VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on

Machine Learning (23CS6PCMAL)

Submitted by

Yashraj Sinha (1BM22CS335)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "Machine Learning (23CS6PCMAL)" carried out by Yashraj Sinha (1BM22CS335), who is bonafide student of B.M.S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Machine Learning (23CS6PCMAL) work prescribed for the said degree.

Ramya KM	Dr. Kavitha Sooda
Assistant Professor	Professor & HOD
Department of CSE, BMSCE	Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	3-3-2025	Write a python program to import and export data using Pandas library functions	1 - 3
2	10/03/25	Demonstrate various data pre-processing techniques for a given dataset	4 - 8
3	10-3-2025	Implement Linear and Multi-Linear Regression algorithm using appropriate dataset	9 - 15
4	17-3-2025	Build Logistic Regression Model for a given dataset	16 - 24
5	24-3-2025	Use an appropriate data set for building the decision tree (ID3) and apply this knowledge to classify a new sample.	25 - 34
6	7-4-2025	Build KNN Classification model for a given dataset.	35 - 39
7	21-4-2025	Build Support vector machine model for a given dataset	40 - 47
8	5-5-2025	Implement Random forest ensemble method on a given dataset.	48 - 51
9	5-5-2025	Implement Boosting ensemble method on a given dataset.	52 - 56
10	05/05/25	Build k-Means algorithm to cluster a set of data stored in a .CSV file.	57 - 59
11	05/05/25	Implement Dimensionality reduction using Principal Component Analysis (PCA) method.	60 - 65

Github Link:

Code:

https://github.com/sweet-er-est/ML-Lab

Program 1

Write a python program to import and export data using Pandas library functions

```
import pandas as pd
# Method-1: Initializing values directly into DataFrame
data_method1 = {'USN': ['1JS17CS001', '1JS17CS002', '1JS17CS003',
'1JS17CS004', '1JS17CS005'],
'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
'Marks': [90, 85, 92, 78, 88]}
df_method1 = pd.DataFrame(data_method1)
print("Method-1:")
print(df_method1)
print("-" * 20)
# Method-2: Importing datasets from sklearn.datasets
from sklearn.datasets import load_diabetes
diabetes_data = load_diabetes()
df_method2 = pd.DataFrame(data=diabetes_data.data,
columns=diabetes_data.feature_names)
df_method2['target'] = diabetes_data.target
print("Method-2:")
print(df_method2.head())
print("-" * 20)
# Method-3: Importing datasets from a specific .csv file
try:
```

```
df_method3 = pd.read_csv('sample_sales_data.csv')
print("Method-3:")
print(df_method3.head())
print("-" * 20)
except FileNotFoundError:
print("sample_sales_data.csv not found. Please upload the file.")
print("-" * 20)
import yfinance as yf
import matplotlib.pyplot as plt
tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]
start_date = "2024-01-01"
end_date = "2024-12-30"
data = yf.download(tickers, start=start_date, end=end_date)
closing_prices = data['Close']
daily_returns = closing_prices.pct_change().dropna()
plt.figure(figsize=(12, 6))
closing_prices.plot()
plt.title('Closing Prices (2024)')
plt.xlabel('Date')
plt.ylabel('Price (INR)')
plt.grid(True)
plt.show()
plt.figure(figsize=(12, 6))
daily_returns.plot()
plt.title('Daily Returns (2024)')
plt.xlabel('Date')
```

plt.ylabel('Daily Return')
plt.grid(True)
plt.show()

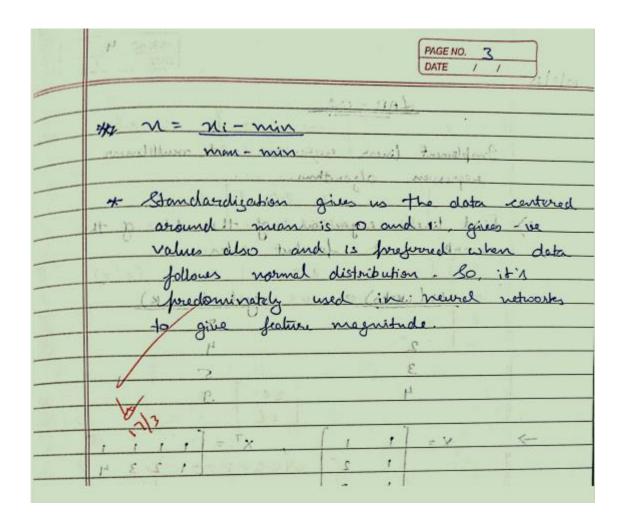
Program 2

Demonstrate various data pre-processing techniques for a given dataset.

Screenshot:

10/3/25	Pegessinh
_	DAR- OI - DATA PROCESSING
	Constant Austria est of manufes with the
	Load CSV file considering housing csv Import pandas as pd Load CSV file considering housing csv Import pandas as pd
Y	Load CSV file considering housing csv
192411	that A have retarded I trusted with 15 world
-7	import pandas as pd
100	ay 2 pd. read_csv(housing.csv')
161	all allies colored colored did to
000	Information of all columns
-7	allydos information =talfingo()throis all
1000	him w. C. frild (information) atter
A SPORTS	41 00 rehappy gab without Daylor
100	Parallel I I don't all numerical calculations
- 37	Statistical informed all numerical calculations
_	state = df. describe () the form
>	stats = df. describe C)
1150	Carlette and a second sound for "Ocean Provincity"
386	forist (comque) himas our
- 141	The care
-	To contrabute count attribute values having
- 57	10 Contratate organic than gers (0)
-	missing values in control
-	nissing values = of ismall () sum ()
>	missing values = of ismel (). sum ().
100	missing values - of the said of him
1	and the short of the second selection of
THE PARTY OF	11/4 11/20/20 20/10/10

P.	A SOUTH ATTENDED
15	which column in the dataset had missing values. How did you handle them?
	values. How did you handle them?
	was remark as all some that I have to
\rightarrow	None of the dataset (Diabetes and Adult iway)
	None of the dataset (Diabetes and Adult income) had missing values:
2>	Which categorical columns did you destify
	which categorical columns did you identify in the datasets. How olid you woode their?
→	1/ In dialectes datest, Grender and Class were
	the cottogonical variables. I ve used
	ordinal encoding for Grender as it
A.A.	had only 19 values (like M. or F)
	so that they get encoded to 0 1
	respectively. Order of the
	Class was encoded using one hot accoder as
	the values were more or they didn't have
1)	any ordered importance.
	is In adult income, there are 9 categorical
	variables and mariful-status, relationship,
	gender income house ordinally encoded,
	next were anoded with one hot sneeder.
21	1212 121
10	What is the adjustice between Min-More Scaling
	Januardiahon when would
U	you use one above the other.
->	* Min-Man Scaline cal H II all about
	* Min-Man Scaling scales the data between 0 tol
	in sheetic erange, such as Data Learning of
	State of the state



from google.colab import files

uploaded = files.upload()

uploaded = files.upload()

