

**Dr Ambedkar Institute of Technology**  
**Bangalore-56**



**Department of  
Computer Science & Engineering**

**Laboratory Manual**

*For*  
**Machine Learning Laboratory**  
**(18CSL66)**  
**(6<sup>th</sup> Semester B.E., CSE)**

**2020-21**

*Lab in charge:*

*Prof. Asha K N*

*Prof. Asha Rani K P*



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering  
**Machine Learning [18CSL66] Lab Manual**

## A. LABORATORYOVERVIEW

Degree:	<b>BE</b>	Programme:	<b>CS</b>
Semester:	<b>6</b>	Academic Year:	<b>2020-21</b>
Laboratory Title:	<b>Machine Learning Laboratory</b>	Laboratory Code:	<b>18CSL66</b>
L-T-P-S:	<b>0-0-1-0</b>	Duration of SEE:	<b>3Hrs</b>
CIE Marks:	<b>50</b>	SEE Marks:	<b>50</b>
Lab Manual Author:	<b>Asha K N Asha Rani K P</b>	Sign	

## B. DESCRIPTION

### 1.PREREQUISITES:

- Creative thinking, sound mathematical insight and programming skills.
- Design and Analysis of Algorithms
- Python Programming

### 2.BASE COURSE:

- **Machine Learning(18CS62)**

### 3.COURSE OBJECTIVES/OUTCOMES:

#### Course Objectives:

This course will enable students to

1. Implement the machine learning algorithms using the Data Set.
2. Learn to use Various python tools for Machine Learning
3. Analyze and interpret the outcomes of the machine learning algorithms.

#### Course Outcomes:

Course Outcomes	Description	RBT Levels
The students should be able to:		
<b>CO1</b>	Understand and interpret the implementation procedures and python Libraries for the machine learning algorithms.	<b>L2</b>
<b>CO2</b>	Analyse the correctness of the datasets to apply appropriate Machine Learning algorithms.	<b>L3</b>
<b>CO3</b>	Design and implement Machine Learning algorithms to solve real world problems.	<b>L4</b>
<b>CO4</b>	Evaluate and interpret the results of the machine learning algorithms.	<b>L5</b>



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 4. RESOURCES REQUIRED:

- Hardware resources
  - Desktop PC
  - Windows/Linux operating system
- Software resources
  - Python
  - Anaconda IDE with Jupiter
- Datasets from standard repositories (Ex: <https://archive.ics.uci.edu/ml/datasets.html>)

**Prepared by**

**Asha K N**

**Asha Rani K P**

**Dr. Siddaraju**

**HOD, CSE**

### 5. RELEVANCE OF THE COURSE:

- To carry out Mini project and Main Project work

### 6. GENERAL INSTRUCTIONS:

- Implement the program in Python IDE – Jupiter and demonstrate the same.
- External practical examination.
  - All laboratory experiments are to be included
  - Students are allowed to pick one experiment from the lot.
  - Marks distribution: Procedure + Conduction + Viva: 10+30+10(50)
  - Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 7.CONTENTTS:

ExptNo.	Title of the Experiments	RBT	CO
1	Implement and demonstrate the <b>FIND-S algorithm</b> for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.	L3	CO1, 2,3,4
2	For a given set of training data examples stored in a .CSV file, implement and demonstrate the <b>Candidate-Elimination algorithm</b> to output a description of the set of all hypotheses consistent with the training examples.	L3	CO1, 2,3,4
3	Write a program to demonstrate the working of the <b>decision tree</b> based <b>ID3 algorithm</b> . Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.	L3	CO1, 2,3,4
4	Build an <b>Artificial Neural Network</b> by <b>implementing the Back propagation algorithm</b> and test the same using appropriate data sets.	L3	CO1, 2,3,4
5	Write a program to implement the naïve <b>Bayesian classifier</b> for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.	L3	CO1, 2,3,4
6	Assuming a set of documents that need to be classified, use the <b>naïve Bayesian Classifier model</b> to perform this task. Built-in API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.	L3	CO1, 2,3,4
7	Write a program to construct a <b>Bayesian network</b> considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Python ML library classes/API.	L3	CO1, 2,3,4
8	Apply <b>EM algorithm</b> to cluster a set of data stored in a .CSV file. Use the same data set for clustering <b>using k-Means algorithm</b> . Compare the results of these two algorithms and comment on the quality of clustering. You can add Python ML library classes/API in the program.	L3	CO1, 2,3,4
9	Write a program to implement <b>k-Nearest Neighbour algorithm</b> to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.	L3	CO1, 2,3,4
10	Implement the non-parametric <b>Locally Weighted Regression algorithm</b> in order to fit data points. Select appropriate data set for your experiment and draw graphs.	L3	CO1, 2,3,4



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 8.REFERENCE:

1. Tom M.Mitchell, Machine Learning, India Edition2013,Mc GrawHill Education.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, h The Elements of Statistical Learning, 2nd edition, springer series in statistics.
3. Ethem Alpaydin, Introduction to machine learning, second edition, MIT press.

### C. EVALUATIONSCHEME

1. LaboratoryCIE:50Marks  
(Record writing,Viva-voce,Writing and Execution )
2. SEE: 50Marks

### D.ARTICULATIONMATRIX

Mapping of CO to PO

COs	POs											
	1	2	3	4	5	6	7	8	9	10	11	12
Understand and interpret the implementation procedures and python Libraries for the machine learning algorithms.	3	3	2		3							
Analyse the correctness of the data sets to apply appropriate Machine Learning algorithms.	3	3	3	3	3							
Design and implement Machine Learning algorithms to solve real world problems.	3	3	3	3	3	2						2
Evaluate and interpret the results of the machine learning algorithms.	3	3		3	3							



### E.EXPERIMENTS

#### 1.EXPERIMENT NO:1

#### 2.TITLE:FIND-SALGORITHM

#### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

#### 4. AIM:

- Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

#### 5. THEORY:

- The concept learning approach in machine learning, can be formulated as “Problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples”.
- Find-S algorithm for concept learning is one of the most basic algorithms of machine learning.

#### 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 in h 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

- It is Guaranteed to output the most specific hypothesis within H that is consistent with the positive training examples.
- Also Notice that negative examples are ignored.

#### Limitations of the Find-S algorithm:

- No way to determine if the only final hypothesis (found by Find-S) is consistent with data or there are more hypothesis that is consistent with data.
- Inconsistent sets of training data can mislead the finds algorithm as it ignores negative data samples.
- A good concept learning algorithm should be able to backtrack the choice of hypothesis found so that the resulting hypothesis can be improved over time. Unfortunately, Find-S provide no such method.

#### 6. LEARNING OUTCOMES :

- Students will be able to apply Find-S algorithm to the real world problem and find the most specific hypothesis from the training data.

#### 7. APPLICATION AREAS:

- Classification based problems.

#### 8. PROGRAMME



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

```
1 #read training data from csv file
2
3 import csv
4 with open('enjoysport.csv', 'r') as f:
5     reader = csv.reader(f)
6     data = list(reader)           #convert data into list of rows
7
8 #Training data from CSV file
9 print("Training data\n")
10 for row in data:
11     print(row)
12
13 attr_len=len(data[0])-1
14 h = ['0']*attr_len          # Initialize h to the most specific hypothesis in H
15
16 print("h=",h)
17 k=0
18
19 print("\nThe Hypothesis are\n")
20 for row in data:
21     if row[-1] == 'yes':      #For each positive training instance x
22         j = 0
23         for col in row:       #For each attribute constraint a, in h
24             if col != 'yes':    #replace a, in h by the next more general constraint
25                 if col != h[j] and h[j] == '0':
26                     h[j] = col
27                 elif col != h[j] and h[j] != '0':
28                     h[j] = '?'
29
30         j = j + 1
31
32     print("h",k,"=",h) #print all Hypothesis
33     k=k+1
34 print('\nMaximally Specific Hypothesis: \n',h,k-1,"=", h)      #print final hypothesis
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)

BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering

**Machine Learning [18CSL66] Lab Manual**

## 9. OUTPUT

Training data

```
['sunny', 'warm', 'normal', 'strong', 'warm', 'same', 'yes']
['sunny', 'warm', 'high', 'strong', 'warm', 'same', 'yes']
['rainy', 'cold', 'high', 'strong', 'warm', 'change', 'no']
['sunny', 'warm', 'high', 'strong', 'cool', 'change', 'yes']
h= ['0', '0', '0', '0', '0', '0']
```

The Hypothesis are

