Machine Learning**		
Name of the Instructor	**Mr.RAGHAVENDRACHAR S**	**Dept**	**CSE**

Assignment No: 1
Date of Issue: 30/09/2020

Total marks:10
Date of Submission: 07/10/2020

<table>
<tr><th>Sl.No</th><th>Assignment Questions</th><th>K Level</th><th>CO</th><th>Marks</th></tr>
<tr><td>1.</td><td>a. Identify various applications of Machine Learning.
b. Make use of following examples to explain well posed learning problem.
i. Checkers learning problem
ii. Handwriting Recognition learning problem
iii. Robot Driving learning problem</td><td>Applying</td><td>CO1</td><td>1</td></tr>
<tr><td>2.</td><td>Determine the following with respect to checkers learning problem.
i. Choosing training experience
ii. Choosing Target function
iii. Choosing Representation of Target function
iv. Choosing function Approximation Algorithm</td><td>Applying</td><td>CO1</td><td>1</td></tr>
<tr><td>3.</td><td>a. Design Checkers learning system using four distinct program modules.
b. Identify different issues in machine learning.</td><td>Applying</td><td>CO1</td><td>1</td></tr>
<tr><td>4.</td><td>a. Design Find – S Algorithm.
b. Apply Find – S Algorithm for the given below target concept Enjoy sport.
<table>
<tr><th>Example</th><th>Sky</th><th>AirTemp</th><th>Humidity</th><th>Wind</th><th>Water</th><th>Forecast</th><th>EnjoySport</th></tr>
<tr><td>1</td><td>Sunny</td><td>Warm</td><td>Normal</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>2</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>3</td><td>Rainy</td><td>Cold</td><td>High</td><td>Strong</td><td>Warm</td><td>Change</td><td>No</td></tr>
<tr><td>4</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Cool</td><td>Change</td><td>Yes</td></tr>
</table></td><td>Applying</td><td>CO1</td><td>1</td></tr>
</table>

<table>
<tr><td>5.</td><td>

a. **Design** Candidate Elimination Algorithm.
b. **Apply** Candidate Elimination Algorithm for the given below target concept Enjoy sport.

<table>
<tr><th>Example</th><th>Sky</th><th>AirTemp</th><th>Humidity</th><th>Wind</th><th>Water</th><th>Forecast</th><th>EnjoySport</th></tr>
<tr><td>1</td><td>Sunny</td><td>Warm</td><td>Normal</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>2</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>3</td><td>Rainy</td><td>Cold</td><td>High</td><td>Strong</td><td>Warm</td><td>Change</td><td>No</td></tr>
<tr><td>4</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Cool</td><td>Change</td><td>Yes</td></tr>
</table>

</td><td>Applying</td><td>CO1</td><td>1</td></tr>
<tr><td>6.</td><td>

a. **Design** List-Then Eliminate Algorithm
b. **Determine** Inductive Bias With respect to the following
i. Biased Hypothesis Space
ii. Unbiased Learner
iii. Futility of Bias-Free Learning

</td><td>Applying</td><td>CO1</td><td>1</td></tr>
<tr><td>7.</td><td>**Identify** Appropriate problems for decision tree learning.</td><td>Applying</td><td>CO2</td><td>1</td></tr>
<tr><td>8.</td><td>**Design** ID3 Algorithm</td><td>Applying</td><td>CO2</td><td>1</td></tr>
<tr><td>9.</td><td>**Identify** the necessary measure required to select the attributes for building decision tree using ID3 Algorithm.</td><td>Applying</td><td>CO2</td><td>1</td></tr>
<tr><td>10.</td><td>

Apply ID3 Algorithm to construct the decision tree for the given target concept Play tennis.

<table>
<tr><th>Day</th><th>Outlook</th><th>Temperature</th><th>Humidity</th><th>Wind</th><th>PlayTennis</th></tr>
<tr><td>D1</td><td>Sunny</td><td>Hot</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D2</td><td>Sunny</td><td>Hot</td><td>High</td><td>Strong</td><td>No</td></tr>
<tr><td>D3</td><td>Overcast</td><td>Hot</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D4</td><td>Rain</td><td>Mild</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D5</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D6</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Strong</td><td>No</td></tr>
<tr><td>D7</td><td>Overcast</td><td>Cool</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D8</td><td>Sunny</td><td>Mild</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D9</td><td>Sunny</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D10</td><td>Rain</td><td>Mild</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D11</td><td>Sunny</td><td>Mild</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D12</td><td>Overcast</td><td>Mild</td><td>High</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D13</td><td>Overcast</td><td>Hot</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D14</td><td>Rain</td><td>Mild</td><td>High</td><td>Strong</td><td>No</td></tr>
</table>

</td><td>Applying</td><td>CO2</td><td>1</td></tr>
</table>

S.Raj

Course in charge **Module Coordinator** **HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
ASSIGNMENT I 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

Degree	: **B.E**	**Semester**	: **VII A&B**
Branch	: **Computer Science & Engineering**	**Course Code**	: **17CS73**
Course Title	: **Machine Learning**	**Max Marks**	: **10**

Q.NO.	POINTS	Marks
1 a	**Applications of Machine Learning** • Learning to recognize spoken words • Learning to drive an autonomous vehicle • Learning to classify new astronomical structures • Learning to play world-class backgammon Explanation of each application in detail	
b	**Well-Defined Learning Problem** Explanation of individual problem in detail ***A checkers learning problem:*** • Task T: playing checkers • Performance measure P: percent of games won against opponents • Training experience E: playing practice games against itself ***A handwriting recognition learning problem***: • Task T: recognizing and classifying handwritten words within images • Performance measure P: percent of words correctly classified • Training experience E: a database of handwritten words with given classifications ***A robot driving learning problem***: • Task T: driving on public four-lane highways using vision sensors • Performance measure P: average distance travelled before an error (as judged by human overseer) • Training experience E: a sequence of images and steering commands recorded while observing a human driver	1
2.	**Designing a Learning System** **Explanation of each step-in detail** (i)Choosing the Training Experience There are three attributes which impact on success or failure of the learner 1. Whether the training experience provides direct or indirect feedback regarding the choices made by the performance system. 2. The degree to which the learner controls the sequence of training examples How well it represents the distribution of examples over which the final system performance P must be measured	

(ii)Choosing the Target Function

Let *ChooseMove* be the target function and the notation is 1

$ChooseMove : B \quad M$

which indicate that this function accepts as input any board from the set of legal board states B and produces as output some move from the set of legal moves M.

An alternative target function is an *evaluation function* that assigns a *numerical score* to any given board state

Let the target function V and the notation

$V : B \quad R$

which denote that V maps any legal board state from the set B to some real value

(iii)Choosing a Representation for the Target Function

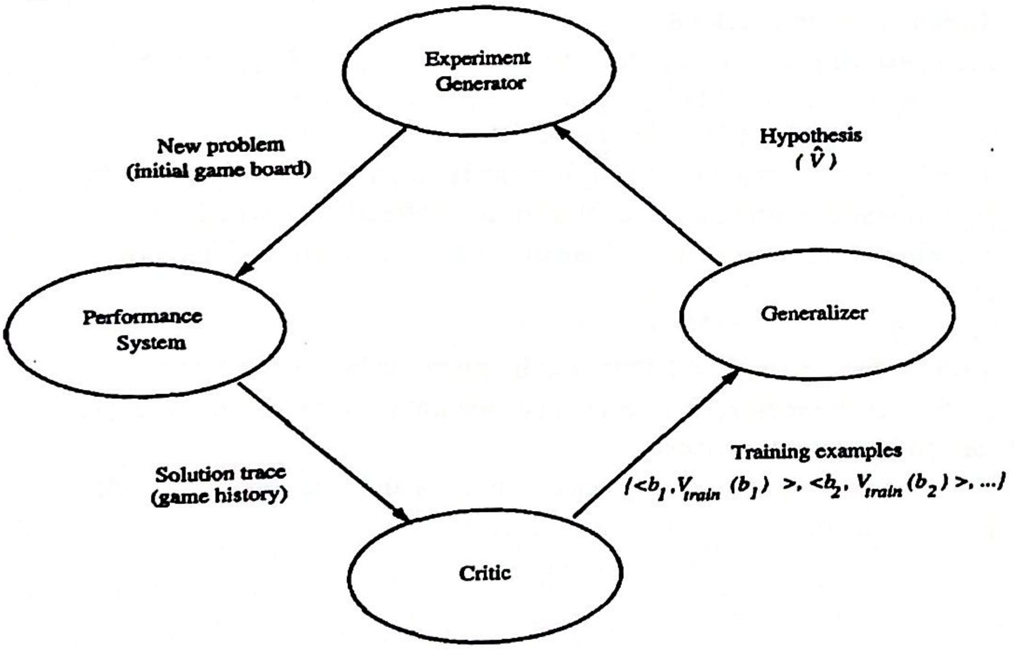
Learning program will represent as a linear function of the form

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

(iv)Choosing a Function Approximation Algorithm

1. Derive training examples from the indirect training experience available to the learner
2. Adjusts the weights w_i to best fit these training examples

3.a The final design of checkers learning system can be described by four distinct program modules that represent the central components in many learning systems



1

b Issues in Machine Learning

- What algorithms exist for learning general target functions from specific training examples??
- How much training data is sufficient?
- When and how can prior knowledge held by the learner guide the process of generalizing from examples?
- What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy alter the complexity of the

learning problem?
- What is the best way to reduce the learning task to one or more function approximation problems?
- How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

FIND-S Algorithm

4 a

```
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
For each attribute constraint a_i in h
If the constraint a_i is satisfied by x
Then do nothing
Else replace a_i in h by the next more general constraint that is satisfied by x
3. Output hypothesis h
```

1

b

x_1 = <Sunny Warm Normal Strong Warm Same>, +
Observing the first training example, it is clear that our hypothesis is too specific. In particular, none of the "Ø" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example
h_1 = <Sunny Warm Normal Strong Warm Same>
This h is still very specific; it asserts that all instances are negative except for the single positive training example
x_2 = <Sunny, Warm, High, Strong, Warm, Same>, +
The second training example forces the algorithm to further generalize h, this time substituting a "?' in place of any attribute value in h that is not satisfied by the new example
h_2 = <Sunny Warm ? Strong Warm Same>
x3 = <Rainy, Cold, High, Strong, Warm, Change>, -
Upon encountering the third training the algorithm makes no change to h.
The FIND-S algorithm simply ignores every negative example.
h3 = < Sunny Warm ? Strong Warm Same>
x4 = <Sunny Warm High Strong Cool Change>, +
The fourth example leads to a further generalization of h
h4 = < Sunny Warm ? Strong ? ? >

5 a

The CANDIDATE-ELIMINTION algorithm computes the ***version space*** containing all hypotheses from H that are consistent with an observed sequence of training examples

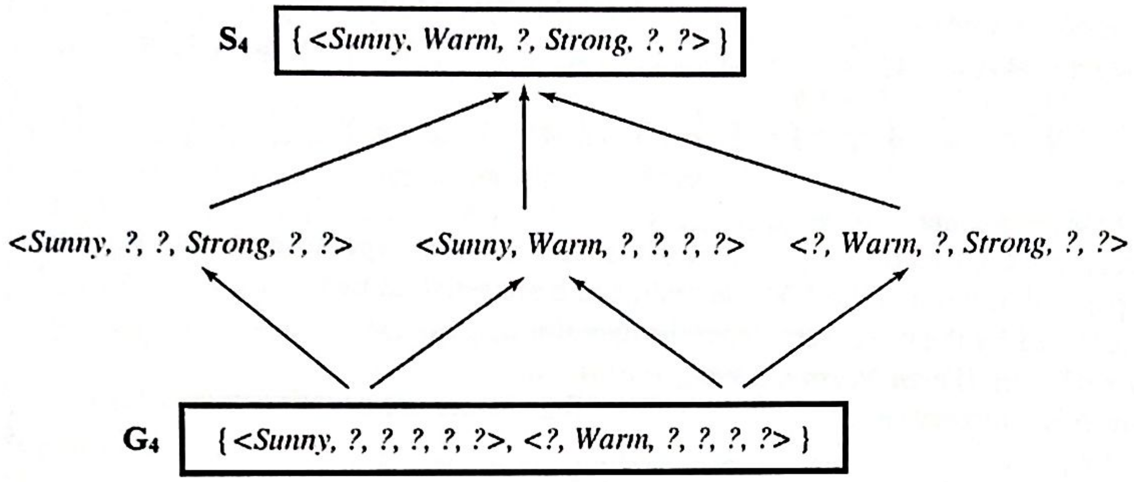
Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H 1
For each training example d, do

- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      - h is consistent with d, and some member of G is more general than h

    - Remove from S any hypothesis that is more general than another hypothesis in S
    - If d is a negative example
- Remove from S any hypothesis inconsistent with d
- For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that
        - h is consistent with d, and some member of S is more specific than h

Remove from G any hypothesis that is less general than another hypothesis in G

b



6a

LIST-THEN-ELIMINATE Algorithm

1. ***VersionSpace c a*** list containing every hypothesis in H
2. For each training example, ***(x, c(x))***

remove from ***VersionSpace*** any hypothesis ***h*** for which ***h(x) ≠ c(x)***

3. Output the list of hypotheses in ***VersionSpace*** 1

b

Explanation of Inductive Bias With respect to the following

- Biased Hypothesis Space
- Unbiased Learner
- Futility of Bias-Free Learning

7

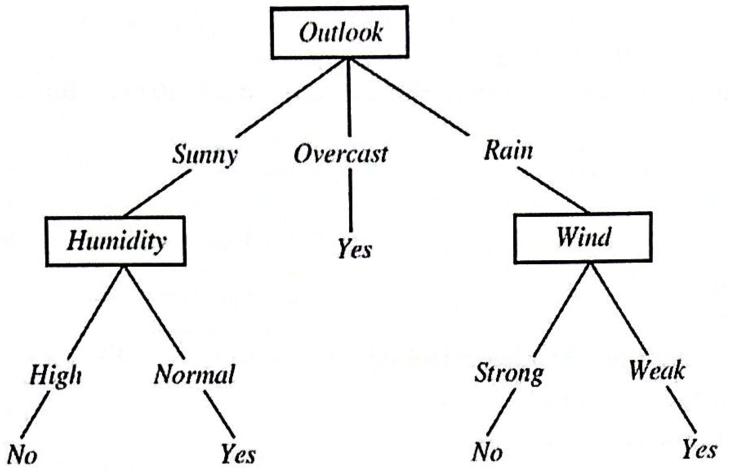
Decision tree learning is generally best suited to problems with the following characteristics:

1. ***Instances are represented by attribute-value pairs*** – Instances are described by a fixed set of attributes and their values
2. ***The target function has discrete output values*** – The decision tree assigns a Boolean classification (e.g., yes or no) to each example. Decision tree methods easily extend to learning functions with more than two possible output values. 1
3. ***Disjunctive descriptions may be required***

4.The training data may contain errors – Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.

5.The training data may contain missing attribute values – Decision tree methods can be used even when some training examples have unknown values

8	***ID3(Examples, Target_attribute, Attributes)*** Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples. • Create a Root node for the tree • If all Examples are positive, Return the single-node tree Root, with label = + • If all Examples are negative, Return the single-node tree Root, with label = - • If Attributes is empty, Return the single-node tree Root, with label = most common value of Target_attribute in Examples • Otherwise Begin • A ← the attribute from Attributes that best* classifies Examples • The decision attribute for Root ← A • For each possible value, v_i, of A, • Add a new tree branch below *Root*, corresponding to the test A = v_i • Let *Examples* $_{vi}$, be the subset of Examples that have value v_i for *A* • If *Examples* $_{vi}$, is empty • Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples • Else below this new branch add the subtree ID3(*Examples* $_{vi}$, Targe_tattribute, Attributes – {A})) • End • Return Root	1
9	The central choice in the ID3 algorithm is selecting which attribute to test at each node in the tree. • A statistical property called ***information gain*** that measures how well a given attribute separates the training examples according to their target classification. • ID3 uses ***information gain*** measure to select among the candidate attributes at each step while growing the tree. • To define information gain, we begin by defining a measure called entropy. • *Entropy measures the impurity of a collection of examples.* • Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is • $Entropy\ (S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$	1
10	The information gain values for all four attributes are • Gain(S, Outlook)=0.246 • Gain(S, Humidity)=0.151 • Gain(S, Wind)=0.048 • Gain(S, Temperature)=0.029 Final decision tree after analysing entire training dataset	1



Course in charge

Module Coordinator

HOD

KSIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ASSIGNMENT QUESTIONS

Academic Year	**2020-2021 [Odd Semester]**		
Batch	**2017-2021**		
Year/Semester/section	**III/VII/A**		
Subject Code-Title	**17CS73- MACHINE LEARNING**		
Name of the Instructor	**Mr.RAGHAVENDRACHAR S**	**Dept**	**CSE**

Assignment No: 2
Date of Issue: 10/11/2020
Total marks:10
Date of Submission: 17/11/2020

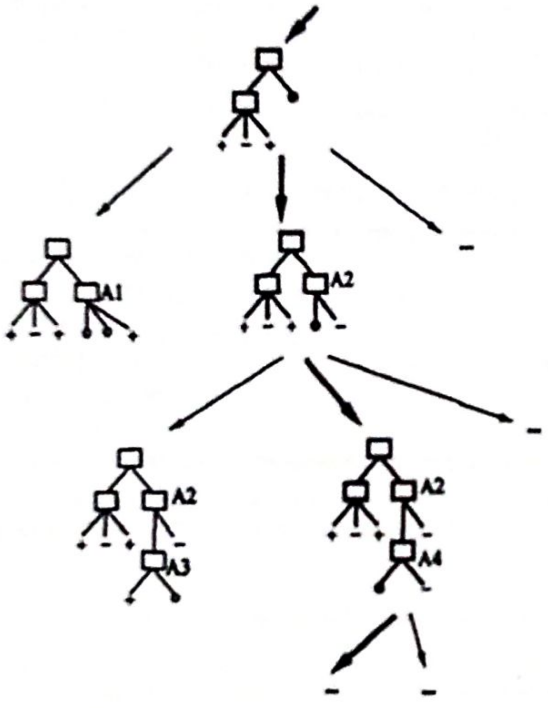
Sl.No	**Assignment Questions**	**K Level**	**CO**	**Marks**
1.	**Make use of** suitable example, explain the following **i.** Hypothesis Space Search In Decision Tree Learning **ii.** Inductive Bias In Decision Tree Learning **iii.** Issues In Decision Tree Learning	**Applying**	**CO2**	**2**
2.	**i. Determine** the application of neural network which is used for learning to steer an autonomous vehicle. **ii. Identify** the appropriate problem for Neural Network Learning.	**Applying**	**CO3**	**2**
3.	**i. Make use of** suitable diagram, explain the concept of single perceptron. **ii. Design** and derive The Gradient Descent Rule	**Applying**	**CO3**	**2**
4.	**i. Determine** why stochastic approximation is needed to Gradient descent rule? **i. Design** Back propagation algorithm.	**Applying**	**CO3**	**2**
5.	**ii. Derive** Back propagation Rule **(Case i and Case ii)**. **iii. Identify** the remarks on Back propagation Algorithm.	**Applying**	**CO3**	**2**
6.	**i. Identify** features of Bayesian learning methods **ii. Determine** Brute Force MAP Learning Algorithm	**Applying**	**CO4**	**2**

Course In charge **Module Coordinator** **HOD**

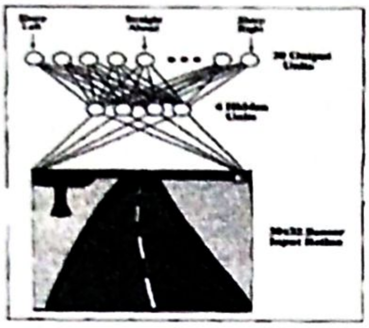
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
ASSIGNMENT II 2020 – 21 ODD SEMESTER

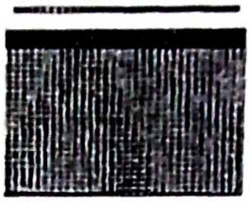
SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**VII A&B**
Branch	:	**Computer Science & Engineering**	**Course Code**	:	**17CS73**
Course Title	:	**Machine Learning**	**Max Marks**	:	**10**

Q.NO.	POINTS	Marks
1	i.  Hypothesis space search by ID3. ID3 searches through the space of possible decision trees from simplest to increasingly complex, guided by the information gain heuristic. ii. a. ID3 Search Strategy b. Approximate Inductive Bias of ID3 c. A closer approximation to the inductive bias of ID3. iii. a. Avoiding Over fitting the Data b. Incorporating Continuous-Valued Attributes c. Alternative Measures for Selecting Attributes d. Handling Training Examples with Missing Attribute Values e. Handling Attributes with Differing Costs	2

2. i.





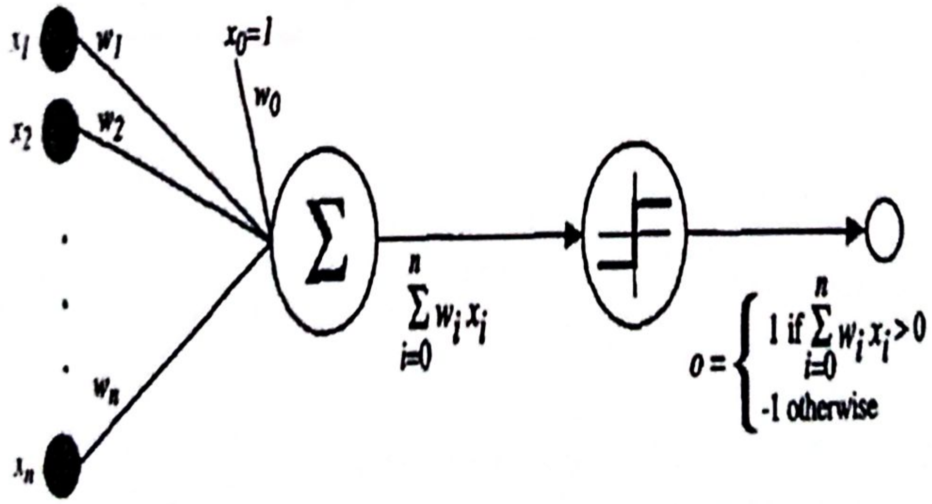
2

ii.

a. Instances are represented by many attribute value pairs
b. Target function output may be discrete valued, real valued or a vector of several real or discrete values attributes.
c. Long training times are acceptable
d. Fast evaluation of the learned target function may be required
e. The ability of humans to understand the learned target function is not important.

3

i.



ii.

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 \\
&= \frac{1}{2}\sum_{d\in D}\frac{\partial}{\partial w_i}(t_d - o_d)^2 \\
&= \frac{1}{2}\sum_{d\in D}2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) \\
&= \sum_{d\in D}(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w}\cdot\vec{x}_d) \\
\frac{\partial E}{\partial w_i} &= \sum_{d\in D}(t_d - o_d)(-x_{id})
\end{aligned}
$$

2

where x_{id} denotes the single input component x_i for training example d. We now have an equation that gives $\frac{\partial E}{\partial w_i}$ in terms of the linear unit inputs x_{id}, outputs O_d, and target values t_d associated with the training examples. Substituting Equation (4.6) into Equation (4.5) yields the weight update rule for gradient descent

$$\Delta w_i = \eta\sum_{d\in D}(t_d - o_d)\ x_{id}$$

4

i.

Gradient descent is an important general paradigm for learning. It is a strategy for searching through a large or infinite hypothesis space that can be applied whenever (1) the hypothesis space contains continuously parameterized hypotheses (e.g., the weights in a linear unit), and (2) the error can be differentiated with respect to these hypothesis parameters. The key practical difficulties in applying gradient descent are (1) converging to a local minimum can sometimes be quite slow (i.e., it can require many thousands of gradient descent steps), and (2) if there are multiple local minima in the error surface, then there is no guarantee that the procedure will find the global minimum.

ii.

```
BACKPROPAGATION(training_examples, $\eta$, $n_{in}$, $n_{out}$, $n_{hidden}$)
    Each training example is a pair of the form $(\vec{x}, \vec{t})$, where $\vec{x}$ is the vector of network input
    values, and $\vec{t}$ is the vector of target network output values.
    $\eta$ is the learning rate (e.g., .05). $n_{in}$ is the number of network inputs, $n_{hidden}$ the number of
    units in the hidden layer, and $n_{out}$ the number of output units.
    The input from unit i into unit j is denoted $x_{ji}$, and the weight from unit i to unit j is denoted
    $w_{ji}$.
• Create a feed-forward network with $n_{in}$ inputs, $n_{hidden}$ hidden units, and $n_{out}$ output units.
• Initialize all network weights to small random numbers (e.g., between -.05 and .05).
• Until the termination condition is met, Do
    • For each $(\vec{x}, \vec{t})$ in training_examples, Do
        Propagate the input forward through the network:
        1. Input the instance $\vec{x}$ to the network and compute the output $o_u$ of every unit u in
           the network.
        Propagate the errors backward through the network:
        2. For each network output unit k, calculate its error term $\delta_k$
                $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$                        (T4.3)
        3. For each hidden unit h, calculate its error term $\delta_h$
                $\delta_h \leftarrow o_h(1 - o_h) \sum_{k\in outputs} w_{kh}\delta_k$                (T4.4)
        4. Update each network weight $w_{ji}$
                $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$
           where
                $\Delta w_{ji} = \eta\,\delta_j\,x_{ji}$                             (T4.5)
```

2

5 i.

Case 1: Training Rule for Output Unit Weights. Just as w_{ji} can influence the rest of the network only through net_j, net_j can influence the network only through o_j. Therefore, we can invoke the chain rule again to write

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial net_j} \tag{4.23}$$

To begin, consider just the first term in Equation (4.23)

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$$

The derivatives $\frac{\partial}{\partial o_j}(t_k - o_k)^2$ will be zero for all output units k except when $k = j$. We therefore drop the summation over output units and simply set $k = j$.

$$\begin{aligned} \frac{\partial E_d}{\partial o_j} &= \frac{\partial}{\partial o_j} \frac{1}{2}(t_j - o_j)^2 \\ &= \frac{1}{2} 2(t_j - o_j) \frac{\partial (t_j - o_j)}{\partial o_j} \\ &= -(t_j - o_j) \end{aligned} \tag{4.24}$$

2

Next consider the second term in Equation (4.23). Since $o_j = \sigma(net_j)$, the derivative $\frac{\partial o_j}{\partial net_j}$ is just the derivative of the sigmoid function, which we have already noted is equal to $\sigma(net_j)(1 - \sigma(net_j))$. Therefore,

$$\begin{aligned} \frac{\partial o_j}{\partial net_j} &= \frac{\partial \sigma(net_j)}{\partial net_j} \\ &= o_j(1 - o_j) \end{aligned} \tag{4.25}$$

Substituting expressions (4.24) and (4.25) into (4.23), we obtain

$$\frac{\partial E_d}{\partial net_j} = -(t_j - o_j)\ o_j(1 - o_j) \tag{4.26}$$

and combining this with Equations (4.21) and (4.22), we have the stochastic gradient descent rule for output units

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = \eta\ (t_j - o_j)\ o_j(1 - o_j) x_{ji} \tag{4.27}$$

Case 2: Training Rule for Hidden Unit Weights. In the case where j is an internal, or hidden unit in the network, the derivation of the training rule for w_{ji} must take into account the indirect ways in which w_{ji} can influence the network outputs and hence E_d. For this reason, we will find it useful to refer to the set of all units immediately downstream of unit j in the network (i.e., all units whose direct inputs include the output of unit j). We denote this set of units by *Downstream*(j). Notice that net_j can influence the network outputs (and therefore E_d) only through the units in *Downstream*(j). Therefore, we can write

$$\begin{aligned} \frac{\partial E_d}{\partial net_j} &= \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j} \\ &= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial net_j} \\ &= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} \\ &= \sum_{k \in Downstream(j)} -\delta_k\ w_{kj} \frac{\partial o_j}{\partial net_j} \\ &= \sum_{k \in Downstream(j)} -\delta_k\ w_{kj}\ o_j(1 - o_j) \end{aligned} \tag{4.28}$$

	ii. a. Convergence and Local Minima b. Representational Power of Feedforward Networks c. Hypothesis Space Search and Inductive Bias d. Hidden Layer Representations e. Generalization, Overfitting, and Stopping Criterion	
6	i. • Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct. This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example. • Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided by asserting (1) a prior probability for each candidate hypothesis, and (2) a probability distribution over observed data for each possible hypothesis. • Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery"). • New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities. • Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.	2
	ii. We can design a straightforward concept learning algorithm to output the maximum a posteriori hypothesis, based on Bayes theorem, as follows: **BRUTE-FORCE MAP LEARNING algorithm** 1. For each hypothesis h in H, calculate the posterior probability $P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$ 2. Output the hypothesis h_{MAP} with the highest posterior probability $h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\, P(h \mid D)$	

Course in charge

Module Coordinator

HOD

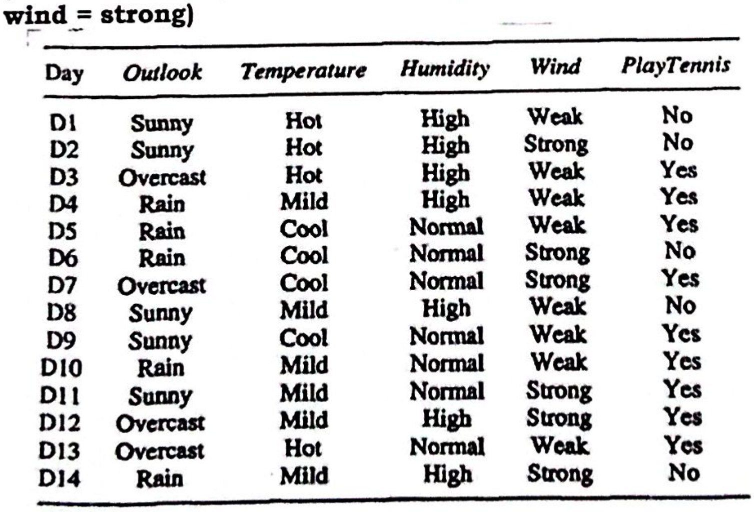
KSIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ASSIGNMENT QUESTIONS

<table>
<tr><td>Academic Year</td><td colspan="3">2020-2021 [Odd Semester]</td></tr>
<tr><td>Batch</td><td colspan="3">2017-2021</td></tr>
<tr><td>Year/Semester/section</td><td colspan="3">IV/VII/A</td></tr>
<tr><td>Subject Code-Title</td><td colspan="3">17CS73- MACHINE LEARNING</td></tr>
<tr><td>Name of the Instructor</td><td>Mr.RAGHAVENDRACHAR S</td><td>Dept</td><td>CSE</td></tr>
</table>

<table>
<tr><td colspan="5">Assignment No: 3
Date of Issue: 29/12/2020
Total marks:10
Date of Submission: 05/01/2021</td></tr>
<tr><th>Sl.No</th><th>Assignment Questions</th><th>K Level</th><th>CO</th><th>Marks</th></tr>
<tr><td>1.</td><td>

i. **Determine** Minimum Description Length Principle
ii. **Apply** naïve bayes classifier for the given training data to classify the instance **(outlook = sunny, temperature = cool, humidity = high, wind = strong)**



<table>
<tr><th>Day</th><th><i>Outlook</i></th><th><i>Temperature</i></th><th><i>Humidity</i></th><th><i>Wind</i></th><th><i>PlayTennis</i></th></tr>
<tr><td>D1</td><td>Sunny</td><td>Hot</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D2</td><td>Sunny</td><td>Hot</td><td>High</td><td>Strong</td><td>No</td></tr>
<tr><td>D3</td><td>Overcast</td><td>Hot</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D4</td><td>Rain</td><td>Mild</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D5</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D6</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Strong</td><td>No</td></tr>
<tr><td>D7</td><td>Overcast</td><td>Cool</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D8</td><td>Sunny</td><td>Mild</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D9</td><td>Sunny</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D10</td><td>Rain</td><td>Mild</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D11</td><td>Sunny</td><td>Mild</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D12</td><td>Overcast</td><td>Mild</td><td>High</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D13</td><td>Overcast</td><td>Hot</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D14</td><td>Rain</td><td>Mild</td><td>High</td><td>Strong</td><td>No</td></tr>
</table>

iii. Determine Bayesian belief network with representation

</td><td>Applying</td><td>CO4</td><td>2</td></tr>
<tr><td>2.</td><td>

Determine the following.
i. Conditional independence
ii. EM Algorithm
iii. Derivations of K means Algorithm

</td><td>Applying</td><td>CO4</td><td>2</td></tr>
</table>

3.	**Determine** the following **i.** Estimating Hypothesis Accuracy **ii.** Sample Error and True Error **iii.** Binomial Distribution	**Applying**	**CO5**	**2**
4.	**Determine** the following **i.** Confidence Intervals **ii.** Central limit Theorem **iii.** Comparing learning algorithms with paired t Tests.	**Applying**	**CO5**	**2**
5.	**Determine** the following **i.** Reinforcement learning **ii.** Learning Task **iii.** Q Learning (Q Function and Algorithm)	**Applying**	**CO5**	**2**

Course In charge **Module Co coordinator** **HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
ASSIGNMENT III 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

Degree	:	**B.E**	**Semester**	:	**VII A&B**
Branch	:	**Computer Science & Engineering**	**Course Code**	:	**17CS73**
Course Title	:	**Machine Learning**	**Max Marks**	:	**10**

Q.NO.	POINTS	Marks
1	i. The Minimum Description Length principle is motivated by interpreting the definition of h_{MAP} in the light of basic concepts from information theory. Consider again the now familiar definition of h_{MAP}. $h_{MAP} = \underset{h\in H}{\operatorname{argmax}}\ P(D\|h)P(h)$ which can be equivalently expressed in terms of maximizing the $\log_2$ $h_{MAP} = \underset{h\in H}{\operatorname{argmax}}\ \log_2 P(D\|h) + \log_2 P(h)$ or alternatively, minimizing the negative of this quantity $h_{MAP} = \underset{h\in H}{\operatorname{argmin}}\ -\log_2 P(D\|h) - \log_2 P(h)$	
	ii. Our task is to predict the target value (*yes* or *no*) of the target concept *PlayTennis* for this new instance. Instantiating Equation (6.20) to fit the current task, the target value v_{NB} is given by $v_{NB} = \underset{v_j\in\{yes,no\}}{\operatorname{argmax}}\ P(v_j)\prod_i P(a_i\|v_j)$ $= \underset{v_j\in\{yes,no\}}{\operatorname{argmax}}\ P(v_j) \quad P(Outlook = sunny\|v_j)P(Temperature = cool\|v_j)$ $P(Humidity = high\|v_j)P(Wind = strong\|v_j) \quad (6.21)$ Notice in the final expression that a_i has been instantiated using the particular attribute values of the new instance. To calculate v_{NB} we now require 10 probabilities that can be estimated from the training data. First, the probabilities of the different target values can easily be estimated based on their frequencies over the 14 training examples $P(PlayTennis = yes) = 9/14 = .64$ $P(PlayTennis = no) = 5/14 = .36$	2

Similarly, we can estimate the conditional probabilities. For example, those for $Wind = strong$ are

$$P(Wind = strong|PlayTennis = yes) = 3/9 = .33$$

$$P(Wind = strong|PlayTennis = no) = 3/5 = .60$$

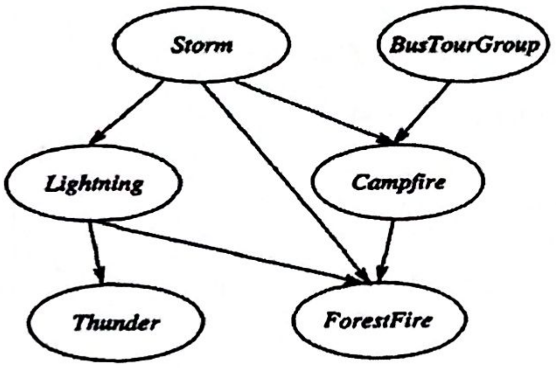
Using these probability estimates and similar estimates for the remaining attribute values, we calculate v_{NB} according to Equation (6.21) as follows (now omitting attribute names for brevity)

$$P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

$$P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) \quad = .0206$$

Thus, the naive Bayes classifier assigns the target value $PlayTennis = no$ to this new instance, based on the probability estimates learned from the training data.

iii.