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder

from sklearn.preprocessing import StandardScaler, MinMaxScaler

from scipy import stats

import pandas as pd

```
df1=pd.read_csv("adult.csv")
print(df1.head())
df2=pd.read_csv("Dataset of Diabetes .csv")
print(df2.head())
df1["education"].value_counts()
ordinal_encoder = OrdinalEncoder(categories=[["HS-grad",
"Some-college", "Bachelors", "Masters", "Assoc-voc", "11th", "Assoc-acdm", "10th", "7t
h-8th", "Prof-school", "9th", "12th", "Doctorate", "5th-6th", "1st-4th", "Preschool"]]
)
df1["Education_Encoded"] = ordinal_encoder.fit_transform(df1[["education"]])
onehot_encoder = OneHotEncoder()
encoded_data =
onehot_encoder.fit_transform(df1[["gender","relationship","workclass","occupati
on", "race", "native-country", "income", "marital-status"]])
encoded_array = encoded_data.toarray()
encoded_df = pd.DataFrame(encoded_array,
columns=onehot_encoder.get_feature_names_out(["gender","relationship","workclas
s", "occupation", "race", "native-country", "income", "marital-status"]))
df_encoded = pd.concat([df1, encoded_df], axis=1)
df_encoded.drop(["education","gender","workclass","relationship","occupation","
race", "native-country", "income", "marital-status"], axis=1, inplace=True)
print(df_encoded.head())
normalizer = MinMaxScaler()
```

```
df_encoded[["fnlwgt","educational-num","capital-gain","capital-loss","hours-per
-week"]] =
normalizer.fit_transform(df_encoded[["fnlwgt","educational-num","capital-gain",
"capital-loss", "hours-per-week"]])
df_encoded.head()
df2.isnull().sum()
df2['Gender'] = df2['Gender'].replace('f', 'F')
ordinal_encoder = OrdinalEncoder(categories=[["F", "M"]])
df2["Gender_Encoded"] = ordinal_encoder.fit_transform(df2[["Gender"]])
onehot_encoder = OneHotEncoder()
encoded_data = onehot_encoder.fit_transform(df2[["CLASS"]])
encoded array = encoded data.toarray()
encoded_df = pd.DataFrame(encoded_array,
columns=onehot encoder.get feature names out(["CLASS"]))
df_encoded = pd.concat([df2, encoded_df], axis=1)
df2 = pd.concat([df2, encoded_df], axis=1)
df2.drop("CLASS", axis=1, inplace=True)
df2.drop("Gender", axis=1, inplace=True)
print(df2.head())
normalizer = MinMaxScaler()
df_encoded[["No_Pation","AGE","Urea","Cr", "HbA1c",
"Chol", "TG", "HDL", "LDL", "VLDL", "BMI"]] =
normalizer.fit_transform(df_encoded[["No_Pation","AGE","Urea","Cr", "HbA1c",
"Chol", "TG", "HDL", "LDL", "VLDL", "BMI"]])
df_encoded.head()
```

Program 3

Implement Linear and Multi-Linear Regression algorithm using appropriate dataset Screenshot:

111	
10/3/25	
-	Implement linear regression and multilinear
-	Implement linear regression
-	310978551
horet	of the date of the
a>	Find linear seguestion sales
-1 s F s	work and lander south
3.1	nu (weeks) yelsales in K)
- Herouth	20 cl
-	2 4
	3 5
	4 9
	1 24 4/1
→	$X = \begin{bmatrix} 1 & 1 \end{bmatrix}, X^T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$
	1 2 1 2 3 4
	1 3
	1 4
	x x = 1 1 1 1 1 = 4 10
	1 2 3 4 1 2 LIO 80]
	1 3
	$(x^{T}x)^{-1} = [30 - 10]_{V}$
	[-10 4 [120-100]
	tis = [15 -0.5]
	-0.5 0.2
	0.5 0.2

_	(x1x) x = 105 -05 [1+1 11 1]
	$(x^Tx)^Tx^T = \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 0.2 \end{bmatrix} \begin{bmatrix} 1.1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix}$
	= 1 0.5 0 -0.5]
	.: and MoME 0.3 450.1 Och mold 0.53 Jan all
	(of 10 day) and find " china " construction"
	(x1x) x11 =11 1 0.5 0 -0.5] [2]
_	C 2
-+	1 to 1 to 3 - 6/1 01 02 4
	enter missing 5
1	fra much and and the 13 1
5	manger of the private of the second of the granged
1	(0) -05)
	2.2
U2 15	meid = ((c) to ba) mais that I proported
outlant)	1111 4 = 70.5 + 2.2 K + + 1
1	in Also considered colorisal enforcement in
3	
-	is summer formet in historical
-	

>	Considering the 3 data csv diles (amade perception
	THEORY SCIENTIFICAL MAIN, CONT.
	furform any data preprocessing steps
→	
	the are explacing suppose Man values in
	enfusioner and test score (out of 10) in
	Shiring csv winy Sillna () Sunction: The other a files don't contain any missing value
	i missing value
	hising-of [enfurious] = hising-of [enfurious).
	20 - Hilling (a)
	2.6 9.000
	hiring - of [test- score (out of 10)] = him
	hiring-of [test owne (out of 10)] = hiring of [test scone (out of 10)] . fillna(mon)
Abs	2 G. Surva (mion)
8/	Also meading categorical cuperiona values in numeric format in hiring.csv.
	in numeric format in biring car.
	we also
	use also encoded "State" column in 1000 compenies
	thing on shot One Hot cucading.
a>	For much 1
	For canada-per capita-income csv reisualize the regression line with the date points? What close the plat toll you about the
	closes the blat till date points? What
	does the plot toll you about the relationship
	Tot copied Income
→	A sugrassion (:
	There is a positive linear relationship between year and her capit
	increases own to

The 3 - documents "Decreesings" (trainer & margan
3) For hiring csv, what is the predicted
salary for a candidate with 12 years
of touheriousing, to tost stone and 10
I interescent characterist in hand high as
I want was not below wo mayor strips
For 120 years of temporiques, 10 test seems
and 10] interview scare 8:0 = 10 homo
Pardicted solary; \$122,126.12 \$ 92268.07
Delight a logich recommended to
4) For 1000 companies csv did god imende
Cotegorical values?
- (W)P=01=
-> The state columns was encoded using
One- Hot Encoding as it's cutegorical data.
occupation = 1
(30 -0 -0) +1 (3 2 +2-)- +1
101
10, 11-10 665 10 1240 10 0601 11
tost distribution at datural (d)
TO SO MICH + RUE AS AN AS

import pandas as pd

import numpy as np

from sklearn import linear_model

import matplotlib.pyplot as plt

 $df = pd.read_csv('/content/housing_area_price.csv')$

df

plt.xlabel('area')

plt.ylabel('price')

```
plt.scatter(df.area,df.price,color='red',marker='+')
new_df = df.drop('price',axis='columns')
new_df
price = df.price
price
reg = linear_model.LinearRegression()
reg.fit(new_df,price)
print(reg.coef_)
print(reg.intercept_)
reg.predict([[5000]])
df_mlr=pd.read_csv('/content/homeprices_Multiple_LR.csv')
df_mlr
df_mlr.bedrooms.median()
df_mlr.bedrooms = df_mlr.bedrooms.fillna(df_mlr.bedrooms.median())
print(df_mlr)
reg = linear_model.LinearRegression()
reg.fit(df_mlr.drop('price',axis='columns'),df_mlr.price)
reg.coef_
reg.intercept_
#Find price of home with 3000 sqr ft area, 3 bedrooms, 40 year old
reg.predict([[3000, 3, 40]])
112.06244194*3000 + 23388.88007794*3 + -3231.71790863*40 + 221323.00186540384
canada=pd.read_csv('/content/canada_per_capita_income.csv')
canada
```

```
plt.xlabel('year')
plt.ylabel('per capita income (US$)')
canada.rename(columns={'per capita income (US$)':'income'},inplace=True)
plt.scatter(canada.year,canada.income ,color='red',marker='+')
new_df_canada = canada.drop('income',axis='columns')
new_df_canada.sample(5)
income = canada.income
income.sample(5)
reg = linear_model.LinearRegression()
reg.fit(new_df_canada,income)
print(reg.coef_)
print(reg.intercept_)
reg.predict([[2020]])
hiring=pd.read_csv('/content/hiring.csv')
hiring
hiring['experience'].fillna(0, inplace=True)
hiring['test_score(out of 10)'].fillna(hiring['test_score(out of 10)'].mean(),
inplace=True)
def convert_to_int(word):
word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6,
'seven':7, 'eight':8,
'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
return word_dict[word]
hiring['experience'] = hiring['experience'].apply(lambda x : convert_to_int(x))
```

```
hiring
reg = linear_model.LinearRegression()
hiring.rename(columns={'salary($)':'salary'},inplace=True)
reg.fit(hiring.drop('salary',axis='columns'),hiring.salary)
print(reg.coef_)
print(reg.intercept_)
#What is the predicted salary for a candidate with 12 years of experience, 10
test score, and 10 interview score?
reg.predict([[12, 10, 10]])
comp=pd.read_csv('/content/1000_Companies.csv')
comp
comp.isnull().sum()
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(sparse_output=False, handle_unknown='ignore')
state_encoded = ohe.fit_transform(comp[['State']])
state_encoded_df = pd.DataFrame(state_encoded,
columns=ohe.get_feature_names_out(['State']))
comp = pd.concat([comp, state_encoded_df], axis=1).drop(columns=['State'])
print(comp)
reg = linear_model.LinearRegression()
reg.fit(comp.drop('Profit',axis='columns'),comp.Profit)
print(reg.coef_)
print(reg.intercept_)
reg.predict([[91694.48, 515841.3, 11931.24,0,1,0]])
```