```
h 0 = ['sunny', 'warm', 'normal', 'strong', 'warm', 'same']
h 1 = ['sunny', 'warm', '?', 'strong', 'warm', 'same']
h 2 = ['sunny', 'warm', '?', 'strong', 'warm', 'same']
h 3 = ['sunny', 'warm', '?', 'strong', '?', '?']
```

Maximally Specific Hypothesis:

```
h 3 = ['sunny', 'warm', '?', 'strong', '?', '?']
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)

BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering

## Machine Learning [18CSL66] Lab Manual

### 1. EXPERIMENT NO: 2

### 2. TITLE: CANDIDATE-ELIMINATION ALGORITHM

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

- For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

### 5. THEORY:

- The key idea in the Candidate-Elimination algorithm is to output a description of the set of all hypotheses consistent with the training examples.
- It computes the description of this set without explicitly enumerating all of its members.
- This is accomplished by using the more-general-than partial ordering and maintaining a compact representation of the set of consistent hypotheses.
- The 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.
- A version space can be represented with its general and specific boundary sets.
- The Candidate-Elimination algorithm represents the version space by storing only its most general members G and its most specific members S.
- Given only these two sets S and G, it is possible to enumerate all members of a version space by generating hypotheses that lie between these two sets in general-to-specific partial ordering over hypotheses. Every member of the version space lies between these boundaries

### Algorithm

1. Initialize G to the set of maximally general hypotheses in H
2. Initialize S to the set of maximally specific hypotheses in H
3. 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, 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

### 6. LEARNING OUTCOMES :

- The students will be able to apply candidate elimination algorithm and output a description of the set of all hypotheses consistent with the training examples

### 7. APPLICATION AREAS:

- Classification based problems.

### 8. PROGRAMME



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering

### Machine Learning [18CSL66] Lab Manual

```
1 import csv
2
3 with open('enjoysport.csv', 'r') as f:
4     reader = csv.reader(f)
5     data = list(reader)
6
7 #Training data from CSV file
8 print("Training data\n")
9 for row in data:
10     print(row)
11 print("-----")
12 attr_len=len(data[0])-1
13
14 #Initialize Specific and General Hypothesis
15 S = ['0']*attr_len
16 G = ['?']*attr_len
17 temp=[]           # altered G
18
19 print("The Hypothesis are\n")
20 print("S=",S)
21 print("G=",G)
22 print("-----")
23 for row in data:
24     if row[-1] == 'yes':
25         j = 0
26         for col in row:
27             if col != 'yes':
28                 if col != S[j] and S[j] == '0':
29                     S[j] = col
30                 elif col != S[j] and S[j] != '0':
31                     S[j] = '?'
32             j = j + 1
33
34         for j in range(0,attr_len):
35             for k in temp:
36                 if k[j] != S[j] and k[j] != '?':
37                     temp.remove(k)
38
39     if row[-1]=='no':
40         j = 0
41         for col in row:
42             if col != 'no':
43                 if col!= S[j] and S[j] != '?':
44                     G[j]=S[j]
45                     temp.append(G)
46                     G=['?']*attr_len
47             j = j + 1
48 print("S=",S)
49 if len(temp)==0:
50     print("G=",G)
51 else:
52     print("G=",temp)
53 print('-----')
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka  
Aided By Govt. of Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### Output:

Training data

```
['sunny', 'warm', 'normal', 'strong', 'warm', 'same', 'yes']
['sunny', 'warm', 'high', 'strong', 'warm', 'same', 'yes']
['rainy', 'cold', 'high', 'strong', 'warm', 'change', 'no']
['sunny', 'warm', 'high', 'strong', 'cool', 'change', 'yes']
```

The Hypothesis are

```
S= ['0', '0', '0', '0', '0', '0']
G= ['?', '?', '?', '?', '?', '?']
```

```
S= ['sunny', 'warm', 'normal', 'strong', 'warm', 'same']
G= ['?', '?', '?', '?', '?', '?']
```

```
S= ['sunny', 'warm', '?', 'strong', 'warm', 'same']
G= ['?', '?', '?', '?', '?', '?']
```

```
S= ['sunny', 'warm', '?', 'strong', 'warm', 'same']
G= [['sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?'], ['?', '?', '?', '?', '?', 'same']]
```

```
S= ['sunny', 'warm', '?', 'strong', '?', '?']
G= [['sunny', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?']]
```



## 1. EXPERIMENT NO: 3

## 2. TITLE: ID3 ALGORITHM

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

- Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify anew sample.

### 5. THEORY:

- ID3 algorithm is a basic algorithm that learns decision trees by constructing them topdown, beginning with the question "which attribute should be tested at the root of the tree?".
- To answer this question, each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root node of the tree.
- A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node (i.e., down the branch corresponding to the example's value for this attribute).
- The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree.
- A simplified version of the algorithm, specialized to learning boolean-valued functions (i.e., concept learning), is described below.

#### 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
  - $A \leftarrow$  the attribute from *Attributes* that best\* classifies *Examples*
  - The decision attribute for *Root*  $\leftarrow A$
  - For each possible value,  $v_i$ , of *A*,
    - Add a new tree branch below *Root*, corresponding to the test  $A = v_i$
    - Let  $Examples_{v_i}$  be the subset of *Examples* that have value  $v_i$  for *A*
    - If  $Examples_{v_i}$  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  $ID3(Examples_{v_i}, Target\_attribute, Attributes - \{A\})$
- End
- Return *Root*



### 6. LEARNING OUTCOMES :

- The student will be able to demonstrate the working of the decision tree based ID3 algorithm, use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

### 7. APPLICATION AREAS:

- Classification related problem areas

### 8. PROGRAMME:

```
1 import pandas as pd
2 import math
3
4 df = pd.read_csv('id3.csv')
5 print("\n Input Data Set is:\n", df)
6
7 t = df.keys()[-1]
8 print('Target Attribute is: ', t)
9 attribute_names = list(df.keys())
10 attribute_names.remove(t)
11 print('Predicting Attributes: ', attribute_names)
12
13
14 def entropy(probs):
15     return sum( [-prob*math.log(prob, 2) for prob in probs])
16
17
18 def entropy_of_list(ls,value):
19     from collections import Counter
20     cnt = Counter(x for x in ls)
21     print('Target attribute class count(Yes/No)=',dict(cnt))
22     total_instances = len(ls)
23     print("Total no of instances/records associated with {0} is: {1}".format(value,total_instances ))
24     probs = [x / total_instances for x in cnt.values()]
25     return entropy(probs)
26
27 def information_gain(df, split_attribute, target_attribute,battr):
28     print("\n\n-----Information Gain Calculation of ",split_attribute, " -----")
29     df_split = df.groupby(split_attribute)
30     glist=[]
31     for gname,group in df_split:
32         print('Grouped Attribute Values \n',group)
33         glist.append(gname)
34     nobs = len(df.index)
35     df_agg1=df_split.agg({target_attribute:lambda x:entropy_of_list(x, glist.pop())})
36     df_agg2=df_split.agg({target_attribute :lambda x:len(x)/nobs})
37     df_agg1.columns=['Entropy']
38     df_agg2.columns=['Proportion']
39     new_entropy = sum( df_agg1['Entropy'] * df_agg2['Proportion'])
40     if battr !='S':
41         old_entropy = entropy_of_list(df[target_attribute],'S-'+df.iloc[0][df.columns.get_loc(battr)])
42     else:
43         old_entropy = entropy_of_list(df[target_attribute],battr)
44     return old_entropy - new_entropy
45
46
```