	S,B	S,¬B	¬S,B	¬S,¬B
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

FIGURE
A Bayesian belief network. The network on the left represents a set of conditional independence assumptions. In particular, each node is asserted to be conditionally independent of its nondescendants, given its immediate parents. Associated with each node is a conditional probability table, which specifies the conditional distribution for the variable given its immediate parents in the graph. The conditional probability table for the *Campfire* node is shown at the right, where *Campfire* is abbreviated to *C*, *Storm* abbreviated to *S*, and *BusTourGroup* abbreviated to *B*.

i.

2.

Let us begin our discussion of Bayesian belief networks by defining precisely the notion of conditional independence. Let X, Y, and Z be three discrete-valued random variables. We say that X is *conditionally independent* of Y given Z if the probability distribution governing X is independent of the value of Y given a value for Z; that is, if

$$(\forall x_i, y_j, z_k)\ P(X = x_i|Y = y_j, Z = z_k) = P(X = x_i|Z = z_k)$$

2

where $x_i \in V(X)$, $y_j \in V(Y)$, and $z_k \in V(Z)$. We commonly write the above expression in abbreviated form as $P(X|Y, Z) = P(X|Z)$. This definition of conditional independence can be extended to sets of variables as well. We say that the set of variables $X_1 \ldots X_l$ is conditionally independent of the set of variables $Y_1 \ldots Y_m$ given the set of variables $Z_1 \ldots Z_n$ if

$$P(X_1 \ldots X_l|Y_1 \ldots Y_m, Z_1 \ldots Z_n) = P(X_1 \ldots X_l|Z_1 \ldots Z_n)$$

Note the correspondence between this definition and our use of conditional independence in the definition of the naive Bayes classifier. The naive Bayes classifier assumes that the instance attribute A_1 is conditionally independent of instance attribute A_2 given the target value V. This allows the naive Bayes classifier to calculate $P(A_1, A_2|V)$ in Equation (6.20) as follows

$$P(A_1, A_2|V) = P(A_1|A_2, V)P(A_2|V) \tag{6.23}$$

$$= P(A_1|V)P(A_2|V) \tag{6.24}$$

Equation (6.23) is just the general form of the product rule of probability from Table 6.1. Equation (6.24) follows because if A_1 is conditionally independent of A_2 given V, then by our definition of conditional independence $P(A_1|A_2, V) = P(A_1|V)$.

ii.

Step 1: Calculate the expected value $E[z_{ij}]$ of each hidden variable z_{ij}, assuming the current hypothesis $h = \langle \mu_1, \mu_2 \rangle$ holds.

Step 2: Calculate a new maximum likelihood hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$, assuming the value taken on by each hidden variable z_{ij} is its expected value $E[z_{ij}]$ calculated in Step 1. Then replace the hypothesis $h = \langle \mu_1, \mu_2 \rangle$ by the new hypothesis $h' = \langle \mu'_1, \mu'_2 \rangle$ and iterate.

iii.

$$E[\ln P(Y|h')] = E\left[\sum_{i=1}^{m}\left(\ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i - \mu'_j)^2\right)\right]$$

$$= \sum_{i=1}^{m}\left(\ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}](x_i - \mu'_j)^2\right)$$

To summarize, the function $Q(h'|h)$ for the k means problem is

$$Q(h'|h) = \sum_{i=1}^{m}\left(\ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}](x_i - \mu'_j)^2\right)$$

where $h' = \langle \mu'_1, \ldots, \mu'_k \rangle$ and where $E[z_{ij}]$ is calculated based on the current hypothesis h and observed data X. As discussed earlier

$$E[z_{ij}] = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}} \tag{6.29}$$

2

3

i.

When evaluating a learned hypothesis we are most often interested in estimating the accuracy with which it will classify future instances. At the same time, we would like to know the probable error in this accuracy estimate (i.e., what error bars to associate with this estimate).

we consider the following setting for the learning problem. There is some space of possible instances X (e.g., the set of all people) over which various target functions may be defined (e.g., people who plan to purchase new skis this year). We assume that different instances in X may be encountered with different frequencies. A convenient way to model this is to assume there is some unknown probability distribution $\mathcal{D}$ that defines the probability of encountering each instance in X

2

ii.

Definition: The **sample error** (denoted $error_S(h)$) of hypothesis h with respect to target function f and data sample S is

$$error_S(h) \equiv \frac{1}{n} \sum_{x \in S} \delta(f(x), h(x))$$

Where n is the number of examples in S, and the quantity $\delta(f(x), h(x))$ is 1 if $f(x) \neq h(x)$, and 0 otherwise.

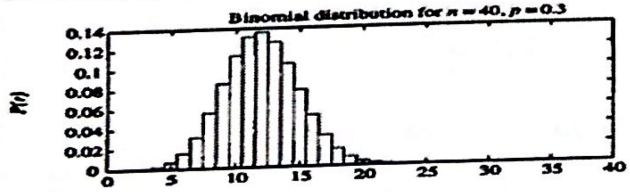
The *true error* of a hypothesis is the probability that it will misclassify a single randomly drawn instance from the distribution $\mathcal{D}$.

Definition: The **true error** (denoted $error_{\mathcal{D}}(h)$) of hypothesis h with respect to target function f and distribution $\mathcal{D}$, is the probability that h will misclassify an instance drawn at random according to $\mathcal{D}$.

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[f(x) \neq h(x)]$$

Here the notation $\Pr_{x \in \mathcal{D}}$ denotes that the probability is taken over the instance distribution $\mathcal{D}$.

iii.



A ***Binomial distribution*** gives the probability of observing r heads in a sample of n independent coin tosses, when the probability of heads on a single coin toss is p. It is defined by the probability function

$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

If the random variable X follows a Binomial distribution, then:

- The probability $\Pr(X = r)$ that X will take on the value r is given by $P(r)$
- The expected, or mean value of X, $E[X]$, is

$$E[X] = np$$

- The variance of X, $Var(X)$, is

$$Var(X) = np(1-p)$$

- The standard deviation of X, σ_X, is

$$\sigma_X = \sqrt{np(1-p)}$$

For sufficiently large values of n the Binomial distribution is closely approximated by a Normal distribution (see Table 5.4) with the same mean and variance. Most statisticians recommend using the Normal approximation only when $np(1-p) \geq 5$.

The Binomial distribution.

4

i.

2

One common way to describe the uncertainty associated with an estimate is to give an interval within which the true value is expected to fall, along with the probability with which it is expected to fall into this interval. Such estimates are called *confidence interval* estimates.

Definition: An $N\%$ **confidence interval** for some parameter p is an interval that is expected with probability $N\%$ to contain p.

ii.

Theorem **Central Limit Theorem.** Consider a set of independent, identically distributed random variables $Y_1 \ldots Y_n$ governed by an arbitrary probability distribution with mean μ and finite variance σ^2. Define the sample mean, $\bar{Y}_n \equiv \frac{1}{n}\sum_{i=1}^{n} Y_i$.

Then as $n \rightarrow \infty$, the distribution governing

$$\frac{\bar{Y}_n - \mu}{\frac{\sigma}{\sqrt{n}}}$$

approaches a Normal distribution, with zero mean and standard deviation equal to 1.

iii.

Above we described one procedure for comparing two learning methods given a fixed set of data. This section discusses the statistical justification for this procedure, and for the confidence interval defined by Equations (5.17) and (5.18). It can be skipped or skimmed on a first reading without loss of continuity.

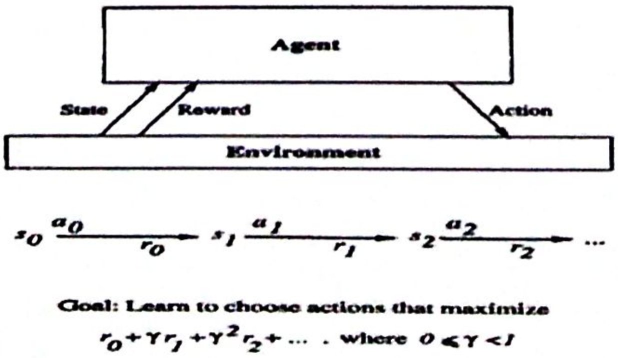
The best way to understand the justification for the confidence interval estimate given by Equation (5.17) is to consider the following estimation problem:

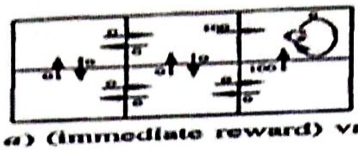
- We are given the observed values of a set of independent, identically distributed random variables $Y_1, Y_2, \ldots, Y_k$.
- We wish to estimate the mean μ of the probability distribution governing these Y_i.
- The estimator we will use is the sample mean $\bar{Y}$

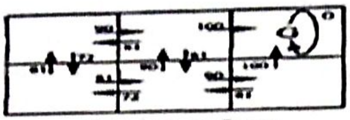
$$\bar{Y} \equiv \frac{1}{k}\sum_{i=1}^{k} Y_i$$

5

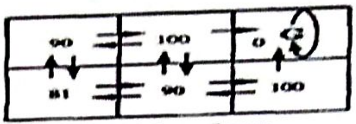
i.



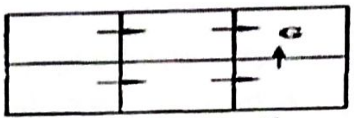
ii. 2

r(s, a) (immediate reward) values



Q(s, a) values

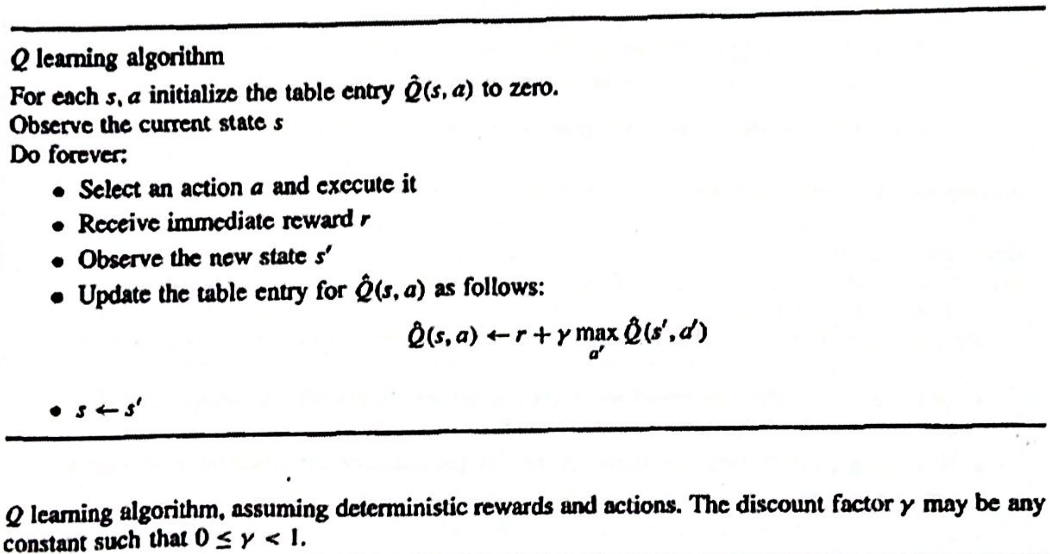


V*(s) values



One optimal policy

iii.



Q learning algorithm

```
For each s, a initialize the table entry $\hat{Q}(s,a)$ to zero.
Observe the current state s
Do forever:
    • Select an action a and execute it
    • Receive immediate reward r
    • Observe the new state s'
    • Update the table entry for $\hat{Q}(s,a)$ as follows:
          $\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$
    • $s \leftarrow s'$
```

Q learning algorithm, assuming deterministic rewards and actions. The discount factor γ may be any constant such that $0 \leq \gamma < 1$.

Course In charge

Module Coordinator

HOD

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109

I SESSIONAL TEST QUESTION PAPER 2020 ODD SEMESTER

Set A

USN										

Degree : B.E
Branch : Computer Science and Engineering
Course Title : Machine Learning
Duration : 90 Minutes

Semester : VII
Course Code : 17CS73
Date : 06-10-2020
Max Marks : 30

Note: Answer ONE full question from each part.

<table>
<tr><th>Q No.</th><th>Question</th><th>Marks</th><th>CO mapping</th><th>K-Level</th></tr>
<tr><td colspan="5">PART-A</td></tr>
<tr><td>1(a)</td><td>Identify various applications of Machine Learning.</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design Checkers learning system using four distinct program modules.</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(c)</td><td>Design and apply Find – S Algorithm for the given target concept Enjoy sport.
<table>
<tr><th>Example</th><th>Sky</th><th>AirTemp</th><th>Humidity</th><th>Wind</th><th>Water</th><th>Forecast</th><th>EnjoySport</th></tr>
<tr><td>1</td><td>Sunny</td><td>Warm</td><td>Normal</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>2</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>3</td><td>Rainy</td><td>Cold</td><td>High</td><td>Strong</td><td>Warm</td><td>Change</td><td>No</td></tr>
<tr><td>4</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Cool</td><td>Change</td><td>Yes</td></tr>
</table></td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>2(a)</td><td>Determine the following with respect to checkers learning problem.
i. Choosing training experience
ii. Choosing Target function</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Identify different issues in machine learning.</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(c)</td><td>Design and apply Candidate Elimination Algorithm for the given target concept Enjoy sport (Refer Question 1(c)).</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">PART-B</td></tr>
</table>

<table>
<tr><td>3(a)</td><td>Identify Appropriate problems for decision tree learning.</td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design ID3 Algorithm</td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>4(a)</td><td>Identify the necessary measure required to select the attributes for building decision tree using ID3 Algorithm.</td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Apply ID3 Algorithm to construct the decision tree for the given target concept Play tennis.
<table>
<tr><th>Day</th><th><i>Outlook</i></th><th><i>Temperature</i></th><th><i>Humidity</i></th><th><i>Wind</i></th><th><i>PlayTennis</i></th></tr>
<tr><td>D1</td><td>Sunny</td><td>Hot</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D2</td><td>Sunny</td><td>Hot</td><td>High</td><td>Strong</td><td>No</td></tr>
<tr><td>D3</td><td>Overcast</td><td>Hot</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D4</td><td>Rain</td><td>Mild</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D5</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D6</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Strong</td><td>No</td></tr>
<tr><td>D7</td><td>Overcast</td><td>Cool</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D8</td><td>Sunny</td><td>Mild</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D9</td><td>Sunny</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D10</td><td>Rain</td><td>Mild</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D11</td><td>Sunny</td><td>Mild</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D12</td><td>Overcast</td><td>Mild</td><td>High</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D13</td><td>Overcast</td><td>Hot</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D14</td><td>Rain</td><td>Mild</td><td>High</td><td>Strong</td><td>No</td></tr>
</table></td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
</table>

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

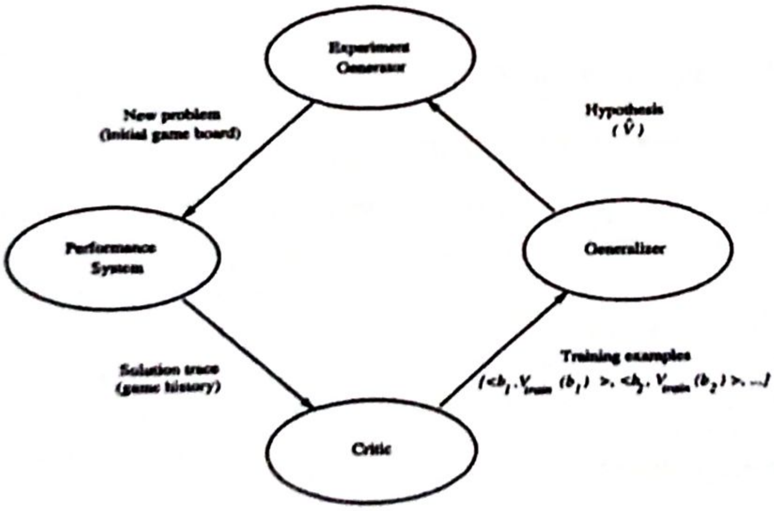
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

Set A

Degree	**:**	**B.E**	**Semester**	**:**	**VII**
Branch	**:**	**CSE**	**Course Code**	**:**	**17CS73**
Course Title	**:**	**Machine Learning**	**Max Marks**	**:**	**30**

Q.NO.	POINTS	MARKS
1(a)	**Applications of Machine Learning** • Learning to recognize spoken words • Learning to drive an autonomous vehicle • Learning to classify new astronomical structures • Learning to play world-class backgammon Explanation of each application in detail	6M
(b)	The final design of checkers learning system can be described by four distinct program modules that represent the central components in many learning systems 	6M
(c)	<u>**FIND-S Algorithm**</u> 1. Initialize *h* to the most specific hypothesis in *H* 2. For each positive training instance *x* For each attribute constraint a_i in *h* If the constraint a_i is satisfied by *x*	3+3=6

Then do nothing

Else replace a_i in *h* by the next more general constraint that is satisfied by *x*

3. Output hypothesis *h*

Analysis of EnjoySport

x_1 = *<Sunny Warm Normal Strong Warm Same>*, +

Observing the first training example, it is clear that our hypothesis is too specific. In particular, none of the "Ø" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example

h_1 = <Sunny Warm Normal Strong Warm Same>

This h is still very specific; it asserts that all instances are negative except for the single positive training example

x_2 = *<Sunny, Warm, High, Strong, Warm, Same>*, +

The second training example forces the algorithm to further generalize h, this time substituting a "?' in place of any attribute value in h that is not satisfied by the new example

h_2 = <Sunny Warm ? Strong Warm Same>

x3 = *<Rainy, Cold, High, Strong, Warm, Change>*, -

Upon encountering the third training the algorithm makes no change to h.

The FIND-S algorithm simply ignores every negative example.

h3 = < Sunny Warm ? Strong Warm Same>

x4 = *<Sunny Warm High Strong Cool Change>*, +

The fourth example leads to a further generalization of h

h4 = < Sunny Warm ? Strong ? ? >

2(a) **Designing a Learning System** 6

Explanation of each step-in detail

	(i)Choosing the Training Experience There are three attributes which impact on success or failure of the learner 1. Whether the training experience provides direct or indirect feedback regarding the choices made by the performance system. 2. The degree to which the learner controls the sequence of training examples.How well it represents the distribution of examples over which the final system performance P must be measured 3. (ii)Choosing the Target Function Let *ChooseMove* be the target function and the notation is $ChooseMove : B \rightarrow M$ which indicate that this function accepts as input any board from the set of legal board states B and produces as output some move from the set of legal moves M. An alternative target function is an *evaluation function* that assigns a *numerical score* to any given board state Let the target function V and the notation $V : B \rightarrow R$ which denote that V maps any legal board state from the set B to some real value	
(b)	Issues in Machine Learning • What algorithms exist for learning general target functions from specific training examples?? • How much training data is sufficient? • When and how can prior knowledge held by the learner guide the process of generalizing from examples? • What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy alter the complexity of the learning problem? • What is the best way to reduce the learning task to one or more function approximation problems? • How can the learner automatically alter its representation to improve its ability to represent and learn the target function?	6M.
(c)	The CANDIDATE-ELIMINTION algorithm computes the ***version space*** containing all hypotheses from H that are consistent with an observed sequence of	3+3=6M

training examples

Initialize G to the set of maximally general hypotheses in H

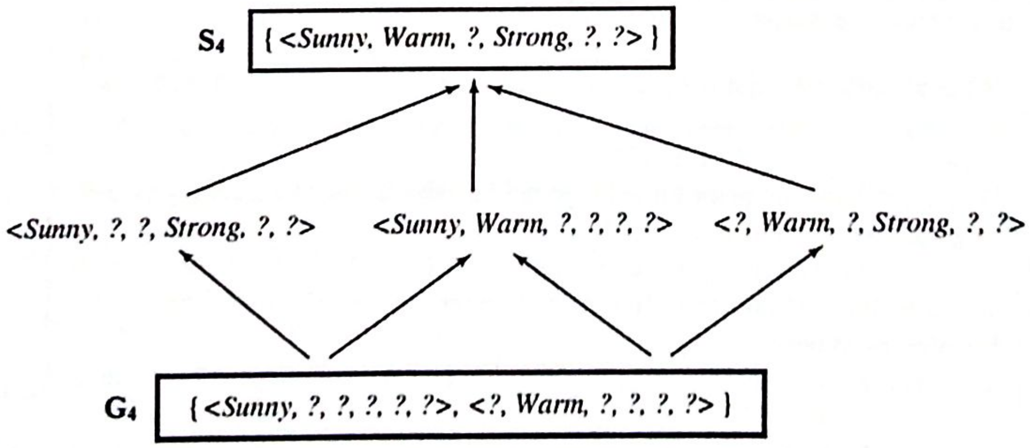
Initialize S to the set of maximally specific hypotheses in H

For each training example d, do

- If d is a positive example
    - Remove from G any hypothesis inconsistent with d
    - For each hypothesis s in S that is not consistent with d
        - Remove s from S
        - Add to S all minimal generalizations h of s such that
            - h is consistent with d, and some member of G is more general than h
        - Remove from S any hypothesis that is more general than another hypothesis in S
        - If d is a negative example
    - Remove from S any hypothesis inconsistent with d
    - For each hypothesis g in G that is not consistent with d
        - Remove g from G
        - Add to G all minimal specializations h of g such that
            - h is consistent with d, and some member of S is more specific than h

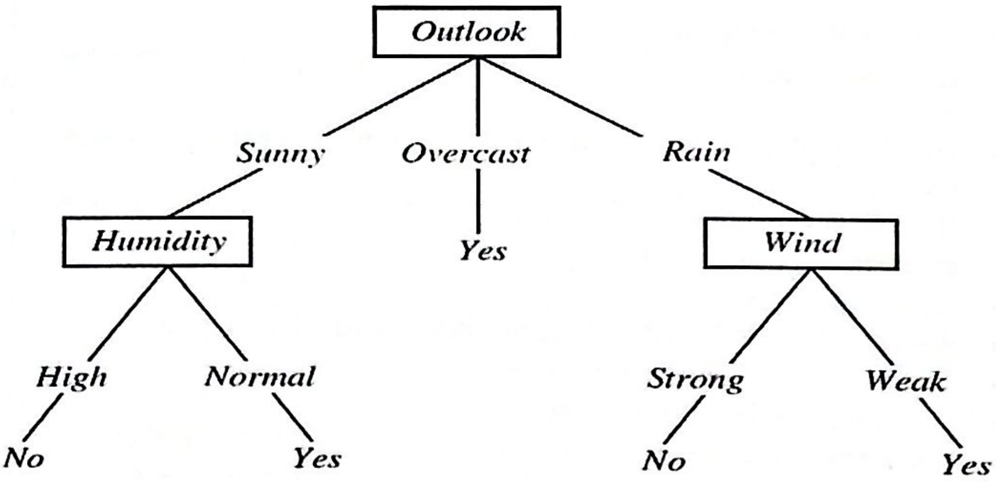
Remove from G any hypothesis that is less general than another hypothesis in G

Analysis of Enjoy Sport



3(a)	Decision tree learning is generally best suited to	6M

	problems with the following characteristics: 1. ***Instances are represented by attribute-value pairs*** – Instances are described by a fixed set of attributes and their values 2. ***The target function has discrete output values*** – The decision tree assigns a Boolean classification (e.g., yes or no) to each example. Decision tree methods easily extend to learning functions with more than two possible output values. 3. ***Disjunctive descriptions may be required*** ***4.The training data may contain errors*** – Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples. ***5.The training data may contain missing attribute values*** – Decision tree methods can be used even when some training examples have unknown values	
(b)	***ID3(Examples, Target_attribute, Attributes)*** Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples. • Create a Root node for the tree • If all Examples are positive, Return the single-node tree Root, with label = + • If all Examples are negative, Return the single-node tree Root, with label = - • If Attributes is empty, Return the single-node tree Root, with label = most common value of Target_attribute in Examples • Otherwise Begin • A ← the attribute from Attributes that best* classifies Examples • The decision attribute for Root ← A • For each possible value, v_i, of A, • Add a new tree branch below *Root*, corresponding to the test A = v_i • Let *Examples* $_{vi}$, be the subset of Examples that have value v_i for *A* • If *Examples* $_{vi}$, is empty • Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples • Else below this new branch add the subtree ID3(*Examples* $_{vi}$, Targe_tattribute, Attributes – {A})) • End Return Root	6M
4(a)	The central choice in the ID3 algorithm is selecting which attribute to test at each	6M

	node in the tree. • A statistical property called ***information gain*** that measures how well a given attribute separates the training examples according to their target classification. • ID3 uses ***information gain*** measure to select among the candidate attributes at each step while growing the tree. • To define information gain, we begin by defining a measure called entropy. • *Entropy measures the impurity of a collection of examples.* • Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is $Entropy\ (S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$	
(b)	The information gain values for all four attributes are • Gain(S, Outlook)=0.246 • Gain(S, Humidity)=0.151 • Gain(S, Wind)=0.048 • Gain(S, Temperature)=0.029 Final decision tree after analysing entire training dataset 	6M

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109

I SESSIONAL TEST QUESTION PAPER 2020 ODD SEMESTER

Set B

USN ☐☐☐☐☐☐☐☐☐☐

Degree	**:**	**B.E**	**Semester :**	**VII**
Branch	**:**	**Computer Science and Engineering**	**Course Code :**	**17CS73**
Course Title	**:**	**Machine Learning**	**Date :**	**06-10-2020**
Duration	**:**	**90 Minutes**	**Max Marks :**	**30**

Note: Answer ONE full question from each part.