Program 4

Build Logistic Regression Model for a given dataset.

Screenshot:

12/3/25	PAGE NO. 8 DATE / /
	dAB 3 - downstie REGRESSIAN (BINARY & MULTIPLE
	Consider a binary classification problem where we want to predict a student will pen
	or fail based on their study hower to
	logistic regression model has been trained to the learned parameter are as = -5: (interest)
4	and at = 018 (co. efficient for study hours).
- 6	@ White a logistic regression equation to
- 40	of this of problem is recorded and to
-	$P(x) = \frac{-(\tilde{\alpha}_{1} + \tilde{\alpha}_{1} + \tilde{\alpha}_{2})}{(1 + \tilde{\alpha}_{2} + \tilde{\alpha}_{2} + \tilde{\alpha}_{2})} $
ė	determination of an industrial of
-	equation = 1 -(-5+0.8 m) 1+e (5-0.3 m
	1+0 11+0
	(b) Colo let - + -
	(b) Calculate the probability that a student who studies for 7 hours will pass.
	-> P(N=3)= 1 = 0.645
	The same of the lands of
	O Determine the predicted class (pass or fail) for this student based on threshold of as
	-> Threshold is 0.5. Hence, pass (64.5)

40	Consider 7= [2,1,0] for three classes. Apply
-	estman function to find the probability
	values of three classes to de
_	modern was booken senate where treath
	21 /2 2 0 /21 1 1 1
_	Stran (2)
	Softman(2) = Circu I reidendre Zei Eilendreider ged all (
	the time collision and sell to
	The second of the second of the second of
	1. 11. P(2) 17 18 et luit 011665tol.
- Au	1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
9,85	well as the proce + e'+ co
-oden	likely to lease the
1000	of Juli P(1) = 4 more into de Frago de 245th &
1	is also well to the test to the sail
	the company when the in
-	Je 10 = 0 0 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	1000 PC 102+e'+e'
-	, 33me). C +C (C
	2 6 5 12 202
- 43	b[510] + [0.902 1.0.502
0	to initiate adold hours love to the
	the the sale was the
	En consideration
1	13 sent may be produced at one today (1)
7 6	1) The same of the
+	The doing of S library williams

For dataset dile "HR_commas sepresis". :> which variable did you identify having a direct and clear impact on employee returnion? why? / () -> The key variables are: · Satisfaction level > Employees with bour "s " Saksfortion buste are likely to have the compo · no - of - project => oreviework is more likely to leave · Time spent in -> Employees who short more time in the comp are mon likely to 000.0 "3+3+5 leave. Balony -> Employees with bien salery have higher attrition rate. ii) what was the occuracy of your logistic regression model? Do you think is it good accuracy? why or why not? -> Accuracy of the model: 74.83%. The occuracy is pirty good for classification problems but not ideal. accuracy = accuracy - scare (y-test, y-pred)

47	For Zoo detast report and doldes to
10	Fort Zoo detast refet male doldes di
-	> Did you person any dater profracessing stops?
	If yes, then what were they and why were
	they necessary?
-	to all indicate and the stand
	> Dralled received whom a collision of the
	> Dropped animal name column as it's not
	a useful frature for classification.
	n: and molimon of each 21 212
- 130 130	No missing votes was found
	· We meanged class typest with the dataset
3333	to map class-type to human-oreadable
1	(aluls (like mammal, bird etc).
	the state of the second of the
-	if were there any missing or incomistant values
	in the detaset? How did you handle
	them!
1897	
_	-> No missing values were found in the
Acres 197	dataset. The data was claim and
3850	ready for modelling.
7630 13	
333.	is collect does the contraine material till
13885	about the hestormance of your model?
350	and the programmance of going winder.
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	-> te -Ti
13500	The confusion matrin shows that the
17/17/11	
	being correct.
	Some misclassification like amphibians and
1000	mammals, suggest there may be overlaps
	in the features lastreen those classes

	is which class types were most frequent
	misclassified? My do you think the
Suple	happened ?
1532 6	so has just must text meat the is
	-> Amphibians and Mammals were
	most broughthe mis clouitied
tow i	amphibians being confused as marinely
	and the same
	This is due to similarities in certin
	patures (like fail, leg & etc) between
40-	These two closses.
	1 (1: 1 1 1 2 col) down and
	A f sta louid from our est st) what
alone d	Marina in a partient of methods Kit.
-01.	al los the Lat P. J.

import pandas as pd

from matplotlib import pyplot as plt

from seaborn import regplot

import seaborn as sns

df1=pd.read_csv('HR_comma_sep.csv')

df1.sample(10)

plt.figure(figsize=(8, 6))

sns.barplot(x='salary_encoded', y='left', data=df1)

plt.title('Impact of Employee Salary on Retention')

plt.xlabel('Salary Level (Encoded)')

plt.ylabel('Proportion of Employees Left')

```
plt.show()
plt.figure(figsize=(12, 6))
sns.barplot(x='Department', y='left', data=df1)
plt.title('Employee Retention Rate by Department')
plt.xlabel('Department')
plt.ylabel('Proportion of Employees Left')
plt.xticks(rotation=45, ha='right')
plt.show()
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
import numpy as np
import seaborn as sns
ohe = OneHotEncoder(handle_unknown='ignore', sparse_output=False)
department_encoded = ohe.fit_transform(df1[['Department']])
department encoded df = pd.DataFrame(department encoded,
columns=ohe.get_feature_names_out(['Department']))
df1 = pd.concat([df1, department_encoded_df], axis=1)
df1 = df1.drop('Department', axis=1)
ordinal_encoder = OrdinalEncoder(categories=[['low', 'medium', 'high']],
dtype=np.int64)
salary_encoded = ordinal_encoder.fit_transform(df1[['salary']])
df1['salary_encoded'] = salary_encoded
df1 = df1.drop('salary', axis=1)
df1.head()
correlation_matrix = df1.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
```

```
plt.title('Correlation Matrix of Features')
plt.show()
correlation\_threshold = 0.1
correlated_features = correlation_matrix['left'].abs() > correlation_threshold
highly_correlated_features =
correlated_features[correlated_features].index.tolist()
new_df = df1[highly_correlated_features]
print(new_df.head())
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
X = \text{new\_df.drop('left', axis=1)}
y = new_df['left']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
example_data = X_test.iloc[0].values.reshape(1, -1)
prediction = model.predict(example_data)
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
zoo_data = pd.read_csv('zoo-data.csv')
zoo_class = pd.read_csv('zoo-class-type.csv')
merged_data = pd.merge(zoo_data, zoo_class, left_on='class_type',
right_on='Class_Number')
merged_data = merged_data.drop(['Animal_Names',
'Number_Of_Animal_Species_In_Class',
'Class_Number','class_type','animal_name'], axis=1)
X = merged_data.drop('Class_Type', axis=1)
y = merged_data['Class_Type']
print(merged_data.head())
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

```
import matplotlib.pyplot as plt
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

Program 5

Use an appropriate data set for building the decision tree (ID3) and apply this knowledge to classify a new sample.