```
47 def id3(df, target_attribute, attribute_names, default_class=None, default_attr='S'):
48     from collections import Counter
49     cnt = Counter(x for x in df[target_attribute])      # class of YES /NO
50     if len(cnt) == 1:
51         return next(iter(cnt))
52     elif df.empty or (not attribute_names):
53         return default_class
54     else:
55         default_class = max(cnt.keys())
56         gainz=[]
57         for attr in attribute_names:
58             ig= information_gain(df, attr, target_attribute, default_attr)
59             gainz.append(ig)
60             print('Information gain of ',attr,' is : ',ig)
61         index_of_max = gainz.index(max(gainz))
62         best_attr = attribute_names[index_of_max]
63         print("\nAttribute with the maximum gain is: ", best_attr)
64         tree = {best_attr:{}}
65         remaining_attribute_names =[i for i in attribute_names if i != best_attr]
66         for attr_val, data_subset in df.groupby(best_attr):
67             subtree = id3(data_subset,target_attribute, remaining_attribute_names, default_class, best_attr)
68             tree[best_attr][attr_val] = subtree
69         return tree
70
71
72 from pprint import pprint
73
74 tree = id3(df,t,attribute_names)
75 print("\nThe Resultant Decision Tree is:")
76 pprint(tree)
```

## Output:

Input Data Set is:

	Outlook	Temperature	Humidity	Wind	Answer
0	sunny	hot	high	weak	no
1	sunny	hot	high	strong	no
2	overcast	hot	high	weak	yes
3	rain	mild	high	weak	yes
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no
6	overcast	cool	normal	strong	yes
7	sunny	mild	high	weak	no
8	sunny	cool	normal	weak	yes
9	rain	mild	normal	weak	yes
10	sunny	mild	normal	strong	yes
11	overcast	mild	high	strong	yes
12	overcast	hot	normal	weak	yes
13	rain	mild	high	strong	no

Target Attribute is: Answer

Predicting Attributes: ['Outlook', 'Temperature', 'Humidity', 'Wind']



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

-----Information Gain Calculation of Outlook -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
2	overcast	hot	high	weak	yes
6	overcast	cool	normal	strong	yes
11	overcast	mild	high	strong	yes
12	overcast	hot	normal	weak	yes

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
3	rain	mild	high	weak	yes
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no
9	rain	mild	normal	weak	yes
13	rain	mild	high	strong	no

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
0	sunny	hot	high	weak	no
1	sunny	hot	high	strong	no
7	sunny	mild	high	weak	no
8	sunny	cool	normal	weak	yes
10	sunny	mild	normal	strong	yes

Target attribute class count(Yes/No)= {'yes': 4}

Total no of instances/records associated with sunny is: 4

Target attribute class count(Yes/No)= {'yes': 3, 'no': 2}

Total no of instances/records associated with rain is: 5

Target attribute class count(Yes/No)= {'no': 3, 'yes': 2}

Total no of instances/records associated with overcast is: 5

Target attribute class count(Yes/No)= {'no': 5, 'yes': 9}

Total no of instances/records associated with S is: 14

Information gain of Outlook is : 0.2467498197744391

-----Information Gain Calculation of Temperature -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no
6	overcast	cool	normal	strong	yes
8	sunny	cool	normal	weak	yes

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
0	sunny	hot	high	weak	no
1	sunny	hot	high	strong	no
2	overcast	hot	high	weak	yes
12	overcast	hot	normal	weak	yes



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
3	rain	mild	high	weak	yes
7	sunny	mild	high	weak	no
9	rain	mild	normal	weak	yes
10	sunny	mild	normal	strong	yes
11	overcast	mild	high	strong	yes
13	rain	mild	high	strong	no

Target attribute class count(Yes/No)= {'yes': 3, 'no': 1}

Total no of instances/records associated with mild is: 4

Target attribute class count(Yes/No)= {'no': 2, 'yes': 2}

Total no of instances/records associated with hot is: 4

Target attribute class count(Yes/No)= {'yes': 4, 'no': 2}

Total no of instances/records associated with cool is: 6

Target attribute class count(Yes/No)= {'no': 5, 'yes': 9}

Total no of instances/records associated with S is: 14

Information gain of Temperature is : 0.029222565658954647

### -----Information Gain Calculation of Humidity -----

### Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
0	sunny	hot	high	weak	no
1	sunny	hot	high	strong	no
2	overcast	hot	high	weak	yes
3	rain	mild	high	weak	yes
7	sunny	mild	high	weak	no
11	overcast	mild	high	strong	yes
13	rain	mild	high	strong	no

### Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no
6	overcast	cool	normal	strong	yes
8	sunny	cool	normal	weak	yes
9	rain	mild	normal	weak	yes
10	sunny	mild	normal	strong	yes
12	overcast	hot	normal	weak	yes

Target attribute class count(Yes/No)= {'no': 4, 'yes': 3}

Total no of instances/records associated with normal is: 7

Target attribute class count(Yes/No)= {'yes': 6, 'no': 1}

Total no of instances/records associated with high is: 7

Target attribute class count(Yes/No)= {'no': 5, 'yes': 9}

Total no of instances/records associated with S is: 14

Information gain of Humidity is : 0.15183550136234136



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

-----Information Gain Calculation of Wind -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
1	sunny	hot	high	strong	no
5	rain	cool	normal	strong	no
6	overcast	cool	normal	strong	yes
10	sunny	mild	normal	strong	yes
11	overcast	mild	high	strong	yes
13	rain	mild	high	strong	no

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
0	sunny	hot	high	weak	no
2	overcast	hot	high	weak	yes
3	rain	mild	high	weak	yes
4	rain	cool	normal	weak	yes
7	sunny	mild	high	weak	no
8	sunny	cool	normal	weak	yes
9	rain	mild	normal	weak	yes
12	overcast	hot	normal	weak	yes

Target attribute class count(Yes/No)= {'no': 3, 'yes': 3}

Total no of instances/records associated with weak is: 6

Target attribute class count(Yes/No)= {'no': 2, 'yes': 6}

Total no of instances/records associated with strong is: 8

Target attribute class count(Yes/No)= {'no': 5, 'yes': 9}

Total no of instances/records associated with S is: 14

Information gain of Wind is : 0.04812703040826927

Attribute with the maximum gain is: Outlook

-----Information Gain Calculation of Temperature -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
3	rain	mild	high	weak	yes
9	rain	mild	normal	weak	yes
13	rain	mild	high	strong	no

Target attribute class count(Yes/No)= {'yes': 1, 'no': 1}

Total no of instances/records associated with mild is: 2

Target attribute class count(Yes/No)= {'yes': 2, 'no': 1}

Total no of instances/records associated with cool is: 3

Target attribute class count(Yes/No)= {'yes': 3, 'no': 2}

Total no of instances/records associated with S-rain is: 5

Information gain of Temperature is : 0.01997309402197489



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

-----Information Gain Calculation of Humidity -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
3	rain	mild	high	weak	yes
13	rain	mild	high	strong	no

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no
9	rain	mild	normal	weak	yes

Target attribute class count(Yes/No)= {'yes': 1, 'no': 1}

Total no of instances/records associated with normal is: 2

Target attribute class count(Yes/No)= {'yes': 2, 'no': 1}

Total no of instances/records associated with high is: 3

Target attribute class count(Yes/No)= {'yes': 3, 'no': 2}

Total no of instances/records associated with S-rain is: 5

Information gain of Humidity is : 0.01997309402197489

-----Information Gain Calculation of Wind -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
5	rain	cool	normal	strong	no
13	rain	mild	high	strong	no

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
3	rain	mild	high	weak	yes
4	rain	cool	normal	weak	yes
9	rain	mild	normal	weak	yes

Target attribute class count(Yes/No)= {'no': 2}

Total no of instances/records associated with weak is: 2

Target attribute class count(Yes/No)= {'yes': 3}

Total no of instances/records associated with strong is: 3

Target attribute class count(Yes/No)= {'yes': 3, 'no': 2}

Total no of instances/records associated with S-rain is: 5

Information gain of Wind is : 0.9709505944546686

Attribute with the maximum gain is: Wind

-----Information Gain Calculation of Temperature -----

Grouped Attribute Values

	Outlook	Temperature	Humidity	Wind	Answer
8	sunny	cool	normal	weak	yes

Grouped Attribute Values



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

Outlook Temperature Humidity Wind Answer  
0 sunny hot high weak no  
1 sunny hot high strong no  
Grouped Attribute Values  
Outlook Temperature Humidity Wind Answer  
7 sunny mild high weak no  
10 sunny mild normal strong yes  
Target attribute class count(Yes/No)= {'yes': 1}  
Total no of instances/records associated with mild is: 1  
Target attribute class count(Yes/No)= {'no': 2}  
Total no of instances/records associated with hot is: 2  
Target attribute class count(Yes/No)= {'no': 1, 'yes': 1}  
Total no of instances/records associated with cool is: 2  
Target attribute class count(Yes/No)= {'no': 3, 'yes': 2}  
Total no of instances/records associated with S-sunny is: 5  
Information gain of Temperature is : 0.5709505944546686

-----Information Gain Calculation of Humidity -----  
Grouped Attribute Values  
Outlook Temperature Humidity Wind Answer  
0 sunny hot high weak no  
1 sunny hot high strong no  
7 sunny mild high weak no  
Grouped Attribute Values  
Outlook Temperature Humidity Wind Answer  
8 sunny cool normal weak yes  
10 sunny mild normal strong yes  
Target attribute class count(Yes/No)= {'no': 3}  
Total no of instances/records associated with normal is: 3  
Target attribute class count(Yes/No)= {'yes': 2}  
Total no of instances/records associated with high is: 2  
Target attribute class count(Yes/No)= {'no': 3, 'yes': 2}  
Total no of instances/records associated with S-sunny is: 5  
Information gain of Humidity is : 0.9709505944546686

-----Information Gain Calculation of Wind -----  
Grouped Attribute Values  
Outlook Temperature Humidity Wind Answer  
1 sunny hot high strong no  
10 sunny mild normal strong yes  
Grouped Attribute Values  
Outlook Temperature Humidity Wind Answer  
0 sunny hot high weak no  
7 sunny mild high weak no  
8 sunny cool normal weak yes



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

Target attribute class count(Yes/No)= {'no': 1, 'yes': 1}

Total no of instances/records associated with weak is: 2

Target attribute class count(Yes/No)= {'no': 2, 'yes': 1}

Total no of instances/records associated with strong is: 3

Target attribute class count(Yes/No)= {'no': 3, 'yes': 2}

Total no of instances/records associated with S-sunny is: 5

Information gain of Wind is : 0.01997309402197489

Attribute with the maximum gain is: Humidity

The Resultant Decision Tree is:

```
{'Outlook': {'overcast': 'yes',
              'rain': {'Wind': {'strong': 'no', 'weak': 'yes'}},
              'sunny': {'Humidity': {'high': 'no', 'normal': 'yes'}}}}
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 1. EXPERIMENT NO: 4

### 2. TITLE: BACKPROPAGATION ALGORITHM

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

- Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

### 5. THEORY:

- Artificial neural networks (ANNs) provide a general, practical method for learning real-valued, discrete-valued, and vector-valued functions from examples.
- Algorithms such as BACKPROPAGATION gradient descent to tune network parameters to best fit a training set of input-output pairs.
- ANN learning is robust to errors in the training data and has been successfully applied to problems such as interpreting visual scenes, speech recognition, and learning robot control strategies.

### Backpropagation algorithm:

1. Input the instance  $x$ , 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 unit  $k$ , calculate its error term  $\delta_k$

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

3. For each network unit  $h$ , calculate its error term  $\delta_h$

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{h,k} \delta_k$$

4. Update each network weight  $w_{ji}$

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

Where

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

### 6. LEARNING OUTCOMES :

- The student will be able to build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

### 7. APPLICATION AREAS:

- Speech recognition, Character recognition, Human Face recognition

### 8. PROGRAMME:



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering

### Machine Learning [18CSL66] Lab Manual

```
1 import numpy as np
2
3 X = np.array(([2, 9], [1, 5], [3, 6]), dtype=float)
4 y = np.array(([92], [86], [89]), dtype=float)
5 X = X/np.amax(X, axis=0) #maximum of X array longitudinally
6 y = y/100
7
8 #Sigmoid Function
9 def sigmoid (x):
10     return 1/(1 + np.exp(-x))
11
12 #Derivative of Sigmoid Function
13 def derivatives_sigmoid(x):
14     return x * (1 - x)
15
16 #Variable initialization
17 epoch=5 #Setting training iterations
18 lr=0.1 #Setting learning rate
19
20 inputlayer_neurons = 2 #number of features in data set
21 hiddenlayer_neurons = 3 #number of hidden layers neurons
22 output_neurons = 1 #number of neurons at output layer
23
24 #weight and bias initialization
25 wh=np.random.uniform(size=(inputlayer_neurons,hiddenlayer_neurons))
26 bh=np.random.uniform(size=(1,hiddenlayer_neurons))
27 wout=np.random.uniform(size=(hiddenlayer_neurons,output_neurons))
28 bout=np.random.uniform(size=(1,output_neurons))
29 #draws a random range of numbers uniformly of dim x*y
30 for i in range(epoch):
31     #Forward Propogation
32     hinp1=np.dot(X,wh)
33     hinp=hinp1 + bh
34     hlayer_act = sigmoid(hinp)
35     outinp1=np.dot(hlayer_act,wout)
36     outinp= outinp1+bout
37     output = sigmoid(outinp)
38
39     #Backpropagation
40     EO = y-output
41     outgrad = derivatives_sigmoid(output)
42     d_output = EO * outgrad
43     EH = d_output.dot(wout.T)
44     hiddengrad = derivatives_sigmoid(hlayer_act)#how much hidden layer wts contributed to error
45     d_hiddenlayer = EH * hiddengrad
46
47     wout += hlayer_act.T.dot(d_output) *lr    # dotproduct of nextlayererror and currentlayerop
48     bout += np.sum(d_output, axis=0,keepdims=True) *lr
49     wh += X.T.dot(d_hiddenlayer) *lr
50     bh += np.sum(d_hiddenlayer, axis=0,keepdims=True) *lr
51 print("Input: \n",X)
52 print("Actual Output: \n",y)
53 print("Predicted Output: \n",output)
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering

**Machine Learning [18CSL66] Lab Manual**

## Output:

**Input:**

```
[[0.66666667 1.      ]
[0.33333333 0.55555556]
[1.      0.66666667]]
```

**Actual Output:**

```
[[0.92]
[0.86]
[0.89]]
```

**Predicted Output:**

```
[[0.90659823]
[0.89630074]
[0.90144612]]
```



## 1. EXPERIMENT NO: 5

## 2. TITLE: NAÏVE BAYESIAN CLASSIFIER

### **3. LEARNING OBJECTIVES:**

- Make use of Data sets in implementing the machine learning algorithms.
  - Implement ML concepts and algorithms in Python

#### **4. AIM:**

- Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

## 5. THEORY:

**Naive Bayes algorithm:** Naive Bayes algorithm is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naïve Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naïve Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability  $P(c|x)$  from  $P(c)$ ,  $P(x)$  and  $P(x|c)$ . Look at the equation below:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

where

$P(c|x)$  is the posterior probability of class (c, target) given predictor (x, attributes).  $P(c)$  is the prior probability of class.

$P(x|c)$  is the likelihood which is the probability of predictor given class.  $P(x)$  is the prior probability of predictor.

The naive Bayes classifier applies to learning tasks where each instance  $x$  is described by a conjunction of attribute values and where the target function  $f(x)$  can take on any value from some finite set  $V$ . A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values  $(a_1, a_2, \dots, a_n)$ . The learner is asked to predict the target value, or classification, for this new instance. The Bayesian approach to classifying the new instance is to assign the most probable target value,



## Algorithm

### Algorithm:

**NaiveBaiseClassifier(training\_examples, New\_Instance)**

Each instance  $x$  is described by a conjunction of attribute values( $a_i$ ) and the target V can take j finite set of values.

- For each value j in target estimate the  $P(V_j)$
- For each attribute in the training example estimate Estimate the  $P(a_i|V_j)$
- Classify each instance as per the rule in equation

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

Where  $V_{NB}$  denotes the target value output by the naive Bayes classifier

- Output  $V_{NB}$

## 6. LEARNING OUTCOMES :

- The student will be able to apply naive Bayesian classifier for the relevant problem and analyse the results.

## 7. APPLICATION AREAS:

- Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

## 8. PROGRAMME



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering

Machine Learning [18CSL66] Lab Manual