<table>
<tr><th>Q No.</th><th>Question</th><th>Marks</th><th>CO mapping</th><th>K-Level</th></tr>
<tr><td colspan="5">PART-A</td></tr>
<tr><td>1(a)</td><td>Design List-Then Eliminate Algorithm</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design Checkers learning system using four distinct program modules.</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(c)</td><td>Apply Find – S Algorithm for the given target concept Enjoy sport.
<table>
<tr><th><i>Example</i></th><th><i>Sky</i></th><th><i>AirTemp</i></th><th><i>Humidity</i></th><th><i>Wind</i></th><th><i>Water</i></th><th><i>Forecast</i></th><th><i>EnjoySport</i></th></tr>
<tr><td>1</td><td>Sunny</td><td>Warm</td><td>Normal</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>2</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Warm</td><td>Same</td><td>Yes</td></tr>
<tr><td>3</td><td>Rainy</td><td>Cold</td><td>High</td><td>Strong</td><td>Warm</td><td>Change</td><td>No</td></tr>
<tr><td>4</td><td>Sunny</td><td>Warm</td><td>High</td><td>Strong</td><td>Cool</td><td>Change</td><td>Yes</td></tr>
</table>
</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>2(a)</td><td>Determine the following with respect to checkers learning problem.
i. Choosing training experience
ii. Choosing Target function</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design Find – S Algorithm.</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td>(c)</td><td>Design and apply Candidate Elimination Algorithm for the given target concept Enjoy sport (Refer Question 1(c)).</td><td>6</td><td>CO1</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">PART-B</td></tr>
</table>

<table>
<tr><td>3(a)</td><td>Identify Appropriate problems for decision tree learning.</td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design ID3 Algorithm</td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>4(a)</td><td>Identify the necessary measure required to select the attributes for building decision tree using ID3 Algorithm.</td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Apply ID3 Algorithm to construct the decision tree for the given target concept Play tennis.
<table>
<tr><th>Day</th><th><i>Outlook</i></th><th><i>Temperature</i></th><th><i>Humidity</i></th><th><i>Wind</i></th><th><i>PlayTennis</i></th></tr>
<tr><td>D1</td><td>Sunny</td><td>Hot</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D2</td><td>Sunny</td><td>Hot</td><td>High</td><td>Strong</td><td>No</td></tr>
<tr><td>D3</td><td>Overcast</td><td>Hot</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D4</td><td>Rain</td><td>Mild</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D5</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D6</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Strong</td><td>No</td></tr>
<tr><td>D7</td><td>Overcast</td><td>Cool</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D8</td><td>Sunny</td><td>Mild</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D9</td><td>Sunny</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D10</td><td>Rain</td><td>Mild</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D11</td><td>Sunny</td><td>Mild</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D12</td><td>Overcast</td><td>Mild</td><td>High</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D13</td><td>Overcast</td><td>Hot</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D14</td><td>Rain</td><td>Mild</td><td>High</td><td>Strong</td><td>No</td></tr>
</table></td><td>6</td><td>CO2</td><td>Applying(K3)</td></tr>
</table>

Signature of Course in charge

Signature of Module Coordinator

Signature of HOD

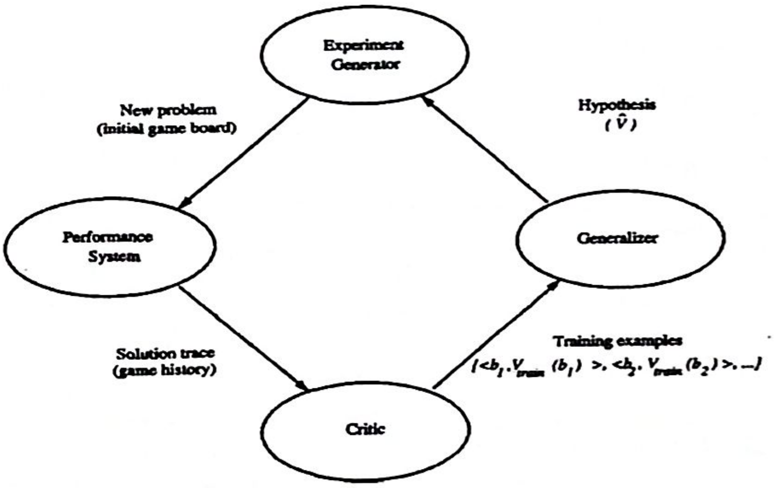
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

Set B

Degree	**:**	**B.E**	**Semester**	**:**	**VII**
Branch	**:**	**CSE**	**Course Code**	**:**	**17CS73**
Course Title	**:**	**Machine Learning**	**Max Marks**	**:**	**30**

Q.NO.	POINTS	MARKS
1(a)	Mention the list then algorithm with its steps and explain briefly how the algorithm works with example	6M
(b)	The final design of checkers learning system can be described by four distinct program modules that represent the central components in many learning systems  	6M
(c)	Analysis of EnjoySport x_1 = <*Sunny Warm Normal Strong Warm Same*>, + Observing the first training example, it is clear that our hypothesis is too specific. In particular, none of the "Ø" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example ***h_1 = <Sunny Warm Normal Strong Warm Same>*** This h is still very specific; it asserts that all instances are negative except for the	3+3=6

	single positive training example x_2 = <*Sunny, Warm, High, Strong, Warm, Same*>, + The second training example forces the algorithm to further generalize h, this time substituting a "?' in place of any attribute value in h that is not satisfied by the new example ***h_2 = <Sunny Warm ? Strong Warm Same>*** *x3 = <Rainy, Cold, High, Strong, Warm, Change>*, - Upon encountering the third training the algorithm makes no change to h. The FIND-S algorithm simply ignores every negative example. ***h3 = < Sunny Warm ? Strong Warm Same>*** *x4 = <Sunny Warm High Strong Cool Change>*, + The fourth example leads to a further generalization of h ***h4 = < Sunny Warm ? Strong ? ? >***	
2(a)	**Designing a Learning System** **Explanation of each step-in detail** (i)Choosing the Training Experience There are three attributes which impact on success or failure of the learner 1. Whether the training experience provides direct or indirect feedback regarding the choices made by the performance system. 2. The degree to which the learner controls the sequence of training examples.How well it represents the distribution of examples over which the final system performance P must be measured 3. (ii)Choosing the Target Function Let *ChooseMove* be the target function and the notation is *ChooseMove : B*→ *M* which indicate that this function accepts as input any board from the set of legal	6

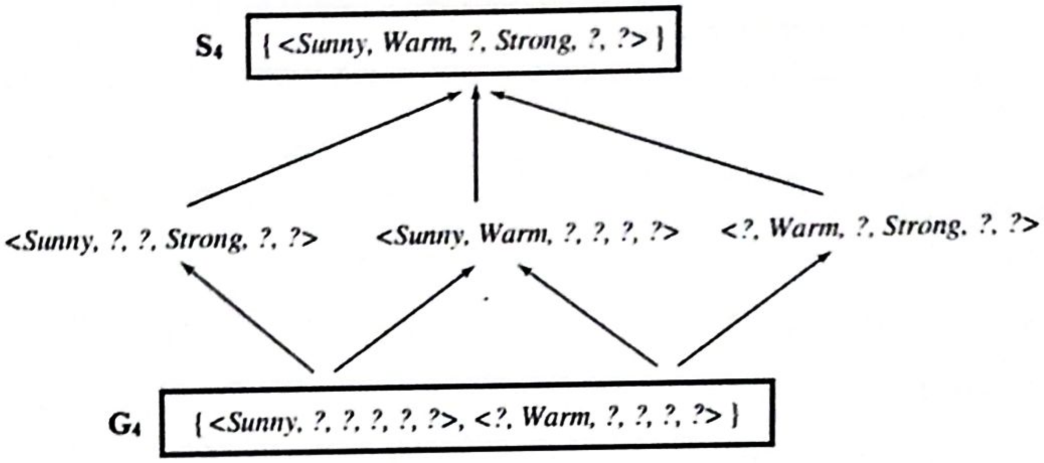
	board states B and produces as output some move from the set of legal moves M. An alternative target function is an *evaluation function* that assigns a *numerical score* to any given board state Let the target function V and the notation $V : B \rightarrow R$ which denote that V maps any legal board state from the set B to some real value	
(b)	**FIND-S Algorithm** 1.Initialize *h* to the most specific hypothesis in *H* 2.For each positive training instance *x* For each attribute constraint a_i in *h* If the constraint a_i is satisfied by *x* Then do nothing Else replace a_i in *h* by the next more general constraint that is satisfied by *x* 3.Output hypothesis *h*	6M

(c)	The CANDIDATE-ELIMINTION algorithm computes the ***version space*** containing all hypotheses from H that are consistent with an observed sequence of training examples
	Initialize G to the set of maximally general hypotheses in H
	Initialize S to the set of maximally specific hypotheses in H
	For each training example d, do
	• If d is a positive example
	  • Remove from G any hypothesis inconsistent with d
	  • For each hypothesis s in S that is not consistent with d
	    • Remove s from S
	    • Add to S all minimal generalizations h of s such that
	      • h is consistent with d, and some member of G is more general than h
	    • Remove from S any hypothesis that is more general than another hypothesis in S
	    • If d is a negative example
	  • Remove from S any hypothesis inconsistent with d
	  • For each hypothesis g in G that is not consistent with d
		3+3=6M

- Remove g from G
- Add to G all minimal specializations h of g such that
    - h is consistent with d, and some member of S is more specific than h

Remove from G any hypothesis that is less general than another hypothesis in G

Analysis of Enjoy Sport



3(a) Decision tree learning is generally best suited to problems with the following characteristics: 6M

1. ***Instances are represented by attribute-value pairs*** – Instances are described by a fixed set of attributes and their values
2. ***The target function has discrete output values*** – The decision tree assigns a Boolean classification (e.g., yes or no) to each example. Decision tree methods easily extend to learning functions with more than two possible output values.
3. ***Disjunctive descriptions may be required***

4.The training data may contain errors – Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.

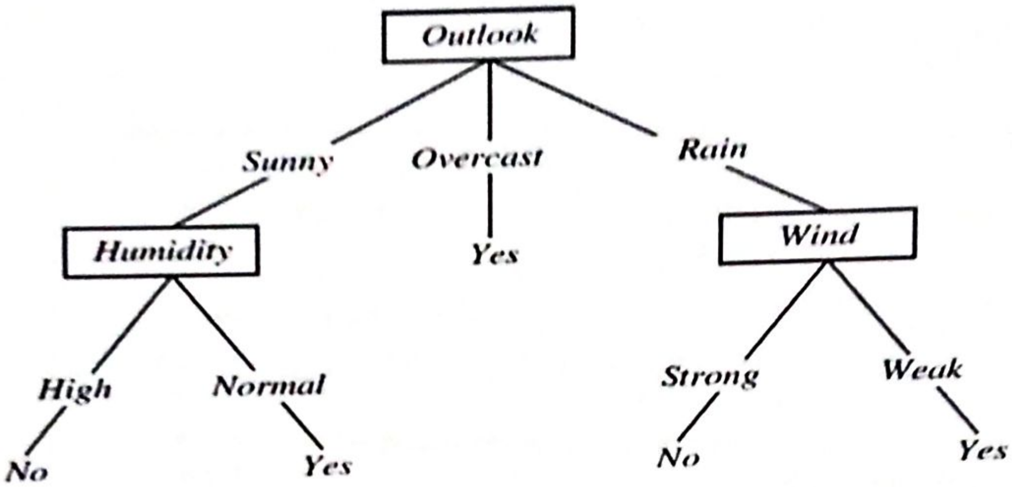
5.The training data may contain missing attribute values – Decision tree methods can be used even when some training examples have unknown values

(b) *ID3(Examples, Target_attribute, Attributes)* 6M

	Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples. • Create a Root node for the tree • If all Examples are positive, Return the single-node tree Root, with label = + • If all Examples are negative, Return the single-node tree Root, with label = - • If Attributes is empty, Return the single-node tree Root, with label = most common value of Target_attribute in Examples • Otherwise Begin • A ← the attribute from Attributes that best* classifies Examples • The decision attribute for Root ← A • For each possible value, v_i, of A, • Add a new tree branch below *Root*, corresponding to the test A = v_i • Let *Examples* $_{vi}$, be the subset of Examples that have value v_i for *A* • If *Examples* $_{vi}$, is empty • Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples • Else below this new branch add the subtree ID3(*Examples* $_{vi}$, Targe_tattribute, Attributes – {A})) • End Return Root	
4(a)	The central choice in the ID3 algorithm is selecting which attribute to test at each node in the tree. • A statistical property called ***information gain*** that measures how well a given attribute separates the training examples according to their target classification. • ID3 uses ***information gain*** measure to select among the candidate attributes at each step while growing the tree. • To define information gain, we begin by defining a measure called entropy. • *Entropy measures the impurity of a collection of examples.* • Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is $Entropy\ (S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$	6M
(b)	The information gain values for all four attributes are • Gain(S, Outlook)=0.246	6M

- Gain(S, Humidity)=0.151
- Gain(S, Wind)=0.048
- Gain(S, Temperature)=0.029

Final decision tree after analysing entire training dataset



Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109
II SESSIONAL TEST QUESTION PAPER 2020 ODD SEMESTER
Set A

USN

Degree : **B.E**
Branch : **Computer Science and Engineering**
Course Title : **Machine Learning**
Duration : **90 Minutes**

Semester : **VII**
Course Code : **17CS73**
Date : **18-11-2020**
Max Marks : **30**

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Make use of** suitable diagram, explain the concept of single perceptron.	6	CO3	Applying(K3)
(b)	**Determine** the application of neural network which is used for learning to steer an autonomous vehicle.	6	CO3	Applying(K3)
(c)	**Design** Back propagation algorithm.	6	CO3	Applying(K3)
	OR			
2(a)	**Design** and derive Gradient Descent Rule	6	CO3	Applying(K3)
(b)	**Identify** the appropriate problem for Neural Network Learning	6	CO3	Applying(K3)
(c)	**Derive** Back propagation Rule **(i) with respect to output layer**	6	CO3	Applying(K3)
	PART-B			
3(a)	**Make use of** suitable example, explain the Hypothesis Space Search in Decision Tree Learning	6	CO2	Applying(K3)
(b)	**Identify** features of Bayesian learning methods	6	CO4	Applying(K3)
	OR			
4(a)	**Identify** various issues In Decision Tree Learning	6	CO2	Applying(K3)
(b)	**Determine** Brute Force MAP Learning Algorithm	6	CO4	Applying(K3)

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

Set A

Degree	:	**B.E**	**Semester**	:	**VII**
Branch	:	**CSE**	**Course Code**	:	**17CS73**
Course Title	:	**Machine Learning**	**Max Marks**	:	**30**

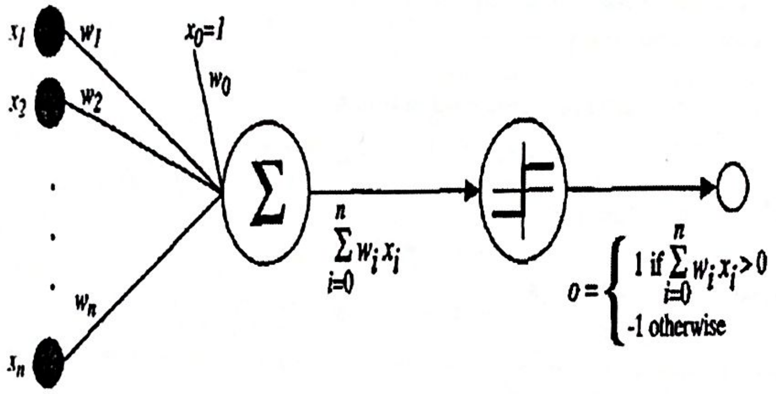
<table>
<tr><th>Q.NO.</th><th>POINTS</th><th>MARKS</th></tr>
<tr><td>1(a)</td><td>


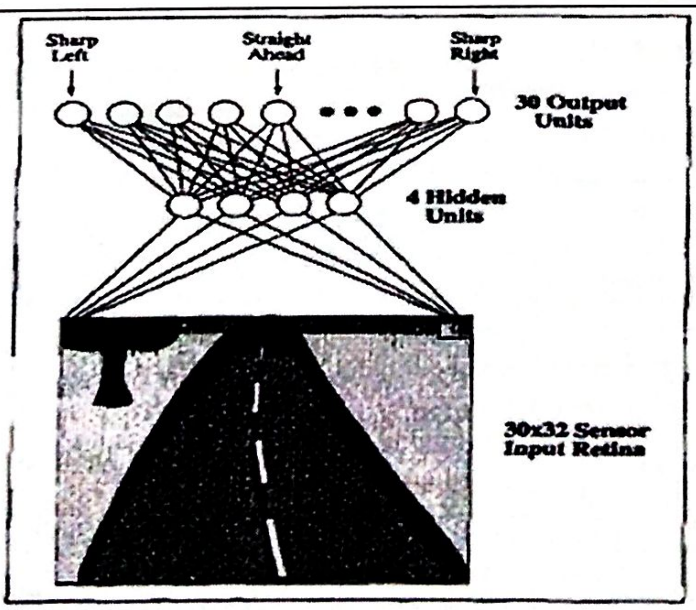
Diagram and explanation</td><td>2M + 4M</td></tr>
<tr><td>(b)</td><td>


Diagram and explanation</td><td>2M + 4M</td></tr>
</table>

(c)

```
BACKPROPAGATION(training_examples, $\eta$, $n_{in}$, $n_{out}$, $n_{hidden}$)
    Each training example is a pair of the form $\langle \vec{x}, \vec{t} \rangle$, where $\vec{x}$ is the vector of network input values, and $\vec{t}$ is the vector of target network output values.
    $\eta$ is the learning rate (e.g., .05). $n_{in}$ is the number of network inputs, $n_{hidden}$ the number of units in the hidden layer, and $n_{out}$ the number of output units.
    The input from unit i into unit j is denoted $x_{ji}$, and the weight from unit i to unit j is denoted $w_{ji}$.
• Create a feed-forward network with $n_{in}$ inputs, $n_{hidden}$ hidden units, and $n_{out}$ output units.
• Initialize all network weights to small random numbers (e.g., between -.05 and .05).
• Until the termination condition is met, Do
    • For each $\langle \vec{x}, \vec{t} \rangle$ in training_examples, Do
        Propagate the input forward through the network:
        1. Input the instance $\vec{x}$ to the network and compute the output $o_u$ of every unit u in the network.
        Propagate the errors backward through the network:
        2. For each network output unit k, calculate its error term $\delta_k$
               $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$
        3. For each hidden unit h, calculate its error term $\delta_h$
               $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$
        4. Update each network weight $w_{ji}$
               $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$
           where
               $\Delta w_{ji} = \eta \delta_j x_{ji}$
```

3+3=6

2(a)

```
GRADIENT-DESCENT(training_examples, $\eta$)
    Each training example is a pair of the form $\langle \vec{x}, t \rangle$, where $\vec{x}$ is the vector of input values, and t is the target output value. $\eta$ is the learning rate (e.g., .05).
• Initialize each $w_i$ to some small random value
• Until the termination condition is met, Do
    • Initialize each $\Delta w_i$ to zero.
    • For each $\langle \vec{x}, t \rangle$ in training_examples, Do
        • Input the instance $\vec{x}$ to the unit and compute the output o
        • For each linear unit weight $w_i$, Do
              $\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$
    • For each linear unit weight $w_i$, Do
              $w_i \leftarrow w_i + \Delta w_i$
```

Design and Derivation

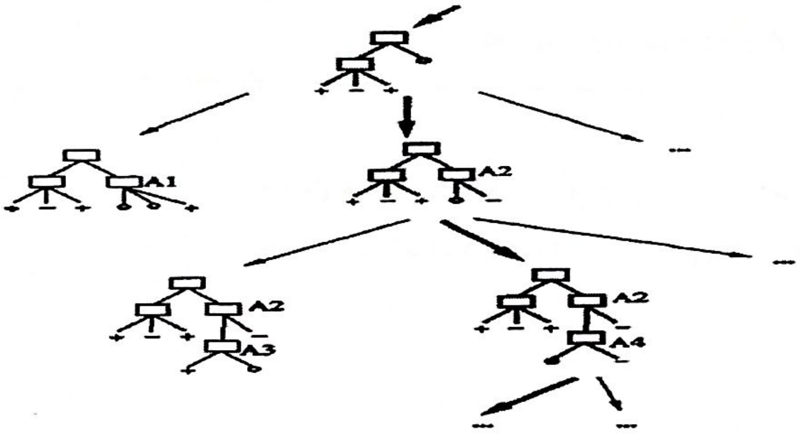
3+3=6

(b)

Appropriate problems for Neural Network Learning

- *Instances are represented by many attribute-value pairs.*
- *The target function output may be discrete-valued, real-valued, or a vector of several real- or discrete-valued attributes.*

1 * 6 = 6M

	• *The training examples may contain errors.* • *Long training times are acceptable.* • *Fast evaluation of the learned target function may be required.* • *The ability of humans to understand the learned target function is not important.*	
(c)	Back propagation Rule **(i) with respect to output layer** Derivation and Explanation	3+3=6M
3(a)	 Diagram and explanation	3+3=6M
(b)	Features of Bayesian learning methods • Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct. This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example. • Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided by asserting (1) a prior probability for each candidate hypothesis, and (2) a probability distribution over observed data for each possible hypothesis. • Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery"). • New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities. • Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.	6M

4(a)	various issues In Decision Tree Learning	1 * 6 = 6M
(b)	**BRUTE-FORCE MAP LEARNING algorithm** 1. For each hypothesis h in H, calculate the posterior probability $P(h\|D) = \frac{P(D\|h)P(h)}{P(D)}$ 2. Output the hypothesis h_{MAP} with the highest posterior probability $h_{MAP} = \underset{h \in H}{\operatorname{argmax}} P(h\|D)$ Formula description and Explanation	3+3=6M

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109

II SESSIONAL TEST QUESTION PAPER 2020 ODD SEMESTER

Set B

USN										

Degree	:	**B.E**	**Semester**	:	**VII**
Branch	:	**Computer Science and Engineering**	**Course Code**	:	**17CS73**
Course Title	:	**Machine Learning**	**Date**	:	**18-11-2020**
Duration	:	**90 Minutes**	**Max Marks**	:	**30**

Note: Answer ONE full question from each part.

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Determine** why stochastic approximation is needed to Gradient descent rule	6	CO3	Applying(K3)
(b)	**Determine** the application of neural network which is used for learning to steer an autonomous vehicle.	6	CO3	Applying(K3)
(c)	**Design** Back propagation algorithm.	6	CO3	Applying(K3)
	OR			
2(a)	**Identify** the appropriate problem for Neural Network Learning	6	CO3	Applying(K3)
(b)	**Design** and derive Gradient Descent Rule	6	CO3	Applying(K3)
(c)	**Derive** Back propagation Rule **(i) with respect to output layer**	6	CO3	Applying(K3)
	PART-B			
3(a)	**Make use of** suitable example, explain the Hypothesis Space Search in Decision Tree Learning	6	CO2	Applying(K3)
(b)	**Determine** Brute Force MAP Learning Algorithm	6	CO4	Applying(K3)
	OR			
4(a)	**Identify** various issues In Decision Tree Learning	6	CO2	Applying(K3)
(b)	**Identify** features of Bayesian learning methods	6	CO4	Applying(K3)

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

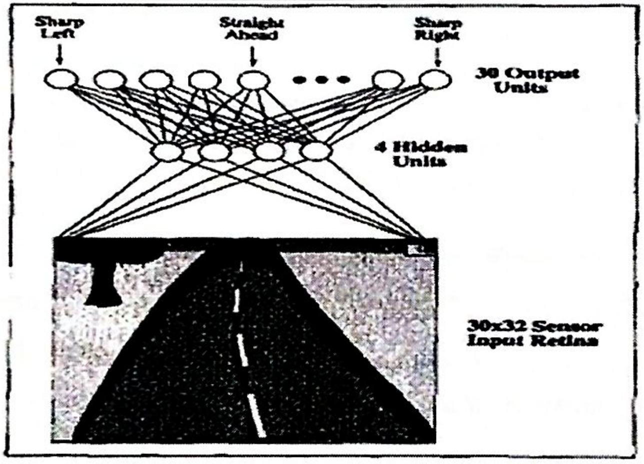
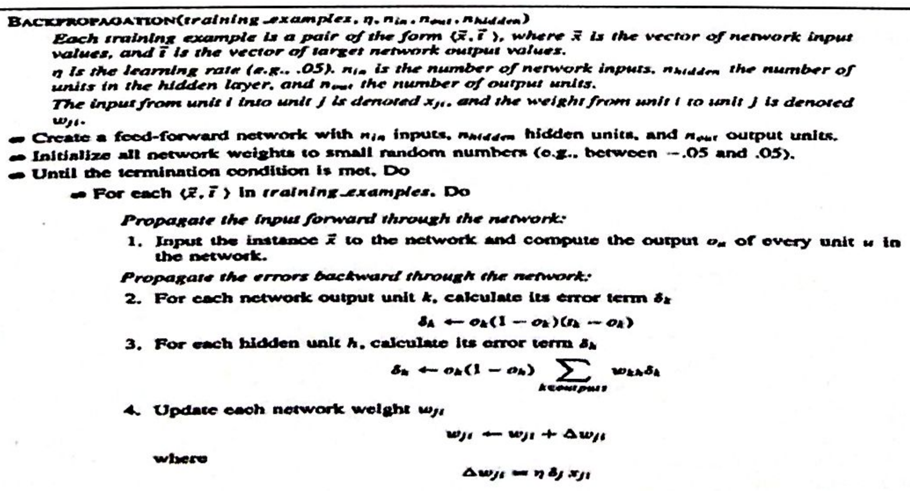
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

Set B

Degree	:	B.E	Semester	:	VII
Branch	:	CSE	Course Code	:	17CS73
Course Title	:	Machine Learning	Max Marks	:	30

Q.NO.	POINTS	MARKS
1(a)	Mentioning about advantages of stochastic approximation with explanation	6*1=6M
(b)	 Diagram and explanation	2M + 4M
(c)		3+3=6

```
BACKPROPAGATION(training_examples, $\eta$, $n_{in}$, $n_{out}$, $n_{hidden}$)
    Each training example is a pair of the form $\langle \vec{x}, \vec{t} \rangle$, where $\vec{x}$ is the vector of network input values, and $\vec{t}$ is the vector of target network output values.
    $\eta$ is the learning rate (e.g., .05). $n_{in}$ is the number of network inputs, $n_{hidden}$ the number of units in the hidden layer, and $n_{out}$ the number of output units.
    The input from unit $i$ into unit $j$ is denoted $x_{ji}$, and the weight from unit $i$ to unit $j$ is denoted $w_{ji}$.
- Create a feed-forward network with $n_{in}$ inputs, $n_{hidden}$ hidden units, and $n_{out}$ output units.
- Initialize all network weights to small random numbers (e.g., between -.05 and .05).
- Until the termination condition is met, Do
    - For each $\langle \vec{x}, \vec{t} \rangle$ in training_examples, Do
        Propagate the input forward through the network:
        1. Input the instance $\vec{x}$ to the network and compute the output $o_u$ of every unit $u$ in the network.
        Propagate the errors backward through the network:
        2. For each network output unit $k$, calculate its error term $\delta_k$
               $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$
        3. For each hidden unit $h$, calculate its error term $\delta_h$
               $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh} \delta_k$
        4. Update each network weight $w_{ji}$
               $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$
           where
               $\Delta w_{ji} = \eta \delta_j x_{ji}$
```

2(a)

```
GRADIENT-DESCENT(training_examples, $\eta$)
    Each training example is a pair of the form $\langle \vec{x}, t \rangle$, where $\vec{x}$ is the vector of input values, and
    $t$ is the target output value. $\eta$ is the learning rate (e.g., .05).
• Initialize each $w_i$ to some small random value
• Until the termination condition is met, Do
    • Initialize each $\Delta w_i$ to zero.
    • For each $\langle \vec{x}, t \rangle$ in training_examples, Do
        • Input the instance $\vec{x}$ to the unit and compute the output $o$
        • For each linear unit weight $w_i$, Do
                $\Delta w_i \leftarrow \Delta w_i + \eta (t - o) x_i$
    • For each linear unit weight $w_i$, Do
                $w_i \leftarrow w_i + \Delta w_i$
```

2(a)	Design and Derivation	3+3=6
(b)	Appropriate problems for Neural Network Learning • *Instances are represented by many attribute-value pairs.* • *The target function output may be discrete-valued, real-valued, or a vector of several real- or discrete-valued attributes.* • *The training examples may contain errors.* • *Long training times are acceptable.* • *Fast evaluation of the learned target function may be required.* • *The ability of humans to understand the learned target function is not important.*	1 * 6 = 6M
(c)	Back propagation Rule **(i) with respect to output layer** Derivation and Explanation	3+3=6M
3(a)	 Diagram and explanation	3+3=6M

(b)	Features of Bayesian learning methods - Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct. This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example. - Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided by asserting (1) a prior probability for each candidate hypothesis, and (2) a probability distribution over observed data for each possible hypothesis. - Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery"). - New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities. - Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.	6M
4(a)	Various issues In Decision Tree Learning	1 * 6 = 6M
(b)	**BRUTE-FORCE MAP LEARNING algorithm** 1. For each hypothesis h in H, calculate the posterior probability $P(h\|D) = \frac{P(D\|h)P(h)}{P(D)}$ 2. Output the hypothesis h_{MAP} with the highest posterior probability $h_{MAP} = \underset{h\in H}{\operatorname{argmax}} P(h\|D)$ Formula description and Explanation	3+3=6M

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

SET A

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109

III SESSIONAL TEST QUESTION PAPER 2020 ODD SEMESTER

USN										

Degree : **B.E**
Branch : **Computer Science and Engineering**
Course Title : **Machine Learning**
Duration : **90 Minutes**

Semester : **VII**
Course Code : **17CS73**
Date : **06-01-2021**
Max Marks : **30**

Note: Answer ONE full question from each part.

<table>
<tr><th>Q No.</th><th>Question</th><th>Marks</th><th>CO mapping</th><th>K-Level</th></tr>
<tr><td colspan="5">PART-A</td></tr>
<tr><td>1(a)</td><td>Apply naïve bayes classifier for the given training data to classify the instance
(outlook = sunny, temperature = cool, humidity = high, wind = strong)
<table>
<tr><th>Day</th><th><i>Outlook</i></th><th><i>Temperature</i></th><th><i>Humidity</i></th><th><i>Wind</i></th><th><i>PlayTennis</i></th></tr>
<tr><td>D1</td><td>Sunny</td><td>Hot</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D2</td><td>Sunny</td><td>Hot</td><td>High</td><td>Strong</td><td>No</td></tr>
<tr><td>D3</td><td>Overcast</td><td>Hot</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D4</td><td>Rain</td><td>Mild</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D5</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D6</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Strong</td><td>No</td></tr>
<tr><td>D7</td><td>Overcast</td><td>Cool</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D8</td><td>Sunny</td><td>Mild</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D9</td><td>Sunny</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D10</td><td>Rain</td><td>Mild</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D11</td><td>Sunny</td><td>Mild</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D12</td><td>Overcast</td><td>Mild</td><td>High</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D13</td><td>Overcast</td><td>Hot</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D14</td><td>Rain</td><td>Mild</td><td>High</td><td>Strong</td><td>No</td></tr>
</table>
</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design Bayesian belief network with its representation</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
<tr><td>(c)</td><td>Identify Sample Error and True Error in hypothesis</td><td>6</td><td>CO5</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>2(a)</td><td>Determine conditional independence in Bayesian belief network</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design EM Algorithm and derive equations for the algorithm</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
</table>

(c)	**Identify** confidence interval for discrete valued hypothesis	6	CO5	Applying(K3)
PART-B				
3(a)	**Make use of** suitable example, explain the Reinforcement Learning	6	CO5	Applying(K3)
(b)	**Determine** Comparing learning algorithms with paired t Tests.	6	CO5	Applying(K3)
OR				
4(a)	**Design** Q Learning (Q Function and Algorithm)	6	CO5	Applying(K3)
(b)	**Determine** Binomial Distribution for sampling theory	6	CO5	Applying(K3)

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
III SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION (SET A)

Degree	:	**B.E**	**Semester**	:	**VII**
Branch	:	**CSE**	**Course Code**	:	**17CS73**
Course Title	:	**Machine Learning**	**Max Marks**	:	**30**

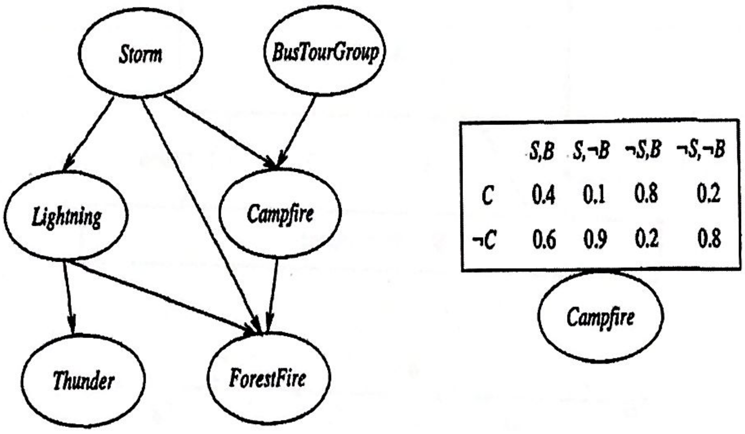
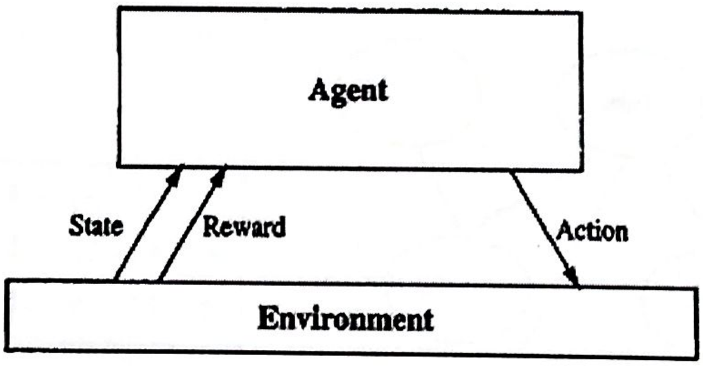
<table>
<tr><th>Q.NO.</th><th>POINTS</th><th>MARKS</th></tr>
<tr><td>1(a)</td><td>$P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$
$P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0206$</td><td>2M + 4M</td></tr>
<tr><td>(b)</td><td>

Diagram and explanation</td><td>2M + 4M</td></tr>
<tr><td>(c)</td><td>Sample Error : Definition and Explanation
True Error : Definition and Explanation</td><td>3 M
3 M</td></tr>
<tr><td>2(a)</td><td>Conditional independence in Bayesian belief network
Definition and Explanation</td><td>2M + 4M</td></tr>
</table>

(b)	**Step 1:** ***Estimation (E) step:*** Calculate $Q(h'\|h)$ using the current hypothesis h and the observed data X to estimate the probability distribution over Y. $Q(h'\|h) \leftarrow E[\ln P(Y\|h')\|h, X]$ **Step 2:** ***Maximization (M) step:*** Replace hypothesis h by the hypothesis h' that maximizes this Q function. $h \leftarrow \underset{h'}{\operatorname{argmax}}\, Q(h'\|h)$ Algorithm and Explanation	2M + 4M
(c)	**Confidence interval for discrete valued hypothesis** Formula and Explanation	4+2=6M
3(a)	 $s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \ldots$ **Goal: Learn to choose actions that maximize** $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$, where $0 \leq \gamma < 1$ Diagram and explanation	3+3=6M
(b)	**Comparing learning algorithms with paired t Tests** Comparison and Explanation	3+3=6M

4(a)	**Q learning algorithm** For each s, a initialize the table entry $\hat{Q}(s, a)$ to zero. Observe the current state s Do forever: • Select an action a and execute it • Receive immediate reward r • Observe the new state s' • Update the table entry for $\hat{Q}(s, a)$ as follows: $$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$ • $s \leftarrow s'$ Algorithm and Explanation	3+3=6M
(b)	**Binomial Distribution for sampling theory** Points and Explanation	3+3=6M

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

SET B

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109

III SESSIONAL TEST QUESTION PAPER 2020 ODD SEMESTER

USN										

Degree : **B.E**
Branch : **Computer Science and Engineering**
Course Title : **Machine Learning**
Duration : **90 Minutes**

Semester : **VII**
Course Code : **17CS73**
Date : **03-02-2021**
Max Marks : **30**

Note: Answer ONE full question from each part.