Screenshot:

2313h	5	7	AB-4		PAGE NO. 12 DATE / /			
	Decision	Tree	(Clo	orification)	Link Control			
9)	menty	gain whether	The	splitting v	culate satory rule "Classification" rade should be	_		
	Instruce	Q2	α3	Classification	m			
/1	1 2 2 4 1	hot	high high	No No				
	7	Cool hot	high high	No No				
→			normal .	Yes.				
	Entropy of entire dataset $5[1+, 4=]$ $\frac{1}{5} \log_2(\frac{1}{5}) - \frac{3}{5} \log_2(\frac{1}{5})$							
	5 0-3310							
	Superior Sain for az, value [Hot. cold]							
	Spot [1+,3-] = F(Smot) - 1 log (1) - 3 log (3)							
	Scord lo+				A COLOR			
	2000	(25)		(col)				

Crain (5, az) = E(5) - 2 501 200						
+						
= 0.7719 - 1 E (Sweld) + 69 4 E (Sweld)						
= 0.7719 - [151 E (Swild) + 694 E (Shul) = 5						
Translated to the supplied the						
= 0.7219 - 0.64396.						
= 9.0713						
The state of the s						
Informain gain for az value [High, normal]						
Informain Jain for as value - 115h, hornes						
2 6 2						
S'nuh [0+, 4-] = 3						
Snormal [1+, 0] = 2						
Imparmation gain of as wat entire dataset						
Charles of V= Flor - 5 Isul Elect						
Chain (s, as) = E(s) - 5 51 E(s)						
131						
1/4/-0 =						
= 0.7219 - 4 (0) +1 (0)						
I have told I release on many my try some and I						
= 0 · 7219						
- 5 / 1 1 m 1 - 1 x 1 = 1 m 1 1 / 2 m						
(Crain (s, az) = 0.0728						
Gain (sa) - 22 a						
Gain (s,9) -0.2219						
W. X.						
high (az)						
high Normal						
(No) (Yes)						
1,2,6,7						

i	For iris csv date set
7	and the state of t
	a what was the accuracy awar for the inis delast?
	-> The accuracy sweet for the inic data.
	> The accuracy sure for the isis deternet is 1 (100%), which indicates that the
	model connectly clarified all the test
	semples.
	Continion matrin ! [10 0 0)
	Confusion matrin! [10 0 0 0]
	0 0 100 11
	and Audah Province
THE T	accuracy (cetosa) = 10+(11+5) = 1
	10+ 11+9+
	and analysis depleading the con-
	Occuracy (versicolor) = 9+(10+11) = 1 10+9+11
	10+9+11
	Series of the se
	occuracy (virginica) = 11+ (10+5) =1
	10-15+11
- 1.	TXT LE LE MAN - MAN - TXT
	duly my of (2 min (A)
	6 What does the confusion matrix tell's you
	about the modelin performance? Were there
	any misclandication? It so alich to
	were most confused.
-	Olive T more than the section of the
	-> Confusion matrix: [10 0 0)
_	-> Confusion matrin: [10 0 0]
1	10011
A TOP OF	

1	1 Edit Orbid
	The rows represent the actual class and
+	columns represent Anedicked class
	constitute classified instance for each
	All values outside the diagonal element
	whit other than diagonal element
	outside other than diagonal elements are o meaning there were no mischary.
	6 mantes
	lince, no mis dessification occurred, no dem
	Since, no mis clessification occurred, no day
	here conjused.
مرد	For "petrol comumphon" csv daknet
	1 ((())) () ()
	@ Can you interpret the Regression True What
	are the most impostant features from
	predicting patrol consumption?
	14.8 1 01
	-> > Population driver's license -> More driver
	leads to more petrol consumption.
	ii) Freome per cepita > Higher income may led (Avg. income) to more which usage.
-	to more while usage.
10	Posted Mail
The state of	iii) Road length (km) -> Moore would may
	conselete with ince of the demander
	iii) Read lingth (km) -> More roads may conrelate with increased ful demande
	6) How does the Regression True handle
	the A target variables compared to
	the Decision True classifier.

-	ix Regression True bredicts continuous coluce								
	i) Regression Trus predicts continuous values, while Classification True predicts categories								
	Land Categories								
-115	iix Read	union True v	naimina .	1 minus (
-	Const	ul 122221	instant of	mani mizing					
-	44	ما الأناب	No cición	Ti. Suns iii					
31/108				Tree manisizes					
3,30	class	huritz							
tele			n Piles is	. 1	, .				
- y est	mit Regul	ssion true f	redicts la nu	meric Value while	<u> </u>				
16	Du	ision tree	spredicts a	class label.					
1-631	1/	6.04	277	2					
	ist Regn	ession tree use	1 the mean	of sampling Sample	1				
	int	we leaf noc	le while	Decision tres					
	anie	no the mo	st frequent	velue accessed in					
	the	leal node.							
		led node.	Mark 1 .	21 12					
-	, 0	18 2 - TABLES	Ch. + Newson 11.	- 14					
	714	1 2 Table 13							
	11.01	1 317 (151-28)	+1 40- 4011	- 41					
	. 0 8	7 12 12 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1							

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, classification_report

 $data = {$

'a1': [True, True, False, False, False, True, True, True, False, False],

'a2': ['Hot', 'Hot', 'Hot', 'Cool', 'Cool', 'Cool', 'Hot', 'Hot', 'Cool',

'Cool'],

```
'a3': ['High', 'High', 'Normal', 'Normal', 'High', 'High',
'Normal', 'Normal', 'High'],
'Classification': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'No', 'Yes',
'Yes', 'Yes']
}
data
df = pd.DataFrame(data)
label_encoders = {}
for column in df.columns:
le = LabelEncoder()
df[column] = le.fit_transform(df[column])
label_encoders[column] = le
df
X = df.drop('Classification', axis=1)
y = df['Classification']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
clf = DecisionTreeClassifier(criterion='entropy')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print(classification_report(y_test, y_pred, target_names=['No', 'Yes']))
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(12,8))
plot_tree(clf, filled=True, feature_names=X.columns, class_names=['No', 'Yes'])
plt.show()
iris=pd.read_csv("/content/iris - Copy.csv")
iris
le = LabelEncoder()
iris["species"] = le.fit_transform(iris["species"])
label_encoders[column] = le
iris
X = iris.drop('species', axis=1)
y = iris['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
clf = DecisionTreeClassifier(criterion='entropy')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print(classification_report(y_test, y_pred, target_names=['Iris-setosa',
'Iris-versicolor', 'Iris-virginica',]))
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
```

```
xticklabels=['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'],
yticklabels=['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
plt.figure(figsize=(12,8))
plot_tree(clf, filled=True, feature_names=X.columns,
class_names=['Iris-setosa', 'Iris-versicolor','Iris-virginica'])
plt.show()
drug=pd.read_csv("/content/drug - Copy.csv")
print(drug)
drug["Drug"].unique()
le = LabelEncoder()
drug["Sex"] = le.fit_transform(drug["Sex"])
drug["BP"] = le.fit_transform(drug["BP"])
drug["Cholesterol"] = le.fit_transform(drug["Cholesterol"])
#drug["Drug"] = le.fit_transform(drug["Drug"])
label_encoders[column] = le
drug
X = drug.drop('Drug', axis=1)
y = drug['Drug']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
clf = DecisionTreeClassifier(criterion='entropy')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print(classification_report(y_test, y_pred, target_names=['drugY', 'drugC',
'drugX', 'drugA', 'drugB']))
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['drugY', 'drugC', 'drugX', 'drugA', 'drugB'],
yticklabels=['drugY', 'drugC', 'drugX', 'drugA', 'drugB'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
petrol=pd.read_csv("/content/petrol_consumption - Copy.csv")
petrol
X = petrol.drop('Petrol_Consumption', axis=1)
y = petrol['Petrol_Consumption']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
clf = DecisionTreeRegressor()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred)**0.5
print(f'Mean Absolute Error: {mae}')
print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
plt.figure(figsize=(30, 30))
plot_tree(clf, filled=True, feature_names=X.columns, fontsize=10)
plt.show()
```