```
1 import numpy as np
2 import math
3 import csv
4
5 def read_data(filename):
6     with open(filename,"r") as csvfile:
7         datareader=csv.reader(csvfile)
8         traindata=list(datareader)
9         metadata=traindata[0] #attributes name
10        traindata=traindata[1:] #training examples
11        return (metadata, traindata)
12
13 def splitdataset(dataset, splitratio):
14     trainsize=int(len(dataset)*splitratio)
15     trainset=[]
16     testset=list(dataset)
17     i=0
18     while len(trainset)<trainsize:
19         trainset.append(testset.pop(i))
20     return [trainset,testset]
21
22 def classify(data,test):
23     totalsize=data.shape[0]
24     print("\n")
25     print("training data size=",totalsize)
26     print("test data size=",test.shape[0])
27
28     countyes=0
29     countno=0
30     probyes=0
31     probno=0
32     print("\n")
33     print("target \t count \t probality")
34     for x in range(data.shape[0]):
35         if data[x,data.shape[1]-1] =='yes':
36             countyes+=1
37         if data[x,data.shape[1]-1] =='no':
38             countno+=1
39     probYes=countyes/totalsize
40     probNo=countno/totalsize
41
42     print("Yes \t", countyes, "\t", probYes)
43     print("No \t", countno, "\t", probNo)
44
45     prob0=np.zeros((test.shape[1]-1))
46     prob1=np.zeros((test.shape[1]-1))
47
48     accuracy=0
49     print("\n")
50     print("instance \t prediction \t target")
51
52     for t in range(test.shape[0]):
53         for k in range (test.shape[1]-1):
54             count1=count0=0
55             for j in range(data.shape[0]):
56                 #how many times appeared with no
57                 if test[t,k]==data[j,k] and data[j,data.shape[1]-1]=='no':
58                     count0+=1
59                     #how many times appeared with yes
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering

Machine Learning [18CSL66] Lab Manual

```
60             if test[t,k]==data[j,k] and data[j,data.shape[1]-1]=='yes':
61                 count1+=1
62                 prob0[k]=count0/countno
63                 prob1[k]=count1/countrys
64                 probno=probNo
65                 probyes=probYes
66                 for i in range(test.shape[1]-1):
67                     probno=probno*prob0[i]
68                     probyes=probyes*prob1[i]
69                 if probno>probyes:
70                     predict='no'
71                 else:
72                     predict='yes'
73
74                 print(" ",t+1," \t\t",predict," \t\t",test[t,test.shape[1]-1])
75                 if predict==test[t,test.shape[1]-1]:
76                     accuracy+=1
77                 finalaccuracy=(accuracy/test.shape[0])*100
78                 print("\n Accuracy=",finalaccuracy,"%")
79
80
81
82 metadata,traindata=read_data("enjoysport.csv")
83 print("\n The attribute names of training data are:",metadata)
84 splitratio=0.7
85 trainset, testset=splitdataset(traindata, splitratio)
86 training=np.array(trainset)
87
88 print("\n The Training data set are:")
89 for x in training:
90     print(x)
91
92 testing=np.array(testset)
93 print("\n The Test data set are:")
94 for x in testing:
95     print(x)
96
97 classify(training, testing)
```

## Output:



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

Department of Computer Science & Engineering

## Machine Learning [18CSL66] Lab Manual

The attribute names of training data are: ['Outlook', 'Temperature', 'Humidity', 'Wind', 'Answer']

The Training data set are:

```
['sunny' 'hot' 'high' 'weak' 'no']
['sunny' 'hot' 'high' 'strong' 'no']
['overcast' 'hot' 'high' 'weak' 'yes']
['rain' 'mild' 'high' 'weak' 'yes']
['rain' 'cool' 'normal' 'weak' 'yes']
['rain' 'cool' 'normal' 'strong' 'no']
['overcast' 'cool' 'normal' 'strong' 'yes']
['sunny' 'mild' 'high' 'weak' 'no']
['sunny' 'cool' 'normal' 'weak' 'yes']
```

The Test data set are:

```
['rain' 'mild' 'normal' 'weak' 'yes']
['sunny' 'mild' 'normal' 'strong' 'yes']
['overcast' 'mild' 'high' 'strong' 'yes']
['overcast' 'hot' 'normal' 'weak' 'yes']
['rain' 'mild' 'high' 'strong' 'no']
```

```
training data size= 9
test data size= 5

instance          prediction      target
  1                yes            yes
  2                no             yes
  3                yes            yes
  4                yes            yes
  5                no             no

Accuracy= 80.0 %

target    count    probolaity
Yes      5        0.5555555555555556
No       4        0.4444444444444444
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 1. EXPERIMENT NO: 6

### 2. TITLE: DOCUMENT CLASSIFICATION USING NAÏVE BAYESIAN CLASSIFIER

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

- Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

### 5. THEORY:

#### Naïve Bayes classifier algorithm for text classification

##### LEARN\_NAIVE\_BAYES\_TEXT(*Examples*, *V*)

*Examples* is a set of text documents along with their target values. *V* is the set of all possible target values. This function learns the probability terms  $P(w_k|v_j)$ , describing the probability that a randomly drawn word from a document in class  $v_j$  will be the English word  $w_k$ . It also learns the class prior probabilities  $P(v_j)$ .

1. collect all words, punctuation, and other tokens that occur in *Examples*

- *Vocabulary*  $\leftarrow$  the set of all distinct words and other tokens occurring in any text document from *Examples*

2. calculate the required  $P(v_j)$  and  $P(w_k|v_j)$  probability terms

- For each target value  $v_j$  in *V* do

- *docs<sub>j</sub>*  $\leftarrow$  the subset of documents from *Examples* for which the target value is  $v_j$

$$\bullet P(v_j) \leftarrow \frac{|\text{docs}_j|}{|\text{Examples}|}$$

- *Text<sub>j</sub>*  $\leftarrow$  a single document created by concatenating all members of *docs<sub>j</sub>*

- $n \leftarrow$  total number of distinct word positions in *Text<sub>j</sub>*

- for each word  $w_k$  in *Vocabulary*

- $n_k \leftarrow$  number of times word  $w_k$  occurs in *Text<sub>j</sub>*

$$\bullet P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|\text{Vocabulary}|}$$

##### CLASSIFY\_NAIVE\_BAYES\_TEXT(*Doc*)

Return the estimated target value for the document *Doc*.  $a_i$  denotes the word found in the *i*th position within *Doc*.

- *positions*  $\leftarrow$  all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return  $v_{NB}$ , where

$$v_{NB} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_{i \in \text{positions}} P(a_i | v_j)$$

### Confusion Matrix

A confusion matrix, also called a contingency table or error matrix, is used to visualize the performance of a classifier.

		Confusion Matrix		Predicted classes	
				negative	positive
		Actual	classes	negative	TN
Actual	classes	negative			FP
		positive		FN	TP



## ► Accuracy

Accuracy is a statistical measure which is defined as the quotient of correct predictions (both True positives (TP) and True negatives (TN)) made by a classifier divided by the sum of all predictions made by the classifier, including False positives (FP) and False negatives (FN).

The formula:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

► **Precision:** Precision is the ratio of the correctly identified positive cases to all the predicted positive cases, i.e. the correctly and the incorrectly predicted cases as positive. Precision is the fraction of retrieved documents that are relevant to the query.

► The formula:

$$\text{precision} = \frac{TP}{TP + FP}$$

► **Recall**, also known as sensitivity, is the ratio of the correctly identified positive cases to all the actual positive cases, which is the sum of the "False Negatives" and "True Positives".

► The formula:

$$\text{recall} = \frac{TP}{TP + FN}$$

### 6. LEARNING OUTCOMES:

- The student will be able to apply naive bayesian classifier for document classification and analyse the results.

### 7. APPLICATION AREAS:

- Applicable in document classification

### 8. PROGRAMME



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

```
1 import pandas as pd
2 msg=pd.read_csv('D:\\data6.csv',names=['message','label']) #Tabular form data
3 print('Total instances in the dataset:',msg.shape[0])
4
5 msg['labelnum']=msg.label.map({'pos':1,'neg':0})
6 #print(msg)
7 X=msg.message
8 Y=msg.labelnum
9
10 #print(X)
11 #print(Y)
12
13
14 # Splitting the dataset into train and test data
15 from sklearn.model_selection import train_test_split
16 xtrain,xtest,ytrain,ytest=train_test_split(X,Y)
17
18 print('\nDataset is split into Training and Testing samples')
19 print('Total training instances :', ytrain.shape[0])
20 print('Total testing instances :', ytest.shape[0])
21
22 # Output of count vectoriser is a sparse matrix
23 # CountVectorizer - stands for 'feature extraction'
24 from sklearn.feature_extraction.text import CountVectorizer
25 count_vect = CountVectorizer()
26 xtrain_dtm = count_vect.fit_transform(xtrain) #Sparse matrix
27 xtest_dtm = count_vect.transform(xtest)
28 print('\nTotal features extracted using CountVectorizer:',xtrain_dtm.shape[1])
29
30 print('\nThe words or Tokens in the text documents\n')
31 print(count_vect.get_feature_names())
32
33
34 # Training Naive Bayes (NB) classifier on training data.
35 from sklearn.naive_bayes import MultinomialNB
36 clf = MultinomialNB().fit(xtrain_dtm,ytrain)
37 predicted = clf.predict(xtest_dtm)
38
39 #printing accuracy metrics
40 from sklearn import metrics
41 print('\nAccuracy metrics')
42 print('=====')
43 print('Accuracy of the classifier is',metrics.accuracy_score(ytest,predicted))
44
45 print('Recall :',metrics.recall_score(ytest,predicted), '\nPrecision :',metrics.precision_score(ytest,predicted))
46 print('Confusion matrix')
47 print('=====')
48 print(metrics.confusion_matrix(ytest,predicted))
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka  
Aided By Govt. of Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### Output:

Total instances in the dataset: (18, 2)

Dataset is split into Training and Testing samples

Total training instances : 13

Total testing instances : 5

Total features extracted using CountVectorizer: 42

The words or Tokens in the text documents

```
['am', 'amazing', 'an', 'awesome', 'bad', 'best', 'boss', 'can', 'dance', 'deal', 'do', 'fun', 'good', 'great', 'have', 'holida
y', 'horrible', 'is', 'juice', 'like', 'locality', 'love', 'my', 'not', 'of', 'place', 'restaurant', 'sandwich', 'stay', 'stuf
f', 'taste', 'that', 'the', 'this', 'tired', 'to', 'tomorrow', 'we', 'what', 'will', 'with', 'work']
```

Accuracy metrics

=====

Accuracy of the classifier is 0.6

Recall : 1.0

Precision : 0.5

Confusion matrix

=====

```
[[1 2]
 [0 2]]
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 1. EXPERIMENT NO: 7

### 2. TITLE: BAYESIAN NETWORK

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

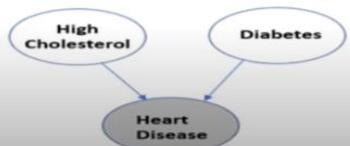
### 4. AIM:

- Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.

### 5. THEORY:

#### What Is A Bayesian Network?

- A Bayesian Network falls under the category of Probabilistic Graphical Modelling (PGM) technique that is used to compute uncertainties by using the concept of probability.
- Popularly known as Belief Networks, Bayesian Networks are used to model uncertainties by using Directed Acyclic Graphs (DAG).



## Steps used to build the Bayesian Network

- **Step 1:** Identify the variables which is set of attributes specified in the dataset(ex Medical Dataset)
- **Step2:** Determine the domain of each variable that is set of values a variable may take
- **Step3:** Create a directed graph network of nodes where each node represents the attribute and edges represent parent child relationship. Edge represents that the child variable is conditionally dependent on the parent.
- **Step4 :** determine the prior and conditional probability for each attribute
- **Step5 :** perform the inference on the model and determine the marginal probabilities

### 6. LEARNING OUTCOMES:

- The student will be able to apply bayesian network for the medical data and demonstrate the diagnosis of heart patients using standard Heart Disease Data Set.

### 7. APPLICATION AREAS:

- Applicable in prediction and classification
- Gene Regulatory Networks
- Medicine
- Biomonitoring
- Document Classification
- Information Retrieval
- Semantic Search



### 8. PROGRAMME

```
1 import pandas as pd
2 import numpy as np
3
4 #read Cleveland Heart Disease data
5 data = pd.read_csv('heart.csv')
6 data = data.replace('?',np.nan)
7 #display the data
8 print('Sample instances from the dataset are given below')
9 print(data.head())
10 #display the Attributes names and datatypes
11 print('\n Attributes and datatypes')
12 print(data.dtypes)
13
14 from pgmpy.estimators import MaximumLikelihoodEstimator
15 from pgmpy.models import BayesianModel
16 from pgmpy.inference import VariableElimination
17
18 #Create Model- Bayesian Network
19 #Defining the model structure. We can define the network by just -
20 #passing a List of edges.
21 model =BayesianModel([('age','heartdisease'),('sex','heartdisease'),
22 ('exang','heartdisease'),('cp','heartdisease'),
23 ('heartdisease','restecg'),('heartdisease','chol')])
24 import networkx as nx
25 import matplotlib.pyplot as plt
26 nx.draw(model, with_labels = True);
27 plt.show()
28
29 #Learning CPDs using Maximum Likelihood Estimators for all the variables
30 print('\n Learning CPD using Maximum likelihood estimators')
31 model.fit(data,estimator=MaximumLikelihoodEstimator)
32
33 #print(model.get_cpds('cp'))
34
35 # Inferencing with Bayesian Network
36 print('\n Inferencing with Bayesian Network:')
37 infer = VariableElimination(model)
38
39 #computing the Probability of HeartDisease given restecg
40 print('\n 1. Probability of HeartDisease given evidence=restecg :1')
41 q1=infer.query(variables=['heartdisease'],evidence={'restecg':1})
42 print(q1)
43
44 #computing the Probability of HeartDisease given cp
45 print('\n 2. Probability of HeartDisease given evidence= cp:2 ')
46 q2=infer.query(variables=['heartdisease'],evidence={'cp':2})
47 print(q2)
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka  
Aided By Govt. of Karnataka

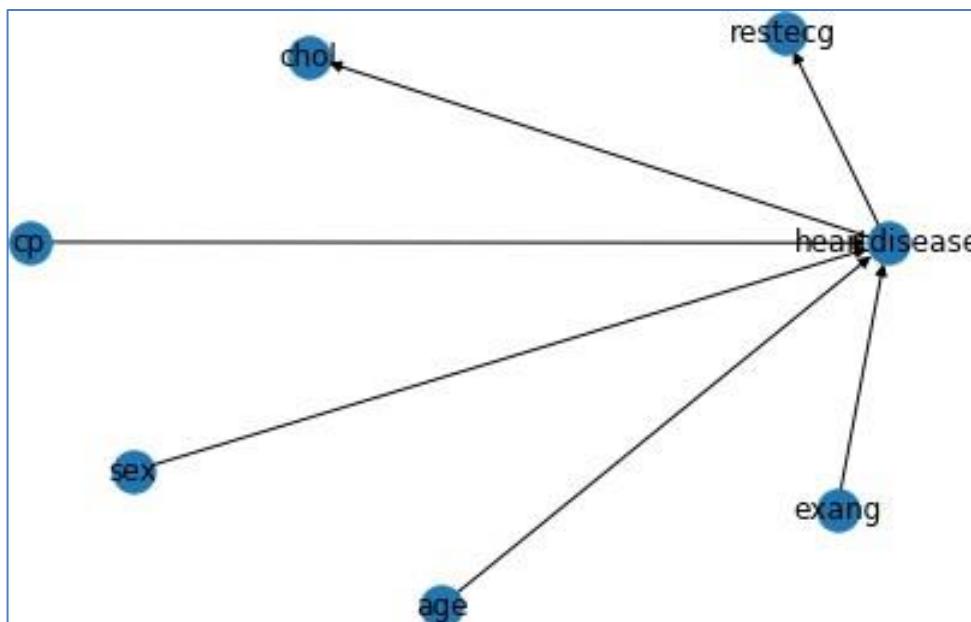
## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### Output:

```
Sample instances from the dataset are given below
    age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope \
0    63    1    1      145   233    1      2     150      0     2.3     3
1    67    1    4      160   286    0      2     108      1     1.5     2
2    67    1    4      120   229    0      2     129      1     2.6     2
3    37    1    3      130   250    0      0     187      0     3.5     3
4    41    0    2      130   204    0      2     172      0     1.4     1

ca  thal  heartdisease
0    0    6          0
1    3    3          2
2    2    7          1
3    0    3          0
4    0    3          0

Attributes and datatypes
age            int64
sex            int64
cp             int64
trestbps       int64
chol           int64
fbs            int64
restecg        int64
thalach        int64
exang          int64
oldpeak        float64
slope          int64
ca             object
thal            object
heartdisease  int64
dtype: object
```





# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

Learning CPD using Maximum likelihood estimators

Finding Elimination Order: : 100%

Eliminating: cp: 100%

5/5 [00:00<00:00, 1666.79it/s]

5/5 [00:00<00:00, 277.74it/s]

Inferencing with Bayesian Network:

1. Probability of HeartDisease given evidence=restecg :1

heartdisease	phi(heartdisease)
heartdisease(0)	0.1012
heartdisease(1)	0.0000
heartdisease(2)	0.2392
heartdisease(3)	0.2015
heartdisease(4)	0.4581