<table>
<tr><th>Q No.</th><th>Question</th><th>Marks</th><th>CO mapping</th><th>K-Level</th></tr>
<tr><td colspan="5">PART-A</td></tr>
<tr><td>1(a)</td><td>Design EM Algorithm and derive equations for the algorithm</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Determine conditional independence in Bayesian belief network</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
<tr><td>(c)</td><td>Determine estimating hypothesis accuracy</td><td>6</td><td>CO5</td><td>Applying(K3)</td></tr>
<tr><td colspan="5">OR</td></tr>
<tr><td>2(a)</td><td>Apply naïve bayes classifier for the given training data to classify the instance
(outlook = sunny, temperature = cool, humidity = high, wind = strong)
<table>
<tr><th><i>Day</i></th><th><i>Outlook</i></th><th><i>Temperature</i></th><th><i>Humidity</i></th><th><i>Wind</i></th><th><i>PlayTennis</i></th></tr>
<tr><td>D1</td><td>Sunny</td><td>Hot</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D2</td><td>Sunny</td><td>Hot</td><td>High</td><td>Strong</td><td>No</td></tr>
<tr><td>D3</td><td>Overcast</td><td>Hot</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D4</td><td>Rain</td><td>Mild</td><td>High</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D5</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D6</td><td>Rain</td><td>Cool</td><td>Normal</td><td>Strong</td><td>No</td></tr>
<tr><td>D7</td><td>Overcast</td><td>Cool</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D8</td><td>Sunny</td><td>Mild</td><td>High</td><td>Weak</td><td>No</td></tr>
<tr><td>D9</td><td>Sunny</td><td>Cool</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D10</td><td>Rain</td><td>Mild</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D11</td><td>Sunny</td><td>Mild</td><td>Normal</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D12</td><td>Overcast</td><td>Mild</td><td>High</td><td>Strong</td><td>Yes</td></tr>
<tr><td>D13</td><td>Overcast</td><td>Hot</td><td>Normal</td><td>Weak</td><td>Yes</td></tr>
<tr><td>D14</td><td>Rain</td><td>Mild</td><td>High</td><td>Strong</td><td>No</td></tr>
</table>
</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
<tr><td>(b)</td><td>Design Bayesian belief network with its representation</td><td>6</td><td>CO4</td><td>Applying(K3)</td></tr>
</table>

(c)	**Identify** confidence interval for discrete valued hypothesis	6	CO5	Applying(K3)
PART-B				
3(a)	**Design** Q Learning (Q Function and Algorithm)	6	CO5	Applying(K3)
(b)	**Determine** central limit theorem	6	CO5	Applying(K3)
OR				
4(a)	**Design** K means algorithm with its derivations	6	CO5	Applying(K3)
(b)	**Make use of** suitable example, explain the Reinforcement Learning	6	CO5	Applying(K3)

Signature of course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2020 – 21 ODD SEMESTER

SCHEME AND SOLUTION

(SET B)

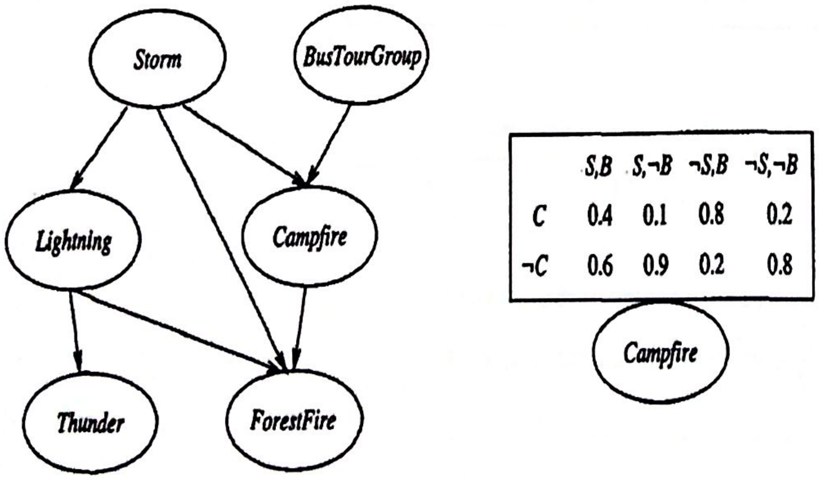
Degree	:	**B.E**	**Semester**	:	**VII**
Branch	:	**CSE**	**Course Code**	:	**17CS73**
Course Title	:	**Machine Learning**	**Max Marks**	:	**30**

<table>
<tr><th>Q.NO.</th><th>POINTS</th><th>MARKS</th></tr>
<tr><td>1(a)</td><td>EM algorithm explanation with steps.
Explanation of each equation</td><td>3M + 3M</td></tr>
<tr><td>(b)</td><td>Conditional independence in Bayesian belief network
Definition and Explanation</td><td>2M + 4M</td></tr>
<tr><td>(c)</td><td>Mentioning of equations to achieve accuracy
Explanation of each term and procedure to achieve accuracy</td><td></td></tr>
<tr><td>2 (a)</td><td>

$$P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

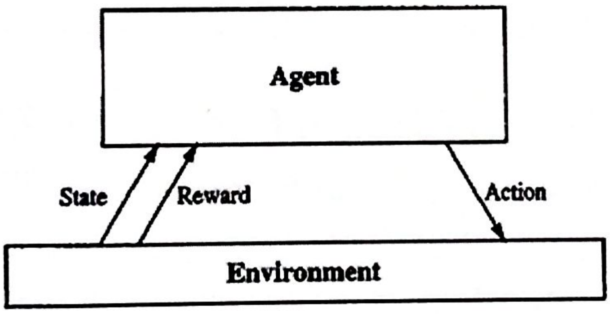
$$P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0206$$

</td><td>2M + 4M</td></tr>
<tr><td>(b)</td><td>



	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

</td><td>2M + 4M</td></tr>
<tr><td>(c)</td><td>**Confidence interval for discrete valued hypothesis**
Formula and Explanation</td><td>4+2=6M</td></tr>
</table>

3(a)	**Q learning algorithm** **For each s, a initialize the table entry $\hat{Q}(s, a)$ to zero.** **Observe the current state s** **Do forever:** • **Select an action a and execute it** • **Receive immediate reward r** • **Observe the new state s'** • **Update the table entry for $\hat{Q}(s, a)$ as follows:** $$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$ • $s \leftarrow s'$	3+3=6M
(b)	Mentioning of central limit theorem Explanation of steps	3+3=6M
4 (a)	Mentioning of K mean algorithm with its steps Explanation of algorithm	
4(b)	 $$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \dots$$ **Goal: Learn to choose actions that maximize** $r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$, where $0 \leq \gamma < 1$	3+3=6M

Signature of Course in charge **Signature of Module Coordinator** **Signature of HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	IV / VII
COURSE TITLE	Machine Learning
COURSE CODE	17CS73
ACADEMIC YEAR	2020-2021
INTERNALS	I

Sl. NO .	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
1	1KS17CS001	AAFREEN HUSSAIN				5	6	6	6	5			28
2	1KS17CS002	ABHISHEK GOWDA.M.V	6	6	6				6	6			30
3	1KS17CS003	AKSHATHA RAMESH	6	6	6				6	5			29
4	1KS17CS004	AKSHITHA.B.S	6	6	6				6	6			30
5	1KS17CS005	AMOGH.R	5	6	6				6	6			29
6	1KS17CS006	AMOGHA MANJUNATHA.K	4	6	6				6	6			28
7	1KS17CS007	AMRUTHA.V.DESHPANDE	6	6	6				6	6			30
8	1KS17CS008	ANOOP.P.S				5	6	6	6	6			29
9	1KS17CS010	ANUSHA.A.G				5	6	6	6	6			29
10	1KS17CS011	ANUSHREE.J	6	6	6				6	6			30
11	1KS17CS013	ASHISH.K.AMAR	4	6	6				6	6			28
12	1KS17CS014	LAKSHMI PRASANNA.B	6	6	6			6	6	6			30
13	1KS17CS016	BHAVESH BHANSALI				5	6	6	6	6			29
14	1KS17CS017	CHAITRA				6	5	6	6	6			29
15	1KS17CS018	CHANDANA.B.R	6	6	6				6	6			30
16	1KS17CS019	CHENNA KESHAVA.N.T				5	6	6	6	6			29

Sl. NO .	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
17	1KS17CS020	DARSHAN.S				6	6	6	6	6			30
18	1KS17CS021	DEEKSHITHA.R				5	5	6	6	5			27
19	1KS17CS022	DEEPIKA.S.H	6	6	6				6	6			30
20	1KS17CS023	DIVYA YASHASWI KANNEY	6	6	6				6	6			30
21	1KS17CS024	GANESH.G.B	6	6	6				6	5			29
22	1KS17CS025	GANESH MAUDGHALYA.H.G				6	5	6	4	5			26
23	1KS17CS026	GAUTHAM.C.R	5	6	6				4	5			26
24	1KS17CS027	H.PRIYANKA	6	6	6				6	6			30
25	1KS17CS028	HANUMESH.V.T				5	5	6	6	6			28
26	1KS17CS029	HARSHITHA.V	6	6	6				6	6			30
27	1KS17CS030	INDRASENA KALYANAM				5	6	6	6	6			29
28	1KS17CS032	KARAN RAGHUNATH				5	5	6	6	5			27
29	1KS17CS033	KARTHIK.T.C	6	6	6				6	6			30
30	1KS17CS034	KAVITHA.S	6	6	6				6	6			30
31	1KS17CS035	KEERTHI.N	6	6	6				6	6			30
32	1KS17CS036	KRITHIKA JAGANNATH	6	6	6				6	6			30
33	1KS17CS037	LAVANYA.V	6	6	3				6	6			27
34	1KS17CS038	LOKESH.B.M	4	6	6				6	6			28
35	1KS17CS040	MANJUNATH.A				5	6	6	6	5			28
36	1KS17CS041	MEGHANA.C.V	6	6	6				6	6			30
37	1KS17CS042	MEGHANA.G	6	6	6				6	6			30
38	1KS17CS043	MEGHANA.G.R	6	2	6				6	6			26
39	1KS17CS044	MOUNIKA.M.K.L	6	6	6				6	6			30
40	1KS17CS045	NEHA.K				5	6	6	6	6			29
41	1KS17CS046	NIKHIL SUBRAMANYA.K				5	5	6	6	6			28
42	1KS17CS047	NIKITHA KATARI	6	6	6				6	5			29
43	1KS17CS048	NISCHITHA.C	6	6	6				6	6			30
44	1KS17CS049	NITISH KUMAR GUPTA	4	6	2				6	5			23
45	1KS17CS050	NYDILE.G.R	6	6	6				6	6			30
46	1KS17CS051	P.KISHORE				5	6	6	6	6			29

SI. NO .	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO1	CO1	CO1	CO1	CO1	CO1	CO2	CO2	CO2	CO2	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
47	1KS17CS102	SHRIRAKSHA.S.KANAGO	6	5	6				6	6			29
48	1KS18CS401	KRUTHIKA.B.M	6	5	6				6	5			28
49	1KS16CS042	MEGHANA.H.S				5	6	6	6	5			28
50	1KS15CS050	LAXMI.K.V	6	6	6					2			20

Course incharge

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	**IV / VII**
COURSE TITLE	**Machine Learning**
COURSE CODE	**17CS73**
ACADEMIC YEAR	**2020-2021**
INTERNALS	**II**

Sl. NO.	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
1	1KS17CS001	AAFREEN HUSSAIN	6	6	5				6	5			28
2	1KS17CS002	ABHISHEK GOWDA.M.V	6	6	5				6	5			28
3	1KS17CS003	AKSHATHA RAMESH	6	6	5				6	5			28
4	1KS17CS004	AKSHITHA.B.S	6	6	5						5	6	28
5	1KS17CS005	AMOGH.R	5	6	5				6	5			28
6	1KS17CS006	AMOGHA MANJUNATHA.K	6	6	5				6	5			28
7	1KS17CS007	AMRUTHA.V.DESHPANDE	6	6	5				6	5			28
8	1KS17CS008	ANOOP.P.S	6	6	6				6	5			29
9	1KS17CS010	ANUSHA.A.G	6	6	5				6	5			28
10	1KS17CS011	ANUSHREE.J	6	6	5				6	5			28
11	1KS17CS013	ASHISH.K.AMAR	6	6	5				5	5			27
12	1KS17CS014	LAKSHMI PRASANNA.B	6	6	5				6	6			29
13	1KS17CS016	BHAVESH BHANSALI	6	6	5				6	5			28
14	1KS17CS017	CHAITRA	6	6	5				6	5			28
15	1KS17CS018	CHANDANA.B.R				6	5	6	6	5			28
16	1KS17CS019	CHENNA KESHAVA.N.T	6	6	5				6	5			28

Sl. NO.	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
17	1KS17CS020	DARSHAN.S				5	5	5	6	5			26
18	1KS17CS021	DEEKSHITHA.R	6	6	5						6	6	29
19	1KS17CS022	DEEPIKA.S.H	6	6	5				6	5			28
20	1KS17CS023	DIVYA YASHASWI KANNEY	6	6	5				6	5			28
21	1KS17CS024	GANESH.G.B	6	6	5				6	5			28
22	1KS17CS025	GANESH MAUDGHALYA.H.G	6	6	5				6	5			28
23	1KS17CS026	GAUTHAM.C.R				6	5	5	6	5			27
24	1KS17CS027	H.PRIYANKA				6	5	5	6	5			27
25	1KS17CS028	HANUMESH.V.T	6	6	5				6	5			28
26	1KS17CS029	HARSHITHA.V	6	6	5				6	5			28
27	1KS17CS030	INDRASENA KALYANAM	6	6	6				6	5			29
28	1KS17CS032	KARAN RAGHUNATH	6	6	5				6	5			28
29	1KS17CS033	KARTHIK.T.C	6	6	5						5	5	27
30	1KS17CS034	KAVITHA.S	6	6	5						6	5	28
31	1KS17CS035	KEERTHI.N	6	6	5				6	5			28
32	1KS17CS036	KRITHIKA JAGANNATH	6	6	5						6	5	28
33	1KS17CS037	LAVANYA.V	6	6	5				6	5			28
34	1KS17CS038	LOKESH.B.M	6	6	5				6	5			28
35	1KS17CS040	MANJUNATH.A	6	6	5				6	5			28
36	1KS17CS041	MEGHANA.C.V	6	6	5				6	5			28
37	1KS17CS042	MEGHANA.G	6	6	5				6	5			28
38	1KS17CS043	MEGHANA.G.R	6	6	5				6	5	6	5	28
39	1KS17CS044	MOUNIKA.M.K.L	6	6	5				6	5			28
40	1KS17CS045	NEHA.K				6	5	6	6	5			28
41	1KS17CS046	NIKHIL SUBRAMANYA.K	6	6	5				6	5			28
42	1KS17CS047	NIKITHA KATARI	5	6	5						5	5	26
43	1KS17CS048	NISCHITHA.C	6	6	5				6	5			28
44	1KS17CS049	NITISH KUMAR GUPTA	6	6	4				6	5			27
45	1KS17CS050	NYDILE.G.R	6	6	5				6	5			28
46	1KS17CS051	P.KISHORE	6	6	5				6	5			28

Sl. NO.	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO3	CO3	CO3	CO3	CO3	CO3	CO2	CO4	CO2	CO4	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
47	1KS17CS102	SHRIRAKSHA.S.KANAGO	6	6	5				6	5			28
48	1KS18CS401	KRUTHIKA.B.M	6	6	5				6	5			28
49	1KS16CS042	MEGHANA.H.S	6	6	5				6	5			28
50	1KS15CS050	LAXMI.K.V	6	6	5				6	5			28
51	1KS16CS090	SHASHANK KAVUR	6	6	5				6	5			28

Course In-Charge

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	IV / VII
COURSE TITLE	Machine Learning
COURSE CODE	17CS73
ACADEMIC YEAR	2020-2021
INTERNALS	III

Sl. NO .	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
1	1KS17CS001	AAFREEN HUSSAIN	6	1	6				0				13
2	1KS17CS002	ABHISHEK GOWDA.M.V	6	6	6						6	6	30
3	1KS17CS003	AKSHATHA RAMESH	6	6	6						6		24
4	1KS17CS004	AKSHITHA.B.S				6	6		6				18
5	1KS17CS005	AMOGH.R	6	6	6						6	2	26
6	1KS17CS006	AMOGHA MANJUNATHA.K	6	6	6						6	6	30
7	1KS17CS007	AMRUTHA.V.DESHPANDE	6	6	6							6	24
8	1KS17CS008	ANOOP.P.S	6	6	6						6		24
9	1KS17CS010	ANUSHA.A.G	6	6					6				18
10	1KS17CS011	ANUSHREE.J				6	6		6				18
11	1KS17CS013	ASHISH.K.AMAR	6	6	6						6	6	30
12	1KS17CS014	LAKSHMI PRASANNA.B	6	6	6						3	6	27
13	1KS17CS016	BHAVESH BHANSALI	6	6					6				18
14	1KS17CS017	CHAITRA				6	6	6			3	6	27
15	1KS17CS018	CHANDANA.B.R				6	6	6			6	6	30
16	1KS17CS019	CHENNA KESHAVA.N.T	6		6				6	6			24

Sl. NO .	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
17	1KS17CS020	DARSHAN.S	6						6				12
18	1KS17CS021	DEEKSHITHA.R	6	6					6		1		18
19	1KS17CS022	DEEPIKA.S.H	6	6	6						6	4	28
20	1KS17CS023	DIVYA YASHASWI KANNEY	ABSENT										
21	1KS17CS024	GANESH.G.B	6	6					6				18
22	1KS17CS025	GANESH MAUDGHALYA.H.G				6	6	6				6	24
23	1KS17CS026	GAUTHAM.C.R				6	6	3			3		18
24	1KS17CS027	H.PRIYANKA	6	6	6				6		6	6	30
25	1KS17CS028	HANUMESH.V.T	6	6	1						6	6	25
26	1KS17CS029	HARSHITHA.V	6	6	6				6	1			25
27	1KS17CS030	INDRASENA KALYANAM	6	6	6						4	6	28
28	1KS17CS032	KARAN RAGHUNATH	6	6	6				6				24
29	1KS17CS033	KARTHIK.T.C				6	6				6	6	24
30	1KS17CS034	KAVITHA.S	6	6	6							6	24
31	1KS17CS035	KEERTHI.N	6	6	6							6	24
32	1KS17CS036	KRITHIKA JAGANNATH	6	6	6				6	6			30
33	1KS17CS037	LAVANYA.V	6	6	6						4	6	28
34	1KS17CS038	LOKESH.B.M	6	4	4						4	1	19
35	1KS17CS040	MANJUNATH.A	6	6							6	6	24
36	1KS17CS041	MEGHANA.C.V	6	6	6				6				24
37	1KS17CS042	MEGHANA.G	6	6	6						6	6	30
38	1KS17CS043	MEGHANA.G.R	6	6	6				6	6			30
39	1KS17CS044	MOUNIKA.M.K.L				6	6		6				18
40	1KS17CS045	NEHA.K	6	6					6				18
41	1KS17CS046	NIKHIL SUBRAMANYA.K	6	6	6						6	2	26
42	1KS17CS047	NIKITHA KATARI	6	5	5				6	6			28
43	1KS17CS048	NISCHITHA.C				6	6		6				18
44	1KS17CS049	NITISH KUMAR GUPTA				6	6	6			3	6	27
45	1KS17CS050	NYDILE.G.R				6	6		6				18
46	1KS17CS051	P.KISHORE				6	6	0			6	6	24

Sl. NO.	USN	NAME	Q.N.1 (a)	Q.N.1(b)	Q.N.1 (c)	Q.N.2 (a)	Q.N.2(b)	Q.N.2 (c)	Q.N.3(a)	Q.N.3(b)	Q.N.4(a)	Q.N.4(b)	TOTAL
			CO4	CO4	CO5	CO4	CO4	CO5	CO5	CO5	CO5	CO5	
			6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	6 Marks	30 Marks
47	1KS17CS102	SHRIRAKSHA.S.KANAGO	6	6	6				6	1			25
48	1KS18CS401	KRUTHIKA.B.M	ABSENT										
49	1KS16CS042	MEGHANA.H.S				6	6					6	18
50	1KS15CS050	LAXMI.K.V	ABSENT										
51	1KS16CS090	SHASHANK KAVUR	ABSENT										

Course incharge

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU-560 109
DEPARTMENT OF COMPUTER SCIENCE & ENGG.
FOR THE ACADEMIC YEAR 2020-2021
ML (15/17CS73) – IA Marks Detail

Sl.No	USN	Name of the Student	First Test Marks (30 M)	Second Test Marks (30 M)	Third Test Marks (30 M)	Final Avg Marks (30 M)	Assignment Marks (10 M)	Final Marks (40 M)
1	1KS17CS001	AAFREEN HUSSAIN	28	28	13	23	10	33
2	1KS17CS002	ABHISHEK GOWDA.M.V	30	28	30	30	10	40
3	1KS17CS003	AKSHATHA RAMESH	29	28	24	27	10	37
4	1KS17CS004	AKSHITHA.B.S	30	28	18	26	10	36
5	1KS17CS005	AMOGH.R	29	28	26	28	10	38
6	1KS17CS006	AMOGHA MANJUNATHA.K	28	28	30	29	10	39
7	1KS17CS007	AMRUTHA.V.DESHPANDE	30	28	24	28	10	38
8	1KS17CS008	ANOOP.P.S	29	29	24	28	10	38
9	1KS17CS010	ANUSHA.A.G	29	28	18	25	10	35
10	1KS17CS011	ANUSHREE.J	30	28	18	26	10	36
11	1KS17CS013	ASHISH.K.AMAR	28	27	30	29	10	39
12	1KS17CS014	LAKSHMI PRASANNA.B	30	29	27	29	10	39
13	1KS17CS016	BHAVESH BHANSALI	29	28	18	25	10	35
14	1KS17CS017	CHAITRA	29	28	27	28	10	38
15	1KS17CS018	CHANDANA.B.R	30	28	30	30	10	40
16	1KS17CS019	CHENNA KESHAVA.N.T	29	28	24	27	10	37
17	1KS17CS020	DARSHAN.S	30	26	12	23	10	33
18	1KS17CS021	DEEKSHITHA.R	27	29	18	25	10	35
19	1KS17CS022	DEEPIKA.S.H	30	28	28	29	10	39
20	1KS17CS023	DIVYA YASHASWI KANNEY	30	28	AB	20	10	30
21	1KS17CS024	GANESH.G.B	29	28	18	25	10	35
22	1KS17CS025	GANESH MAUDGHALYA.H.G	26	28	24	26	10	36
23	1KS17CS026	GAUTHAM.C.R	26	27	18	24	10	34
24	1KS17CS027	H.PRIYANKA	30	27	30	29	10	39
25	1KS17CS028	HANUMESH.V.T	28	28	25	27	10	37
26	1KS17CS029	HARSHITHA.V	30	28	25	28	10	38
27	1KS17CS030	INDRASENA KALYANAM	29	29	28	29	10	39
28	1KS17CS032	KARAN RAGHUNATH	27	28	24	27	10	37
29	1KS17CS033	KARTHIK.T.C	30	27	24	27	10	37
30	1KS17CS034	KAVITHA.S	30	28	24	28	10	38
31	1KS17CS035	KEERTHI.N	30	28	24	30	10	40
32	1KS17CS036	KRITHIKA JAGANNATH	30	28	30	30	10	40
33	1KS17CS037	LAVANYA.V	27	28	28	28	10	38
34	1KS17CS038	LOKESH.B.M	28	28	19	25	10	35
35	1KS17CS040	MANJUNATH.A	28	28	24	27	10	37
36	1KS17CS041	MEGHANA.C.V	30	28	24	28	10	38
37	1KS17CS042	MEGHANA.G	30	28	30	30	10	40
38	1KS17CS043	MEGHANA.G.R	26	28	30	28	10	38
39	1KS17CS044	MOUNIKA.M.K.L	30	28	18	26	10	36
40	1KS17CS045	NEHA.K	29	28	18	25	10	35
41	1KS17CS046	NIKHIL SUBRAMANYA.K	28	28	26	28	10	38
42	1KS17CS047	NIKITHA KATARI	29	26	28	28	10	38
43	1KS17CS048	NISCHITHA.C	30	28	18	26	10	36
44	1KS17CS049	NITISH KUMAR GUPTA	23	27	27	26	10	36
45	1KS17CS050	NYDILE.G.R	30	28	18	26	10	36
46	1KS17CS051	P.KISHORE	29	28	24	27	10	37
47	1KS17CS102	SHRIRAKSHA.S.KANAGO	29	28	25	28	10	38

48	1KS18CS401	KRUTHIKA.B.M	28	28	AB	19	10	29
49	1KS16CS042	MEGHANA.H.S	28	28	18	25	10	35
50	1KS15CS050	LAXMI.K.V	10(15)	14 (15)	AB	12	05 (05)	17(20)
51	1KS16CS090	SHASHANK KAVUR	AB	15 (15)	AB	08	05 (05)	13(20)

Course incharge

Visvesvaraya Technological University

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE

Branch : CS Scheme : 2017 Semester : 7

Sl NO.	USN	17CS71	17CS72	17CS73	17CS743	17CS754	17CSL76	17CSL77	17CSP78	STUDENT SIGNATURE
1	1KS16CS042	34	35	35	33	31	37	36	88	Meghana
2	1KS16CS110	35	38	35	36	36	40	40	88	
3	1KS17CS001	38	37	33	38	35	39	40	91	Afrees
4	1KS17CS002	40	35	40	39	39	38	40	94	
5	1KS17CS003	38	35	37	37	38	40	40	89	
6	1KS17CS004	38	34	36	38	34	38	40	90	
7	1KS17CS005	40	33	38	36	35	40	40	89	
8	1KS17CS006	38	37	39	39	40	39	40	97	
9	1KS17CS007	37	37	38	35	35	40	40	90	
10	1KS17CS008	39	36	38	37	35	38	40	90	PJ
11	1KS17CS010	40	34	35	36	36	38	40	91	
12	1KS17CS011	39	37	36	36	31	40	40	88	Anushree I
13	1KS17CS013	40	34	39	39	36	40	40	94	
14	1KS17CS014	40	37	39	38	37	40	39	91	
15	1KS17CS016	39	34	35	34	34	40	40	92	Bharath
16	1KS17CS017	39	36	38	37	36	40	40	91	Chintan
17	1KS17CS018	40	35	40	38	40	40	40	95	Chandana B.R
18	1KS17CS019	38	38	37	38	36	39	40	95	NT
19	1KS17CS020	35	33	33	32	33	40	39	88	
20	1KS17CS021	35	36	35	35	39	39	39	91	Deekshita
21	1KS17CS022	37	35	39	35	35	40	39	91	Deepika
22	1KS17CS023	30	27	30	28	29	38	40	91	
23	1KS17CS024	37	33	35	33	33	40	37	88	
24	1KS17CS025	35	34	36	34	34	39	40	92	
25	1KS17CS026	40	32	34	36	36	38	40	88	CR
26	1KS17CS027	40	37	39	39	39	38	40	89	
27	1KS17CS028	39	33	37	33	32	38	34	87	
28	1KS17CS029	38	35	38	36	37	38	40	94	
29	1KS17CS030	40	38	39	40	40	40	40	95	
30	1KS17CS032	36	35	37	32	34	40	40	92	
31	1KS17CS033	37	33	37	38	38	40	40	90	
32	1KS17CS034	37	35	38	37	37	38	37	85	Karthik
33	1KS17CS035	39	35	40	36	35	38	37	89	
34	1KS17CS036	39	37	40	40	40	40	40	95	
35	1KS17CS037	39	36	38	33	37	40	37	91	Lavanya
36	1KS17CS038	37	34	35	35	33	40	39	94	
37	1KS17CS040	38	30	37	34	33	38	40	90	

	USN	17CS71	17CS72	17CS73	17CS743	17CS754	17CSL76	17CSL77	17CSP78	STUDENT SIGNATURE
9	1KS17CS042	40	37	40	39	40	40	40	94	
40	1KS17CS043	40	37	38	39	39	40	40	89	
41	1KS17CS044	39	36	36	35	38	39	40	91	Mounika
42	1KS17CS045	38	34	35	36	33	40	37	91	Neha K.
43	1KS17CS046	40	34	38	36	32	39	35	89	Nikhil
44	1KS17CS047	37	36	38	33	34	38	37	94	
45	1KS17CS048	36	32	36	29	32	38	40	88	
46	1KS17CS049	36	36	36	37	33	40	37	87	
47	1KS17CS050	40	36	36	35	34	40	40	88	
48	1KS17CS051	37	35	37	35	34	40	40	92	
49	1KS17CS052	40	40	37	40	38	40	40	90	
50	1KS17CS053	36	40	39	38	39	40	39	91	
51	1KS17CS055	38	40	40	39	40	40	40	89	R. Priya
52	1KS17CS056	36	36	34	32	25	40	38	89	
53	1KS17CS057	33	34	29	33	37	40	40	84	Praveen
54	1KS17CS058	38	35	30	35	37	40	40	89	Praveen
55	1KS17CS059	34	32	37	32	32	37	34	84	
56	1KS17CS060	36	38	37	34	39	40	39	88	
57	1KS17CS061	34	37	34	36	34	36	37	89	
58	1KS17CS062	38	34	37	35	30	36	37	89	
59	1KS17CS063	35	35	34	32	33	40	36	88	
60	1KS17CS064	38	40	38	37	39	40	40	94	
61	1KS17CS065	38	38	39	34	38	40	40	88	Roshini R.
62	1KS17CS066	37	40	38	39	39	40	40	90	Ruchi
63	1KS17CS067	37	40	35	36	33	40	40	87	
64	1KS17CS069	36	36	36	35	36	39	36	84	
65	1KS17CS070	40	40	40	38	40	40	40	93	
66	1KS17CS071	36	38	38	38	36	40	40	97	
67	1KS17CS072	38	36	38	34	35	38	36	84	Sakshi
8	1KS17CS074	26	29	24	27	27	38	39	88	
69	1KS17CS075	40	39	40	39	38	39	40	90	
70	1KS17CS076	37	34	36	36	33	40	40	93	Shreya
71	1KS17CS077	34	30	30	33	30	38	35	89	
72	1KS17CS078	36	40	40	40	38	40	39	88	Sindhu
73	1KS17CS079	37	40	36	36	33	40	40	88	Sindhu
74	1KS17CS081	39	40	38	38	37	39	40	90	Spoorthi K
75	1KS17CS082	30	30	30	30	30	39	40	93	Spoorthi. V
76	1KS17CS083	40	40	39	38	38	38	40	97	S. Kala
77	1KS17CS084	39	40	38	38	35	40	40	89	Srushti A
78	1KS17CS085	37	38	40	34	39	39	40	87	
79	1KS17CS086	40	38	39	38	36	39	40	92	
80	1KS17CS087	38	36	40	36	37	39	39	87	K. Supriya
81	1KS17CS088	40	36	38	39	37	40	40	87	
82	1KS17CS089	40	40	40	38	38	40	40	97	
83	1KS17CS090	40	37	36	37	37	40	40	88	

.	USN	17CS71	17CS72	17CS73	17CS743	17CS754	17CSL76	17CSL77	17CSP78	STUDENT SIGNATURE
85	1KS17CS092	40	40	40	40	40	40	40	89	
86	1KS17CS093	36	37	35	36	32	39	37	88	
87	1KS17CS094	29	31	33	29	25	40	37	87	
88	1KS17CS095	36	40	38	36	35	40	40	87	
89	1KS17CS096	39	37	40	36	34	40	40	90	
90	1KS17CS097	35	37	37	34	34	40	37	84	
91	1KS17CS098	27	35	35	33	35	40	37	84	
92	1KS17CS099	35	34	34	33	31	40	40	84	
93	1KS17CS100	40	35	36	36	34	40	40	93	
94	1KS17CS101	40	38	40	36	36	40	40	93	
95	1KS17CS102	39	36	38	37	34	39	36	89	
96	1KS18CS401	35	30	29	28	27	38	29	88	
--x--	Faculty Signature									----XXXXXXX----

* - values are either optional subjects or the faculty has not yet entered the marks

26/2/2021

HOD

Seal and Signature

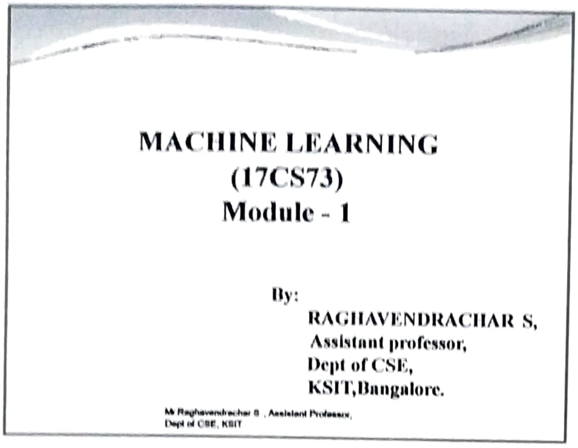
Head of the Department
ept. of Computer Science & Engg.
K.S. Institute of Technology
Bengaluru -560 109

26/2/2021

PRINCIPAL

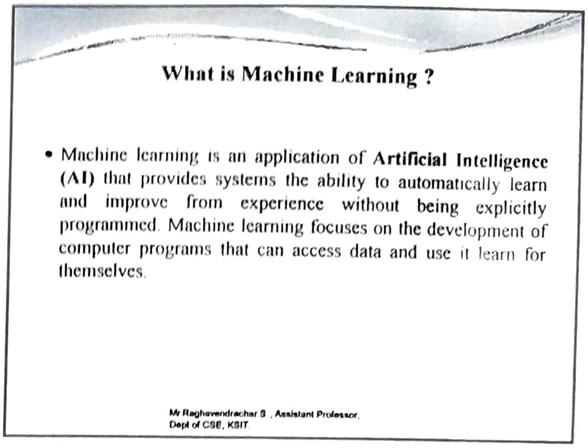
Seal and Signature

PRINCIPAL
K.S. INSTITUTE OF TECHNOLOGY
BENGALURU - 560 109



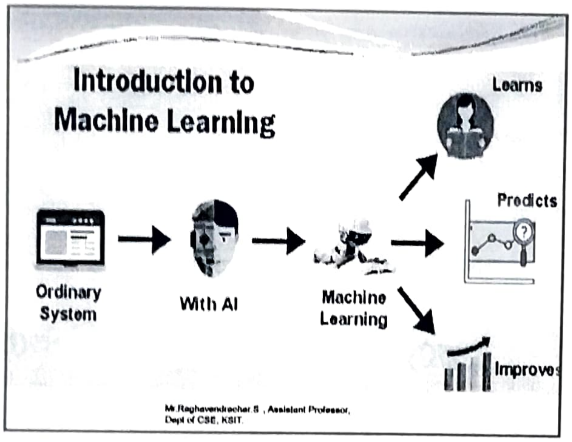
MACHINE LEARNING
(17CS73)
Module - 1

By:
RAGHAVENDRACHAR S,
Assistant professor,
Dept of CSE,
KSIT,Bangalore.