Program 6

Build KNN Classification model for a given dataset.

not:					
		JAE-05			1
3 19125	and the same of th				
	K:	NIN CLASSE	CATION		1
			and the state of t	1 -	
-	Penson	Age	Sclary K	Toney	tu
					1
	1 19 00	a declar	L	1	-
	A	18	56	H	502
	В	_23	517	M	465
1		24	30	1 11	319
1000	2	41	6	17	40.4
N . 10	O E COM	43	30	7	31
	F	38	40	2	Co:1
101-0	X	35	100		_
	F 7.55	-			
	Use, 1	(=3,	-in-	-	_
			2		
	Formula	2 J (NL-	u,)2+(y2-y,)2	2	_
		11. 22 6	35-13)2: 52	0	-
	DA =	1 (100-20) +1	25-18) = 5 L	11/1	
	N =	[haz =0) 14	(35-323) Le 4	h. b	
		1600 335 1	(3.3)		
	De E	1 (100 - 70)2 t	13-22) = z	31.9	
		1			
	D) =	[(100 - 60)]	1 135-45) =	40·4	
			The second of		
	DE :	1 (100 - 70	12+ (35-41) 1 =	30 31	
			/		
	Dr =	1 100 - 4	52 + (35/38)2 =	60.1	+
					_
	do				-
	No. of the last of	T DOWNSON BY	of point x from		

Value of the same	
-	Meanst 3 neighbours are C, B, E and
-	their traget is al, my, it mesherively
T	taget vanable for
	Hence, X will be Y.
	god ad and driene las I willy will
	and the second second second
	al accessor that the second of
-9	For inis dalant, how to choose the k value?
	Demonstrate using accuracy rate and error
_	out .
	To a subject to the late of the
->	Too small - y very sensitive to roise
	Teo lance - man ismore local bettern
-total	Value
wild.	Too large -> may ignore local fattern
	For accuracy to be 35%, best value of k
	Should be 5 1. c. 1 = 5 (if accuracy
	15 93% at k=3)
	Frenz sate = 1 - accuracy
1	For error tate to be optimum, then k=5 i.c. 5". error should be
- A 13-11	k=5 i.c. 5% erron should be
	optimum (if error note is to 1. at k=3).
	A TOTAL CONTRACTOR OF A
10>	For delecter date of set, what is the purpose
- 4/	of feature scaling? How to furthern 17?
	of justice steering.
->	Feature scaling is good for dialuter datast
	0 0

1	to grance of independent hat
1 bin	to normalize the grange of independent features
buste	(input variables), as pecially when using distance - based algorithm like KAIN, SVM etc.
6	distance - based algorithm
	Many feations if not scaled the transfer
	Account the followings
	with higher glate value will dismonate
5 900	no dissense.
46.4	looking helps also features to contribute
	equally.
	, 0
	Drive Standard scalar mother de
	Using Standard scalar method we will use the formula
	May a sent and the
	7 - n- mean - / ton Normaline
100	7 = n- mean (for Normaligate std. deviation distribution
- 31	Wary MinMan scalar mathodisce will
G G	the lands of the state of the s
	use the formula to NED 21
	4 7- 10-halla /154 4
	15 man - min - distribution
V	solved - man - man
1	a 1. 7 1 st a come Vida mentala
79.	/Z => 7-score which transforms a value so
Lasta	I that the whole feature his
V.	mean 20 and standard aleutation = 1
-3-1-	

import pandas as pd

iris=pd.read_csv('iris.csv')

iris.head()

 $from \ sklearn.model_selection \ import \ train_test_split$

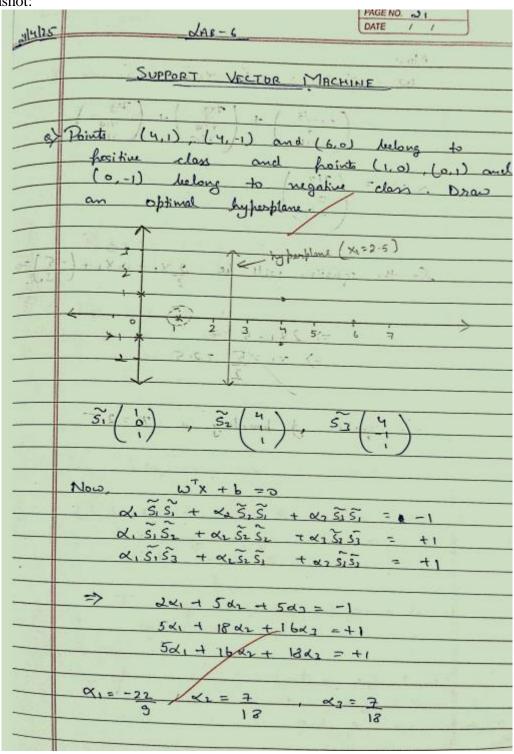
 $from \ sklearn.neighbors \ import \ KNeighbors Classifier$

```
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
X = iris.drop('species', axis=1)
y = iris['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
best_k = 1
best_accuracy = 0
for k in range(1, 11):
   knn = KNeighborsClassifier(n_neighbors=k)
   knn.fit(X_train, y_train)
   y_pred = knn.predict(X_test)
   accuracy = accuracy_score(y_test, y_pred)
   print(f''Accuracy for k=\{k\}: {accuracy}, Error Rate for k=\{k\}:
   {1-accuracy}")
   if accuracy > best_accuracy:
           best_accuracy = accuracy
           best_k = k
print(f"Best k value: {best_k}")
knn = KNeighborsClassifier(n_neighbors=best_k)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt='d',
cmap='Blues',xticklabels=iris['species'].unique(),
yticklabels=iris['species'].unique())
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

Program 7

Build Support vector machine model for a given dataset



P. D.	AGE NO. 22
3 3/1/2	
Nlow,	
W = 12 xisi = 0xisi + 225, + 22	53
= (-22/5 \ + / 2x4)+ (1 EVE
= (-72/5) + (7x4/18) + (-71/8) + (-71/8)	-3/18
2/3	01
blastid builde	1.00
(2013)	
Company and the many and	mat war
So, the equation will be 2 X, +	0 X2+ (-5)=
3	(3)
The same of the sa	
=> 2 × 1 - 5 70	
$= \frac{1}{2} \times 1 - 5 = 2.5$	
7 2	
equation of boundary is Xi	= 2.5
	J
1-9 -2- 41	
0= d+ x'c	sald:
1-1-135-00 - 3.5.2 - 3.5.3	

1	For insecs datast,
-	and the state of t
	a what is the accuracy score of the classifier using the linear kernel and RBF kernel?
	using the linear formal and RBF kernel?
	1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
	-> Linear kund: 1.0(100:1) 344
	RBF kornel: 1.0 (100%.)
0	Description of the second of t
	(b) which kernel gove better histornance on the
	inis dataset? why 1
	> Both permet gave the same preformance
	with 100 % accuracy to the
	data data data
	Since the dataset is not complen or
	averlepping, both linear and REF ternel
	forform will the st MV2 C-
	and dead and and and
_2>	For letter-recognition.csv dataset,
	ALLE COLLEGE STATE OF THE STATE
~	@ Bresent and interpret the confusion motorn Are
	there any specific letters that are becausely
	in others!
	The superior of the state of th
	-> The diagonal climents are convect predictions.
	Garagonal elements show entron.
19,1	son to it be now a series of the
	Letter 'G' is sometimes conjused with 'C' h'a
	12 10 10 10 10 10 10 10 10 10 10 10 10 10
	detter like 'O' and 'a' are misclarified.

-	1 what is AUC score, and how closes it
	10 what is AUC Kent, humance
	grefact 1
	how well the
	-> AUC Score shows lecture clane
	-> AUC score shouss letteren clanes.
	AUC value scarie classe to 1.0 means
	better ferformance.
	Auc = 0.5 -> model is guessing randomly
- 4/42	AUC = 0.8 -> herfed classifier
	1 3 3 4 1 1 1
	AUC Score = 0.994.
1000	AUC scare = 0.551
	@ How does the funformence of the SVM model on this clotast compare to its TOIS detect
191	performance on the IRIS dateset
1000	
	SVIM IS USER TOR
	1 the land brokken
	tudah was will asse that a
	On JRIS detest it's cary because
2019	there was a subject of
	A A O V & A CON VARIANTE
,	Santin After Associa
	Ox detter datout, performance is
THE	Hill strong despite 26 closes.
42.50	1 addies
	This shows the hower of SVM maners
	multi-class problems officiently.
9	214
useries i	and of the of add added

import pandas as pd

iris=pd.read_csv('iris.csv')

iris.head()

iris.isnull().sum()

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
iris = pd.read_csv('iris.csv')
X = iris.drop('species', axis=1)
y = iris['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
svm_rbf = SVC(kernel='rbf', gamma='scale')
svm_rbf.fit(X_train, y_train)
y_pred_rbf = svm_rbf.predict(X_test)
accuracy_rbf = accuracy_score(y_test, y_pred_rbf)
print(f"Accuracy (RBF Kernel): {accuracy_rbf}")
cm_rbf = confusion_matrix(y_test, y_pred_rbf)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_rbf, annot=True, fmt="d", cmap="Blues",
xticklabels=iris['species'].unique(),
yticklabels=iris['species'].unique())
plt.title("Confusion Matrix (RBF Kernel)")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
svm_linear = SVC(kernel='linear')
```

```
svm_linear.fit(X_train, y_train)
y_pred_linear = svm_linear.predict(X_test)
accuracy_linear = accuracy_score(y_test, y_pred_linear)
print(f"Accuracy (Linear Kernel): {accuracy_linear}")
cm_linear = confusion_matrix(y_test, y_pred_linear)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_linear, annot=True, fmt="d", cmap="Blues",
xticklabels=iris['species'].unique(),
yticklabels=iris['species'].unique())
plt.title("Confusion Matrix (Linear Kernel)")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
df2=pd.read_csv('letter-recognition.csv')
df2.head()
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelBinarizer
X = df2.drop('letter', axis=1)
y = df2['letter']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
svm_classifier = SVC(kernel='linear', probability=True)
svm_classifier.fit(X_train, y_train)
y_pred = svm_classifier.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap="Blues")
plt.title('Confusion Matrix for SVM')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
lb = LabelBinarizer()
lb.fit(y_test)
y_test_lb = lb.transform(y_test)
y_pred_prob = svm_classifier.predict_proba(X_test)
fpr = \{\}
tpr = \{\}
thresh = \{ \}
roc_auc = dict()
n_{class} = y_{test_lb.shape[1]}
for i in range(n_class):
   fpr[i], tpr[i], thresh[i] = roc_curve(y_test_lb[:,i], y_pred_prob[:,i])
   roc_auc[i] = auc(fpr[i], tpr[i])
```

```
plt.plot(fpr[0], tpr[0], linestyle='--',color='orange', label='SVM (AUC = %0.2f)' % roc_auc[0])

plt.title('ROC Curve for Class 0')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive rate')

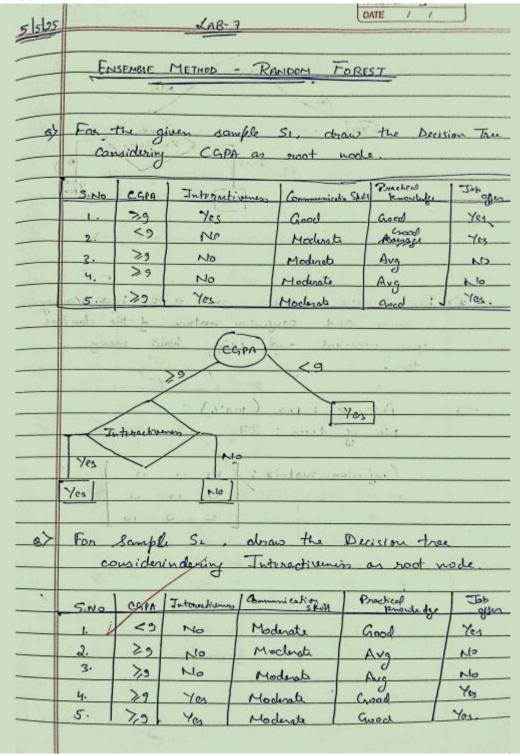
plt.legend(loc='best')

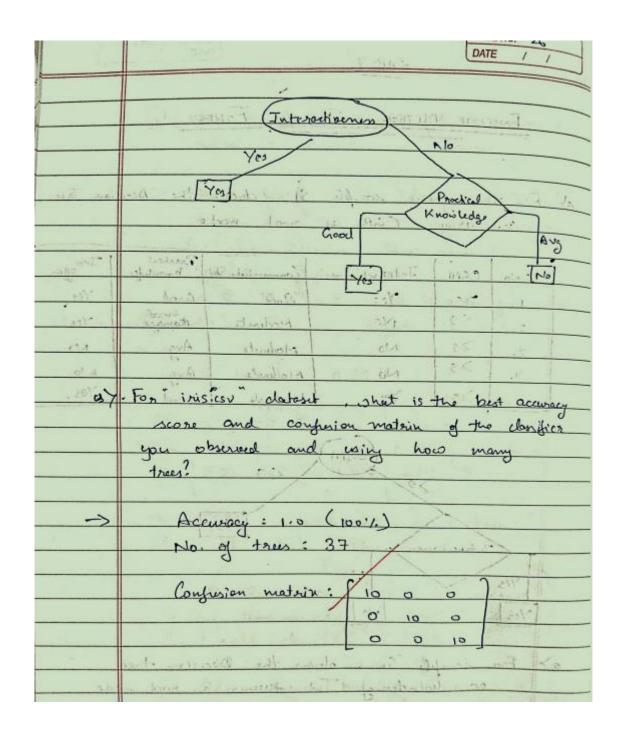
plt.show()

print(f''AUC score for class 0: {roc_auc[0]}'')
```

Program 8

Implement Random forest ensemble method on a given dataset.





import pandas as pd

from sklearn.model_selection import train_test_split

 $from \ sklearn.ensemble \ import \ Random Forest Classifier$

from sklearn.metrics import accuracy_score

data = pd.read_csv('iris.csv')

```
X = data.drop('species', axis=1)
y = data['species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
rf_classifier = RandomForestClassifier(random_state=42)
rf_classifier.fit(X_train, y_train)
y_pred = rf_classifier.predict(X_test)
default_accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy with default n_estimators (10): {default_accuracy}")
best_n_estimators = 10
best_accuracy = default_accuracy
for n_{estimators} in range(1, 101):
   rf classifier = RandomForestClassifier(n estimators=n estimators,
random_state=42)
   rf_classifier.fit(X_train, y_train)
   y_pred = rf_classifier.predict(X_test)
   accuracy = accuracy_score(y_test, y_pred)
   if accuracy > best_accuracy:
           best_accuracy = accuracy
           best_n_estimators = n_estimators
print(f"Best accuracy: {best_accuracy} achieved with n_estimators:
{best_n_estimators}")
from sklearn.metrics import precision_score, recall_score, f1_score
best_rf_classifier = RandomForestClassifier(n_estimators=best_n_estimators,
random_state=42)
```