2. Probability of HeartDisease given evidence= cp:2

Finding Elimination Order: : 100%

Eliminating: chol: 100%

5/5 [00:00<00:00, 4983.73it/s]

5/5 [00:00<00:00, 454.64it/s]

heartdisease	phi(heartdisease)
heartdisease(0)	0.3610
heartdisease(1)	0.2159
heartdisease(2)	0.1373
heartdisease(3)	0.1537
heartdisease(4)	0.1321



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### 1. EXPERIMENT NO: 8

### 2. TITLE: CLUSTERING BASED ON EM ALGORITHM AND K-MEANS

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

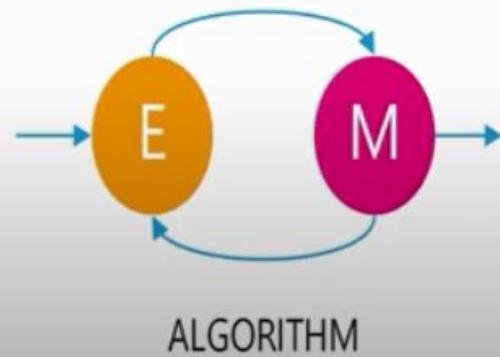
Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

### 5. THEORY:

## Expectation-Maximization Algorithm

### Algorithm:

1. Given a set of incomplete data, consider a set of starting parameters.
2. **Expectation step (E - step):** Using the observed available data of the dataset, estimate (guess) the values of the missing data.
3. **Maximization step (M - step):** Complete data generated after the expectation (E) step is used in order to update the parameters.
4. Repeat step 2 and step 3 until convergence.





## K-Means Algorithm

- **K-Means** algorithm is an iterative algorithm that tries to partition the dataset into  $K$  pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**.
- It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.
- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum.

### Algorithm 1 $k$ -means algorithm

- 1: Specify the number  $k$  of clusters to assign.
- 2: Randomly initialize  $k$  centroids.
- 3: **repeat**
- 4:   **expectation:** Assign each point to its closest centroid.
- 5:   **maximization:** Compute the new centroid (mean) of each cluster.
- 6: **until** The centroid positions do not change.

### 6. LEARNINGOUTCOMES:

- The students will be capable to apply EM algorithm and k-Means algorithm for clustering and analyse the results.

### 7. APPLICATIONAREAS:

- Text mining
- Pattern recognition
- Image analysis

### 8. PROGRAMME:



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

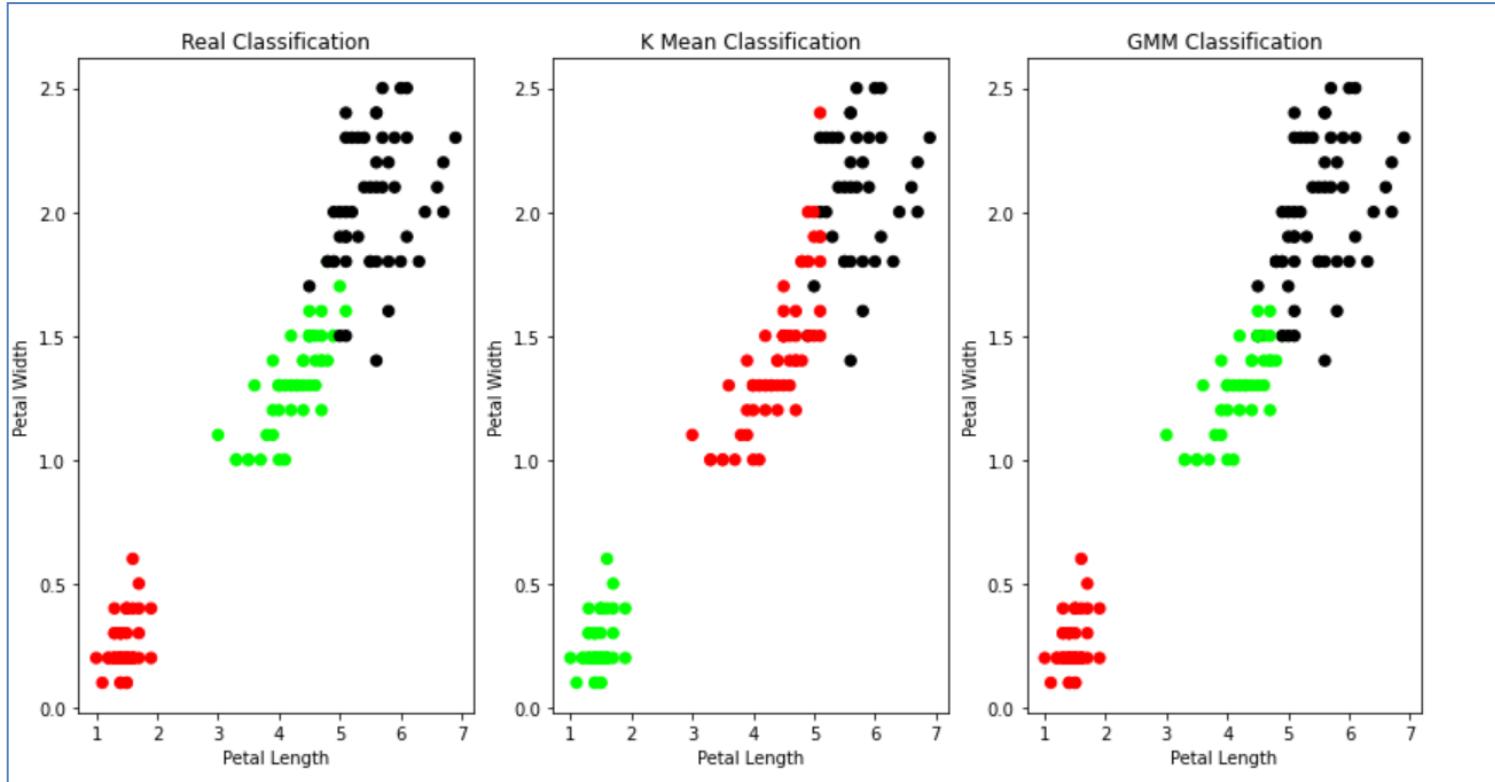
## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

```
1 import matplotlib.pyplot as plt
2 from sklearn import datasets
3 import sklearn.metrics as sm
4 import pandas as pd
5 import numpy as np
6
7 iris = datasets.load_iris()
8
9 X = pd.DataFrame(iris.data)
10 X.columns = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']
11
12 y = pd.DataFrame(iris.target)
13 y.columns = ['Targets']
14
15 plt.figure(figsize=(14,7))
16 colormap = np.array(['red', 'lime', 'black'])
17
18 # Plot the Original Classifications
19 plt.subplot(1, 3, 1)
20 plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y.Targets], s=40)
21 plt.title('Real Classification')
22 plt.xlabel('Petal Length')
23 plt.ylabel('Petal Width')
24
25
26 # Plot the Models Classifications
27 #KMeans
28 from sklearn.cluster import KMeans
29 model = KMeans(n_clusters=3)
30 model.fit(X)
31 plt.subplot(1, 3, 2)
32 plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[model.labels_], s=40)
33 plt.title('K Mean Classification')
34 plt.xlabel('Petal Length')
35 plt.ylabel('Petal Width')
36 print('The accuracy score of K-Mean: ',sm.accuracy_score(y, model.labels_))
37 print('The Confusion matrixof K-Mean: ',sm.confusion_matrix(y, model.labels_))
38
39 #EM(GMM)
40 from sklearn.mixture import GaussianMixture
41 gmm = GaussianMixture(n_components=3)
42 gmm.fit(X)
43 y_gmm = gmm.predict(X)
44 plt.subplot(1, 3, 3)
45 plt.scatter(X.Petal_Length, X.Petal_Width, c=colormap[y_gmm], s=40)
46 plt.title('GMM Classification')
47 plt.xlabel('Petal Length')
48 plt.ylabel('Petal Width')
49
50 print('The accuracy score of EM: ',sm.accuracy_score(y, y_gmm))
51 print('The Confusion matrix of EM: ',sm.confusion_matrix(y, y_gmm))
```



## Output:

```
The accuracy score of K-Mean: 0.24
The Confusion matrix of K-Mean: [[ 0 50 0]
[48 0 2]
[14 0 36]]
The accuracy score of EM: 0.9666666666666667
The Confusion matrix of EM: [[50 0 0]
[ 0 45 5]
[ 0 0 50]]
```





### 1. EXPERIMENT NO: 9

### 2. TITLE: K-NEAREST NEIGHBOUR

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

- Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

### 5. THEORY:

- K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.
- It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data.