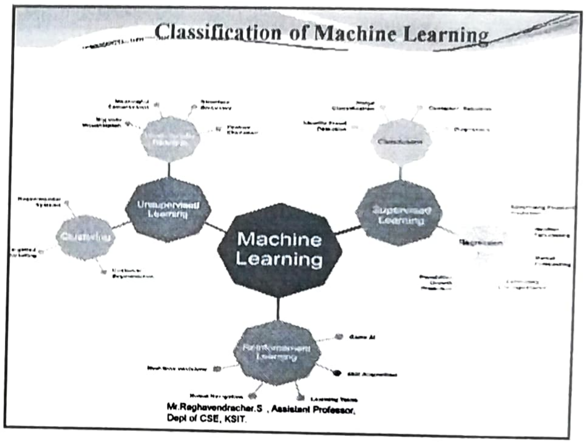


What is Machine Learning ?

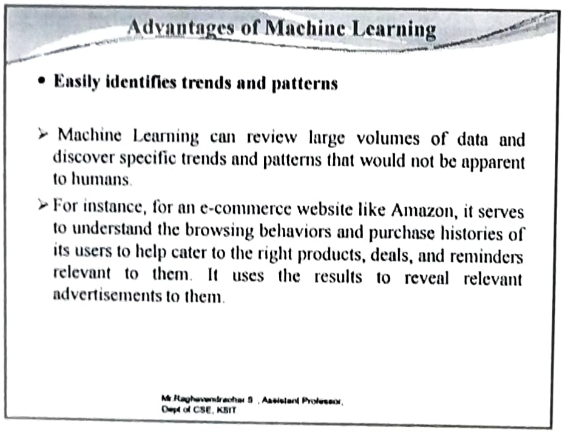
- Machine learning is an application of **Artificial Intelligence (AI)** that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.



Introduction to Machine Learning



Classification of Machine Learning

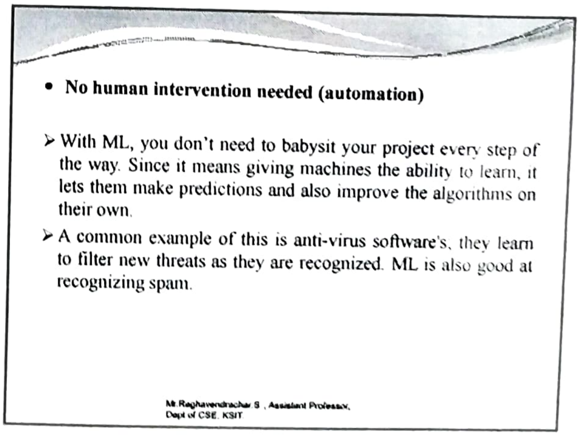


Advantages of Machine Learning

- **Easily identifies trends and patterns**

➢ Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans.

➢ For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.



- **No human intervention needed (automation)**

➢ With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own.

➢ A common example of this is anti-virus software's, they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

- **Continuous Improvement**

- As ML algorithms gain experience, they keep improving in accuracy and efficiency.
- This lets them make better decisions. Say you need to make a weather forecast model.
- As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

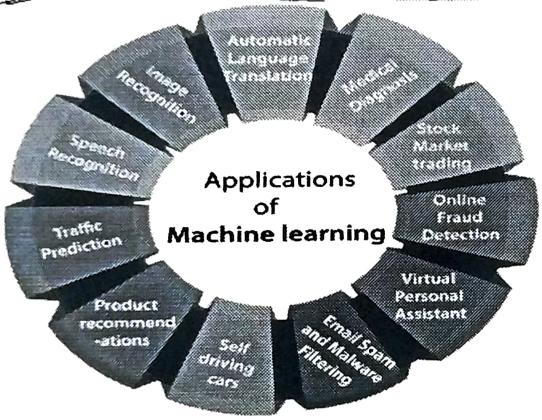
- **Handling multi-dimensional and multi-variety data**

- Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

- **Wide Applications**

- You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Applications of Machine Learning



MACHINE LEARNING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)
SEMESTER – VII

Subject Code	17CS73	IA Marks	40
Number of Lecture Hours/Week	03	Exam Marks	60
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Module – 1	Teaching Hours
Introduction: Well posed learning problems. Designing a Learning system. Perspective and Issues in Machine Learning. **Concept Learning:** Concept learning task. Concept learning as search. Find-S algorithm. Version space. Candidate Elimination algorithm. Inductive Bias. **Text Book1. Sections: 1.1 – 1.3, 2.1-2.5, 2.7**	10 Hours
Module – 2	
Decision Tree Learning: Decision tree representation. Appropriate problems for decision tree learning. Basic decision tree learning algorithm. hypothesis space search in decision tree learning. Inductive bias in decision tree learning. Issues in decision tree learning. **Text Book1. Sections: 3.1-3.7**	10 Hours
Module – 3	
Artificial Neural Networks: Introduction. Neural Network representation. Appropriate problems. Perceptrons. Backpropagation algorithm. **Text book 1, Sections: 4.1 – 4.6**	08 Hours

Module – 4	
Bayesian Learning: Introduction. Bayes theorem. Bayes theorem and concept learning. ML and LS error hypothesis. ML for predicting probabilities, MDL principle. Naive Bayes classifier. Bayesian belief networks. EM algoritlm **Text book 1, Sections: 6.1 – 6.6, 6.9, 6.11, 6.12**	10 Hours
Module – 5	
Evaluating Hypothesis: Motivation, Estimating hypothesis accuracy. Basics of sampling theorem. General approach for deriving confidence intervals. Difference in error of two hypothesis. Comparing learning algorithms. **Instance Based Learning:** Introduction. k-nearest neighbor learning. locally weighted regression. radial basis function, cased-based reasoning **Reinforcement Learning:** Introduction. Learning Task. Q Learning **Text book 1, Sections: 5.1-5.6, 8.1-8.5, 13.1-13.3**	12 Hours

Course Outcomes: After studying this course, students will be able to

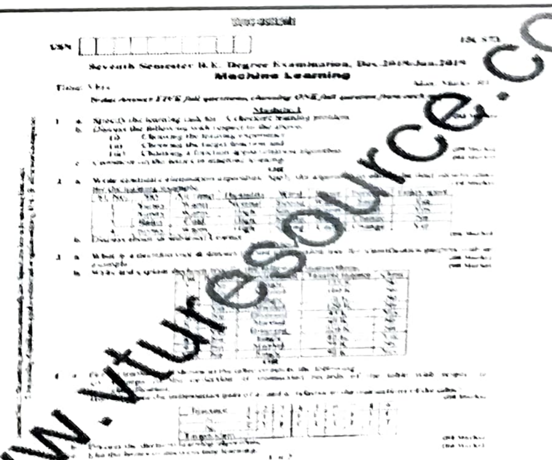
- Recall the problems for machine learning. And select the either supervised, unsupervised or reinforcement learning
- Understand theory of probability and statistics related to machine learning
- Illustrate concept learning, ANN, Bayes classifier, k nearest neighbor, Q.

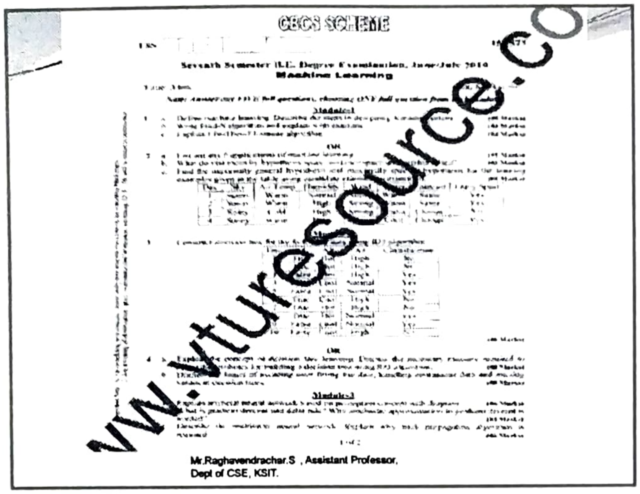
Text Books:

1. Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Education

Reference Books:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, h The Elements of Statistical Learning, 2nd edition, springer series in statistics.
2. Ethem Alpaydın, Introduction to machine learning, second edition, MIT press.

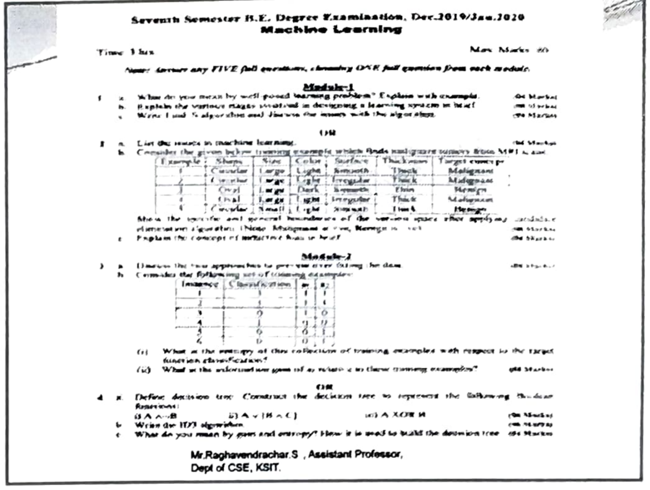

Machine Learning



CBCS SCHEME

Seventh Semester B.E. Degree Examination, June/July 2019

Machine Learning



Seventh Semester B.E. Degree Examination, Dec.2019/Jan.2020

Machine Learning

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. What do you mean by well-posed learning problem? Explain with example.

OR

Module-2

OR

Applications of Machine Learning

- Learning to recognize spoken words.

  All of the most successful speech recognition systems employ machine learning in some form. For example, the SPHINX system (e.g., Lee 1989) learns speaker-specific strategies for recognizing the primitive sounds (phonemes) and words from the observed speech signal. Neural network learning methods (e.g., Waibel et al. 1989) and methods for learning hidden Markov models (e.g., Lee 1989) are effective for automatically customizing to individual speakers, vocabularies, microphone characteristics, background noise, etc. Similar techniques have potential applications in many signal-interpretation problems.
- Learning to drive an autonomous vehicle.

  Machine learning methods have been used to train computer-controlled vehicles to steer correctly when driving on a variety of road types. For example, the ALVINN system (Pomerleau 1989) has used its learned strategies to drive unassisted at 70 miles per hour for 90 miles on public highways among other cars. Similar techniques have possible applications in many sensor-based control problems.
- Learning to classify new astronomical structures.

  Machine learning methods have been applied to a variety of large databases to learn general regularities implicit in the data. For example, decision tree learning algorithms have been used by NASA to learn how to classify celestial objects from the second Palomar Observatory Sky Survey (Fayyad et al. 1995). This system is now used to automatically classify all objects in the Sky Survey, which consists of three terrabytes of image data.
- Learning to play world-class backgammon.

  The most successful computer programs for playing games such as backgammon are based on machine learning algorithms. For example, the world's top computer program for backgammon, TD-GAMMON (Tesauro 1992, 1995), learned its strategy by playing over one million practice games against itself. It now plays at a level competitive with the human world champion. Similar techniques have applications in many practical problems where very large search spaces must be examined efficiently.

TABLE
Some successful applications of machine learning.

Well – Posed Learning Problems

- **Definition:** A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.
- **A checkers learning problem:**

1. **Task T:** playing checkers
2. **Performance measure P:** percent of games won against opponents
3. **Training experience E:** playing practice games against itself

- **A handwriting recognition learning problem:**

1. **Task T:** recognizing and classifying handwritten words within images
2. **Performance measure P:** percent of words correctly classified
3. **Training experience E:** a database of handwritten words with given classifications

- **A robot driving learning problem:**

1. **Task T:** driving on public four-lane highways using vision sensors
2. **Performance measure P:** average distance traveled before an error (as judged by human overseer)
3. **Training experience E:** a sequence of images and steering commands recorded while observing a human driver

Designing a Learning System

- Choosing the Training Experience
- Choosing the Target Function
- Choosing a Representation for the Target Function
- Choosing a Function Approximation Algorithm
- The Final Design

Choosing the Training Experience

- The type of training experience available can have a significant impact on success or failure of the learner.
- One key attribute is whether the training experience provides direct or indirect feedback regarding the choices made by the performance system.
- For example, in learning to play checkers, the system might learn from direct training example consisting of individual checkers board states and the correct move for each.
- Indirect information consisting of the move sequences and final outcomes of various games played.

- A second important attribute of the training experience is the degree to which the learner controls the sequence of training examples.
- For example, the learner might rely on the teacher to select informative board states and to provide the correct move for each.
- Alternatively, the learner might itself propose board states that it finds particularly confusing and ask the teacher for the correct move.
- Learner may have complete control over both the board states and (indirect) training classifications

---

- A third important attribute of the training experience is how well it represents the distribution of examples over which the final system performance P must be measured.
- In general, learning is most reliable when the training examples follow a distribution similar to that of future test examples.
- In our checkers learning scenario, the performance metric P is the percent of games the system wins in the world tournament.
- If its training experience E consists only of games played against itself, there is an obvious danger that this training experience might not be fully representative of the distribution of situations over which it will later be tested.

- To proceed with our design, let us decide that our system will train by playing games against itself. This has the advantage that no external trainer need be present, and it therefore allows the system to generate as much training data as time permits. We now have a fully specified learning task.
- **A checkers learning problem:**

1. **Task T:** playing checkers
2. **Performance measure P:** percent of games won in the world tournament
3. **Training experience E:** games played against itself

- In order to complete the design of the learning system, we must now choose

1. the exact type of knowledge to be, learned
2. a representation for this target knowledge
3. a learning mechanism

Choosing the Target Function

- The next design choice is to determine exactly what type of knowledge will be learned and how this will be used by the performance program
- For example, checkers-playing program that can generate the legal moves from any board state. The program needs only to learn how to choose the best move from among these legal moves
- This learning task is representative of a large class of tasks for which the legal moves that define some large search space are known a priori, but for which the best search strategy is not known.
- Many optimization problems fall into this class, such as the problems of scheduling and controlling manufacturing processes where the available manufacturing steps are well understood, but the best strategy for sequencing them is not.

- The function **ChooseMove** : **B** -> **M** to indicate that this function accepts as input any board from the set of legal board states **B** and produces as output some move from the set of legal moves **M**.
- Throughout our discussion of machine learning we will find it useful to reduce the problem of improving performance **P** at task **T** to the problem of learning some particular target function such as **ChooseMove**.
- The choice of the target function will therefore be a key design choice.

- we will find it useful to define one particular target function V among the many that produce optimal play. As we shall see, this will make it easier to design a training algorithm. Let us therefore define the target value V(b) for an arbitrary board state b in B, as follows:

1. if b is a final board state that is won, then V(b) = 100
2. if b is a final board state that is lost, then V(b) = -100
3. if b is a final board state that is drawn, then V(b) = 0
4. if b is a not a final state in the game, then V(b) = V(b'), where b' is the best final board state that can be achieved starting from b and playing optimally until the end of the game (assuming the opponent plays optimally, as well).

Choosing a Representation for the Target Function

- we must choose a representation that the learning program will use to describe the function $\hat{V}$ that it will learn.
- for any given board state, the function $\hat{V}$ will be calculated as a linear combination of the following board features.
  - x_1: the number of black pieces on the board
  - x_2: the number of red pieces on the board
  - x_3: the number of black kings on the board
  - x_4: the number of red kings on the board
  - x_5: the number of black pieces threatened by red (i.e., which can be captured on red's next turn)
  - x_6: the number of red pieces threatened by black

Thus, our learning program will represent $\hat{V}(b)$ as a linear function of the form

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

where w_0 through w_6 are numerical coefficients, or weights, to be chosen by the learning algorithm. Learned values for the weights w_1 through w_6 will determine the relative importance of the various board features in determining the value of the board, whereas the weight w_0 will provide an additive constant to the board value.

Partial design of a checkers learning program:

- Task *T*: playing checkers
- Performance measure *P*: percent of games won in the world tournament
- Training experience *E*: games played against itself
- *Target function: V:Board → ℜ*
- *Target function representation*

$$\hat{V}(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

The first three items above correspond to the specification of the learning task, whereas the final two items constitute design choices for the implementation of the learning program.

Choosing a Function Approximation Algorithm

In order to learn the target function $\hat{V}$ we require a set of training examples, each describing a specific board state b and the training value $V_{train}(b)$ for b. In other words, each training example is an ordered pair of the form $\langle b, V_{train}(b)\rangle$. For instance, the following training example describes a board state b in which black has won the game (note $x_2 = 0$ indicates that red has no remaining pieces) and for which the target function value $V_{train}(b)$ is therefore +100.

$$\langle\langle x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0\rangle, +100\rangle$$

Estimating Training Values

- According to our formulation of the learning problem, the only training information available to our learner is whether the game was eventually won or lost. On the other hand, we require training examples that assign specific scores to specific board states.

- While it is easy to assign a value to board states that correspond to the end of the game, it is less obvious how to assign training values to the more numerous intermediate board states that occur before the game's end.

Despite the ambiguity inherent in estimating training values for intermediate board states, one simple approach has been found to be surprisingly successful. This approach is to assign the training value of $V_{train}(b)$ for any intermediate board state b to be $\hat{V}(Successor(b))$, where $\hat{V}$ is the learner's current approximation to V and where $Successor(b)$ denotes the next board state following b for which it is again the program's turn to move (i.e., the board state following the program's move and the opponent's response). This rule for estimating training values can be summarized as

Rule for estimating training values.

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

Adjusting The Weights

All that remains is to specify the learning algorithm for choosing the weights w_i to best fit the set of training examples $\{\langle b, V_{train}(b)\rangle\}$. As a first step we must define what we mean by the *best fit* to the training data. One common approach is to define the best hypothesis, or set of weights, as that which minimizes the squared error E between the training values and the values predicted by the hypothesis $\hat{V}$.

$$E \equiv \sum_{\langle b, V_{train}(b)\rangle \in \, training \ examples} (V_{train}(b) - \hat{V}(b))^2$$

- Several algorithms are known for finding weights of a linear function that minimize E.

- One such algorithm is called the least mean squares, or LMS **training rule.**

- For each observed training example it adjusts the weights a small amount in the direction that reduces the error on this training example.

LMS weight update rule.

For each training example $\langle b, V_{train}(b)\rangle$

- Use the current weights to calculate $\hat{V}(b)$
- For each weight w_i, update it as

$$w_i \leftarrow w_i + \eta \ (V_{train}(b) - \hat{V}(b)) \ x_i$$

Here η is a small constant (e.g., 0.1) that moderates the size of the weight update. To get an intuitive understanding for why this weight update rule works, notice that when the error $(V_{train}(b) - \hat{V}(b))$ is zero, no weights are changed. When $(V_{train}(b) - \hat{V}(b))$ is positive (i.e., when $\hat{V}(b)$ is too low), then each weight is increased in proportion to the value of its corresponding feature. This will raise the value of $\hat{V}(b)$, reducing the error.

The Final Design

The final design of our checkers learning system can be naturally described by four distinct program modules that represent the central components in many learning systems. These four modules, summarized in Figure , are as follows:

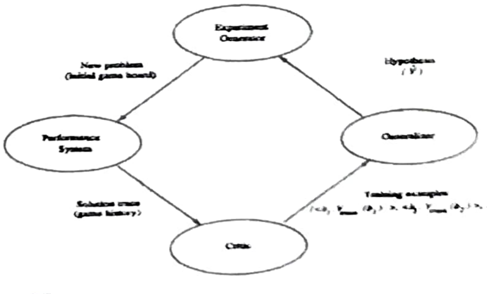


FIGURE
Final design of the checkers learning program.

Performance System

- The Performance System is the module that must solve the given performance task, in this case playing checkers, by using the learned target function(s). It takes an instance of a new problem (new game) as input and produces a trace of its solution (game history) as output.
- In our case, the strategy used by the Performance System to select its next move at each step is determined by the learned p evaluation function. Therefore, we expect its performance to improve as this evaluation function becomes increasingly accurate.

Critic

The **Critic** takes as input the history or trace of the game and produces as output a set of training examples of the target function. As shown in the diagram, each training example in this case corresponds to some game state in the trace, along with an estimate V_{train} of the target function value for this example. In our example, the Critic corresponds to the training rule given by Equation

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

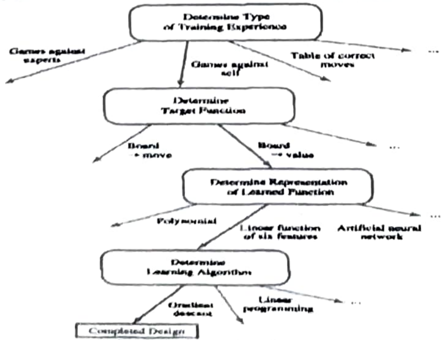
Generalizer

- The Generalizer takes as input the training examples and produces an output hypothesis that is its estimate of the target function.
- It generalizes from the specific training examples, hypothesizing a general function that covers these examples and other cases beyond the training examples.
- In our example, the Generalizer corresponds to the LMS algorithm, and the output hypothesis is the function $\hat{V}$ described by the learned weights $w_0, \ldots, w_6$.

Experiment Generator

- The Experiment Generator takes as input the current hypothesis (currently learned function) and outputs a new problem (i.e., initial board state) for the Performance System to explore.
- Its role is to pick new practice problems that will maximize the learning rate of the overall system.
- In our example, the Experiment Generator follows a very simple strategy: It always proposes the same initial game board to begin a new game.
- More sophisticated strategies could involve creating board positions designed to explore particular regions of the state space.

- The sequence of design choices made for the checkers program is summarized in Figure .



- These design choices have constrained the learning task in a number of ways.
- Furthermore, we have constrained this evaluation function to depend on only the six specific board features provided.
- If the true target function V can indeed be represented by a linear combination of these particular features, then our program has a good chance to learn it. If not, then the best we can hope for is that it will learn a good approximation, since a program can certainly never learn anything that it cannot at least represent.

Perspectives And Issues In Machine Learning

- One useful perspective on machine learning is that it involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner.

- The LMS algorithm for fitting weights achieves this goal by iteratively tuning the weights, adding a correction to each weight each time the hypothesized evaluation function predicts a value that differs from the training value.

Issues in Machine Learning

- What algorithms exist for learning general target functions from specific training examples? In what settings will particular algorithms converge to the desired function, given sufficient training data? Which algorithms perform best for which types of problems and representations?
- How much training data is sufficient? What general bounds can be found to relate the confidence in learned hypotheses to the amount of training experience and the character of the learner's hypothesis space?
- When and how can prior knowledge held by the learner guide the process of generalizing from examples? Can prior knowledge be helpful even when it is only approximately correct?
- What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy alter the complexity of the learning problem?
- What is the best way to reduce the learning task to one or more function approximation problems? Put another way, what specific functions should the system attempt to learn? Can this process itself be automated?
- How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

Concept Learning

- Inducing general functions from specific training examples is a main issue of machine learning.
- Concept Learning – a learning task in which a human or machine learner is trained to classify objects by being shown a set of example objects along with their class labels.
- The Learner simplifies what has been observed by considering it in the form of an example.
- Concept Learning – also known as category learning, concept attainment and concept formation.
- Concept Learning : Acquiring the definition of a general category from given sample of positive and negative training examples of the category.

Formal Definition for Concept Learning

- Inferring a boolean-valued function from training examples of its input and output.

- An example for concept – learning is the learning of bird concept from the given examples of birds (positive examples) and non – birds (negative examples).

- Each concept is a Boolean – Valued function defined over this larger set. [Example : a function defined over all animals whose value is true for birds and false for every other animal.]

A Concept Learning Task

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

TABLE
Positive and negative training examples for the target concept *EnjoySport.*

- consider the example task of learning the target concept "days on which my friend Aldo enjoys his favorite water sport.“

- The previous slide table describes a set of example days, each represented by a set of attributes.

- The attribute Enjoy Sport indicates whether or not Aldo enjoys his favorite water sport on this day.

- The task is to learn to predict the value of EnjoySport for an arbitrary day, based on the values of its other attributes.

- What hypothesis representation shall we provide to the learner in this case?

- Let us begin by considering a simple representation in which each hypothesis consists of a conjunction of constraints on the instance attributes.

- In particular, let each hypothesis be a vector of six constraints, specifying the values of the six attributes Sky, AirTemp, Humidity, Wind, Water, and Forecast. For each attribute, the hypothesis will either
- indicate by a "?" that any value is acceptable for this attribute,
- specify a single required value (e.g., *Warm*) for the attribute, or
- indicate by a "Ø" that no value is acceptable.

If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example ($h(x) = 1$). To illustrate, the hypothesis that Aldo enjoys his favorite sport only on cold days with high humidity (independent of the values of the other attributes) is represented by the expression

$$\langle ?, Cold, High, ?, ?, ? \rangle$$

- To summarize, the EnjoySport concept learning task requires learning the set of days for which **EnjoySport** = yes, describing this set by a conjunction of constraints over the instance attributes.

- In general, any concept learning task can be described by the set of instances over which the target function is defined, the target function, the set of candidate hypotheses considered by the learner, and the set of available training examples.

Notation

- **Given:**
  - Instances X: Possible days, each described by the attributes
    - *Sky* (with possible values *Sunny*, *Cloudy*, and *Rainy*),
    - *AirTemp* (with values *Warm* and *Cold*),
    - *Humidity* (with values *Normal* and *High*),
    - *Wind* (with values *Strong* and *Weak*),
    - *Water* (with values *Warm* and *Cool*), and
    - *Forecast* (with values *Same* and *Change*).
  - Hypotheses H: Each hypothesis is described by a conjunction of constraints on the attributes *Sky*, *AirTemp*, *Humidity*, *Wind*, *Water*, and *Forecast*. The constraints may be "?" (any value is acceptable), "Ø" (no value is acceptable), or a specific value.
  - Target concept c: $EnjoySport : X \rightarrow \{0, 1\}$
  - Training examples D: Positive and negative examples of the target function
- **Determine:**
  - A hypothesis h in H such that $h(x) = c(x)$ for all x in X.

TABLE
The *EnjoySport* concept learning task.

When learning the target concept, the learner is presented a set of *training examples*, each consisting of an instance x from X, along with its target concept value $c(x)$ (e.g., the training examples in Table). Instances for which $c(x) = 1$ are called *positive examples*, or members of the target concept. Instances for which $c(x) = 0$ are called *negative examples*, or nonmembers of the target concept. We will often write the ordered pair $\langle x, c(x)\rangle$ to describe the training example consisting of the instance x and its target concept value $c(x)$. We use the symbol D to denote the set of available training examples.

Given a set of training examples of the target concept c, the problem faced by the learner is to hypothesize, or estimate, c. We use the symbol H to denote the set of *all possible hypotheses* that the learner may consider regarding the identity of the target concept. Usually H is determined by the human designer's choice of hypothesis representation. In general, each hypothesis h in H represents a boolean-valued function defined over X; that is, $h : X \rightarrow \{0, 1\}$. The goal of the learner is to find a hypothesis h such that $h(x) = c(x)$ for a'l x in X.

The Inductive Learning Hypothesis

- Inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data. Lacking any further information, our assumption is that the best hypothesis regarding unseen instances is the hypothesis that best fits the observed training data.

- **The inductive learning hypothesis :** Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Concept Learning As Search

- Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.
- The goal of this search is to find the hypothesis that best fits the training examples.
- It is important to note that by selecting a hypothesis representation, the designer of the learning algorithm implicitly defines the space of all hypotheses that the program can ever represent and therefore can ever learn.

- Consider, for example, the instances X and hypotheses H in the EnjoySport learning task. Given that the attribute Sky has three possible values, and that AirTemp, Humidity, Wind, Water, and Forecast each have two possible values, the instance space X contains exactly **3*2*2*2*2*2 = 96** distinct instances.
- A similar calculation shows that there are **5*4*4*4*4*4 = 5120** syntactically distinct hypotheses within H.
- every hypothesis containing one or more "∅" symbols represents the empty set of instances; that is, it classifies every instance as negative. Therefore, the number of semantically distinct hypotheses is only 1 + (4*3*3*3*3*3) = 973.
- Our EnjoySport example is a very simple learning task, with a relatively small, finite hypothesis space. Most practical learning tasks involve much larger, sometimes infinite, hypothesis spaces.

General-to-Specific Ordering of Hypotheses

- To illustrate the general-to-specific ordering, consider the two hypotheses

$$h_1 = \langle Sunny, ?, ?, Strong, ?, ? \rangle$$

$$h_2 = \langle Sunny, ?, ?, ?, ?, ? \rangle$$

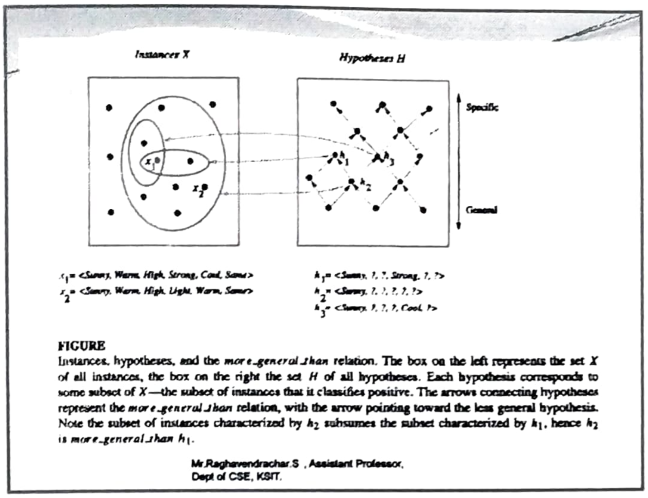
- Now consider the sets of instances that are classified positive by hl and by h2.
- Because h2 imposes fewer constraints on the instance, it classifies more instances as positive.
- In fact, any instance classified positive by hl will also be classified positive by h2.
- Therefore, we say that h2 is more general than hl.

This intuitive "more general than" relationship between hypotheses can be defined more precisely as follows. First, for any instance x in X and hypothesis h in H, we say that x *satisfies* h if and only if $h(x) = 1$. We now define the *more_general_than_or_equal_to* relation in terms of the sets of instances that satisfy the two hypotheses: Given hypotheses h_j and h_k, h_j is *more_general_than_or_equal_to* h_k if and only if any instance that satisfies h_k also satisfies h_j.

Definition: Let h_j and h_k be boolean-valued functions defined over X. Then h_j is more_general_than_or_equal_to h_k (written $h_j \geq_g h_k$) if and only if

$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

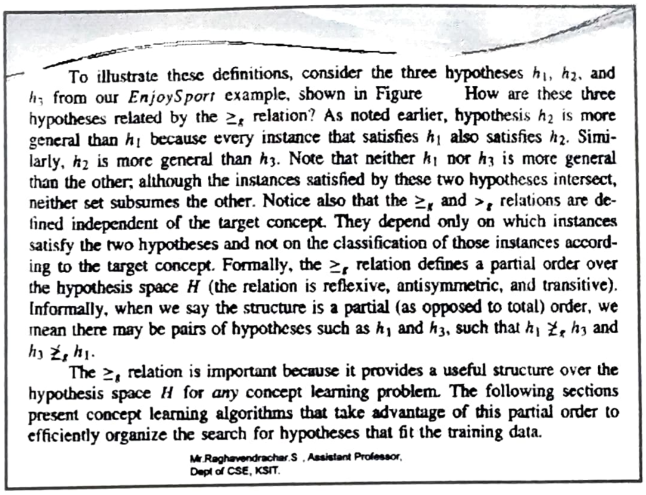
We will also find it useful to consider cases where one hypothesis is strictly more general than the other. Therefore, we will say that h_j is (strictly) *more_general_than* h_k (written $h_j >_g h_k$) if and only if $(h_j \geq_g h_k) \wedge (h_k \ngeq_g h_j)$. Finally, we will sometimes find the inverse useful and will say that h_j is *more_specific_than* h_k when h_k is *more_general_than* h_j.



x_1= <Sunny, Warm, High, Strong, Cool, Same>
x_2= <Sunny, Warm, High, Light, Warm, Same>

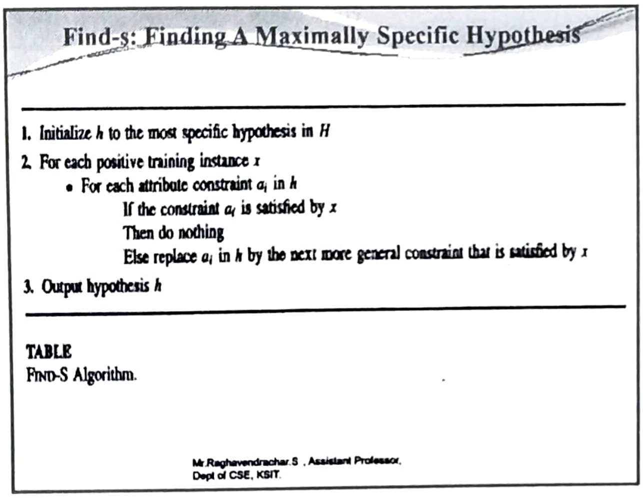
h_1= <Sunny, ?, ?, Strong, ?, ?>
h_2= <Sunny, ?, ?, ?, ?, ?>
h_3= <Sunny, ?, ?, ?, Cool, ?>

FIGURE
Instances, hypotheses, and the *more_general_than* relation. The box on the left represents the set X of all instances, the box on the right the set H of all hypotheses. Each hypothesis corresponds to some subset of X—the subset of instances that it classifies positive. The arrows connecting hypotheses represent the *more_general_than* relation, with the arrow pointing toward the less general hypothesis. Note the subset of instances characterized by h_2 subsumes the subset characterized by h_1, hence h_2 is *more_general_than* h_1.



To illustrate these definitions, consider the three hypotheses h_1, h_2, and h_3 from our *EnjoySport* example, shown in Figure How are these three hypotheses related by the $\geq_g$ relation? As noted earlier, hypothesis h_2 is more general than h_1 because every instance that satisfies h_1 also satisfies h_2. Similarly, h_2 is more general than h_3. Note that neither h_1 nor h_3 is more general than the other; although the instances satisfied by these two hypotheses intersect, neither set subsumes the other. Notice also that the $\geq_g$ and $>_g$ relations are defined independent of the target concept. They depend only on which instances satisfy the two hypotheses and not on the classification of those instances according to the target concept. Formally, the $\geq_g$ relation defines a partial order over the hypothesis space H (the relation is reflexive, antisymmetric, and transitive). Informally, when we say the structure is a partial (as opposed to total) order, we mean there may be pairs of hypotheses such as h_1 and h_3, such that $h_1 \ngeq_g h_3$ and $h_3 \ngeq_g h_1$.

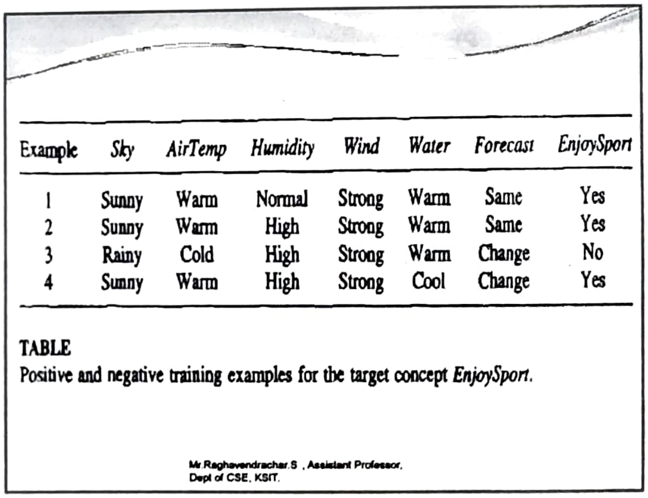
The $\geq_g$ relation is important because it provides a useful structure over the hypothesis space H for *any* concept learning problem. The following sections present concept learning algorithms that take advantage of this partial order to efficiently organize the search for hypotheses that fit the training data.



Find-s: Finding A Maximally Specific Hypothesis

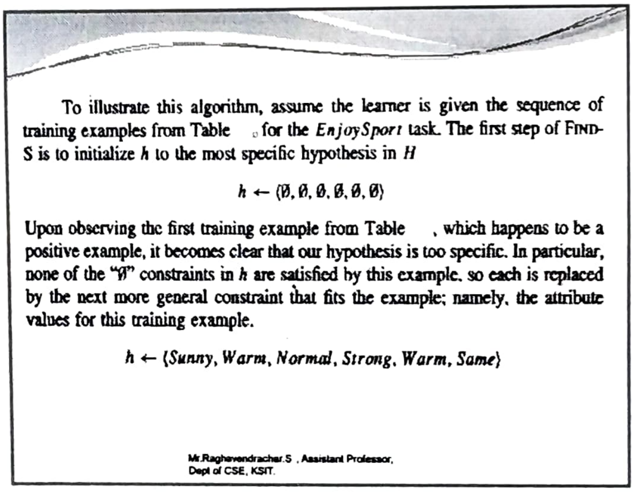
```
1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
      • For each attribute constraint $a_i$ in h
            If the constraint $a_i$ is satisfied by x
            Then do nothing
            Else replace $a_i$ in h by the next more general constraint that is satisfied by x
3. Output hypothesis h
```

TABLE
FIND-S Algorithm.



Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

TABLE
Positive and negative training examples for the target concept *EnjoySport*.

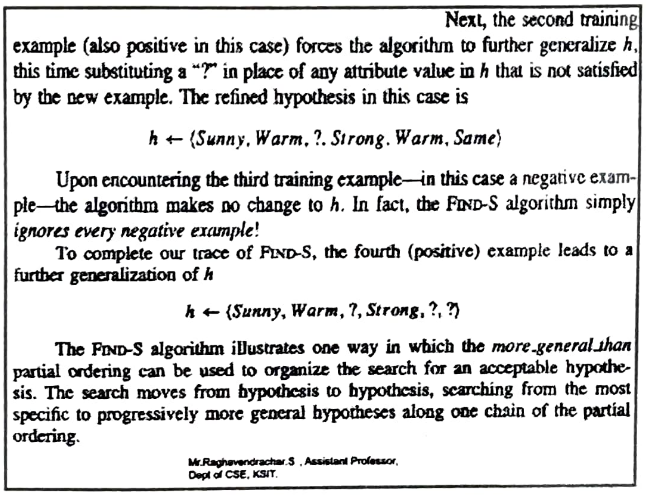


To illustrate this algorithm, assume the learner is given the sequence of training examples from Table for the *EnjoySport* task. The first step of FIND-S is to initialize h to the most specific hypothesis in H

$$h \leftarrow \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

Upon observing the first training example from Table , which happens to be a positive example, it becomes clear that our hypothesis is too specific. In particular, none of the "$\emptyset$" constraints in h are satisfied by this example, so each is replaced by the next more general constraint that fits the example; namely, the attribute values for this training example.

$$h \leftarrow \langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$$



Next, the second training example (also positive in this case) forces the algorithm to further generalize h, this time substituting a "?" in place of any attribute value in h that is not satisfied by the new example. The refined hypothesis in this case is

$$h \leftarrow \langle Sunny, Warm, ?, Strong, Warm, Same \rangle$$

Upon encountering the third training example—in this case a negative example—the algorithm makes no change to h. In fact, the FIND-S algorithm simply *ignores every negative example!*

To complete our trace of FIND-S, the fourth (positive) example leads to a further generalization of h

$$h \leftarrow \langle Sunny, Warm, ?, Strong, ?, ? \rangle$$

The FIND-S algorithm illustrates one way in which the *more_general_than* partial ordering can be used to organize the search for an acceptable hypothesis. The search moves from hypothesis to hypothesis, searching from the most specific to progressively more general hypotheses along one chain of the partial ordering.

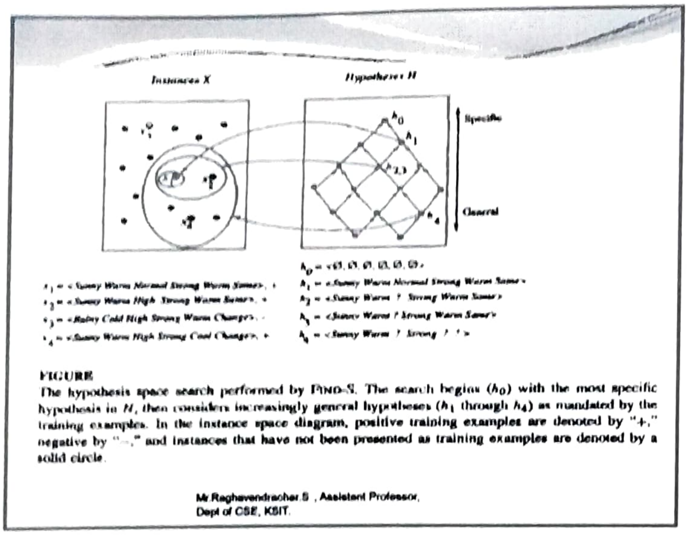
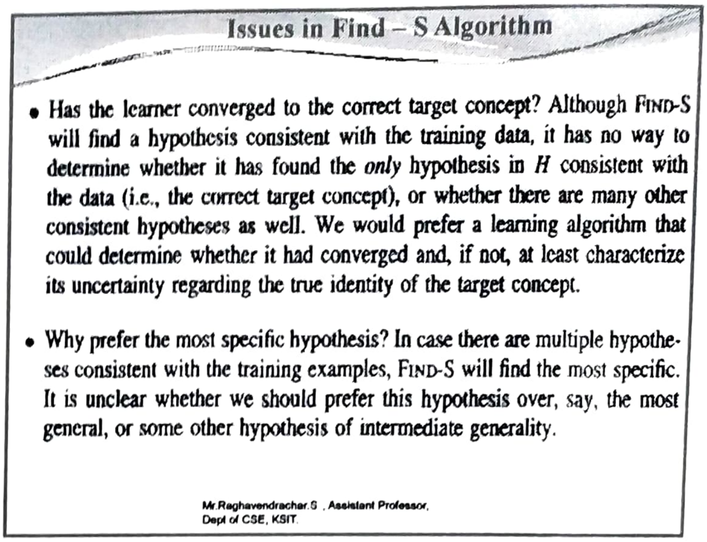


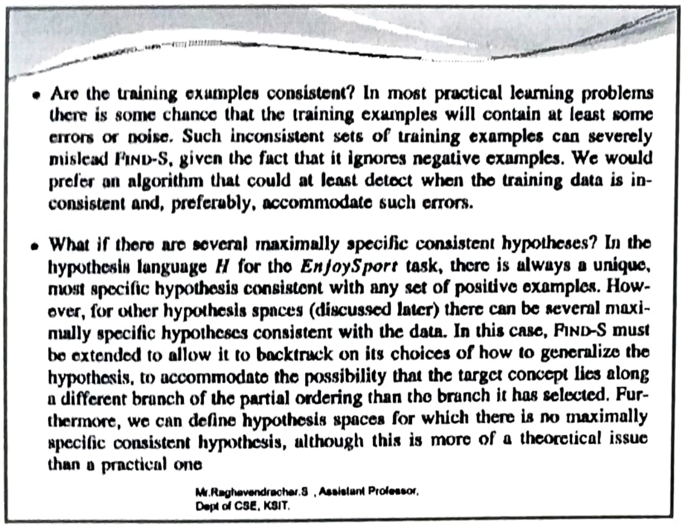
FIGURE
The hypothesis space search performed by FIND-S. The search begins (h_0) with the most specific hypothesis in H, then considers increasingly general hypotheses (h_1 through h_4) as mandated by the training examples. In the instance space diagram, positive training examples are denoted by "+," negative by "−," and instances that have not been presented as training examples are denoted by a solid circle.



Issues in Find – S Algorithm

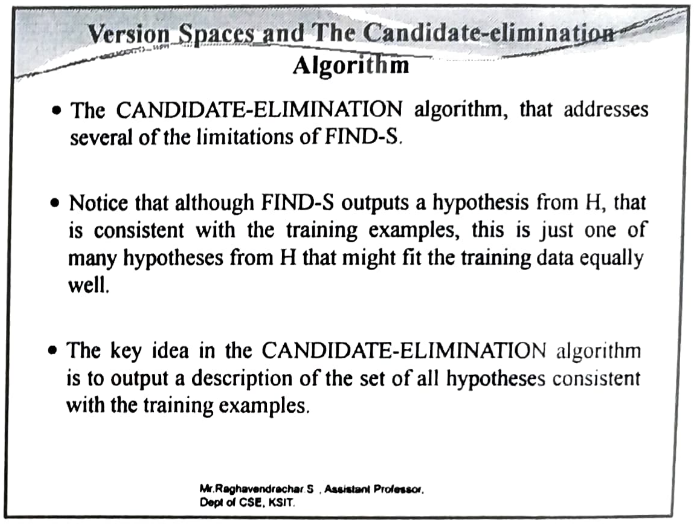
- Has the learner converged to the correct target concept? Although FIND-S will find a hypothesis consistent with the training data, it has no way to determine whether it has found the *only* hypothesis in H consistent with the data (i.e., the correct target concept), or whether there are many other consistent hypotheses as well. We would prefer a learning algorithm that could determine whether it had converged and, if not, at least characterize its uncertainty regarding the true identity of the target concept.

- Why prefer the most specific hypothesis? In case there are multiple hypotheses consistent with the training examples, FIND-S will find the most specific. It is unclear whether we should prefer this hypothesis over, say, the most general, or some other hypothesis of intermediate generality.



- Are the training examples consistent? In most practical learning problems there is some chance that the training examples will contain at least some errors or noise. Such inconsistent sets of training examples can severely mislead FIND-S, given the fact that it ignores negative examples. We would prefer an algorithm that could at least detect when the training data is inconsistent and, preferably, accommodate such errors.

- What if there are several maximally specific consistent hypotheses? In the hypothesis language H for the *EnjoySport* task, there is always a unique, most specific hypothesis consistent with any set of positive examples. However, for other hypothesis spaces (discussed later) there can be several maximally specific hypotheses consistent with the data. In this case, FIND-S must be extended to allow it to backtrack on its choices of how to generalize the hypothesis, to accommodate the possibility that the target concept lies along a different branch of the partial ordering than the branch it has selected. Furthermore, we can define hypothesis spaces for which there is no maximally specific consistent hypothesis, although this is more of a theoretical issue than a practical one

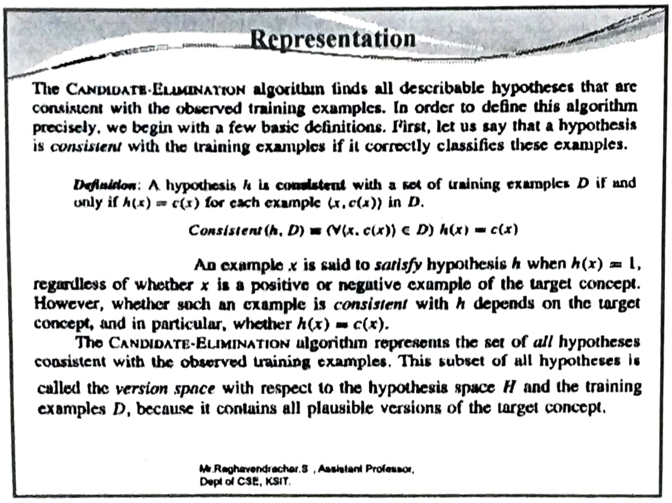


Version Spaces and The Candidate-elimination Algorithm

- The CANDIDATE-ELIMINATION algorithm, that addresses several of the limitations of FIND-S.

- Notice that although FIND-S outputs a hypothesis from H, that is consistent with the training examples, this is just one of many hypotheses from H that might fit the training data equally well.

- The key idea in the CANDIDATE-ELIMINATION algorithm is to output a description of the set of all hypotheses consistent with the training examples.



Representation

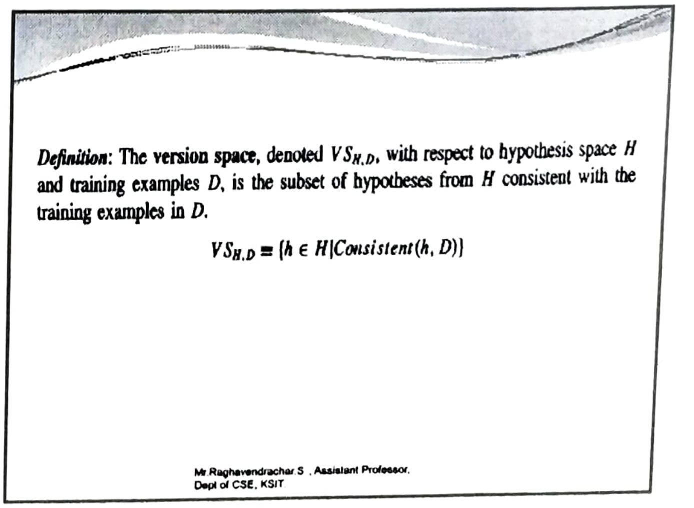
The CANDIDATE-ELIMINATION algorithm finds all describable hypotheses that are consistent with the observed training examples. In order to define this algorithm precisely, we begin with a few basic definitions. First, let us say that a hypothesis is *consistent* with the training examples if it correctly classifies these examples.

Definition: A hypothesis h is **consistent** with a set of training examples D if and only if $h(x) = c(x)$ for each example $\langle x, c(x)\rangle$ in D.

$$Consistent(h, D) \equiv (\forall \langle x, c(x)\rangle \in D)\ h(x) = c(x)$$

An example x is said to *satisfy* hypothesis h when $h(x) = 1$, regardless of whether x is a positive or negative example of the target concept. However, whether such an example is *consistent* with h depends on the target concept, and in particular, whether $h(x) = c(x)$.

The CANDIDATE-ELIMINATION algorithm represents the set of *all* hypotheses consistent with the observed training examples. This subset of all hypotheses is called the *version space* with respect to the hypothesis space H and the training examples D, because it contains all plausible versions of the target concept.



Definition: The version space, denoted $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with the training examples in D.

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

The LIST-THEN-ELIMINATE Algorithm

The List-Then-Eliminate Algorithm

```
1. VersionSpace ← a list containing every hypothesis in H
2. For each training example, ⟨x, c(x)⟩
       remove from VersionSpace any hypothesis h for which h(x) ≠ c(x)
3. Output the list of hypotheses in VersionSpace
```

The List-Then-Eliminate algorithm.

- The LIST-THEN-ELIMINATE Algorithm first initializes the version space to contain all hypotheses in H, then eliminates any hypothesis found inconsistent with any training example.

- The version space of candidate hypotheses thus shrinks as more examples are observed, until ideally just one hypothesis remains that is consistent with all the observed examples.

- The LIST-THEN-ELIMINATE Algorithm it is guaranteed to output all hypotheses consistent with the training data.

A More Compact Representation for Version Spaces

- The CANDIDATE-ELIMINATION Algorithm works on the same principle as the above LIST-THEN-ELIMINATE Algorithm.

- The version space is represented by its most general and least general members.

- These members form general and specific boundary sets that delimit the version space within the partially ordered hypothesis space.

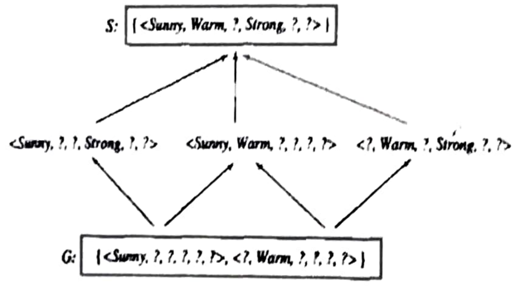


FIGURE
A version space with its general and specific boundary sets. The version space includes all six hypotheses shown here, but can be represented more simply by S and G. Arrows indicate instances of the *more_general_than* relation. This is the version space for the *EnjoySport* concept learning problem and training examples described in Table

To illustrate this representation for version spaces, consider again the *EnjoySport* concept learning problem described in Table . Recall that given the four training examples from Table , Find-S outputs the hypothesis

$$h = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$$

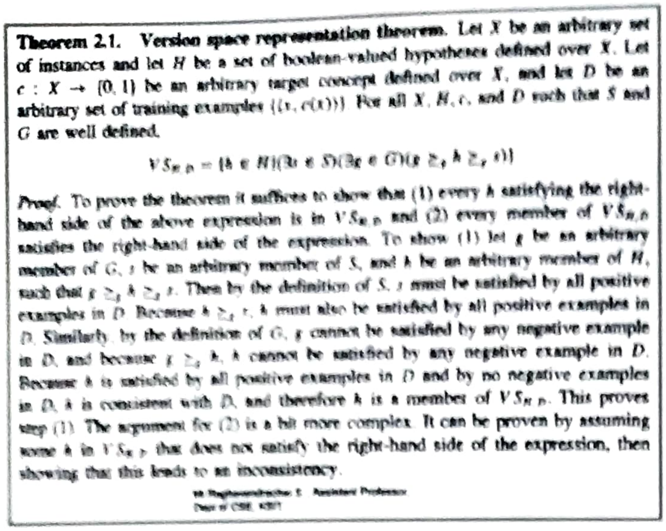
In fact, this is just one of six different hypotheses from H that are consistent with these training examples. All six hypotheses are shown in Figure . They constitute the version space relative to this set of data and this hypothesis representation. The arrows among these six hypotheses in Figure indicate instances of the *more_general_than* relation. The Candidate-Elimination algorithm represents the version space by storing only its most general members (labeled G in Figure) and its most specific (labeled S in the figure). Given only these two sets S and G, it is possible to enumerate all members of the version space as needed by generating the hypotheses that lie between these two sets in the general-to-specific partial ordering over hypotheses.

***Definition*: The general boundary** G, with respect to hypothesis space H and training data D, is the set of maximally general members of H consistent with D.

$$G \equiv \{g \in H | Consistent(g, D) \wedge (\neg\exists g' \in H)[(g' >_g g) \wedge Consistent(g', D)]\}$$

***Definition*: The specific boundary** S, with respect to hypothesis space H and training data D, is the set of minimally general (i.e., maximally specific) members of H consistent with D.

$$S \equiv \{s \in H | Consistent(s, D) \wedge (\neg\exists s' \in H)[(s >_g s') \wedge Consistent(s', D)]\}$$

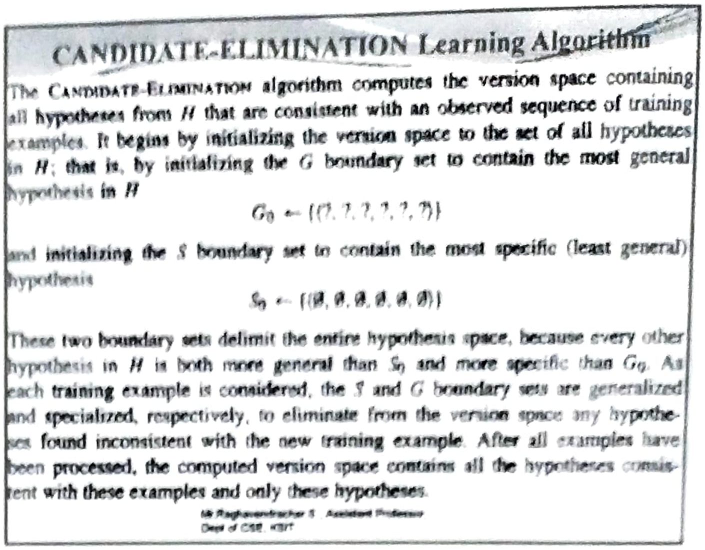


Theorem 2.1. Version space representation theorem. Let X be an arbitrary set of instances and let H be a set of boolean-valued hypotheses defined over X. Let $c : X \rightarrow \{0, 1\}$ be an arbitrary target concept defined over X, and let D be an arbitrary set of training examples $\{\langle x, c(x)\rangle\}$. For all X, H, c, and D such that S and G are well defined,

$$VS_{H,D} = \{h \in H \mid (\exists s \in S)(\exists g \in G)(g \geq_g h \geq_g s)\}$$

Proof. To prove the theorem it suffices to show that (1) every h satisfying the right-hand side of the above expression is in $VS_{H,D}$ and (2) every member of $VS_{H,D}$ satisfies the right-hand side of the expression. To show (1) let g be an arbitrary member of G, s be an arbitrary member of S, and h be an arbitrary member of H, such that $g \geq_g h \geq_g s$. Then by the definition of S, s must be satisfied by all positive examples in D. Because $h \geq_g s$, h must also be satisfied by all positive examples in D. Similarly, by the definition of G, g cannot be satisfied by any negative example in D, and because $g \geq_g h$, h cannot be satisfied by any negative example in D. Because h is satisfied by all positive examples in D and by no negative examples in D, h is consistent with D, and therefore h is a member of $VS_{H,D}$. This proves step (1). The argument for (2) is a bit more complex. It can be proven by assuming some h in $VS_{H,D}$ that does not satisfy the right-hand side of the expression, then showing that this leads to an inconsistency.

Mr.Raghavendrachar S , Assistant Professor,
Dept of CSE, KSIT



CANDIDATE-ELIMINATION Learning Algorithm

The CANDIDATE-ELIMINATION algorithm computes the version space containing all hypotheses from H that are consistent with an observed sequence of training examples. It begins by initializing the version space to the set of all hypotheses in H; that is, by initializing the G boundary set to contain the most general hypothesis in H

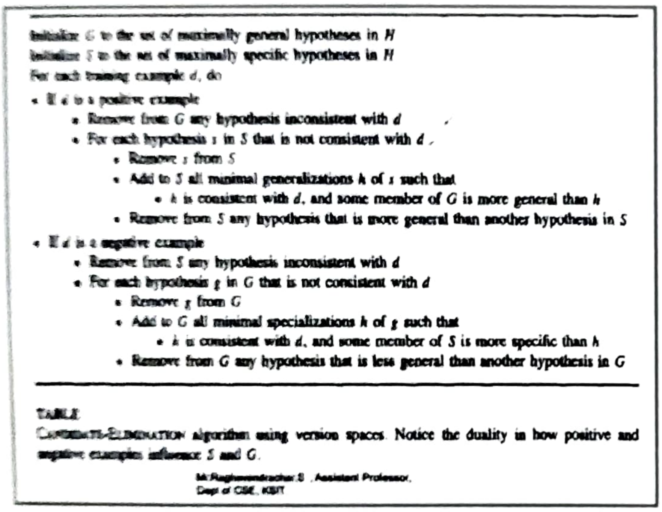
$$G_0 \leftarrow \{\langle ?, ?, ?, ?, ?, ? \rangle\}$$

and initializing the S boundary set to contain the most specific (least general) hypothesis

$$S_0 \leftarrow \{\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle\}$$

These two boundary sets delimit the entire hypothesis space, because every other hypothesis in H is both more general than S_0 and more specific than G_0. As each training example is considered, the S and G boundary sets are generalized and specialized, respectively, to eliminate from the version space any hypotheses found inconsistent with the new training example. After all examples have been processed, the computed version space contains all the hypotheses consistent with these examples and only these hypotheses.

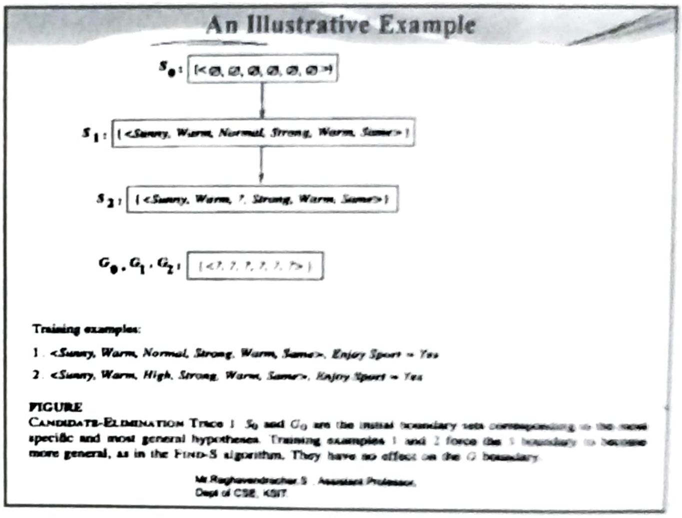
Mr.Raghavendrachar S , Assistant Professor,
Dept of CSE, KSIT



```
Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For each training example d, do
  • If d is a positive example
      • Remove from G any hypothesis inconsistent with d
      • For each hypothesis s in S that is not consistent with d
          • Remove s from S
          • Add to S all minimal generalizations h of s such that
              • h is consistent with d, and some member of G is more general than h
          • Remove from S any hypothesis that is more general than another hypothesis in S
  • If d is a negative example
      • Remove from S any hypothesis inconsistent with d
      • For each hypothesis g in G that is not consistent with d
          • Remove g from G
          • Add to G all minimal specializations h of g such that
              • h is consistent with d, and some member of S is more specific than h
          • Remove from G any hypothesis that is less general than another hypothesis in G
```

TABLE
CANDIDATE-ELIMINATION algorithm using version spaces. Notice the duality in how positive and negative examples influence S and G.

Mr.Raghavendrachar S , Assistant Professor,
Dept of CSE, KSIT



An Illustrative Example

S_0: {<Ø, Ø, Ø, Ø, Ø, Ø>}

S_1: {<Sunny, Warm, Normal, Strong, Warm, Same>}

S_2: {<Sunny, Warm, ?, Strong, Warm, Same>}

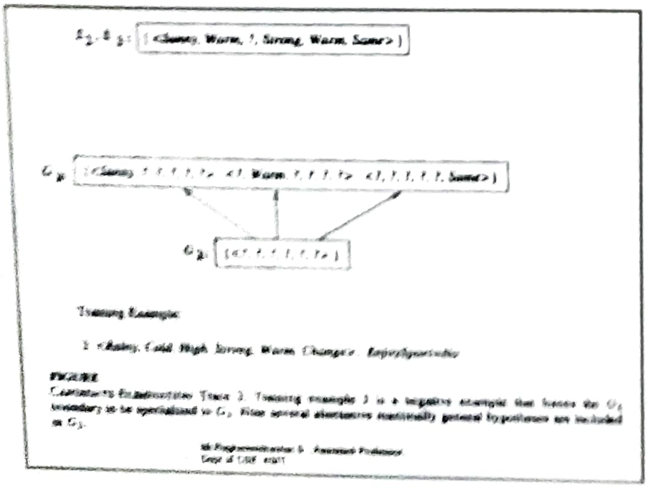
G_0, G_1, G_2: {<?, ?, ?, ?, ?, ?>}

Training examples:

1. <*Sunny, Warm, Normal, Strong, Warm, Same*>, *Enjoy Sport = Yes*
2. <*Sunny, Warm, High, Strong, Warm, Same*>, *Enjoy Sport = Yes*

FIGURE
CANDIDATE-ELIMINATION Trace 1. S_0 and G_0 are the initial boundary sets corresponding to the most specific and most general hypotheses. Training examples 1 and 2 force the S boundary to become more general, as in the FIND-S algorithm. They have no effect on the G boundary.

Mr.Raghavendrachar S , Assistant Professor,
Dept of CSE, KSIT



S_2, S_3: {<Sunny, Warm, ?, Strong, Warm, Same>}

G_3: {<Sunny, ?, ?, ?, ?, ?> <?, Warm, ?, ?, ?, ?> <?, ?, ?, ?, ?, Same>}

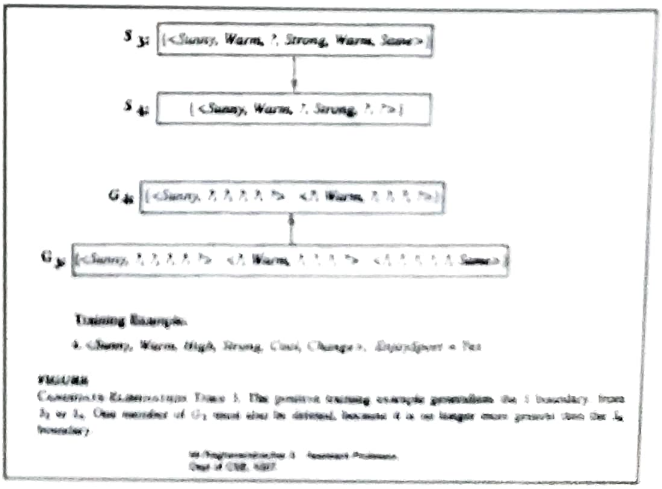
G_2: {<?, ?, ?, ?, ?, ?>}

Training Example:

3. <*Rainy, Cold, High, Strong, Warm, Change*>, *EnjoySport=No*

FIGURE
CANDIDATE-ELIMINATION Trace 2. Training example 3 is a negative example that forces the G_2 boundary to be specialized to G_3. Note several alternative maximally general hypotheses are included in G_3.

Mr.Raghavendrachar S , Assistant Professor,
Dept of CSE, KSIT



S_3: {<Sunny, Warm, ?, Strong, Warm, Same>}

S_4: {<Sunny, Warm, ?, Strong, ?, ?>}

G_4: {<Sunny, ?, ?, ?, ?, ?> <?, Warm, ?, ?, ?, ?>}

G_3: {<Sunny, ?, ?, ?, ?, ?> <?, Warm, ?, ?, ?, ?> <?, ?, ?, ?, ?, Same>}

Training Example:

4. <*Sunny, Warm, High, Strong, Cool, Change*>, *EnjoySport = Yes*

FIGURE
CANDIDATE-ELIMINATION Trace 3. The positive training example generalizes the S boundary, from S_3 to S_4. One member of G_3 must also be deleted, because it is no longer more general than the S_4 boundary.

Mr.Raghavendrachar S , Assistant Professor,
Dept of CSE, KSIT

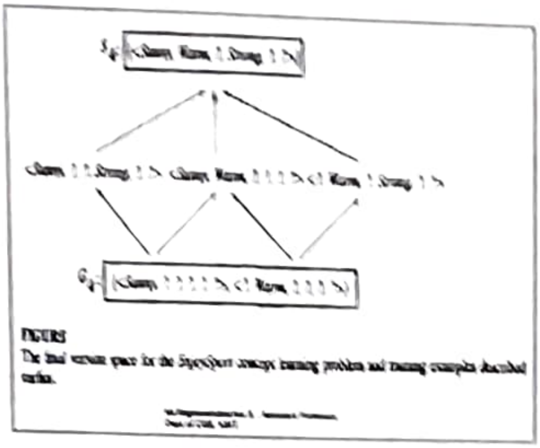
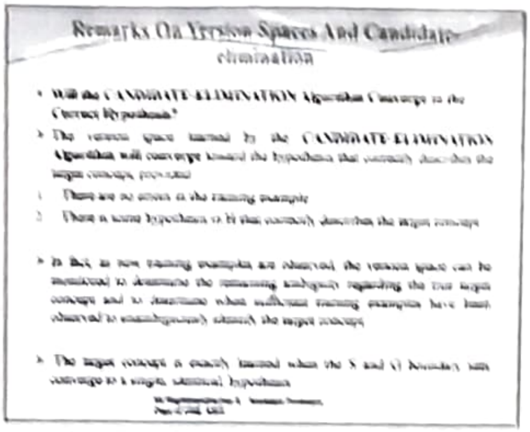
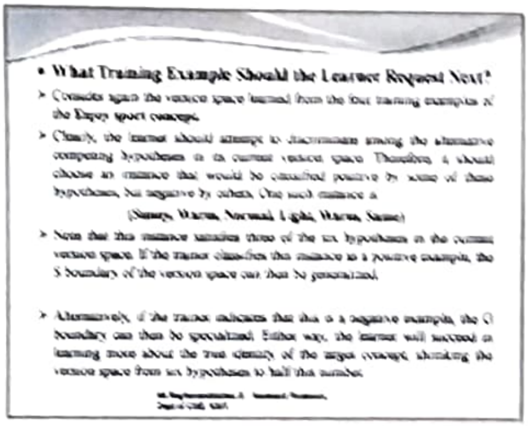


FIGURE
The final version space for the EnjoySport concept learning problem and training examples described earlier.



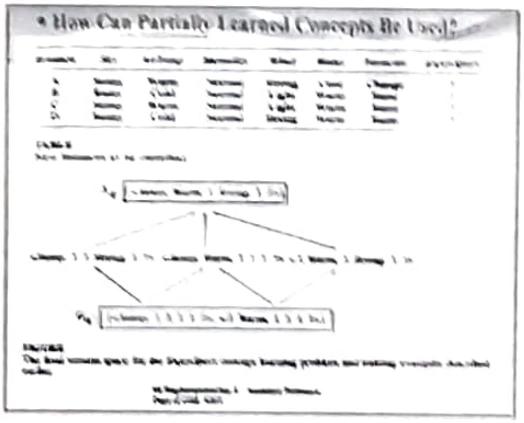
Remarks On Version Spaces And Candidate-elimination

- **Will the CANDIDATE-ELIMINATION Algorithm Converge to the Correct Hypothesis?**
- The version space learned by the CANDIDATE-ELIMINATION **Algorithm will converge** toward the hypothesis that correctly describes the target concept, provided
  1. There are no errors in the training examples
  2. There is some hypothesis in H that correctly describes the target concept
- In fact, as new training examples are observed, the version space can be monitored to determine the remaining ambiguity regarding the true target concept and to determine when sufficient training examples have been observed to unambiguously identify the target concept.
- The target concept is exactly learned when the S and G boundary sets converge to a single, identical, hypothesis.



- **What Training Example Should the Learner Request Next?**
- Consider again the version space learned from the four training examples of the Enjoy sport concept.
- Clearly, the learner should attempt to discriminate among the alternative competing hypotheses in its current version space. Therefore, it should choose an instance that would be classified positive by some of these hypotheses, but negative by others. One such instance is

  (Sunny, Warm, Normal, Light, Warm, Same)
- Note that this instance satisfies three of the six hypotheses in the current version space. If the trainer classifies this instance as a positive example, the S boundary of the version space can then be generalized.
- Alternatively, if the trainer indicates that this is a negative example, the G boundary can then be specialized. Either way, the learner will succeed in learning more about the true identity of the target concept, shrinking the version space from six hypotheses to half this number.

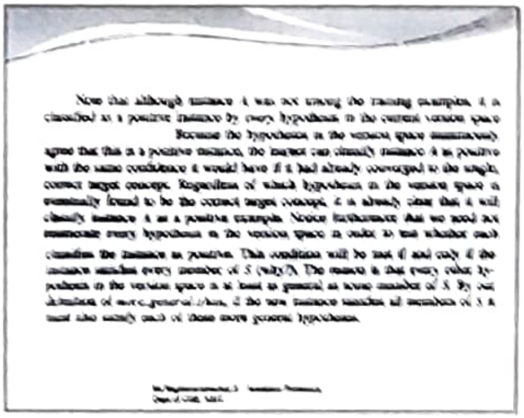


• How Can Partially Learned Concepts Be Used?

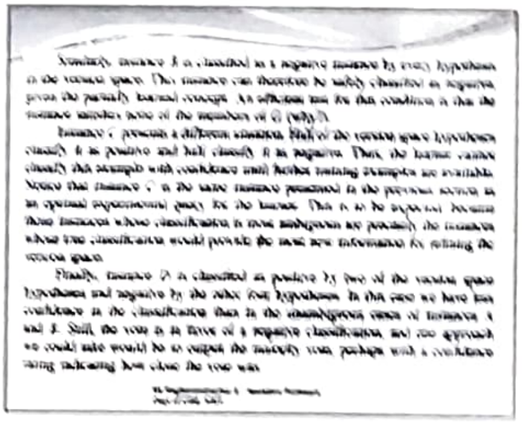
Instance	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	Light	Warm	Same	?
C	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

TABLE
New instances to be classified.

FIGURE
The final version space for the EnjoySport concept learning problem and training examples described earlier.



Note that although instance A was not among the training examples, it is classified as a positive instance by every hypothesis in the current version space. Because the hypotheses in the version space unanimously agree that this is a positive instance, the learner can classify instance A as positive with the same confidence it would have if it had already converged to the single, correct target concept. Regardless of which hypothesis in the version space is eventually found to be the correct target concept, it is already clear that it will classify instance A as a positive example. Notice furthermore that we need not enumerate every hypothesis in the version space in order to test whether each classifies the instance as positive. This condition will be met if and only if the instance satisfies every member of S (why?). The reason is that every other hypothesis in the version space is at least as general as some member of S. By our definition of more-general-than, if the new instance satisfies all members of S it must also satisfy each of these more general hypotheses.



Similarly, instance B is classified as a negative instance by every hypothesis in the version space. This instance can therefore be safely classified as negative, given the partially learned concept. An efficient test for this condition is that the instance satisfies none of the members of G (why?).

Instance C presents a different situation. Half of the version space hypotheses classify it as positive and half classify it as negative. Thus, the learner cannot classify this example with confidence until further training examples are available. Notice that instance C is the same instance presented in the previous section as an optimal experimental query for the learner. This is to be expected, because those instances whose classification is most ambiguous are precisely the instances whose true classification would provide the most new information for refining the version space.

Finally, instance D is classified as positive by two of the version space hypotheses and negative by the other four hypotheses. In this case we have less confidence in the classification than in the unambiguous cases of instances A and B. Still, the vote is in favor of a negative classification, and one approach we could take would be to output the majority vote, perhaps with a confidence rating indicating how close the vote was.

INDUCTIVE BIAS

- The **CANDIDATE-ELIMINATION Algorithm will converge toward** the true target concept provided it is given accurate training examples and provided its initial hypothesis space contains the target concept

- What if the target concept is not contained in the hypothesis space? Can we avoid this difficulty by using a hypothesis space that includes every possible hypothesis? How does the size of this hypothesis space influence the ability of the algorithm to generalize to unobserved instances? How does the size of the hypothesis space influence the number of training examples that must be observed? These are fundamental questions for inductive inference in general.

A Biased Hypothesis Space

- Suppose we wish to assure that the hypothesis space contains the unknown target concept.

- The obvious solution is to enrich the hypothesis space to include **every possible hypothesis.**

- To illustrate, consider again the **Enjoy Sport example in** which we restricted the hypothesis space to include only conjunctions of attribute values.

- Because of this restriction, the hypothesis space is unable to represent even simple disjunctive target concepts such as **"Sky = Sunny or Sky = Cloudy."**

In fact, given the following three training examples of this disjunctive hypothesis, our algorithm would find that there are zero hypotheses in the version space.

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

To see why there are no hypotheses consistent with these three examples, note that the most specific hypothesis consistent with the first two examples *and representable in the given hypothesis space H* is

$$S_2 : \langle ?, Warm, Normal, Strong, Cool, Change\rangle$$

This hypothesis, although it is the maximally specific hypothesis from H that is consistent with the first two examples, is already overly general: it incorrectly covers the third (negative) training example. The problem is that we have biased the learner to consider only conjunctive hypotheses. In this case we require a more expressive hypothesis space.

An Unbiased Learner

In the *EnjoySport* learning task, for example, the size of the instance space X of days described by the six available attributes is 96. How many possible concepts can be defined over this set of instances? In other words, how large is the power set of X? In general, the number of distinct subsets that can be defined over a set X containing $|X|$ elements (i.e., the size of the power set of X) is $2^{|X|}$. Thus, there are 2^{96}, or approximately 10^{28} distinct target concepts that could be defined over this instance space and that our learner might be called upon to learn.

Let us reformulate the *EnjoySport* learning task in an unbiased way by defining a new hypothesis space H' that can represent every subset of instances; that is, let H' correspond to the power set of X. One way to define such an H' is to allow arbitrary disjunctions, conjunctions, and negations of our earlier hypotheses. For instance, the target concept "*Sky = Sunny* or *Sky = Cloudy*" could then be described as

$$\langle Sunny, ?, ?, ?, ?, ?\rangle \vee \langle Cloudy, ?, ?, ?, ?, ?\rangle$$

Given this hypothesis space, we can safely use the CANDIDATE-ELIMINATION algorithm without worrying that the target concept might not be expressible. However, while this hypothesis space eliminates any problems of expressibility, it unfortunately raises a new, equally difficult problem: our concept learning algorithm is now completely unable to generalize beyond the observed examples! To see why, suppose we present three positive examples (x_1, x_2, x_3) and two negative examples (x_4, x_5) to the learner. At this point, the S boundary of the version space will contain the hypothesis which is just the disjunction of the positive examples

$$S : \{(x_1 \vee x_2 \vee x_3)\}$$

---

because this is the most specific possible hypothesis that covers these three examples. Similarly, the G boundary will consist of the hypothesis that rules out only the observed negative examples

$$G : \{\neg(x_4 \vee x_5)\}$$

The problem here is that with this very expressive hypothesis representation, the S boundary will always be simply the disjunction of the observed positive examples, while the G boundary will always be the negated disjunction of the observed negative examples. Therefore, the only examples that will be unambiguously classified by S and G are the observed training examples themselves. In order to converge to a single, final target concept, we will have to present every single instance in X as a training example!

The Futility of Bias-Free Learning

Because inductive learning requires some form of prior assumptions, or inductive bias, we will find it useful to characterize different learning approaches by the inductive bias[1] they employ. Let us define this notion of inductive bias more precisely. The key idea we wish to capture here is the policy by which the learner generalizes beyond the observed training data, to infer the classification of new instances. Therefore, consider the general setting in which an arbitrary learning algorithm L is provided an arbitrary set of training data $D_c = \{\langle x, c(x)\rangle\}$ of some arbitrary target concept c. After training, L is asked to classify a new instance x_i. Let $L(x_i, D_c)$ denote the classification (e.g., positive or negative) that L assigns to x_i after learning from the training data D_c. We can describe this inductive inference step performed by L as follows

$$(D_c \wedge x_i) \succ L(x_i, D_c)$$

where the notation $y \succ z$ indicates that z is inductively inferred from y. For example, if we take L to be the CANDIDATE-ELIMINATION algorithm, D_c to be the training data from Table 2.1, and x_i to be the first instance from Table 2.6, then the inductive inference performed in this case concludes that $L(x_i, D_c) = (EnjoySport = yes)$.

Because L is an inductive learning algorithm, the result $L(x_i, D_c)$ that it infers will not in general be provably correct; that is, the classification $L(x_i, D_c)$ need not follow deductively from the training data D_c and the description of the new instance x_i. However, it is interesting to ask what additional assumptions could be added to $D_c \wedge x_i$ so that $L(x_i, D_c)$ would follow deductively. We define the inductive bias of L as this set of additional assumptions. More precisely, we define the inductive bias of L to be the set of assumptions B such that for all new instances x_i

$$(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)$$

where the notation $y \vdash z$ indicates that z follows deductively from y (i.e., that z is provable from y). Thus, we define the inductive bias of a learner as the set of additional assumptions B sufficient to justify its inductive inferences as deductive inferences. To summarize,

Definition: Consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and let $D_c = \{\langle x, c(x)\rangle\}$ be an arbitrary set of training examples of c. Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c. **The inductive bias** of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

Inductive bias of CANDIDATE- ELIMINATION Algorithm

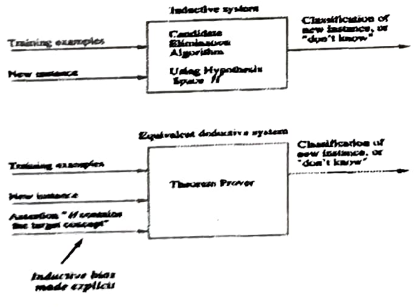


FIGURE
Modeling inductive systems by equivalent deductive systems. The input-output behavior of the CANDIDATE-ELIMINATION algorithm using a hypothesis space H is identical to that of a deductive theorem prover utilizing the assertion "H contains the target concept." This assertion is therefore called the *inductive bias* of the CANDIDATE-ELIMINATION algorithm. Characterizing inductive systems by their inductive bias allows modeling them by their equivalent deductive systems. This provides a way to compare inductive systems according to their policies for generalizing beyond the observed training data.

One advantage of viewing inductive inference systems in terms of their inductive bias is that it provides a nonprocedural means of characterizing their policy for generalizing beyond the observed data. A second advantage is that it allows comparison of different learners according to the strength of the inductive bias they employ. Consider, for example, the following three learning algorithms, which are listed from weakest to strongest bias.

1. ROTE-LEARNER: Learning corresponds simply to storing each observed training example in memory. Subsequent instances are classified by looking them up in memory. If the instance is found in memory, the stored classification is returned. Otherwise, the system refuses to classify the new instance.
2. CANDIDATE-ELIMINATION algorithm: New instances are classified only in the case where all members of the current version space agree on the classification. Otherwise, the system refuses to classify the new instance.
3. FIND-S: This algorithm, described earlier, finds the most specific hypothesis consistent with the training examples. It then uses this hypothesis to classify all subsequent instances.

KSIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

EXHAUSTIVE QUESTION BANK

Academic Year	2020-2021 [Odd Semester]		
Batch	2017-2021		
Year/Semester/section	IV/VII/A & B		
Subject Code-Title	17CS73- Machine Learning		
Name of the Instructor	Mr.RAGHAVENDRACHAR S	Dept	CSE

Questions

Module 1

1. **Identify** various applications of Machine Learning.
2. **Make use of** following examples to explain well posed learning problem.
   - i. Checkers learning problem
   - ii. Handwriting Recognition learning problem
   - iii. Robot Driving learning problem

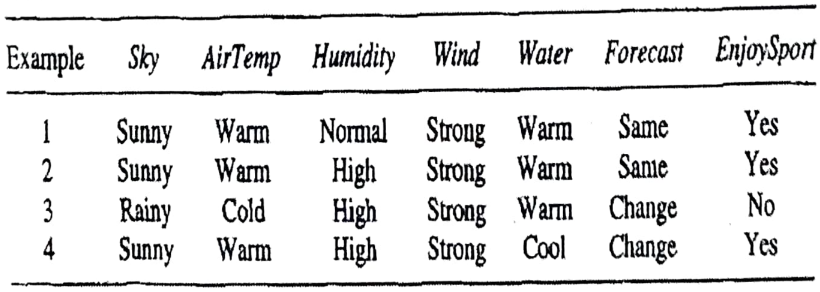
3. **Determine** the following with respect to checkers learning problem.
   - i. Choosing training experience
   - ii. Choosing Target function
   - iii. Choosing Representation of Target function
   - iv. Choosing function Approximation Algorithm

4. **Design** Checkers learning system using four distinct program modules.

5. **Identify** different issues in machine learning.

6. **Design** Find – S Algorithm.

7. **Apply** Find – S Algorithm for the given below target concept Enjoy sport.



Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

8. **Design** Candidate Elimination Algorithm.

9. **Apply** Candidate Elimination Algorithm for the given below target concept Enjoy sport.

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

10. **Design** List-Then Eliminate Algorithm

11. **Determine** Inductive Bias With respect to the following
    - i. Biased Hypothesis Space
    - ii. Unbiased Learner
    - iii. Futility of Bias-Free Learning

Module 2

1. **Identify** Appropriate problems for decision tree learning.

2. **Design** ID3 Algorithm

3. **Identify** the necessary measure required to select the attributes for building decision tree using ID3 Algorithm.

4. **Apply** ID3 Algorithm to construct the decision tree for the given target concept Play tennis.

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

5. **Make use of** suitable example, explain the following
    - **i.** Hypothesis Space Search In Decision Tree Learning
    - **ii.** Inductive Bias In Decision Tree Learning
    - **iii.** Issues In Decision Tree Learning

Module 3

1. **Determine** the application of neural network which is used for learning to steer an autonomous vehicle.

2. **Identify** the appropriate problem for Neural Network Learning.

3. **Make use of** suitable diagram, explain the concept of single perceptron.

4. **Design** and derive The Gradient Descent Rule

5. **Determine** why stochastic approximation is needed to Gradient descent rule?

6. **Design** Back propagation algorithm.

7. **Derive** Back propagation Rule **(Case i and Case ii)**.

8. **Identify** the remarks on Back propagation Algorithm.

Module 4

1. **Identify** features of Bayesian learning methods.

2. **Determine** Brute Force MAP Learning Algorithm.

3. **Determine** Minimum Description Length Principle.

4. **Apply** naïve bayes classifier for the given training data to classify the instance **(outlook = sunny, temperature = cool, humidity = high, wind = strong)**

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

5. **Determine** Bayesian belief network with representation.

6. **Determine** the following.
i. Conditional independence
ii. EM Algorithm
iii. Derivations of K means Algorithm

Module 5

1. **Determine** the following
i. Estimating Hypothesis Accuracy
ii. Sample Error and True Error
iii. Binomial Distribution

2. **Determine** the following
i. Confidence Intervals
ii. Central limit Theorem
iii. Comparing learning algorithms with paired t Tests.

3. **Determine** the following
i. Reinforcement learning
ii. Learning Task
iii. Learning (Q Function and Algorithm)

[COURSE INCHARGE]

CBCS SCHEME

USN

17CS73

Seventh Semester B.E. Degree Examination, July/August 2021

Machine Learning

Time: 3 hrs.

Max. Marks: 100

Note: Answer any FIVE full questions.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

1 a. Explain the designing of a learning system in detail. **(10 Marks)**

b. Define learning. Specify the learning problem for handwriting recognition and robot driving. **(05 Marks)**

c. Explain the issues in machine learning. **(05 Marks)**

2 a. Write the steps involved in find-S algorithm. **(05 Marks)**

b. Apply candidate elimination algorithm to obtain final version space for the training set shown in Table.Q2(b) to infer which books or articles the user reads based on keywords supplied in the article. **(10 Marks)**

Article	Crime	Academes	Local	Music	Reads
a_1	True	False	False	True	True
a_2	True	False	False	False	True
a_3	False	True	False	False	False
a_4	False	False	True	False	False
a_5	True	True	False	False	True

Table.Q2(b)

c. State the inductive bias rote-learner, candidate-elimination and Find-S algorithm. **(05 Marks)**

3 a. Define the following terms with an example for each:

(i) Decision tree (ii) Entropy (iii) Information gain
(iv) Restriction Bias (v) Preference Bias **(10 Marks)**

b. Construct decision tree for the data set shown in Table.Q3(b) to find whether a seed is poisonous or not. **(10 Marks)**

Example	Colour	Toughness	Fungus	Appearance	Poisonous
1	Green	Soft	Yes	Wrinkled	Yes
2	Green	Hard	Yes	Smooth	No
3	Brown	Soft	No	Wrinkled	No
4	Brown	Soft	Yes	Wrinkled	Yes
5	Green	Soft	Yes	Smooth	Yes
6	Green	Hard	No	Wrinkled	No
7	Orange	Soft	Yes	Wrinkled	Yes

Table.Q3(b)

4 a. Explain ID3 algorithm. Give an example. **(10 Marks)**

b. Explain the issues and solutions to those issues in decision tree learning. **(10 Marks)**

5 a. Derive an expression for gradient descent rule to minimize the error. Using the same, write the gradient descent algorithm for training a linear unit. **(10 Marks)**

b. Write back propagation algorithm that uses stochastic gradient descent method. What is the effect of adding momentum to the network? **(10 Marks)**

6 a. List the characteristics of the problems which can be solved using back propagation algorithm. **(05 Marks)**

b. Design a perceptron to implement two input AND function. **(05 Marks)**

c. Derive expressions for training rule of output and hidden unit weights for back propagation algorithm. **(10 Marks)**

7 a. Define Maximum a Posteriori (MAP) hypothesis. Derive an equation for MAP hypothesis using Baye's theorem. **(04 Marks)**

b. Given P(A = True) = 0.3, P(A = False) = 0.7, P(B = True | A = True) = 0.4, P(B = False | A = True) = 0.6, P(B = True | A = False) = 0.6, P(B = False | A = False) = 0.4. Calculate P(A = False/ B = False) using Baye's rule. **(06 Marks)**

c. Given the previous patient's data in the Table.Q7(c). Use Naïve Baye's classifics to classify the new data (Chills = Y, Runny nose = N, Headache = Mild, Fever = Y) to find whether the patient has flue or not. **(10 Marks)**

Chills	Runny nose	Headache	Fever	Flue
Y	N	Mild	Y	N
Y	Y	No	N	Y
Y	N	Strong	Y	Y
N	Y	Mild	Y	Y
N	N	No	N	N
N	Y	Strong	Y	Y
N	Y	Strong	N	N
Y	Y	Mild	Y	Y

Table.Q7(c)

8 a. Describe the features of Bayesian learning methods. **(05 Marks)**

b. A patient takes a lab test and the result comes back positive. It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result is only 97% of the cases. Furthermore only 0.008 of the entire population has this disease.

(i) What is the probability that this patient has cancer?

(ii) What is the probability that he does not have cancer? **(05 Marks)**

c. The Table.Q8(c) provides a set of 14 training examples of the target concept 'Play Tennis' where each day is described by the attributes, outlook, temperature, humidity and wind.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Table.Q8(c)

Use the Naïve Bayes classifier and the training data from this table to classify the following novel instance: <Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong> **(10 Marks)**

9 a. Explain binomial distribution and write the expressions for its probability distribution, mean, variance and standard deviation. **(04 Marks)**

b. Define the following terms:
   (i) Sample error
   (ii) True error
   (iii) N% confidence interval
   (iv) Random variable
   (v) Expected value
   (vi) Variance (06 Marks)

c. Write K-Nearest Neighbour algorithm for approximating a discrete values target function. Apply the same for the following three-dimensional training data instances along with one-dimensional output.

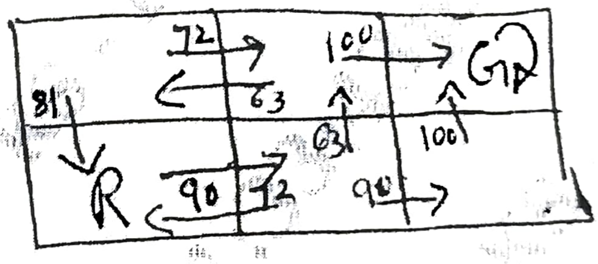
$x_1 = 5, x_2 = 7, x_3 = 3, y = 4$
$x_1 = 2, x_2 = 4, x_3 = 9, y = 8$
$x_1 = 3, x_2 = 8, x_3 = 1, y = 2$
$x_1 = 7, x_2 = 7, x_3 = 2, y = 4$
$x_1 = 1, x_2 = 9, x_3 = 7, y = 8$

Consider the query point ($x_1 = 5, x_2 = 3, x_3 = 4$) and K = 3. (10 Marks)

10 a. List the steps used for deriving confidence intervals. (04 Marks)

b. Explain CADIT system using case based reasoning. (06 Marks)

c. Write Q learning algorithm. Consider the following state s_1. Find $\hat{Q}(s_1, a_{right})$ for R given immediate reward as 0 and $\gamma = 0.9$. (10 Marks)



* * * * *

CBCS SCHEME

USN ☐☐☐☐☐☐☐☐☐☐ **15CS73**

Seventh Semester B.E. Degree Examination, July/August 2021
Machine Learning

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions.

1 a. Explain briefly steps involved in designing the learning system. **(08 Marks)**

b. Use example 1 given below to obtain most generalized hypothesis.
example 1.

x_1 = <Sunny, Warm, Normal, Strong, Warm, Same> +
x_2 = <Sunny, Warm, High, Strong, Warm, Same> +
x_3 = <Rainy, Cold, High, Strong, Warm, Change> –
x_4 = <Sunny, Warm, High, Strong, Cool, Change> + **(04 Marks)**

c. What are the limitations of find-S algorithm? **(04 Marks)**

2 a. Write candidate elimination algorithm and obtain version space description for the example2 given below.
Example 2:
(i) <Sunny, Warm, Normal, Strong, Warm, Same> = Yes
(ii) <Sunny, Warm, High, Strong, Warm, Same> = Yes
(iii) <Rainy, Cold, High, Strong, Warm, Change> = No
(iv) <Sunny, Warm, High, Strong, Cool, Change> = Yes **(10 Marks)**

b. Use example instances given below and obtain the proper classification. Give rational behind the class assigned to each instance. Use the hypothesis determined from example 2 at Q2(a).

x_1 = <Sunny, Warm, Normal, Strong, Cool, Change> = ?
x_2 = <Rain, Cold, Normal, Light, Warm, Same> = ?
x_3 = <Sunny, Warm, Normal, Light, Warm, Same> = ?
x_4 = <Sunny, Cold, Normal, Strong, Warm, Same> = ? **(06 Marks)**

3 a. What are the suitable characteristics to use tree based learning? **(04 Marks)**

b. Write ID3 algorithm and explain steps involved in it. **(06 Marks)**

c. Obtain decision tree for following expression:
(i) $A \wedge \neg B$
(ii) $A \vee [B \wedge C]$
(iii) A XOR B
(iv) $[A \wedge B] \vee [C \wedge D]$ **(06 Marks)**

4 a. Consider the example given in table 1. and determine suitable decision tree.

Table 1

Sl.No.	Outlook	Temperature	Humidity	Wind	Enjoy Sports
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

(12 Marks)

b. Explain limitations of ID3 algorithm. (04 Marks)

5 a. Using perceptron design Neural Network to solve following expression given in below truth table.

x_1	x_2	Output
0	0	0
0	1	0
1	0	0
1	1	1

(04 Marks)

b. Write algorithm for Gradient Descent learning and explain. (08 Marks)

c. What are the advantages of Artificial Neural Network? (04 Marks)

6 a. Derive the solution for weight update rule where the error is expressed as

$$\in_d(\vec{\omega}) = \frac{1}{2}(t_d - O_d)^2$$

(08 Marks)

b. Write Back propagation algorithm and explain. (08 Marks)

7 a. Use the following cancer diagnostic statics and priori probabilities given to determine P(cancer/⊕)

(i) 0.008 people can have cancer.

(ii) Positive test is correct in 98% cases.

(iv) The negative test is correct in 97% cases. (06 Marks)

b. Determine the posteriori probability for concept learning given $h \in H$ and D set of instances as training example. (06 Marks)

c. Show that maximum likelyhood hypothesis is equivalent to sum of minimum squared error.

$$h_{ML} = {}^{\text{argmin}} h \in H \sum_{i=1}^{m} (d_i - h(x_i))^2$$

(04 Marks)

8 a. Use the example given in table 1 at Q.No.4(a) and obtain classification for following instance by applying Naïve Bay's classifier.
Outlook = Sunny, Temperature = Cool, Humidity = High
Wind = Strong **(10 Marks)**

b. Explain Naïve Bay's classifier. **(06 Marks)**

9 a. Explain what is sample error and true error. **(04 Marks)**
b. What is confidence interval? How the confidence interval is determined? **(06 Marks)**
c. Write steps involved in comparing hypothesis using confidence interval. **(06 Marks)**

10 Write short note on following :
a. K-Nearest Neighbourhood learning
b. Locally Weighted Regression
c. Case Based Learning
d. Reinforced Learning **(16 Marks)**

* * * * *

CBCS SCHEME

USN ☐☐☐☐☐☐☐☐☐☐

17CS73

Seventh Semester B.E. Degree Examination, Jan./Feb. 2021

Machine Learning

Time: 3 hrs. Max. Marks: 100

Note: *Answer any FIVE full questions, choosing ONE full question from each module.*

Module-1

1 a. Define machine learning. Mention five applications of machine learning. **(06 Marks)**

b. Explain concept learning task with an example. **(06 Marks)**

c. Apply candidate elimination algorithm and obtain the version space considering the training examples given Table Q1(c).

Eyes	Nose	Head	Fcolor	Hair?	Smile?(TC)
Round	Triangle	Round	Purple	Yes	Yes
Square	Square	Square	Green	Yes	No
Square	Triangle	Round	Yellow	Yes	Yes
Round	Triangle	Round	Green	No	No
Square	Square	Round	Yellow	Yes	Yes

Table Q1(c) **(08 Marks)**

OR

2 a. Explain the following with respect to designing a learning system :

i) Choosing the training experience

ii) Choosing the target function

iii) Choosing a representation for the target function. **(09 Marks)**

b. Write Find–S algorithm. Apply the Find–S for Table Q1(c) to find maximally specific hypothesis. **(06 Marks)**

c. Explain the concept of inductive bias. **(05 Marks)**

Module-2

3 a. Explain the concept of decision tree learning. Discuss the necessary measures required to select the attributed for building a decision tree using ID3 algorithm. **(11 Marks)**

b. Explain the following with respect to decision tree learning :

i) Incorporating continuous-valued attributes

ii) Alternative measures for selecting attributes

iii) Handling training examples with missing attribute values. **(09 Marks)**

OR

4 a. Construct decision tree using ID3 considering the following training examples :

Weekend	Weather	Parental availability	Wealthy	Decision class
H_1	Sunny	Yes	Rich	Cinema
H_2	Sunny	No	Rich	Tennis
H_3	Windy	Yes	Rich	Cinema
H_4	Rainy	Yes	Poor	Cinema
H_5	Rainy	No	Rich	Home
H_6	Rainy	Yes	Poor	Cinema
H_7	Windy	No	Poor	Cinema
H_8	Windy	No	Rich	Shopping
H_9	Windy	Yes	Rich	Cinema
H_{10}	Sunny	No	Rich	Tennis

Table Q4(b) **(12 Marks)**

b. Discuss the issues of avoiding overfitting the data, and handling attributes with differing costs. **(08 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-3

5 a. Discuss the application of neural network which is used to steer an autonomous vehicle. **(06 Marks)**

b. Write Gradient descent algorithm to train a linear unit along with the derivation. **(08 Marks)**

c. Discuss the issues of convergence, local minima and generalization, overfitting and stopping criterion. **(06 Marks)**

OR

6 a. List the appropriate problems for neural network learning. **(05 Marks)**

b. Define preceptron and discuss its training rule. **(05 Marks)**

c. Show the derivation of back propagation training rule for output unit weights. **(10 Marks)**

Module-4

7 a. Explain Bayes theorem and mention the features of Bayesian learning. **(07 Marks)**

b. Prove that a maximum likelihood hypotheses cab be used to predict probabilities. **(08 Marks)**

c. Explain Naïve Bayes classifier. **(05 Marks)**

OR

8 a. Describe MAP learning algorithm. **(08 Marks)**

b. Classify the test data and {Red, SUV, Domestic} using Naine Bayes classifier for the dataset shown in Table Q8(b).

Color	Type	Origin	Stolen
Red	Sports	Domestic	Yes
Red	Sports	Domestic	No
Red	Sports	Domestic	Yes
Yellow	Sports	Domestic	No
Yellow	Sports	Imported	Yes
Yellow	SUV	Imported	No
Yellow	SUV	Imported	Yes
Yellow	SUV	Domestic	No
Red	SUV	Imported	No
Red	Sports	Imported	Yes

Table Q8(b) **(06 Marks)**

c. Write and explain EM algorithm. **(06 Marks)**

Module-5

9 a. Define :

i) Sample error

ii) True error

iii) Confidence intervals. **(06 Marks)**

b. Explain K-nearest neighbor learning algorithm. **(08 Marks)**

c. Write a note no Q – learning. **(06 Marks)**

OR

10 a. Define mean value, variance, standard deviation and estimation bias of a random variable. **(04 Marks)**

b. Explain locally weighted linear regression and radial basis functions. **(10 Marks)**

c. What is reinforcement learning? How it differs from other function approximation tasks? **(06 Marks)**

* * * * *

CBCS SCHEME

15CS73

USN | | | | | | | | | |

Seventh Semester B.E. Degree Examination, Aug./Sept.2020
Machine Learning

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg. 42+8 = 50, will be treated as malpractice.

Module-1

1 a. What is Machine Learning? Explain different perspectives and issues in machine learning. **(06 Marks)**

b. Explain the steps in designing a learning system. **(10 Marks)**

OR

2 a. Describe the Candidate-Elimination algorithm. Explain its working, taking the enjoy sport concept and training instances given below:

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy sport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Clod	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Warm	Change	Yes

(10 Marks)

b. Explain how to model inductive systems by their equivalent deductive systems for Candidate-Elimination Algorithm. **(06 Marks)**

Module-2

3 a. Explain the concepts of entropy and information gain. **(06 Marks)**

b. Describe the ID3 algorithm for decision tree learning. **(10 Marks)**

OR

4 a. Apply ID3 algorithm for constructing decision tree for the following training example.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

(10 Marks)

b. Explain the issues in decision tree learning. **(06 Marks)**

Module-3

5 a. Explain appropriate problems for Neural Network Learning with its characteristics. (10 Marks)
b. Explain the single perceptron with its learning algorithm. (06 Marks)

OR

6 a. Explain Back Propagation algorithm. (10 Marks)
b. Explain the remarks of Back propagation algorithm. (06 Marks)

Module-4

7 a. Explain Naïve Bayes classifier. (10 Marks)
b. Explain Bayesian Belief Networks. (06 Marks)

OR

8 a. Explain EM algorithm. (08 Marks)
b. Explain the derivation of K-means algorithm. (08 Marks)

Module-5

9 a. Explain K-nearest neighbor learning algorithm with example. (10 Marks)
b. Explain case based reasoning with example. (06 Marks)

OR

10 Write short note on:
a. Q learning
b. Radial basis function
c. Locally weighted regression
d. Sampling theory. (16 Marks)

* * * * *

CBCS SCHEME

USN | 1 | K | S | 1 | 6 | C | S | 1 | 0 | 7 |

15CS73

Seventh Semester B.E. Degree Examination, Dec.2019/Jan.2020

Machine Learning

Time: 3 hrs. Max. Marks: 80

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. What do you mean by well-posed learning problem? Explain with example. **(04 Marks)**

b. Explain the various stages involved in designing a learning system in brief. **(08 Marks)**

c. Write Find_S algorithm and discuss the issues with the algorithm. **(04 Marks)**

OR

2 a. List the issues in machine learning. **(04 Marks)**

b. Consider the given below training example which finds malignant tumors from MRI scans.

Example	Shape	Size	Color	Surface	Thickness	Target concept
1	Circular	Large	Light	Smooth	Thick	Malignant
2	Circular	Large	Light	Irregular	Thick	Malignant
3	Oval	Large	Dark	Smooth	Thin	Benign
4	Oval	Large	Light	Irregular	Thick	Malignant
5	Circular	Small	Light	Smooth	Thick	Benign

Show the specific and general boundaries of the version space after applying candidate elimination algorithm. (Note: Malignant is +ve, Benign is –ve). **(08 Marks)**

c. Explain the concept of inductive bias in brief. **(04 Marks)**

Module-2

3 a. Discuss the two approaches to prevent over fitting the data. **(08 Marks)**

b. Consider the following set of training examples:

Instance	Classification	a_1	a_2
1	1	1	1
2	1	1	1
3	0	1	0
4	1	0	0
5	0	0	1
6	0	0	1

(i) What is the entropy of this collection of training examples with respect to the target function classification?

(ii) What is the information gain of a_2 relative to these training examples? **(08 Marks)**

OR

4 a. Define decision tree. Construct the decision tree to represent the following Boolean functions:

i) $A \wedge \neg B$ ii) $A \vee [B \wedge C]$ iii) A XOR B **(06 Marks)**

b. Write the ID3 algorithm. **(06 Marks)**

c. What do you mean by gain and entropy? How it is used to build the decision tree. **(04 Marks)**

Module-3

5 a. Define perceptron. Explain the concept of single perceptron with neat diagram. **(06 Marks)**
b. Explain the back propagation algorithm. Why is it not likely to be trapped in local minima? **(10 Marks)**

OR

6 a. List the appropriate problems for neural network learning. **(04 Marks)**
b. Discuss the perceptron training rule and delta rule that solves the learning problem of perceptron. **(08 Marks)**
c. Write a remark on representation of feed forward networks. **(04 Marks)**

Module-4

7 a. Explain Naïve Bayes classifier. **(08 Marks)**
b. Explain brute force MAP learning algorithm. **(08 Marks)**

OR

8 a. Discuss Minimum Description Length principle in brief. **(08 Marks)**
b. Explain Bayesian belief networks and conditional independence with example. **(08 Marks)**

Module-5

9 a. Define: (i) Simple Error (ii) True Error **(04 Marks)**
b. Explain K-nearest neighbor learning algorithm. **(08 Marks)**
c. What is reinforcement learning? **(04 Marks)**

OR

10 a. Define expected value, variance, standard deviation and estimate bias of a random variable. **(04 Marks)**
b. Explain locally weighted linear regression. **(08 Marks)**
c. Write a note on Q-learning. **(04 Marks)**

* * * * *

CBCS SCHEME

USN | | | | | | | | | |

15CS73

Seventh Semester B.E. Degree Examination, Dec.2018/Jan.2019

Machine Learning

Time: 3 hrs. Max. Marks: 80

Note: *Answer FIVE full questions, choosing ONE full question from each module.*

Module-1

1 a. Specify the learning task for 'A checkers learning problem'. (03 Marks)

b. Discuss the following with respect to the above,

(i) Choosing the training experience.
(ii) Choosing the target function and
(iii) Choosing a function approximation algorithm. (09 Marks)

c. Comment on the issues in machine learning. (04 Marks)

OR

2 a. Write candidate elimination algorithm. Apply the algorithm to obtain the final version space for the training example. (10 Marks)

Sl. No.	Sky	Air temp	Humidity	Wind	Water	Forecast	Enjoy sport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

b. Discuss about an unbiased Learner. (06 Marks)

Module-2

3 a. What is a decision tree & discuss the use of decision tree for classification purpose with an example. (08 Marks)

b. Write and explain decision tree for the following transactions: (08 Marks)

Tid	Refund	Martial status	Taxable Income	Cheat
1	Yes	Single	125 K	No
2	No	Married	100 K	No
3	No	Single	70 K	No
4	Yes	Married	120 K	No
5	No	Divorced	95 K	Yes
6	No	Married	60 K	No
7	Yes	Divorced	220 K	No
8	No	Single	85 K	Yes
9	No	Married	75 K	No
10	No	Single	90 K	Yes

OR

4 a. For the transactions shown in the table compute the following :

(i) Entropy of the collection of transaction records of the table with respect to classification.
(ii) What are the information gain of a_1 and a_2 relative to the transactions of the table? (08 Marks)

Instance	1	2	3	4	5	6	7	8	9
a_1	T	T	T	F	F	F	F	T	F
a_2	T	T	F	F	T	T	F	F	T
Target class	+	+	–	+	–	–	–	+	–

b. Discuss the decision learning algorithm. (04 Marks)

c. List the issues of decision tree learning. (04 Marks)

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg. 42+8 = 50, will be treated as malpractice.

Module-3

5 a. Draw the perceptron network with the notation. Derive an equation of gradient descent rule to minimize the error. **(08 Marks)**

b. Explain the importance of the terms : (i) Hidden layer (ii) Generalization (iii) Overfitting (iv) Stopping criterion **(08 Marks)**

OR

6 a. Discuss the application of Neural network which is used for learning to steer an autonomous vehicle. **(06 Marks)**

b. Write an algorithm for back propagation algorithm which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network. **(10 Marks)**

Module-4

7 a. What is Bayes theorem and maximum posterior hypothesis? **(04 Marks)**

b. Derive an equation for MAP hypothesis using Bayes theorem. **(04 Marks)**

c. Consider a football game between two rival teams: Team 0 and Team 1. Suppose Team 0 wins 95% of the time and Team 1 wins the remaining matches. Among the games won by team 0, only 30% of them come from playing on teams 1's football field. On the otherhand, 75% of the victories for team 1 are obtained while playing at home. If team 1 is to host the next match between the two teams, which team will most likely emerge as the winner? **(08 Marks)**

OR

8 a. Describe Brute-force MAP learning algorithm. **(04 Marks)**

b. Discuss the Naïve Bayees classifier. **(04 Marks)**

c. The following table gives data set about stolen vehicles. Using Naïve bayes classifier classify the new data (Red, SUV, Domestic) **(08 Marks)**

Table

Color	Type	Origin	Stolen
Red	Sports	Domestic	Yes
Red	Sports	Domestic	No
Red	Sports	Domestic	Yes
Yellow	Sports	Domestic	No
Yellow	Sports	Imported	Yes
Yellow	SUV	Imported	No
Yellow	SUV	Imported	Yes
Yellow	SUV	Domestic	No
Red	SUV	Imported	No
Red	Sports	Imported	Yes

Module-5

9 a. Write short notes on the following:

(i) Estimating Hypothesis accuracy.

(ii) Binomial distribution. **(08 Marks)**

b. Discuss the method of comparing two algorithms. Justify with paired to tests method. **(08 Marks)**

OR

10 a. Discuss the K-nearest neighbor language. **(04 Marks)**

b. Discuss locally weighted Regression. **(04 Marks)**

c. Discuss the learning tasks and Q learning in the context of reinforcement learning. **(08 Marks)**

* * * * *

K.S.INSTITUTE OF TECHNOLOGY, BENGALURU - 109
DEPARTMENT OF COMPUTER SCIENCE & ENGG
COURSE END SUVERY -ACADEMIC YEAR (SEP 2020 - JAN 2021 (ODD SEM))
VII Sem A Section

Sub Code:17CS73 **Faculty Name:Mr. RAGHAVENDRACHAR.S** **Subject: MACHINE LEARNING**

QUESTIONARE

1	**To what extent you understand concept learning algorithms ?**
2	**How efficient you understand the concepts of Decesion Tree Learning?**
3	**How efficient you understand the concepts of Artificial Neural Networks?**
4	**How efficient you understand the concepts of Bayesian Learning?**
5	**How efficient you understand the concepts of Instance based Learning?**
6	**How efficient you understand the concepts of Reinforcement Learning?**

Sl. No.	Name of the Student	USN	SEM / SECTION	1	2	3	4	5	6
1	Indrasena kalyanam	1KS17CS030	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
2	P KISHORE	1KS17CS051	VII A	VERY GOOD	VERY GOOD	VERY GOOD	GOOD	EXCELLENT	EXCELLENT
3	DARSHAN S	1KS17CS020	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
4	GANESH MAUDGHALYA H G	1KS17CS025	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
5	AAFREEN HUSSAIN	1KS17CS001	VII A	GOOD	GOOD	SATISFACTORY	SATISFACTORY	GOOD	GOOD
6	AMOGH R	1KS17CS005	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	VERY GOOD	VERY GOOD
7	MEGHANA	1KS17CS041	VII A	VERY GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
8	CHENNAKESHAVA NT	1KS17CS019	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	VERY GOOD	VERY GOOD
9	MANJUNATH A	1KS17CS040	VII A	VERY GOOD	EXCELLENT	EXCELLENT	EXCELLENT	VERY GOOD	VERY GOOD
10	KAVITHA. s	1KS17CS034	VII A	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	GOOD
11	NIKITA KATARI	1KS17CS047	VII A	VERY GOOD	VERY GOOD	GOOD	GOOD	GOOD	GOOD
12	GANESH G B	1KS17CS024	VII A	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
13	AKSHITHA	1KS17CS004	VII A	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD

Sl. No.	Name of the Student	USN	SEM / SECTION	1	2	3	4	5	6
14	KARTHIK T C	1KS17CS033	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	VERY GOOD	GOOD
15	KRITHIKA JAGANNATH	1KS17CS036	VII A	VERY GOOD	VERY GOOD	EXCELLENT	VERY GOOD	VERY GOOD	EXCELLENT
16	ABHISHEK GOWDA M V	1KS17CS002	VII A	EXCELLENT	EXCELLENT	VERY GOOD	EXCELLENT	VERY GOOD	VERY GOOD
17	JANARDHAN RAJU B	1KS18CS400	VII A	EXCELLENT	EXCELLENT	VERY GOOD	VERY GOOD	EXCELLENT	VERY GOOD
18	MEGHANA G R	1KS17CS043	VII A	VERY GOOD	EXCELLENT	EXCELLENT	VERY GOOD	EXCELLENT	EXCELLENT
19	CHANDANA B R	1ks17cs018	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
20	Nikhil subramanya	1KS17CS046	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
21	AMOGHA MANJUNATHA K	1KS17CS006	VII A	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
22	AMRUTHA.V.DESHPANDE	1KS17CS007	VII A	VERY GOOD	VERY GOOD	EXCELLENT	VERY GOOD	EXCELLENT	EXCELLENT
23	DEEPIKA S H	1KS17CS022	VII A	GOOD	SATISFACTORY	GOOD	VERY GOOD	VERY GOOD	VERY GOOD
24	NISHCHITHA C	1KS17CS048	VII A	VERY GOOD	VERY GOOD	VERY GOOD	EXCELLENT	EXCELLENT	EXCELLENT
25	KRUTHIKA B M	1KS18CS401	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
26	SHRIRAKSHA S KANAGO	1KS17CS102	VII A	VERY GOOD	VERY GOOD	EXCELLENT	VERY GOOD	EXCELLENT	EXCELLENT
27	LAVANYA V	1KS17CS037	VII A	EXCELLENT	VERY GOOD	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
28	ASHISH K AMAR	1KS17CS013	VII A	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	EXCELLENT	VERY GOOD
29	B LAKSHMI PRASANNA	1KS17CS014	VII A	GOOD	GOOD	GOOD	GOOD	VERY GOOD	VERY GOOD
30	ANUSHREE J	1KS17CS011	VII A	VERY GOOD	VERY GOOD	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
31	GAUTHAM C R	1KS17CS026	VII A	VERY GOOD	VERY GOOD	VERY GOOD	GOOD	VERY GOOD	VERY GOOD
32	Nitish Kumar Gupta	1KS17CS049	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
33	Anoop P S	1KS17CS008	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
34	ANUSHA AG	1KS17CS010	VII A	GOOD	GOOD	VERY GOOD	GOOD	GOOD	GOOD
35	MEGHANA. H. S	1KS16CS042	VII A	EXCELLENT	VERY GOOD	VERY GOOD	VERY GOOD	EXCELLENT	EXCELLENT

Sl. No.	Name of the Student	USN	SEM / SECTION	1	2	3	4	5	6
36	HARSHITHA V	1KS17CS029	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
37	MEGHANA.G	1KS17CS042	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
38	AKSHATHA RAMESH	1KS17CS003	VII A	GOOD	GOOD	GOOD	VERY GOOD	GOOD	GOOD
39	NYDILE G R	1KS17CS050	VII A	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD
40	BHAVESH BHANSALI	1KS17CS016	VII A	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD	VERY GOOD
41	CHAITRA	1KS17CS017	VII A	VERY GOOD	GOOD	VERY GOOD	GOOD	VERY GOOD	EXCELLENT
42	MOUNIKA M K L	1KS17CS044	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
43	KEERTHI.N	IKS17CS035	VII A	VERY GOOD	GOOD	VERY GOOD	VERY GOOD	EXCELLENT	EXCELLENT
44	anuhya kulkarni	1KS17CS009	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
45	KARAN RAGHUNATH	1KS17CS032	VII A	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT	EXCELLENT
46	LOKESH	1KS17CS038	VII A	VERY GOOD	EXCELLENT	EXCELLENT	VERY GOOD	SATISFACTORY	SATISFACORY
47	NEHA K	1KS17CS045	VII A	VERY GOOD	EXCELLENT	EXCELLENT	EXCELLENT	VERY GOOD	EXCELLENT

Signature of Faculty

K S INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

YEAR / SEMESTER	**IV / VII**
COURSE TITLE	**MACHINE LEARNING**
COURSE CODE	**17CS73**
ACADEMIC YEAR	**2020-2021**

SL No	USN	Name of the Student	17CS73														
			IA1		A1		IA2			A2			IA3		A3		SEE
			CO1	CO2	CO1	CO2	CO2	CO3	CO4	CO2	CO3	CO4	CO4	CO5	CO4	CO5	
			18	**12**	**6**	**4**	**6**	**18**	**6**	**2**	**6**	**2**	**12**	**18**	**4**	**6**	**60**
1	1KS17CS001	AAFREEN HUSSAIN	17	11	6	4	6	17	5	2	6	2	7	6	4	6	31
2	1KS17CS002	ABHISHEK GOWDA.M.V	18	12	6	4	6	17	5	2	6	2	12	18	4	6	38
3	1KS17CS003	AKSHATHA RAMESH	18	11	6	4	6	17	5	2	6	2	12	12	4	6	42
4	1KS17CS004	AKSHITHA.B.S	18	12	6	4	5	17	6	2	6	2	12	6	4	6	22
5	1KS17CS005	AMOGH.R	17	12	6	4	6	16	5	2	6	2	12	14	4	6	25
6	1KS17CS006	AMOGHA MANJUNATHA.K	16	12	6	4	6	17	5	2	6	2	12	18	4	6	29
7	1KS17CS007	AMRUTHA.V.DESHPANDE	18	12	6	4	6	17	5	2	6	2	12	12	4	6	26
8	1KS17CS008	ANOOP.P.S	17	12	6	4	6	18	5	2	6	2	12	12	4	6	36
9	1KS17CS010	ANUSHA.A.G	17	12	6	4	6	17	5	2	6	2	12	6	4	6	41
10	1KS17CS011	ANUSHREE.J	18	12	6	4	6	17	5	2	6	2	12	6	4	6	33
11	1KS17CS013	ASHISH.K.AMAR	16	12	6	4	5	17	5	2	6	2	12	18	4	6	32
12	1KS17CS014	LAKSHMI PRASANNA.B	18	12	6	4	6	17	6	2	6	2	12	15	4	6	36
13	1KS17CS016	BHAVESH BHANSALI	17	12	6	4	6	17	5	2	6	2	12	6	4	6	26
14	1KS17CS017	CHAITRA	17	12	6	4	6	17	5	2	6	2	12	15	4	6	44
15	1KS17CS018	CHANDANA.B.R	18	12	6	4	6	17	5	2	6	2	12	18	4	6	39
16	1KS17CS019	CHENNA KESHAVA.N.T	17	12	6	4	6	17	5	2	6	2	6	18	4	6	40
17	1KS17CS020	DARSHAN.S	18	12	6	4	6	15	5	2	6	2	6	6	4	6	10
18	1KS17CS021	DEEKSHITHA.R	16	11	6	4	6	17	6	2	6	2	12	6	4	6	25
19	1KS17CS022	DEEPIKA.S.H	18	12	6	4	6	17	5	2	6	2	12	16	4	6	26
20	1KS17CS023	DIVYA YASHASWI KANNEY	18	12	6	4	6	17	5	2	6	2	NA	NA	4	6	AB
21	1KS17CS024	GANESH.G.B	18	11	6	4	6	17	5	2	6	2	12	6	4	6	32
22	1KS17CS025	GANESH MAUDGHALYA.H.G	17	9	6	4	6	17	5	2	6	2	12	12	4	6	21
23	1KS17CS026	GAUTHAM.C.R	17	9	6	4	6	16	5	2	6	2	12	6	4	6	33
24	1KS17CS027	H.PRIYANKA	18	12	6	4	6	16	5	2	6	2	12	18	4	6	50
25	1KS17CS028	HANUMESH.V.T	16	12	6	4	6	17	5	2	6	2	12	13	4	6	33
26	1KS17CS029	HARSHITHA.V	18	12	6	4	6	17	5	2	6	2	12	13	4	6	21
27	1KS17CS030	INDRASENA KALYANAM	17	12	6	4	6	18	5	2	6	2	12	16	4	6	24
28	1KS17CS032	KARAN RAGHUNATH	16	11	6	4	6	17	5	2	6	2	12	12	4	6	29

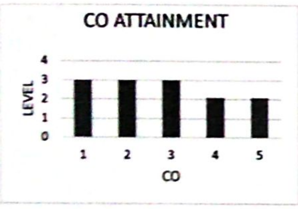
Sl. No	USN	Name of the Student	17CS73														
			IA1		A1		IA2			A2			IA3		A3		SEE
			CO1	CO2	CO1	CO2	CO2	CO3	CO4	CO2	CO3	CO4	CO4	CO5	CO4	CO5	
			18	12	6	4	6	18	6	2	6	2	12	18	4	6	60
29	1KS17CS033	KARTHIK.T.C	18	12	6	4	5	17	5	2	6	2	12	12	4	6	40
30	1KS17CS034	KAVITHA.S	18	12	6	4	6	17	5	2	6	2	12	12	4	6	34
31	1KS17CS035	KEERTHI.N	18	12	6	4	6	17	5	2	6	2	12	12	4	6	26
32	1KS17CS036	KRITHIKA JAGANNATH	18	12	6	4	6	17	5	2	6	2	12	18	4	6	40
33	1KS17CS037	LAVANYA.V	15	12	6	4	6	17	5	2	6	2	12	16	4	6	13
34	1KS17CS038	LOKESH.B.M	16	12	6	4	6	17	5	2	6	2	10	9	4	6	24
35	1KS17CS040	MANJUNATH.A	17	11	6	4	6	17	5	2	6	2	12	12	4	6	29
36	1KS17CS041	MEGHANA.C.V	18	12	6	4	6	17	5	2	6	2	12	18	4	6	32
37	1KS17CS042	MEGHANA.G	18	12	6	4	6	17	5	2	6	2	12	12	4	6	30
38	1KS17CS043	MEGHANA.G.R	14	12	6	4	6	17	5	2	6	2	12	18	4	6	29
39	1KS17CS044	MOUNIKA.M.K.L	18	12	6	4	6	17	5	2	6	2	12	6	4	6	35
40	1KS17CS045	NEHA.K	17	12	6	4	6	17	5	2	6	2	12	6	4	6	31
41	1KS17CS046	NIKHIL SUBRAMANYA.K	16	12	6	4	6	17	5	2	6	2	12	14	4	6	27
42	1KS17CS047	NIKITHA KATARI	18	11	6	4	5	16	5	2	6	2	11	17	4	6	24
43	1KS17CS048	NISCHITHA.C	18	12	6	4	6	17	5	2	6	2	12	6	4	6	21
44	1KS17CS049	NITISH KUMAR GUPTA	12	11	6	4	6	17	NA	2	6	2	12	15	4	6	21
45	1KS17CS050	NYDILE.G.R	18	12	6	4	6	17	5	2	6	2	12	6	4	6	26
46	1KS17CS051	P.KISHORE	17	12	6	4	6	17	5	2	6	2	12	12	4	6	32
47	1KS17CS052	PARTH.P.SHAH	16	12	6	4	6	17	6	2	6	2	6	18	4	6	39
48	1KS17CS053	PAVANI.M	16	12	6	4	6	17	6	2	6	2	10	18	4	6	48
49	1KS17CS055	POOJA.R	17	12	6	4	6	17	6	2	6	2	12	18	4	6	25
50	1KS17CS056	PRASHANTH.K	16	11	6	4	5	18	6	2	6	2	5	10	4	6	33
51	1KS17CS057	PRAVEEN	14	8	6	4	5	15	5	2	6	2	NA	9	4	6	38
52	1KS17CS058	PRAVEEN.A	16	12	6	4	5	18	4	2	6	2	NA	4	4	6	32
53	1KS17CS059	PRAVEEN.S	16	10	6	4	5	18	6	2	6	2	11	14	4	6	21
54	1KS17CS060	RAJASHREE SHIVAKUMAR	18	12	6	4	5	18	6	2	6	2	3	18	4	6	29
55	1KS17CS061	RAKSHITH.R	11	12	6	4	5	18	6	2	6	2	6	12	4	6	28
56	1KS17CS062	ROHITH.K	16	11	6	4	5	18	6	2	6	2	6	18	4	6	21
57	1KS17CS063	ROHITH.R	17	11	6	4	5	18	6	2	6	2	2	12	4	6	24
58	1KS17CS064	ROOPASHREE.N	17	12	6	4	5	18	6	2	6	2	6	18	4	6	42
59	1KS17CS065	ROSHINI.R	17	10	6	4	5	18	5	2	6	2	12	18	4	6	36
60	1KS17CS066	RUCHITHA.G.K	17	12	6	4	5	18	6	2	6	2	6	18	4	6	41
61	1KS17CS067	S.MONIKA	18	11	6	4	5	18	5	2	6	2	6	12	4	6	28
62	1KS17CS069	SAI KUMAR.L.S	16	11	6	4	6	18	6	2	6	2	9	12	4	6	41
63	1KS17CS070	SAI SNEHA.S.V	18	12	6	4	6	18	6	2	6	2	12	18	4	6	33
64	1KS17CS071	SAKSHI KUMARI	18	12	6	4	5	18	6	2	6	2	12	12	4	6	31
65	1KS17CS072	SANDESH NAIKAL	17	12	6	4	5	18	6	2	6	2	9	15	4	6	27
66	1KS17CS074	SHARANYA.H	12	12	6	4	5	6	6	2	6	2	NA	NA	4	6	12
67	1KS17CS075	SHASHANK SHET.K	18	12	6	4	5	18	6	2	6	2	12	18	4	6	33
68	1KS17CS076	SHREYAS.R	18	12	6	4	5	18	6	2	6	2	12	6	4	6	21

SL. No	USN	Name of the Student	17CS73														
			IA1		A1		IA2			A2			IA3		A3		SEE
			CO1	CO2	CO1	CO2	CO2	CO3	CO4	CO2	CO3	CO4	CO4	CO5	CO4	CO5	
			18	**12**	**6**	**4**	**6**	**18**	**6**	**2**	**6**	**2**	**12**	**18**	**4**	**6**	**60**
69	1KS17CS077	SHRI HARSHA KULKARNI	11	12	6	4	5	15	5	2	6	2	NA	12	4	6	25
70	1KS17CS078	SINDHU.H.S	17	12	6	4	6	18	6	2	6	2	12	18	4	6	34
71	1KS17CS079	SINDHU.M	17	12	6	4	5	18	6	2	6	2	8	12	4	6	41
72	1KS17CS081	SPOORTHI.R	18	12	6	4	6	18	6	2	6	2	6	18	4	6	33
73	1KS17CS082	SPOORTHI.V	18	12	6	4	5	18	6	2	6	2	NA	NA	4	6	33
74	1KS17CS083	SRIKALA.K.M	17	12	6	4	5	17	6	2	6	2	12	18	4	6	46
75	1KS17CS084	SRUSHTI.A	18	12	6	4	6	18	6	2	6	2	12	12	4	6	32
76	1KS17CS085	SUJANA.G.N	17	12	6	4	5	18	6	2	6	2	12	18	4	6	26
77	1KS17CS086	SUPREETHA.R.KASHYAP	16	12	6	4	5	18	6	2	6	2	10	18	4	6	30
78	1KS17CS087	SUPRIYA.K	17	12	6	4	5	18	6	2	6	2	12	18	4	6	39
79	1KS17CS088	SURAKSHITHA.M	18	12	6	4	6	18	6	2	6	2	6	18	4	6	43
80	1KS17CS089	SWATI PAI	18	12	6	4	5	18	6	2	6	2	12	18	4	6	32
81	1KS17CS090	T.K.DHANUSHREE	18	12	6	4	5	18	6	2	6	2	6	12	4	6	29
82	1KS17CS091	TALUPULA SIVA SAI CHAITHANYA	17	12	6	4	5	18	6	2	6	2	10	16	4	6	24
83	1KS17CS092	TEJAS.C.S	17	12	6	4	6	18	6	2	6	2	12	18	4	6	51
84	1KS17CS093	THARUN.K	18	12	6	4	5	18	6	2	6	2	2	12	4	6	34
85	1KS17CS094	VARSHINI.N.PRAKASH	14	12	6	4	6	12	6	2	6	2	5	12	4	6	27
86	1KS17CS095	VARSHITHA.S	18	12	6	4	5	18	6	2	6	2	8	15	4	6	27
87	1KS17CS096	VARUN ATTIGANAL VENKATESH	17	12	6	4	5	18	6	2	6	2	12	18	4	6	29
88	1KS17CS097	VARUN.R.REDDY	18	12	6	4	5	18	6	2	6	2	4	16	4	6	8
89	1KS17CS098	VIKRAM SHIVAPPA CHATTARAKI	17	12	6	4	5	18	5	2	6	2	NA	18	4	6	9
90	1KS17CS099	VINAY RAMARAO BIRADAR	17	12	6	4	5	16	5	2	6	2	4	12	4	6	23
91	1KS17CS100	VISHAL.M.S	17	12	6	4	5	18	6	2	6	2	6	12	4	6	29
92	1KS17CS101	VYBHAVI.J	18	12	6	4	5	18	6	2	6	2	12	18	4	6	33
93	1KS17CS102	SHRIRAKSHA.S.KANAGO	17	12	6	4	6	17	5	2	6	2	12	13	4	6	21
94	1KS18CS401	KRUTHIKA.B.M	17	11	6	4	6	17	5	2	6	2	NA	NA	4	6	13
60% of Maximum marks (X)			11	07	04	02	04	11	04	01	04	01	07	11	02	04	36
No. of students above X			94	94	94	94	94	93	93	94	94	94	66	72	94	94	23
Total number of students (Y)			94	94	94	94	94	94	93	94	94	94	86	90	94	94	93
CO Percentage			100.00	100.00	100.00	100.00	100.00	98.94	100.00	100.00	100.00	100.00	76.74	80.00	100.00	100.00	24.73
			CO1	CO2	CO1	CO2	CO2	CO3	CO4	CO2	CO3	CO4	CO4	CO5	CO4	CO5	SEE

CO	CIE	SEE	DIRECT ATTAINMENT	Level	COURSE EXIT SURVEY	LEVEL	ATTAINMENT
CO1	100.00	24.73	62.37	3.00	98.05	3.00	3
CO2	100.00	24.73	62.37	3.00	98.05	3.00	3
CO3	99.47	24.73	62.10	3.00	98.05	3.00	3
CO4	94.19	24.73	59.46	2.00	98.05	3.00	2.1
CO5	90.00	24.73	57.37	2.00	98.05	3.00	2.1
AVERAGE							2.64

	IA1	A1	IA2	A2	IA3	A3	AVG
CO1	100.00	100.00					100.00
CO2	100.00	100.00	100.00	100			100.00
CO3			98.94	100			99.47
CO4			100.00	100	76.74	100	94.19
CO5					80.00	100	90.00

CO Attainment Level	Significance	For Direct attainment , 50% of CIE and 50% of SEE marks are considered.
Level 3	60% and above students should have scored >= 60% of Total marks	For indirect attainment, Course end survey is considered.
Level 2	55% to 59% of students should have scored >= 60% of Total marks	CO attainment is 90%of direct attainment + 10% of Indirect atttainment.
Level 1	50% to 54% of students should have scored >= 60% of Total marks	PO attainment = CO-PO mapping strength/3 * CO attainment .



Co-Po Mapping Table

CO'S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS01	PS02
CO1	3	2	2	-	2	-	-	-	-	1.00	-	1.00	3	2
CO2	3	2	2	-	2	-	-	-	-	1.00	-	1.00	3	2
CO3	3	2	2	-	2	-	-	-	-	1.00	-	1.00	3	2
CO4	3	2	2	-	2	-	-	-	-	1.00	-	1.00	3	2
CO5	3	2	2	-	2	-	-	-	-	1.00	-	1.00	3	2
AVG	3.00	2.00	2.00	-	2.00	-	-	-	-	1.00	-	1.00	3.00	2.00

PO ATTAINMENT TABLE

CO'S	CO Attainment in %	CO RESULT	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO1
CO1	3.00	Y	3.00	2.00	2.00	-	2.00	-	-	-	-	1.00	-	1.00	3.00	2.00
CO2	3.00	Y	3.00	2.00	2.00	-	2.00	-	-	-	-	1.00	-	1.00	3.00	2.00
CO3	3.00	Y	3.00	2.00	2.00	-	2.00	-	-	-	-	1.00	-	1.00	3.00	2.00
CO4	2.10	Y	2.10	1.40	1.40	-	1.40	-	-	-	-	0.70	-	0.70	2.10	1.40
CO5	2.10	Y	2.10	1.40	1.40	-	1.40	-	-	-	-	0.70	-	0.70	2.10	1.40
Average			2.64	1.76	1.76	-	1.76	-	-	-	-	0.88	-	0.88	2.64	1.76

Course Incharge

HOD

Head of the Department
Dept. of Computer Science & Engg
K.S. Institute of Technology
Bengaluru -560 109