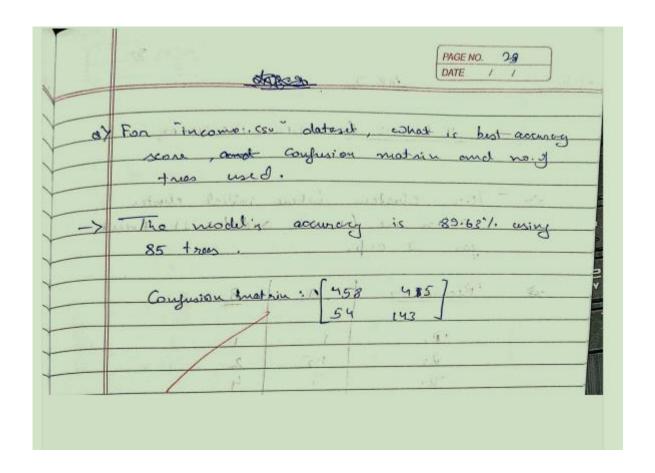
```
best_rf_classifier.fit(X_train, y_train)
best_y_pred = best_rf_classifier.predict(X_test)
precision = precision_score(y_test, best_y_pred, average='weighted')
recall = recall_score(y_test, best_y_pred, average='weighted')
f1 = f1_score(y_test, best_y_pred, average='weighted')
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1-score: {f1}")
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
cm = confusion_matrix(y_test, best_y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
   xticklabels=data['species'].unique(),
   yticklabels=data['species'].unique())
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

Program 9

Implement Boosting ensemble method on a given dataset.

3/5/35		_dr	8-8	DATE /)
		ENISEMBLE P	157HOD- BO	OSTING CADABO	OST AL	رمه
		(No.e)	- Teller and	*		
3>	For	-H3 - 0	ú · · ·		200	
5/	der	ision a	LA 1	mple data the	is to	4
	for	the atts	whate Capa	to Colculation using AdaBo	-20g	3
	al	gositm.		Many Manyo	net-	
	390	ortical is	1-9 8 VVI	× 14 105		
	CGPA	Interactionen	Proactical Knowled	Communication S	1 1 1 1 -	Tob graft
2263				E. S.		420
700	29	To	Good	1 - hood		Yes
	<9 ≥9		Good	Moderate		Yes
	<9	No	Ays som	Moderate		No
14.	7.9	No.	Ays	and		No
4.0	≥9	Yes	Chool	Moderate		Yes
	1	163	Crood	Moderate:		Yes
->	The Tax	tial sut =	2.1 (1. 1.			
	1. 9	all	6	(bri) - market	1.7	
				100	_	
	CGPA	Producte	d Job Open	Actual Job offen		- pdate
MAN A			01313	That dop offer	tes	Lowin
NI L	7,9	Ye		Com Tes	1/4	0.131
	129	No	1 anti-a	Yes	1/6	0.121
	1/3		> Yes	No	1/6	0.520
	< 9	No		N2	1/6	0.250
	7,5	Ye	1	Yes	1/6	0-124
	20	1 46		Yes	1/6	0.124
						10.124

			. (DATE
		-		
	0 0	20		
1 . A	Ecgen = 2 X1	= 0.33	3 11111611	1
weights Erro			-	
Enro	2 4	1	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	
312	×capn = 1h	1- ECCON	= 0.3	47
-cla	Lest March 1	ZCGFA /	pt.	ariah .
	+ 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	49,00 Capp	the at-	And Annual Contract of the Con
		_ 5	· and the	el n
- 4x -1	7 - 6	KHX Ge-E +		
1, 1	oromalizing 6	Partie Value	6	74.55
	Juctor	-0.317	0.29	2
201		X C -0.347 # 1	x2xe"	1 1 5
	stephold	A H Spee	0-1	1 65
eld	1 2: 0-	5428	ela.	168
- NA	2,62	33/5	d and	1.2
1937	up dated 12 (in	nitial wt) x e		for comety
	wt hard	7	-41	friedicated class
	uhdated = (ini)	×11 11 12×	1 -> 1	,
	updated = (inil	rial wolver		incorrectly
	- CI	122/2017	- June	diedred class
hatel t	Actual low class .	- 100 self do	R. Acet	4965
16	es (cot) uppated =		/	
	de d	/	\~	161
pus 6	(cot) years =	0.2501	A Y	120
10000	ivestruct	andres		PX: 2.
S. Arriva	310 314 3	1 1 24		10
2010	24			18 1
-28101	3.1 60%			
	II .			



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('income.csv')

label_encoders = {}
for column in df.columns:
    if df[column].dtype == 'object':
        le = LabelEncoder()
```

df[column] = le.fit_transform(df[column])

label_encoders[column] = le

```
X = df.drop('income\_level', axis=1)
y = df['income_level']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
abc = AdaBoostClassifier(n_estimators=10, random_state=0)
abc.fit(X_train, y_train)
y_pred = abc.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy with n_estimators=10: {accuracy}")
best_accuracy = 0
best_n_estimators = 0
for n_{estimators} in range(1, 101):
   abc = AdaBoostClassifier(n_estimators=n_estimators, random_state=0)
   abc.fit(X_train, y_train)
   y_pred = abc.predict(X_test)
   accuracy = accuracy_score(y_test, y_pred)
   if accuracy > best_accuracy:
           best_accuracy = accuracy
           best_n_estimators = n_estimators
print(f"\nBest accuracy: {best_accuracy} achieved with
n_estimators={best_n_estimators}")
```

```
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
abc = AdaBoostClassifier(n_estimators=73, random_state=0)
abc.fit(X_train, y_train)
y_pred = abc.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy with n_estimators=73: {accuracy}")
print(classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
xticklabels=["<=50K", ">50K"], yticklabels=["<=50K", ">50K"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix Heatmap")
plt.show()
```

Program 10

Build k-Means algorithm to cluster a set of data stored in a .CSV file.

ot:						
1			-			1
215/2		K- MEANS CLUSTE				14 ALGORITHM
1		K- ME	ANI	10-1	1900	-hard de la
1	1000	- 60.5	111111			Area coult
1			100400	1 (1 () () () () ()	ishere	initial cluster
1	> Two	- ch	MA	1.	1)	and (5,2) . Freut
1	Cen	tos	re	سلب.		not as
1	don	2	eu	þ		
1	4		1	1 7		to B was start
1	Reco	nd No		41.	<u>A</u>	
1		VAT.	_	1.81		
-	-	P,	10	1		
-		22		1:3		2
		23		3		4
		Py		5	7	7
		Rs_	H	3	5	5
		Re	4.		.5	5
	R2		3.5		5	4.5
	K)					
						S To Fair Law - To -
				- 1	.le.	alicha er
\rightarrow	We wi	11 use		Zuch	colean	(5,7)
					(2-2	
	Tole Koh		neita	1	1	Assigned Chates
	Record No	Percond No #4		R. C2	-	chister Chister
	18				1/2/2	0.
	RI		1	.21		C1
	R2	1.12	6	.12		21
	R3	3.61		61		21
	RY	7.21			(2
	R5	4.12	2.		(2
	RG	5.31	2.	06	C	.1
	R ₂	4.3	2.	92	C	2
	DE STATE OF STATE					

	For inis.co draw ellow plat. What was
9	the optimal K-value obtained?
	Acroston to cold and
	optimal K value = 3 as the ello
>	Optimal solvible at X=2
(1 F . C.	is clearly visible at X=3.
· = 11.//	1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
	201
	201
	150
-	100
	50 1
	1 2 4 6 8 10
	75 613
	8, 4 10 10 10 10 10 10 10 10 10 10 10 10 10
	9 1 50 1 46.1 10.5
RE	2.12 0.00
RI	2.74 1.07 1 (2
	The state of the s
	few centroid
	1 2 1 1
-/	C1= (1+1.5), (1+2) = (1.25,1.5)
/	
	C1= (3+5+3.5+4.5+35) (4+3+5+5+4.5) = (3.9,5.1)
00	100 A

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

```
from sklearn.preprocessing import MinMaxScaler
df = pd.read_csv('iris.csv')
df = df[['petal_width', 'petal_length']]
scaler = MinMaxScaler()
df[['petal_width', 'petal_length']] = scaler.fit_transform(df[['petal_width',
'petal_length']])
sse = []
k_rng = range(1, 10)
for k in k_rng:
   km = KMeans(n_clusters=k)
   km.fit(df)
   sse.append(km.inertia_)
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng, sse)
plt.show()
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(df)
df['cluster'] = kmeans.labels_
plt.scatter(df['petal_width'], df['petal_length'], c=df['cluster'],
cmap='viridis')
plt.xlabel('Petal Width')
plt.ylabel('Petal Length')
plt.title('K-Means Clustering of Iris Flowers')
plt.show()
```

Program 11

Implement Dimensionality reduction using Principal Component Analysis (PCA) method.

hot:	JAB-10
5/5/14	
	2
	PRINCIPLE COMPONENT ANDLYSIS
	The state of the sent of the s
8>	Given the data in table, reduce the dinensions
	from 2 to 1 using Porinciple Compount
	Analysis CPCA). Compute first principle
	component.
	I workers a figure for the second
_	
_	Frature Frankle 1 Frankle 2 Frankle 4
	X ₁
_	X2 11 1 9 5 - 219 1.
-	5.5 - 11-1
	Consider Eigen value os:
	λν 30.3849
-	
-	22 = 66 6 . 6151
-	01174.5
	Consider eigen vectors as:
	e1= [0.5574] , e2= [0.8303]
	1 -0.2303 1/4 - 1 11 0.55341
	12 22 11 -
\rightarrow	Mean:
	X1 2 4+8+13+3 30 8 0888 08 08 08
	4
	X2 = 11+4+5+17 = 8:57
	7
	Coul. 1 2 16 2 1
	Cov(x1x1)= 17 ((4-8)2+18-8)2+ (3-8)2+ (7-8)2
	28-13
_	= 14
	COV(x, x)= 1/M-8)(11-8:5)+(8-3)(4-8:5)
	Cov(x, x)= 1 (4-8)(11-8:5) + (8-2)(4-8:5) + (13-2)(5-8:5)+(2-8)(14-8:5)
	≥ -1

	Cov (xx, x,) = Cov(x, xx)=, -11
	(2x(x2x)=1/4,2x5)2 (1,2x2)
	(DV(X2, XV) = 1 ((1-8:5) 2+ (4.25)2 (5-8:5)2
-	C(4- 8-2-3-)
-	- 1 - 1 - 1 - 23 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
	Later Crest - 123 car frage
	S = Cov(x1,xxx . cov(x,x)
1 10	Coverson icover, moi
1	118 = 12 0
	1/15 = [14 - 11] -11 23
	L-11 23
0	Auto the second at
	Now, eigen values of the covariance making
1	1016-12 AGE 1- GE MOLAIN
	S->T =0
	Complete the contract of the
	202300 PL 2000 15
	=> 14-2 -11 = 22-372 + 201
	[-1(23-)]
	- nest1 2. 0
	=> >= 30.8489, 6.6151
	Compute Eigen excetors on take the ony
	given.
	gran.
	(CT-12500)
	C.T = (0.5574 (2 0.830)) [4-8]
	11-25
17	
	1) (1) (1) (1) (1) (1) (1) (1) (1) (1) (
-	() () () () () () () () () () () () () (

	Feature	Ex	Exz	Fx3	Exa
	×1	4	8	13	7
	XL	11	4	5	14
	First Principle	-4.3052	3.7361	5.6928	-5-1238
	Compound				
		1			
6 }	For 'heart as 'dataset, export the accuracy				
	score before and after offying PCA.				
	Jier Spring Ten.				
->	Accuracy July 200:				
	Aceuracy before PCA: SVM =: 0.8750				
	devices 2				
	Random forest: 0.8537				
		tandon for	est : 0.0	64.1	
	۸				
	Accuracy often 9cx;				
	SVM : 0-8700				
	Logistic Regardo; 0.2533				
	Random Farest: 0.8532				
	Original features: 15				
	Feshing after P.CA: 13.				
	//				
	1				
	16				

import pandas as pd

 $from\ sklearn.preprocessing\ import\ Label Encoder,\ One Hot Encoder$

df = pd.read_csv("heart.csv")

text_cols = df.select_dtypes(include=['object']).columns

le = LabelEncoder()

ohe = OneHotEncoder(handle_unknown='ignore', sparse_output=False)

```
for col in text_cols:
   df[col + '_le'] = le.fit_transform(df[col])
   ohe_results = ohe.fit_transform(df[[col]])
   df_ohe = pd.DataFrame(ohe_results, columns=[f"{col}_{i}" for i in
range(ohe_results.shape[1])])
   df = pd.concat([df, df_ohe], axis = 1)
df = df.drop(text_cols, axis=1)
print(df.head())
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, MinMaxScaler
df = pd.read_csv("heart.csv")
text_cols = df.select_dtypes(include=['object']).columns
le = LabelEncoder()
ohe = OneHotEncoder(handle_unknown='ignore', sparse_output=False)
for col in text_cols:
df[col + '_le'] = le.fit_transform(df[col])
ohe_results = ohe.fit_transform(df[[col]])
df_ohe = pd.DataFrame(ohe_results, columns=[f"{col}_{i}" for i in
range(ohe_results.shape[1])])
df = pd.concat([df, df_ohe], axis = 1)
df = df.drop(text_cols, axis=1)
scaler = MinMaxScaler()
scaled_values = scaler.fit_transform(df)
df_scaled = pd.DataFrame(scaled_values, columns=df.columns)
print(df_scaled.head())
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
X = df_scaled.drop('target', axis=1)
y = df_scaled['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
svm\_model = SVC()
lr_model = LogisticRegression()
rf_model = RandomForestClassifier()
models = {'SVM': svm_model,
'Logistic Regression': lr_model,
'Random Forest': rf_model
}
results = \{\}
for name, model in models.items():
   model.fit(X_train, y_train)
   y_pred = model.predict(X_test)
   accuracy = accuracy_score(y_test, y_pred)
   results[name] = accuracy
best_model = max(results, key=results.get)
best_accuracy = results[best_model]
print("Model Accuracies:")
for name, accuracy in results.items():
   print(f"{name}: {accuracy}")
print(f"\nBest Model: {best_model} with accuracy: {best_accuracy}")
```

```
from sklearn.decomposition import PCA
pca = PCA(n_components=10)
X_train_pca = pca.fit_transform(X_train)
X_{test_pca} = pca.transform(X_{test})
results_pca = {}
for name, model in models.items():
   model.fit(X_train_pca, y_train)
   y_pred_pca = model.predict(X_test_pca)
   accuracy_pca = accuracy_score(y_test, y_pred_pca)
   results_pca[name] = accuracy_pca
best_model_pca = max(results_pca, key=results_pca.get)
best_accuracy_pca = results_pca[best_model_pca]
print("\nModel Accuracies after PCA:")
for name, accuracy in results_pca.items():
   print(f"{name}: {accuracy}")
print(f"\nBest Model after PCA: {best_model_pca} with accuracy:
{best_accuracy_pca}")
```