## K-NN Algorithm

► **Input:** Let m be the number of training data samples. Let p be an unknown point.

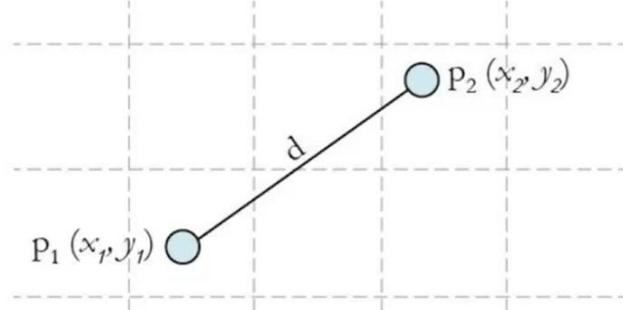
► Method:

- Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).
- for i=0 to m

► **Calculate Euclidean distance** d(arr[i], p).

- Make set S of K smallest distances obtained. Each of these distances correspond to an already classified data point.

► Return the majority label among S.



$$\text{Euclidean distance } (d) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

► 6

### 6. LEARNING OUTCOMES :

- The student will be able to implement k-Nearest Neighbour algorithm to classify the iris data set and Print both correct and wrong predictions.

### 7. APPLICATION AREAS:

- Recommender systems
- Classification problems



## 8. PROGRAMME:

```
1 from sklearn import datasets
2 iris=datasets.load_iris()
3 iris_data=iris.data
4 iris_labels=iris.target
5
6 from sklearn.model_selection import train_test_split
7 x_train,x_test,y_train,y_test=train_test_split(iris_data,iris_labels,test_size=0.30)
8
9 from sklearn.neighbors import KNeighborsClassifier
10 classifier=KNeighborsClassifier(n_neighbors=5)
11 classifier.fit(x_train,y_train)
12 y_pred=classifier.predict(x_test)
13
14 target_names = iris.target_names
15 for pred,actual in zip(y_pred,y_test):
16     print("Prediction is "+str(target_names[pred])+", Actual is "+str(target_names[actual]))
17
18 from sklearn.metrics import classification_report,confusion_matrix
19 print('Confusion matrix is as follows')
20 print(confusion_matrix(y_test,y_pred))
21 print('Accuracy Metrics')
22 print(classification_report(y_test,y_pred))
```

		Predicted			Actual Total
		Iris-setosa	Iris-versicolor	Iris-virginica	
Actual	Iris-setosa	14	0	0	14
	Iris-versicolor	0	13	2	15
		0	1	15	16
Predicted Total		14	14	17	45

## Output:



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

Confusion matrix is as follows

```
[[14  0  0]
 [ 0 13  2]
 [ 0  1 15]]
```

Accuracy Metrics

	precision	recall	f1-score	support
0	1.00	1.00	1.00	14
1	0.93	0.87	0.90	15
2	0.88	0.94	0.91	16
accuracy			0.93	45
macro avg	0.94	0.93	0.94	45
weighted avg	0.93	0.93	0.93	45

Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is versicolor, Actual is versicolor  
Prediction is versicolor, Actual is versicolor  
Prediction is setosa, Actual is setosa  
Prediction is versicolor, Actual is versicolor  
Prediction is versicolor, Actual is versicolor  
Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa

Prediction is virginica, Actual is versicolor

Prediction is versicolor, Actual is versicolor  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is versicolor, Actual is versicolor  
Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is versicolor, Actual is versicolor  
Prediction is virginica, Actual is virginica  
Prediction is versicolor, Actual is versicolor  
Prediction is virginica, Actual is virginica  
Prediction is versicolor, Actual is versicolor  
Prediction is setosa, Actual is setosa  
Prediction is versicolor, Actual is versicolor  
Prediction is versicolor, Actual is versicolor  
Prediction is versicolor, Actual is versicolor  
Prediction is versicolor, Actual is virginica

Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is virginica  
Prediction is virginica, Actual is virginica  
Prediction is setosa, Actual is setosa  
Prediction is virginica, Actual is versicolor



### 1. EXPERIMENT NO: 10

### 2. TITLE: LOCALLY WEIGHTED REGRESSION ALGORITHM

### 3. LEARNING OBJECTIVES:

- Make use of Data sets in implementing the machine learning algorithms.
- Implement ML concepts and algorithms in Python

### 4. AIM:

- Implement the non-parametric Locally Weighted Regression algorithm in order to fit datapoints. Select appropriate data set for your experiment and draw graphs.

### 5. THEORY:

## LOCALLY WEIGHTED REGRESSION

- ▶ Locally weighted linear regression is a supervised learning algorithm.
- ▶ It is a non-parametric algorithm.
- ▶ There exists no training phase. All the work is done during the testing phase/while making predictions.

### ALGORITHM:

1. Read the Given data Sample to X and the curve (linear or non linear) to Y
2. Set the value for Smoothening parameter or free parameter say  $\tau$
3. Set the bias /Point of interest set  $X_0$  which is a subset of X
4. Determine the weight matrix using:

$$w(x, x_o) = e^{-\frac{(x-x_o)^2}{2\tau^2}}$$

5. Determine the value of model term parameter  $\beta$  using :

$$\hat{\beta}(x_o) = (X^T W X)^{-1} X^T W y$$

6. Prediction =  $x_0 * \beta$

### 6. LEARNING OUTCOMES :

- To understand and implement linear regression and analyse the results with change in the parameters

### 7. APPLICATION AREAS:

- Demand analysis in business
- Forecasting
- Optimization of business processes

### 8. PROGRAMME:



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka  
Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi)  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka  
Aided By Govt. of Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

```
1 from numpy import *
2 from os import listdir
3 import matplotlib
4 import matplotlib.pyplot as plt
5 import pandas as pd
6 import numpy as np
7 import numpy.linalg as np
8 from scipy.stats.stats import pearsonr
9
10 def kernel(point,xmat, k):
11     m,n = np.shape(xmat)
12     weights = np.mat(np.eye((m)))
13     for j in range(m):
14         diff = point - X[j]
15         weights[j,j] = np.exp(diff*diff.T/(-2.0*k**2))
16     return weights
17
18 def localWeight(point,xmat,ymat,k):
19     wei = kernel(point,xmat,k)
20     W = (X.T*(wei*X)).I*(X.T*(wei*ymat.T))
21     return W
22
23 def localWeightRegression(xmat,ymat,k):
24     m,n = np.shape(xmat)
25     ypred = np.zeros(m)
26     for i in range(m):
27         ypred[i] = xmat[i]*localWeight(xmat[i],xmat,ymat,k)
28     return ypred
29
30 # Load data points
31 data = pd.read_csv('D://tips.csv')
32 bill = np.array(data.total_bill)
33 tip = np.array(data.tip)
34
35 #preparing and add 1 in bill
36 mbill = np.mat(bill)
37 mtip = np.mat(tip) # mat is used to convert to n dimesional to 2 dimensional array form
38 m= np.shape(mbill)[1]
39 # print(m) 244 data is stored in m
40 one = np.mat(np.ones(m))
41 X= np.hstack((one.T,mbill.T)) # create a stack of bill from ONE
42 #print(X)
43 #set k here
44 ypred = localWeightRegression(X,mtip,0.3)
45 SortIndex = X[:,1].argsort(0)
46 xsort = X[SortIndex][:,0]
47
48 fig = plt.figure()
49 ax = fig.add_subplot(1,1,1)
50 ax.scatter(bill,tip, color='green')
51 ax.plot(xsort[:,1],ypred[SortIndex], color = 'red', linewidth=5)
52 plt.xlabel('Total bill')
53 plt.ylabel('Tip')
54 plt.show();
```



# Dr. Ambedkar Institute of Technology

(An Autonomous Institution, Aided by Government of Karnataka)

Affiliated to Visvesvaraya Technological University, Belgaum & Approved by AICTE, New Delhi  
BDA Outer Ring Road, Near Jnana Bharathi Campus, Mallathahalli, Bengaluru-560056, Karnataka

## Department of Computer Science & Engineering Machine Learning [18CSL66] Lab Manual

### Output:

