# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



### **Machine Learning (23CS6PCMAL)**

Submitted by

Prakruthi B S (1BM23CS414)

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 **Prakruthi B S (1BM23CS414)**, 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.

Lab Faculty Incharge

Name: Ms. Saritha A N Assistant Professor

Department of CSE, BMSCE

Dr. Kavitha Sooda Professor & HOD

Department of CSE, BMSCE

## Index

| Sl.<br>No. | Date                  | Experiment Title   | Page<br>No. |
|------------|-----------------------|--|-------------|
| 1          | Z [ = Z = Z ( ] Z . ) | Write a python program to import and export data using Pandas library functions                                    | 1-7         |
| 2          | 3-3-2025              | Demonstrate various data pre-processing techniques for a given dataset   | 8-13        |
| 3          | 10-3-2023 1           | Implement Linear and Multi-Linear Regression algorithm using appropriate dataset                                   | 14-21       |
| 4          | 17-3-2025             | Build Logistic Regression Model for a given dataset  | 22-24       |
| 5          | 24-3-2025             | Use an appropriate data set for building the decision tree (ID3) and apply this knowledge to classify a new sample | 25-33       |
| 6          | 7-4-2025              | Build KNN Classification model for a given dataset   | 34-37       |
| 7          | 21-4-2025             | Build Support vector machine model for a given dataset   | 38-43       |
| 8          | 5-5-2025              | Implement Random forest ensemble method on a given dataset   | 44-46       |
| 9          | 5-5-2025              | Implement Boosting ensemble method on a given dataset  | 47-48       |
| 10         | 12-5-2025             | Build k-Means algorithm to cluster a set of data stored in a .CSV file   | 49-51       |
| 11         | 1 /.= )=/.(1/. )      | Implement Dimensionality reduction using Principal Component Analysis (PCA) method                                 | 52-54       |

#### Github Link:

 $\underline{https://github.com/prakruthi23/ML.git}$ 

### Program 1

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

| I had be a second as a second | DATE: PAGE   |
|---|--|
| Jabl  | traces we all rading !   |
|   | 100 mil 15 7 later   |
| method 1 . Entitlatizing valu   | res derectly ento patagram   |
| anscut your know value  | es , five nows of data wis   |
| column headings as "us  | N. Name, Marks"  |
| The second second   |  |
| imparts pandas as pd.   |  |
| dasa = & 'USW': ['101', 10:   | 21,7031, 10013,  |
| " Name ": ['Alice! . 'Bob', "   |  |
| (Lou 75,05, 76): " (Spark)  |  |
| dj = pd. Data Frame (data)  |  |
| pilat ("sample datas").   |  |
| punt(d) shead()).   | 2 00 at  |
| Method 2: Emparting datas   | ets from steleman datasets   |
| Loading diabetes datas  | sets skleandafasets, load  |
| Loading diabetes datas  | post load dlabetes.  |
| Loading diabetes datas  | post load dlabetes.  |
| from skiran datasets for deabetes - load deabetes   | post load dlabetes.  |
| hom skiran datasets for<br>deabetes -load deabetrs ()<br>dj = pd patapame (diabetr  | post load deabetes.  |
| from stream datasets for deabetes = load deabetes () all = pd. patapame (diabetes).  1) ['toract'] = deabetes to  | post load deabetes.  |
| from sklearn.datasets for<br>deabetes = load.deabetes()<br>a) = pd. patapame (deabetes)<br>teabure names).<br>dy ['tanget'] = deabetes.tas<br>punt ("Sample data:")   | post load deabetes.  |
| from sklearn.datasets for<br>deabetes = load.deabetes()<br>a) = pd. patapame (deabetes)<br>teabure names).<br>dy ['tanget'] = deabetes.tas<br>punt ("Sample data:")   | post load deabetes.  |
| from sklearn datasets for deabetes = load diabetes ()  a) = pd. Datapame (diabetes ()  b) = pd. Datapame (diabetes ()  feature names),  a) ['tanget'] = deabetes too  public ("Sample data:")  public (dj. head ()).  | post load dlabetes.  es.data, columns=dlabetes  eget                                 |
| from skiran datasets for deabetes = load deabetes () all patapame (deabetes) que l'tanget ") = deabetes to prent ("Sample data:")  prend (df. head ()).   | post load deabetes.  es. data, columns = deabetes  orget                             |
| from skiran datasets for deabetes = load deabetes (deabetes)  de pd. patapame (deabetes)  de ['target'] = deabetes too  prent ("Sample data:")  prend (de head ()).  Method 3: 2 mporting datas  Sample sales data ()   | post load deabetes.  es. data, columns = deabetes  orget  sets from a specific car f |
| from sklearn datasets for deabetes = load diabetes ()  a) = pd. Datapame (diabetes ()  b) = pd. Datapame (diabetes ()  feature names),  a) ['tanget'] = deabetes too  public ("Sample data:")  public (dj. head ()).  | post load deabetes.  es.data, columns = deabetes  rests from a specific columns      |

punt ("Sample duta") prent 1 dj. head (1) perst("In") methody: sowaloading datasets from existing dataset defastraces like raggle, UCT, mendaly, keel etc of = policial content/paraset of Deabeter acer! encoding = "aten-1" punt (" Sample data") prential head () TO 00 2 using the code given in the above sledes, do the exercise of the "Stock Market Data Analysts". considering the following Lettore Bank Utd., Beter Bank Utd, botak Mahandia Bank teckers: ["HOFCBANK, NO", "8 CRCTCANE, NS", "KOTABBANK, NS"] 2. Start data: 2024-01-01, Ad dates 2024-12-30. 3- plot the closing piece and daily return for all the three banks merkened Emport ytenance as 41. import pordas as pd. Emport matplotteb: pyplot as plt trologies : C'HIDECRANK NS", "ZCRCT BANK NS", "KOTALBANK NS" Startdate = "2024-01-01" end-date: "2024-12-30"

```
data= 41. download CECheux, start = start date, end =
    end-date, group by " " " " (bear")
    Kekes-data = data [ Heker].
     hokes -day C'Dasily Rehun's - Kikes -data l'close?
     plt. figure (figure c (12,6)).
     plt supplot (2,1,1)
     Kicker-data ['close']. plot (Herce = j "Stecher 3- closeng
            Pelce")
     plt. subplot (2,1,2).
     Keleer-data C'Dally return' 3. plot Chitle of " & Heller 3.
            Daily Returns", colors 'orange')
     plt. regul-layoute
     ple shows.
```

```
import pandas as pd
data = {
'Name': ['Alice', 'Bob', 'Charlie', 'David'],
'Age': [25, 30, 35, 40],
'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}
df = pd.DataFrame(data)
print("Sample data:")
print(df.head())
```

```
from sklearn.datasets import load_iris
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target
print("Sample data:")
print(df.head())
```

```
file_path = 'data.csv' # Ensure the file exists in the same directory

df = pd.read_csv(file_path)

print("Sample data:")

print(df.head())

print("\n")
```

```
df = pd.read_csv('/content/Mobiles Dataset (2025).csv', encoding='latin-1')
print("Sample data:")
print(df.head())
```

```
import pandas as pd
data = {
     'USN': ['101', '102', '103', '104'],
     'Name': ['Alice', 'Bob', 'Charlie', 'David'],
     'Marks': [25, 30, 35, 40],
}
df = pd.DataFrame(data)
print("Sample data:")
print(df.head())
```

```
from sklearn.datasets import load_diabetes
diabetes = load_diabetes()
df = pd.DataFrame(diabetes.data, columns=diabetes.feature_names)
df['target'] = diabetes.target
print("Sample data:")
print(df.head())
file_path = 'sample_sales_data.csv'
df = pd.read_csv(file_path)
print("Sample data:")
print(df.head())
print(df.head())
```

```
df = pd.read_csv('/content/Dataset of Diabetes .csv', encoding='latin-1')
print("Sample data:")
print(df.head())
```

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
tickers = ["RELIANCE.NS", "TCS.NS", "INFY.NS"]
data = yf.download(tickers, start="2022-10-01", end="2023-10-01",
group_by='ticker')
print("First 5 rows of the dataset:")
print(data.head())
print("\nShape of the dataset:")
print(data.shape)
print("\nColumn names:")
print(data.columns)
reliance data = data['RELIANCE.NS']
print("\nSummary statistics for Reliance Industries:")
print(reliance_data.describe())
reliance_data['Daily Return'] = reliance_data['Close'].pct_change()
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
reliance_data['Close'].plot(title="Reliance Industries - Closing Price")
plt.subplot(2, 1, 2)
reliance_data['Daily Return'].plot(title="Reliance Industries
- Daily Returns", color='orange')
plt.tight_layout()
plt.show()
reliance data.to csv('reliance stock data.csv')
```

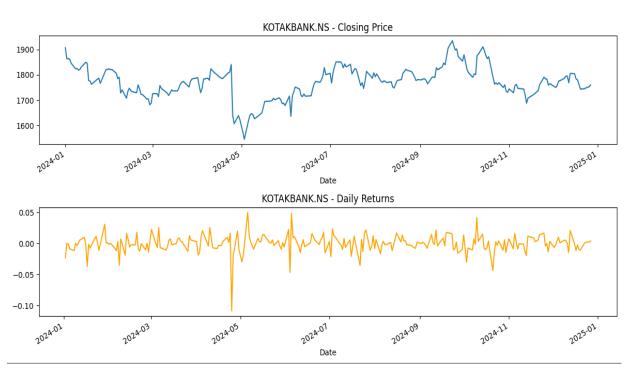
#### print("\nReliance stock data saved to 'reliance\_stock\_data.csv'.")

```
import yfinance as yf import pandas
as pd
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, group_by='ticker')
for ticker in tickers:
    ticker_data = data[ticker]
    ticker_data['Daily Return'] = ticker_data['Close'].pct_change()
    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1) ticker_data['Close'].plot(title=f"{ticker}-Closing Price") plt.subplot(2, 1, 2)
    ticker_data['Daily Return'].plot(title=f"{ticker}-Daily Returns", color='orange')
    plt.tight_layout() plt.show()
```

| Sa | mple data | :   |             |
|----|-----------|-----|-------------|
|    | Name      | Age | City        |
| 0  | Alice     | 25  | New York    |
| 1  | Bob       | 30  | Los Angeles |
| 2  | Charlie   | 35  | Chicago     |
| 3  | David     | 40  | Houston     |
|    |           |     |             |

```
Sample data:
   ΙD
                              City
           Name
                 Age
                  25
                          New York
     1
          Alice
     2
            Bob
                  30 Los Angeles
2
     3
       Charlie
                  35
                           Chicago
3
     4
         David
                  40
                           Houston
                           Phoenix
     5
            Eva
                   28
```

```
Sample data:
  Company Name
                            Model Name Mobile Weight RAM Front Camera \
0
         Apple
                       iPhone 16 128GB
                                                                      12MP
                                                  174g
                                                        6GB
         Apple
                       iPhone 16 256GB
                                                                      12MP
1
2
3
                                                  174g
                                                        6GB
                                                                      12MP
                       iPhone 16 512GB
         Apple
                                                  174g
                                                        6GB
                 iPhone 16 Plus 128GB
         Apple
                                                  203g
                                                        6GB
                                                                      12MP
4
         Apple
                 iPhone 16 Plus 256GB
                                                  203g
                                                        6GB
                                                                      12MP
                 Processor Battery Capacity Screen Size
  Back Camera
0
         48MP
                A17 Bionic
                                    3,600mAh 6.1 inches
         48MP
                A17 Bionic
                                     3,600mAh
                                               6.1 inches
                A17 Bionic
                                     3,600mAh
                                               6.1 inches
3
4
               A17 Bionic
A17 Bionic
                                     4,200mAh
                                               6.7 inches
         48MP
                                     4,200mAh
         48MP
                                               6.7 inches
  Launched Price (Pakistan) Launched Price (India) Launched Price (China)
                 PKR 224,999
                                           INR 79,999
                                                                     CNY 5,799
                 PKR 234,999
PKR 244,999
                                           INR 84,999
INR 89,999
                                                                      CNY 6,099
1
2
3
                                                                      CNY 6,499
                 PKR 249,999
                                           INR 89,999
                                                                      CNY 6,199
                 PKR 259,999
                                           INR 94,999
                                                                      CNY 6,499
  Launched Price (USA) Launched Price (Dubai)
                                                   Launched Year
                USD 799
0
                                       AED 2,799
                                                             2024
                USD 849
                                       AED 2,999
                                                             2024
1
2
3
4
                USD 899
                                       AED 3,199
                                                             2024
                USD 899
                                                             2024
                                       AED 3,199
                USD 949
                                       AED 3,399
                                                             2024
```



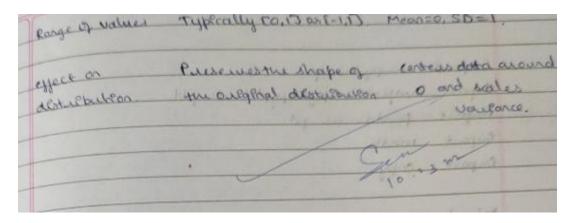
### Program 2

Demonstrate various data pre-processing techniques for a given dataset

| . 1    | ot:  |
|--------|--|
| (03/25 | The state of the s |
|        | Jab-2  |
|        | "h   |
|        | hiere python code, consider plename as "housing. csux  |
|        | Last 12 Louis  |
| Colo   | i. To load. civ file ento the data prame   |
|        | To display information of all columns  |
|        | in to display statistical in primation of all numerical  |
|        | iv. To display the court of unique labels por "ocean   |
| AS 12  | Proxhany column.   |
| -      | V. To display which attributes (colons) in a dataset   |
|        | mining values court greater than zero.   |
| E Same | 39 July Sale Cambar William Sale Sales 19874   |
|        | impost pandas as pd.   |
|        | housing.df=pd.sead.csv("Nowing.csv").  |
|        | pratigramation of all columns").   |
|        | pulled (housing of latas).   |
| -      |  |
|        | prent("In statistical enformation of numerical column")  |
|        | preathouses of Flores Line of runeileal columni)   |
|        | preathousing of t'ocean proximity J. value courts()  |
|        | pullet ("/ court of uneque tabels for " ocean Provenity")  |
|        | pullathousing-of-describers for " ocean Provenity")  |
|        | J-af-aescurbe()3   |
|        | punt (1/2 cales  |
|        | prent ("In column wern messeng values")  |
|        | prest (nesseng values (nesseng - of - Encell) - sunc)  |
|        | prest (nesseng values (messing-values) so)   |
|        | 0 3203   |
|        | 1  |

|       | Fax both diabetes and adult Income   |
|-------|--|
|       | the state of the s |
| 1)    | which columns in the dataset had missing values? How   |
|       | Emport numby as ap   |
|       | diabetes of = police d_csu("Kontent/Dataset of Decides(0.(5")) adult shome of = police d_csu("Kontent/adult.csv")  |
|       | prent ("messag values ")   |
|       | print Carabetes of enulis suncis   |
| - 1-  | prent ("mensing values")   |
|       | pilat(ddult_sacone-df. (saul O. sung)  |
|       | number-cols d'abetes = d'abetes-dj. select-dégles c'éclide = np.   |
| Test  | number colst adult = adult - Encoure of select-dtyper Chiclude =   |
|       | d'apetes de l'numeric cols diabetes J = diabetes de l'numeric  |
|       | cols_deabetes), funacdiabetes of Councile - cols_diabetes).  |
|       | adult Prione of Fruncisc ids adult I adult Prione of   |
|       | connecte cols adults fulna cadult Parone of Enumerice co   |
| 10.15 | adult J. median ())  |
|       | 11-X-12 Start Starts   |
| 2>    | when reteasured columns all you Edentity in the  |
|       | dataset? How died you encode thom?   |
|       |  |

| Porports pordas as pd.  pan skleam preprocessing impart labellinades.  Lawaret of Diabetern  |
|--|
| Emparts pardas as propert land   |
| from secret of Deabeters   |
| diabetes-dy = pol dead-csv ("/contention   |
| print ("plabetes dataseti")  |
| (and the state of  |
| Cou. P. us. 2. 21, 07 = 23d  |
| 1.1 Ell I im and 'Heron'   |
| dishar to close category J. back the   |
| bin = bin, labels - labels, right = False).  |
| Charles College Colleg |
| deabetes - of = pd get dunnies Cdiabetes - of, columns.  |
| CBM2 - category 1)   |
| Et gender en deabetes-et, columns:   |
| tabel encodes chabelencodiess)   |
| deabetes de l'exercision label encoder les transferses   |
| deapetes of C'gender's)  |
| buent ("In peoples dataset are   |
| sucht (alabetes- af e head()).   |
| Carried an   |
| shat & the delience  |
| short is the defferere between mennax scalling and   |
|  |
| armuld x'= x-xmin x'= x-u  |
| X-max-Xasea  |
| The state of the s |
|  |



```
import pandas as pd
housing_df = pd.read_csv("housing.csv")
print("Information of all columns:")
print(housing_df.info())
print("\nStatistical information of numerical columns:")
print(housing_df.describe())
print("\nCount of unique labels for 'Ocean Proximity' column:")
print("\nColumns with missing values:")
missing_values = housing_df.isnull().sum()
print(missing_values[missing_values > 0])
```

```
import pandas as pd
import numpy as np
diabetes_df = pd.read_csv("/content/Dataset of Diabetes (1).csv")
adult_income_df = pd.read_csv("/content/adult.csv")
print("Missing values in Diabetes dataset:")
print(diabetes_df.isnull().sum())
print("\nMissing values in Adult Income dataset:")
print(adult_income_df.isnull().sum())
numeric_cols_diabetes = diabetes_df.select_dtypes(include=np.number).columns
numeric_cols_adult = adult_income_df.select_dtypes(include=np.number).columns
diabetes_df[numeric_cols_diabetes]
diabetes_df[numeric_cols_diabetes].fillna(diabetes_df[numeric_cols_diabetes].median())
adult_income_df[numeric_cols_adult] = adult_income_df[numeric_cols_adult].fillna(adult_income_df[numeric_cols_adult].median())
```

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
diabetes_df = pd.read_csv("/content/Dataset of Diabetes (1).csv")
print("Diabetes dataset (first few rows):")
```

```
print(diabetes_df.head())
bins = [0, 18.5, 24.9, 40]
labels = ['Low', 'Normal', 'High']
diabetes_df['BMI_Category'] = pd.cut(diabetes_df['BMI'], bins=bins, labels=labels, right=False)
diabetes_df = pd.get_dummies(diabetes_df, columns=['BMI_Category'])
if 'Gender' in diabetes_df.columns:
    label_encoder = LabelEncoder()
    diabetes_df['Gender'] = label_encoder.fit_transform(diabetes_df['Gender'])
print("\nDiabetes dataset after encoding:")
print(diabetes_df.head())
```

```
Diabetes dataset (first few rows):
    ID
        No Pation Gender
                            AGE
                                  Urea
                                         Cr
                                             HbA1c
                                                     Cho1
                                                             TG
                                                                 HDL
                                                                       LDL
                                                                             VLDL
                             50
   502
                                                                              0.5
0
             17975
                         F
                                   4.7
                                         46
                                               4.9
                                                      4.2
                                                            0.9
                                                                  2.4
                                                                       1.4
   735
             34221
                         м
                              26
                                   4.5
                                         62
                                               4.9
                                                      3.7
                                                            1.4
                                                                  1.1
                                                                       2.1
                                                                              0.6
                                                      4.2
   420
             47975
                         F
                              50
                                   4.7
                                         46
                                               4.9
                                                            0.9
                                                                  2.4
                                                                       1.4
                                                                              0.5
   680
             87656
                              50
                                   4.7
                                         46
                                               4.9
                                                      4.2
                                                            0.9
                                                                  2.4
                                                                       1.4
                                                                              0.5
4
   504
             34223
                         м
                                   7.1
                                         46
                                               4.9
                                                      4.9
                                                                 0.8
                                                                       2.0
                                                                              0.4
                              33
                                                            1.0
    BMI CLASS
0
   24.0
             N
   23.0
1
             N
  24.0
             N
3
   24.0
             N
   21.0
4
             N
Diabetes dataset after encoding:
        No_Pation Gender
    TD
                              AGE
                                   Urea
                                          Cr
                                              HbA1c
                                                      Cho1
                                                              TG
                                                                  HDI
                                                                        LDL
                                                                              VI DI
0
   502
             17975
                          0
                               50
                                    4.7
                                          46
                                                4.9
                                                       4.2
                                                             0.9
                                                                   2.4
                                                                        1.4
                                                                               0.5
1
   735
             34221
                               26
                                    4.5
                                          62
                                                 4.9
                                                       3.7
                                                             1.4
                                                                   1.1
                                                                        2.1
                                                                               0.6
2 420
             47975
                          ø
                               50
                                    4.7
                                          46
                                                 4.9
                                                       4.2
                                                             0.9
                                                                   2.4
                                                                        1.4
                                                                               0.5
                          0
                                    4.7
                                                       4.2
                                                                        1.4
   680
             87656
                               50
                                          46
                                                 4.9
                                                             0.9
                                                                   2.4
                                                                               0.5
   504
             34223
                                          46
                                                 4.9
                                                       4.9
                                                             1.0
                                                                  0.8
                                                                        2.0
                                                                               0.4
    BMI CLASS
                BMI_Category_Low
                                    BMI_Category_Normal
                                                            BMI_Category_High
0
                            False
                                                                         False
   24.0
             N
                                                     True
   23.0
             N
                            False
                                                     True
                                                                         False
2
  24.0
                            False
                                                                         False
             N
                                                     True
   24.0
                             False
                                                     True
                                                                         False
4
   21.0
             N
                             False
                                                     True
                                                                         False
```

```
Information of all columns:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
# Column
                                     Non-Null Count Dtype
    Avg. Area Income
                                     5000 non-null
                                                      float64
     Avg. Area House Age
                                     5000 non-null
                                                      float64
     Avg. Area Number of Rooms 5000 non-null
Avg. Area Number of Bedrooms 5000 non-null
                                                      float64
                                                      float64
     Area Population
                                     5000 non-null
                                                      float64
     Price
                                     5000 non-null
                                                      float64
   Address
                                     5000 non-null
                                                      object
 6
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
Statistical information of numerical columns:
       Avg. Area Income \, Avg. Area House Age \, Avg. Area Number of Rooms \, \, \
count
            5000.000000
                                   5000.000000
                                                                5000.000000
mean
           68583.108984
                                      5.977222
                                                                   6.987792
           10657.991214
                                      0.991456
                                                                   1.005833
std
           17796.631190
                                      2.644304
                                                                    3.236194
min
                                                                   6.299250
25%
           61480.562390
                                      5.322283
           68804.286405
                                      5.970429
                                                                   7.002902
50%
75%
           75783.338665
                                      6.650808
                                                                    7.665871
max
          107701.748400
                                      9.519088
                                                                   10.759588
       Avg. Area Number of Bedrooms Area Population
                                                                 Price
                                           5000.000000 5.000000e+03
count
                         5000.000000
                                           36163.516039 1.232073e+06
9925.650114 3.531176e+05
172.610686 1.593866e+04
mean
                            3.981330
std
                             1.234137
                             2.000000
min
                             3.140000
                                           29403.928700 9.975771e+05
25%
                             4.050000
50%
                                           36199.406690
                                                         1.232669e+06
75%
                             4.490000
                                           42861.290770 1.471210e+06
                             6.500000
                                           69621.713380 2.469066e+06
max
Count of unique labels for 'Ocean Proximity' column:
Columns with missing values:
Series([], dtype: int64)
```

| Missing  | values | in | Diabetes | dataset: |
|----------|--------|----|----------|----------|
| ID       | 0      |    |          |          |
| No_Patio | on 0   |    |          |          |
| Gender   | 0      |    |          |          |
| AGE      | 0      |    |          |          |
| Urea     | 0      |    |          |          |
| Cr       | 0      |    |          |          |
| HbA1c    | 0      |    |          |          |
| Chol     | 0      |    |          |          |
| TG       | 0      |    |          |          |
| HDL      | 0      |    |          |          |
| LDL      | 0      |    |          |          |
| VLDL     | 0      |    |          |          |
| BMI      | 0      |    |          |          |
| CLASS    | 0      |    |          |          |
| dtype: : | int64  |    |          |          |

Program 3
Implement Linear and Multi-Linear Regression algorithm using appropriate dataset

| Lab-4.                                | DATE                                    | PAGE  |
|---------------------------------------|---|-------|
| 3                                     | and Mark                                |       |
| Linear regression                     | many nge                                | 4     |
| - Tamen                               |   |       |
| Empart parders as pd                  | and a second                            |       |
| Compart numby as prap                 | The same                                |       |
| Emport matplotice, pyplot as put      | 411111111111111111111111111111111111111 |       |
| Emport matplottib axes as ax          | 7 15 11 11 17 13 P                      |       |
| from matplotteb ansmite               | Lastonitus                              |       |
| from matplottibe animation empart     | Funch remaison.                         |       |
| 2020 3220 2010 dea geekspergeers oug  | 1                                       | 4.1   |
| 30300330MASIE (daya ton for con       | trop-rouges (repro                      | dsi   |
| data = pd_rad_csvcist.                | out )                                   |       |
| data-men                              | 100000000000000000000000000000000000000 | S.F.  |
| data = data diopraci                  | un.                                     | 3 911 |
| train enput = np. anay(data, x (o; so |   | 1)    |
| + win-output - np away (data YEO : 50 |   |       |
| test Paper = np. away (data x 1500:   |   |       |
| test output = np. awaylana. Y [ 500   |   |       |
|                                       | dearway place                           |       |
| clas theoregressions                  |   |       |
| def 8086 1804):                       |   |       |
| self, parameterson = 8 }              |   |       |
| del primard propogation (self, tro    |   |       |
| m = self parameters P'm'3             |   |       |
| c- self parameters ('c')              |   |       |
| pudections = np. multiply             |   | 2+6   |
| setus predections.                    |   |       |
| del cost function self, predict       | eary, tuck out                          |       |
| cost sopmean (church-ou               | tout-brodecsen                          | 18416 |
| nehun cost                            | 1                                       | 7.    |
| 50.0038 (48)                          |   |       |

del bareward propagation well, hear laput, train output, predictions): due de l'outer = & 3 of - Chredictions-train-output). don's & " op meaning multiply (train expet, dp) dc = 2 "hp. mean(at). decloate Pidm'3:dm development'de 13 = de achen decloates. des uplate parameters (self, derevales, learning sate): self-parameters C'm's = self-parameters Cm' ) - learnengsate \* der contestion'3 delf-parameters ('c') = pelf. parameters ('c') - learning rate \* derevates ['de'] des train creep, train-expet, train-output, learning-rate Eteus: self parameters ['m'] = repression unsform (0,1) "-1 self. parameters ('c'3 = np. sandom. uniform(0,1)\*-1 Elanol- pad () arogens . + by = xo, pos x-vals-np. Maspacelmentials. Exputs, max (train- Exput) 100) the car plot (x-vals; self parameters ('m'3 = x-vals + self, parameters (1017, and colons red!, labels · Requession (Pre'). ax, matter chala laput, trais-autput, marker'd' colousiques", label = "Tudinling date"). ar set year o, max (territa output) +1)

| -  | act update (home):   |
|----|--|
|    | predictions self poward propagation (training ut)  |
|    | east = Dely cost - processor predection, train-outputs   |
|    | dentoalsares = sey, backward propagations  |
|    | traco coput, traco output, prodections   |
|    | self, update parameter (derevates, leaviling rate).  |
|    | the set-ydata (self, parantete is mi) = x bals +.  |
|    | sell baraneters (1013)   |
|    | self low appendicants.   |
|    | print ("steration - 53, loss & ) itomat (brane +1, 10x+1).   |
|    | return the   |
| -  | on's Eundt Purcanimation ( p. update, prancos Thus, entaval  |
| -  | 200, b(8t = tupe)  |
| -  | one same (lenear reguesson a git, writers thurbed);  |
|    | bit. Makel ('Imput')   |
|    | pt . Ylabell'output')  |
|    | oit. Litte 1' Lenear Regression')  |
|    | nit leneaded   |
|    | with Manch to the state of the  |
| 7  | schus self paranters eself low.  |
| 1  | ACREA DEST. PROPERTY OF THE PR |
| -  | (1200)   |
| 20 | in the word rog. that there is   |
|    |  |
|    | Contract of the second  |

multiple linear requestion. impost numby as ap Emport porter as pd inport matplottibipypiot as pit data= {"Featud": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10), "Feature2": 52, 3,5, 2,11,13, 17, 19, 23, 297; " Feature 3" 13, 6, 9, 12, 15, 18, 21, 24, 27, 307, "Tanget": [8, 9, 15, 20, 31, 111, 53, 66, 80, 96]} dj = pd. Data Francidata) x=df.diop(column=1"Target").values. y-dy ["Target ], values, reshape (-1, 1). X=np. hetacis ((np. ones ((x. shape(03,1)), x)). beta enpelinely, solve (x. Tax + 0. 010 np. Edentry (x. 1) X.TQY) 4- pied = X@ beta. nue = ap-mean (14 - spmean(4-pied) + 2) total-vaulance= np sum ((y-np, mean(u)) =12) explained-vancance = np. sum ((y pred - np. mean(y)) × 2) +0.25 Explained - vancance 1 total vancance point ("model coexecutor" betach: ) - patter()) putral ("Interepris", betgroscos) putal = mean squared Estationse). partie R-squared : 122) pet. scatterly, y-pred, tolons "blue") pl+, plat (y, y, color='red', leaestyle=' 1) pet . Klabel ["Actual values") pet ylabel ("Preached values"). plt. + 8+66 ( "Actual as predected value ") pla . Many

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.axes as ax
from matplotlib.animation import FuncAnimation
```

```
url = 'https://media.geeksforgeeks.org/wp-content/uploads/20240320114716/data_for_lr.csv'
data = pd.read_csv(url)
data
data = data.dropna()
train_input = np.array(data.x[0:500]).reshape(500, 1)
train_output = np.array(data.y[0:500]).reshape(500, 1)
test_input = np.array(data.x[500:700]).reshape(199, 1)
test_output = np.array(data.y[500:700]).reshape(199, 1)
```

```
class LinearRegression:
  def __init__(self):
     self.parameters = {}
  def forward_propagation(self, train_input):
     m = self.parameters['m']
     c = self.parameters['c']
     predictions = np.multiply(m, train_input) + c
     return predictions
  def cost_function(self, predictions, train_output):
     cost = np.mean((train_output - predictions) ** 2)
     return cost
  def backward_propagation(self, train_input, train_output, predictions):
     derivatives = {}
     df = (predictions-train_output)
     dm = 2 * np.mean(np.multiply(train_input, df))
     dc = 2 * np.mean(df)
     derivatives['dm'] = dm
     derivatives['dc'] = dc
     return derivatives
  def update_parameters(self, derivatives, learning_rate):
     self.parameters['m'] = self.parameters['m'] - learning_rate * derivatives['dm']
     self.parameters['c'] = self.parameters['c'] - learning_rate * derivatives['dc']
  def train(self, train_input, train_output, learning_rate, iters):
     self.parameters['m'] = np.random.uniform(0, 1) * -1
     self.parameters['c'] = np.random.uniform(0, 1) * -1
     self.loss = []
     fig, ax = plt.subplots()
```

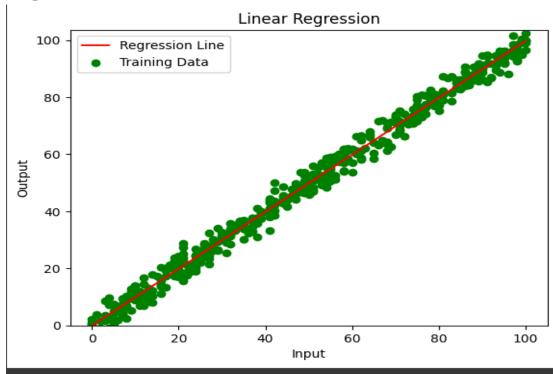
```
x_vals = np.linspace(min(train_input), max(train_input), 100)
line, = ax.plot(x_vals, self.parameters['m'] * x_vals +
          self.parameters['c'], color='red', label='Regression Line')
ax.scatter(train_input, train_output, marker='o',
     color='green', label='Training Data')
ax.set_ylim(0, max(train_output) + 1)
def update(frame):
  predictions = self.forward_propagation(train_input)
  cost = self.cost_function(predictions, train_output)
  derivatives = self.backward_propagation(
     train_input, train_output, predictions)
  self.update_parameters(derivatives, learning_rate)
  line.set_ydata(self.parameters['m']
          * x_vals + self.parameters['c'])
  self.loss.append(cost)
  print("Iteration = { }, Loss = { }".format(frame + 1, cost))
  return line.
ani = FuncAnimation(fig, update, frames=iters, interval=200, blit=True)
ani.save('linear_regression_A.gif', writer='ffmpeg')
plt.xlabel('Input')
plt.ylabel('Output')
plt.title('Linear Regression')
plt.legend()
plt.show()
return self.parameters, self.loss
```

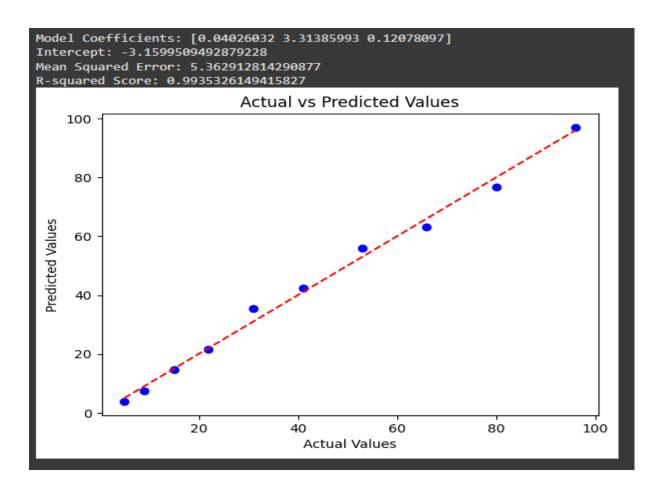
```
linear_reg = LinearRegression()
parameters, loss = linear_reg.train(train_input, train_output, 0.0001, 20)
```

#### Multi Linear Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = {
    "Feature1": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    "Feature2": [2, 3, 5, 7, 11, 13, 17, 19, 23, 29],
    "Feature3": [3, 6, 9, 12, 15, 18, 21, 24, 27, 30],
    "Target": [5, 9, 15, 22, 31, 41, 53, 66, 80, 96]
}
df = pd.DataFrame(data)
X = df.drop(columns=["Target"]).values
y = df["Target"].values.reshape(-1, 1)
```

```
X = np.hstack((np.ones((X.shape[0], 1)), X))
beta = np.linalg.solve(X.T @ X + 0.01 * np.identity(X.shape[1]), X.T @ y)
y_pred = X @ beta
mse = np.mean((y - y_pred) ** 2)
total\_variance = np.sum((y - np.mean(y)) ** 2)
explained_variance = np.sum((y_pred - np.mean(y)) ** 2)
r2 = explained_variance / total_variance
print("Model Coefficients:", beta[1:].flatten())
print("Intercept:", beta[0][0])
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)
plt.scatter(y, y_pred, color='blue')
plt.plot(y, y, color='red', linestyle='--')
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Actual vs Predicted Values")
plt.show()
```





### Program 4

Build Logistic Regression Model for a given dataset

|                                 | DATE  | DACE.  |
|---------------------------------|---|--|
| Logiske Realists                | DATE  | PAGE:  |
| Logroke Regiensson              | oft opposite  |  |
|                                 |   |  |
| Perposit marphotish will        | and the state of the  | 114  |
| The District Colored            |   |  |
| 1000                            | Standard Committee of the   | COLUMN TO THE CO |
| del compute contro              | commentations.  | 119  |
| m=lency)                        | alson my fred the   | 44   |
|                                 | ( Droppet ,   | 09   |
| h = segmord(x@theta)            | 1919 000  | 119  |
| cast=(-1/m) * np. sum(y* s      | p.log(h) +(1-4)   | np.log(1-h)  |
| del graders dexert(x, y, there, |   |  |
| m=len(y)                        | alpha, Herakow)   |  |
| cast hestory = (2               |   |  |
| par en rangellererations        | e   |  |
| gradent = (1/m) = x.T@1         |   | 0 d = V l  |
| theta - = alpha & grade         |   |  |
| cost history appead (com        |   | ((0))  |
| ashun theta, cost washing       |   |  |
| det predect (x itneta):         | account of the  | BARREL S   |
| return (segmord Lx @ the to     | 0>=0.5), astype   | (629   |
| np. wandom. scedius)            | Mary The State of | ALC: UNIVERSE  |
| X= np. sandom. sand(100,1)      | * 10  | 15-12-   |
| y= (x>5) = axtype (5/4)         | Colon   |  |
| x b > np. c. Enp. ones(CX. An   | cx,((),(0)24)   | and mich   |
| thera = np. Zenos (x-b. shapes  |   |  |
| alphazo.i                       | Marie Contract  |  |
| The waters = 1000               |   |  |
| theta, cost history = gradien   | + descert (x-b, y   | theta,   |
| alpha, ? resal?                 |   |  |
| y-pred= predect (x-p, shets     | ()  |  |

```
accuracy * "p. mear (y. pred = = y)

prend (f "Accuracy * & accuracy $ . 2 f 3")

plt . xaster(x 1 4 . color = 'blue', (abel = 'Acetraal data)).

plt . xaster(x 1 4 . pred , color = 'ded); marker = 'x'

label = 'Fredeched label")

plt . xlabel ("Feahrex")

plt . y label ("Claus (a a 1)")

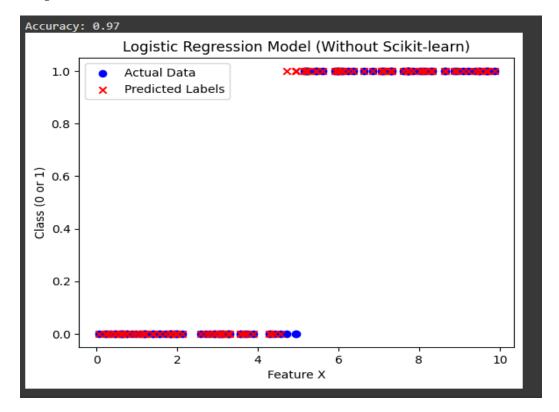
plt . legerd()

plt . title (" 1 R")

plt . show()
```

```
import numpy as np
import matplotlib.pyplot as plt
def sigmoid(z):
  return 1/(1 + np.exp(-z))
def compute\_cost(X, y, theta):
  m = len(y)
  h = sigmoid(X @ theta)
  cost = (-1/m) * np.sum(y * np.log(h) + (1 - y) * np.log(1 - h))
  return cost
def gradient_descent(X, y, theta, alpha, iterations):
  m = len(y)
  cost_history = []
  for _ in range(iterations):
     gradient = (1/m) * X.T @ (sigmoid(X @ theta) - y)
     theta -= alpha * gradient
     cost_history.append(compute_cost(X, y, theta))
  return theta, cost_history
def predict(X, theta):
  return (sigmoid(X @ theta) \geq 0.5).astype(int)
np.random.seed(42)
X = np.random.rand(100, 1) * 10
```

```
y = (X > 5).astype(int).ravel()
X_b = np.c_{np.ones}((X.shape[0], 1)), X
theta = np.zeros(X_b.shape[1])
alpha = 0.1
iterations = 1000
theta, cost_history = gradient_descent(X_b, y, theta, alpha, iterations)
y_pred = predict(X_b, theta)
accuracy = np.mean(y_pred == y)
print(f"Accuracy: {accuracy:.2f}")
plt.scatter(X, y, color='blue', label='Actual Data')
plt.scatter(X, y_pred, color='red', marker='x', label='Predicted Labels')
plt.xlabel("Feature X")
plt.ylabel("Class (0 or 1)")
plt.legend()
plt.title("Logistic Regression Model (Without Scikit-learn)")
plt.show()
```



#### Program 5

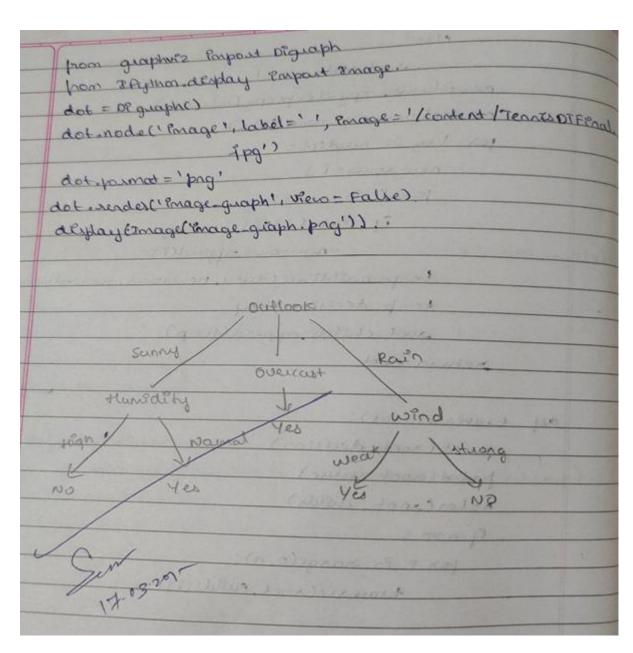
Use an appropriate data set for building the decision tree (ID3) and apply this knowledge to classify a new sample

```
Jab-3.
gasa yamin traga?
Empart modpottib pylotpyplot as plt
Payout pandas as pd.
Papart math
Proport copy.
totaset = pol. sead_csvc /content /Tenors-csv').
X = dataset . floc [ :: ] . values .
attributes = [ 'outlook', 'Temp', 'tunidity', 'wind']
class node Cobject):
     def - salt (self):
      self value = None
     self. decision = None
       self, child = None
det tendent repyldata, nous):
     40=0
     20=0
     Pdr=les(dataCo))-1
     entrophy = 0
     for " or nous "
        Py data (1) Crass - Yes':
              Yes=Yes+1.
         else :
```

```
no=no+1.
      x= yes/ayes+no)
      y=nol(eyes+no)
      ry x1=0 and y1=0:
        est-copy = -1" (x" math lag (x)+ y" math lage (4))
       1=40
      nehin entropy, and
del pendmaxejaen(data, nous, bluma):
    maxepin=0
    2 stidx = -1
     entropy, and Find Entropy (data, sould)
     2) entropy == 0"
      setus maxeals, rettly, and
     you of in columns:
       mydict= {3
        Par= j
        for t to nows ?
          Legis datacissidas
         · if Isey not en mydect:
           nujdect- (teg)=1.
             mydlettkeyJ: mydlettbeyJ+1.
         gain : entropy
          you key ea my dect:
           40:0
            par k in nous:
```

```
() data[x][] == Key:
               Ex duta (1031-13== "Vesto
               yes= yes+1
              classic
        1 +61 = 0
        x = yes/(yestno)
        y= no/(yestno)
        1) x1=0 and 41=00 1;
          gara += (mydect ckey) * (x = mathologzes)
           y math logs(4)))/14.
    1) gain > max eyah ?
       maxejala-gala
         retedr=i
  setus mangain, netide, as.
des buildiseccata, sous, column):
    max yalo, Pdx, and = podmax yalo (x, sous, column)
    moot = Nodec
    E7 = ablanton
    By margain = =0:
    ey ons == 1:
         2001 avalues Yes'
       " 400t svalues 'NO!
    took needed
   2004 value = attilbute [ Ed x ]
   nugdict = & 3
   pos ? la nones?
      bey=datacescedx).
      es key not en mydert:
       mydlet (kg) - 1.
```

| newcolumns = copy deepropy (columns)  newcolumns & semoise (columns)  par tey in mydict:  par t in sours:  if data (c) I columns:  newsours, append(t)  temps build Tree (data, newsours, newcolumn  temp, de cision = key  soot, childs, append (temp)  seturn soot   | else -                           | DATE            | PAGE    |
|--|----------------------------------|-----------------|---------|
| newcolumns = repy-deepcopy (columns)  newcolumns = remove (18x)  par bey in myditet:  newrows = (3)  par i in rows;  if data (12tida) = sitey:  temp= build = sec (data, newrows, newsolumn  temp, decision = toy  return root  bet traverse(root):  puint (root, deesision)  print (root, deesision)  print (root, walue)  n= len (root, cuilds)  if n ro:  traverse(root, cuilds)  columns = (i par i in range(orus))  root = build   recky, rows, rolumns)  root = build   recky, rows, rolumns)  root + deesision = 'start'  |                                  | - Carpera       |         |
| por bey in mydet:  por t in source  if data ( ) I the source  rewrous, append ( )  temps build tree ( data, newrous, newedom  temp, declision = temp  soct. childs, append ( temp)  seture soct  pulat ( soot, deels on)  prent ( soot, deels on)  prent ( soot, deels on)  tox e in sangelo, n):  traverse( soot):  for e in sangelo, n):  traverse( soot, childs)  ef noo:  traverse( soot, childs)  columns = to par 1 in sangelo ( w)  soot = build tree ( y, sours, tolumns)  soot = build tree ( y, sours, tolumns)  soot = build tree ( y, sours, tolumns)  | Noot values += 1.                | the same of the |         |
| pas key in mydret:  newsous=()  par i in sous;  if data(:) Itida)= sitey;  newsous, append(i)  temp= buildTree(data, newsous, newsolum  temp, decision = key  soct. (hilds, append(temp)  seturn soct  punt (soot, deelsion)  punt (soot, deelsion)  punt (soot, deelsion)  for i in songe(o,n);  transpection, (hilds)  (i n >0:  transpection, (hilds)  (column=1: par i in sange(o,u))  column=1: par i in sange(o,u)  soot=buildTree(x, sous, columns)  soot-decision='stari'  | newcommoscopy-deepcopy (roly     | Carrie          | 114     |
| newsous=()  for " " sows"  if data(?) teads of they:  newsous append(")  temp= buildTree (data, newsous, newsolum  temp, decision = key  soot, (hilds, append (temp)  return soot  pulat (soot, deels on)  print (soot, deels on)  print (soot, value)  n=len (soot, value)  for " in sangelo, n);  traverse(soot, childs of the sous elsoot, childs of the soot elsoot, childs of the soot elsoot, childs of the sous elso the sous elsoot, childs of the sous elso the sous elso sous elso the sous elso elso sous elso sous elso sous elso elso elso elso elso elso elso els   | newcolumns e semone (POX)        | and the same    | 100     |
| The sours of the sours of the source of the  | los key so mydects               |                 |         |
| The sours of the sours of the source of the  | newsour = (3                     | Valence of      | 218     |
| rewrous, append(1)  temps build True (data, newrows, newrolum  temp, decision = key  soot, childs, append (temp)  return root  bet traverse (root):  pull (soot, decision)  paint (soot, value)  A=len (soot, childs)  (f n >0:  traverse (root, childs))  (e) calculate():  rows = (i for songel one)  column = (i for songel one)  root = build True (x, rouns prolumns)  root = decision = 'start'   | you ? Po souls?                  | 50 3 10 Barrer  | water   |
| rewsous, append(1)  temps buildTree(data, newsous, newsous,  temp, decision = key  soct.childs, append(temp)  neturn noct  bet traverse(root):  pulat (root, decision)  postat (root, value)  n=ten(root, value)  for 2 in sange(e,n):  traverse(root, childs(1))  def calculate():  sous=1: for 1 in range(onu)  column=1: par 1 in range(onu)  root=buildTree(x, rows, rolumns)  root=decision="Start"   | is : Cross C: Data by 17         | lien:           | 1110    |
| temp buildTree(data, newrows, newrolumo  temp, dectilon = kay  soot, childs, append (temp)  setur soot  pulat (soot, deelsoon)  pulat (soot, deelsoon)  pulat (soot, value)  n=len(soot, childs)  ef n>o:  taxesse(soot, childs[1])  def calculate():  sous=ri for ! in sange(o, n):  column=f ! for ! in sange(o, n):  soot=buildTree(x, sous, columns)  soot=deelson='Start'   | newsousapper                     | der             |         |
| temp, decision = key  soot, childs, append (temp)  seturn soot  def traverse (soot):  pull (soot, deels on)  pull (soot, deels on)  pull (soot, deels on)  pull (soot, deels on):  ten (soot, childs)  ef n so:  tos e la sangelo, n):  traverse (soot, childs on)  def calculate():  sous = li for l'en sangelo, w)  column = l'e for l'en sangelo, w)  soot = build treely, sous, rolumns)  soot = build treely, sous, rolumns)  | temps buildtier (data, no        | whome, no       | mediaca |
| soot. (Wilds, append (temp)  seturn soot  et traverse (2001):  pulat (2001, deels on)  palat (2001, value)  1-len (2001, value)  (f n 20:  tox ? in sange(0,n):  traverse (2001, (Wilds (1)))  def calculate():  20us = [i fax ? in sange(0, w)]  columns = [i fax ? in sange(0, w)]  20ut = build   2 us sange(0, w)   | temp decripon = key              |                 |         |
| return sort  traverse (soot, deels son)  pront (soot, deels son)  pront (soot, value)  n=ten (soot, chelds)  for p in range(o,n):  traverse (soot, chelds son)  tel calculate():  sous=10 for por p in sange(o, us)  columns=10 for por p in sange(o, us)  soot=buildTrue(x, sous, columns)  soot=decls on="Start"   |                                  | . (4            |         |
| pull (soot, deelsoon)  postal (soot, value)  N=len (soot, cullds)  (f n>0:  fox & in sangelo,n):  traverse(soot, cullds(1))  def calculate():  sows=1: fox & in sangelo,w)  column=1: fox & in sangelo,w)  soot=buildTrue(x, sows, valuemas)  soot=decision="staxt"  |                                  |                 |         |
| pull (soot, deelsoon)  postal (soot, value)  N=len (soot, cullds)  (f n>0:  fox & in sangelo,n):  traverse(soot, cullds(1))  def calculate():  sows=1: fox & in sangelo,w)  column=1: fox & in sangelo,w)  soot=buildTrue(x, sows, valuemas)  soot=decision="staxt"  |                                  |                 |         |
| part (2001, value)  12 (2001, curida)  (f n >0:  10x 1: in sange(0,n):  12 (2010116():  20 (2010 | del traverse(root):              |                 |         |
| part (2001, value)  12 (2001, curida)  (f n >0:  10x 1: in sange(0,n):  12 (2010116():  20 (2010 | pulat ( soot, deels on)          | 1               | L. S.   |
| ef n >0 :  tox 1 : n >0 release():  transmerse(xoot, childs(17))  tel calculate():  >0 ms = 1? fax 1 : n xange(0, ms)  columns = 1 : pax 1 : n xange(0, ms)  xoot = buildTree(x, xomx, rolumns)  xoot = decision = 'start'   | pooled (noot, value)             |                 | men il  |
| tel calculate():  Some = [: fax f for sange(o, rul)]  column = [: fax f for sange(o, rul)]  column = [: fax f for sange(o, rul)]  soot = bulldTree(x, rows, rolumns)  root = decision = 'start'  | n=tentapot, cuelda)              |                 | DE L    |
| tel calculate():  Some = [: fax f for sange(o, rul)]  column = [: fax f for sange(o, rul)]  column = [: fax f for sange(o, rul)]  soot = bulldTree(x, rows, rolumns)  root = decision = 'start'  | el n >0 °                        |                 |         |
| tel calculate()=  Some = 12 for 1 for range(o, rul)  column = 12 for 1 for range(o, rul)  column = 12 for 12 for range(o, rul)  Acot = bulldTreely, rows, rolumns)  Acot = decision = 'start'  |                                  | 200             | No I    |
| tel calculate()=  20 us = [? fax ? Pn sange(0,14)]  column= [? fax ? Pn sange(0,14)]  2001 = bulldTree(x, some, columns)  2001 = decesson = 'start'  |                                  | (4)             |         |
| columns = (? pax ? En sange(o, rus))  sout = bulldTree(x, sours, rolumns)  soot decessor = 'start'   |                                  |                 |         |
| columns = (? pax ? En sange(o, rus))  sout = bulldTree(x, sours, rolumns)  soot decessor = 'start'   | del coloniale():                 |                 |         |
| Look-decision - 'Start'  | Canalague of 9 and 573 and       |                 |         |
| 2006-buildTree(x, 20012, columns)  | W. 23-62 102 1 82 2000 0.W       | ).              |         |
| soot decision - 'start'  | Course to fact the sound to have | Can             |         |
|  |                                  |                 |         |
| £.1006398 (3001)   |                                  |                 |         |
|  | £ 7008798 ( 2001)                |                 |         |
| calculates   |                                  | 100000          |         |



```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import math
import copy
```

```
dataset = pd.read_csv('/content/Tennis.csv')

X = dataset.iloc[:,:].values

X
```

```
class Node(object):

def __init__(self):

self.value = None

self.decision = None

self.child = None
```

```
def findEntropy(data, rows):
  yes=0
  no=0
  ans=-1
  idx=len(data[0])-1
  entropy=0
  for i in rows:
    if data[i][idx]=='Yes':
       yes=yes+1
    else:
       no=no+1
  x=yes/(yes+no)
  y=no/(yes+no)
  if x!=0 and y!=0:
    entropy= -1*(x*math.log2(x)+y*math.log2(y))
  if x==1:
    ans = 1
  if y==1:
    ans = 0
  return entropy, ans
```

```
def findMaxGain(data, rows, columns):
    maxGain = 0
    retidx = -1
    entropy, ans = findEntropy(data, rows)
    if entropy == 0:
        """if ans == 1:
        print("Yes")
    else:
        print("No")"""
    return maxGain, retidx, ans
    for j in columns:
        mydict = {}
        idx = j
        for i in rows:
        key = data[i][idx]
        if key not in mydict:
```

```
mydict[key] = 1
     else:
       mydict[key] = mydict[key] + 1
  gain = entropy
  for key in mydict:
     yes = 0
     no = 0
     for k in rows:
       if data[k][j] == key:
          if data[k][-1] == 'Yes':
            yes = yes + 1
          else:
            no = no + 1
     x = yes/(yes+no)
     y = no/(yes+no)
     if x != 0 and y != 0:
       gain += (mydict[key] * (x*math.log2(x) + y*math.log2(y)))/14
  if gain > maxGain:
     maxGain = gain
     retidx = j
return maxGain, retidx, ans
```

```
def buildTree(data, rows, columns):
  maxGain, idx, ans = findMaxGain(X, rows, columns)
  root = Node()
  root.childs = []
  if maxGain == 0:
    if ans == 1:
       root.value = 'Yes'
    else:
       root.value = 'No'
    return root
  root.value = attribute[idx]
  mydict = \{\}
  for i in rows:
    key = data[i][idx]
    if key not in mydict:
       mydict[key] = 1
    else:
       mydict[key] += 1
  newcolumns = copy.deepcopy(columns)
  newcolumns.remove(idx)
  for key in mydict:
    newrows = \prod
```

```
for i in rows:

if data[i][idx] == key:

newrows.append(i)

temp = buildTree(data, newrows, newcolumns)

temp.decision = key

root.childs.append(temp)

return root
```

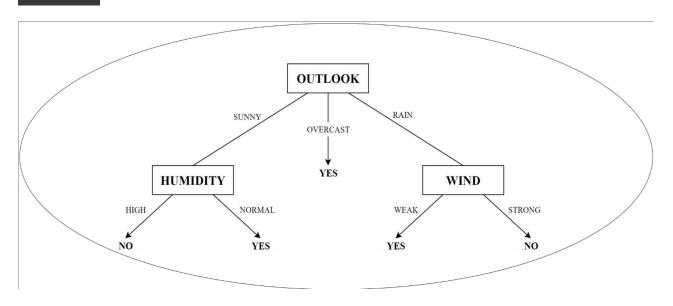
```
def traverse(root):
    print(root.decision)
    print(root.value)
    n = len(root.childs)
    if n > 0:
        for i in range(0, n):
            traverse(root.childs[i])
```

```
def calculate():
    rows = [i for i in range(0, 14)]
    columns = [i for i in range(0, 4)]
    root = buildTree(X, rows, columns)
    root.decision = 'Start'
    traverse(root)
```

#### calculate()

```
from graphviz import Digraph
from IPython.display import Image
dot = Digraph()
dot.node('image', label=", image='/content/TennisDTFinal.jpg')
dot.format = 'png'
dot.render('image_graph', view=False)
display(Image('image_graph.png'))
```

Start Outlook Sunny Humidity High No Normal Yes 0vercast Yes Rain Wind Weak Yes Strong No

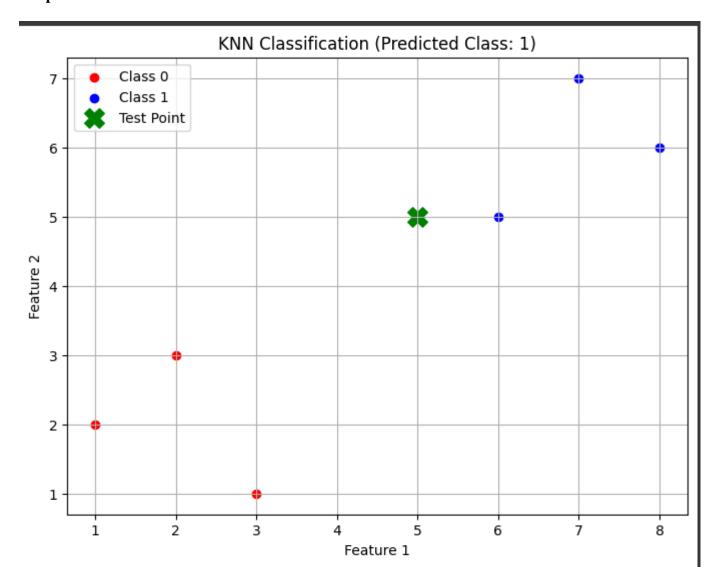


Build KNN Classification model for a given dataset

| -   |  |
|-----|--|
|     | Laber Cara Maria Cara Cara Cara Cara Cara Cara Cara  |
|     | Cat de la constitue con a constitue de la cons |
|     | CC2 1932 yearings resold   |
|     | Empart numby as up   |
|     | impart matplottib piplat as plt.   |
|     | from collection impart courter   |
|     | Control and and controlling  |
|     | def cuclidean-distancely, x2):   |
|     | rehen aposquat Exposur(CXI-76) = "2))  |
|     |  |
|     | class knn:   |
|     | def -9086-(self, k=3);   |
|     | selfile sky i they allo  |
|     | det ket (seep, x, y):  |
| 13  | self. X-train inp. array(X)  |
|     | sey. Y-train-npousagry)  |
|     | del predect (self, x):   |
| 141 | - Sehin Coly-prediction for x en &   |
|     | des = predict(seq', x):  |
|     | distances= [inclideon-distance(x, x-tiam); for   |
|     | Khider on sey x hades  |
|     | b_Praces = np. argsail(distances) (: self: b);   |
|     | to nearest labels : (self , y - Liabelt ) par 1; ?   |
|     | F. indicus   |
|     | most common c Counter (& neareast labels), mast  |
|     | commoncio  |
|     | rehun most common Costos.  |
|     | des some (self, x, y):   |
|     | predections self-bredick(x)  |
|     | setion remeans publications = - 4).  |
|     | Senior Reput   |

15-4-1020 = Np. asray CEE1, 27, 12, 27, 13, 13, (6,5), (5, 7), (8,4)) y times = np. anay (10,0,0,1,1,13) x-test = np. array (((5,5)3). Enn= KNM(k=3) Enn. K+CX-train; y-train) prediction-knn. predict (x-test) ph. 18gure (1892878=(8,6)) you ? in wange (len (x teal n)): plt. xatter Ex-train [?363; x-train [?3613, color = (red' Py y-train to)== o else blue! lable = 1 "clais &y team (?33" Py 1" class Ey-train[7] 9" not in pt.gca().get legent hardles labels () [13 else " ") plt, xatter (x-test 103103, X-test 10311), colon= 'quen', 5=0 marker: 'X', label= 'Test poond!). plt. title (j" Know classification (predicted class spredictions) ("(12 plf. Ylabel ("features") pet. ylabel ("Features") plt. legende) plt, gild (true) plt . Shows

```
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
def euclidean_distance(x1, x2):
  return np.sqrt(np.sum((x1 - x2) ** 2))
class KNN:
  def __init__(self, k=3):
     self.k = k
  def fit(self, X, y):
     self.X_train = np.array(X)
     self.y_train = np.array(y)
  def predict(self, X):
     return [self._predict(x) for x in X]
  def _predict(self, x):
     distances = [euclidean_distance(x, x_train) for x_train in self.X_train]
     k_indices = np.argsort(distances)[:self.k]
     k nearest labels = [self.y train[i] for i in k indices]
     most_common = Counter(k_nearest_labels).most_common(1)
     return most_common[0][0]
  def score(self, X, y):
     predictions = self.predict(X)
     return np.mean(predictions == y)
X_{train} = np.array([[1, 2], [2, 3], [3, 1], [6, 5], [7, 7], [8, 6]])
y_{train} = np.array([0, 0, 0, 1, 1, 1])
X_{\text{test}} = \text{np.array}([[5, 5]])
knn = KNN(k=3)
knn.fit(X_train, y_train)
prediction = knn.predict(X_test)
plt.figure(figsize=(8, 6))
for i in range(len(X_train)):
  plt.scatter(X_train[i][0], X_train[i][1],
          color='red' if y_train[i] == 0 else 'blue',
          label=f"Class {y_train[i]}" if f"Class {y_train[i]}" not in plt.gca().get_legend_handles_labels()[1] else "")
plt.scatter(X_test[0][0], X_test[0][1], color='green', s=200, marker='X', label='Test Point')
plt.title(f"KNN Classification (Predicted Class: {prediction[0]})")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.legend()
plt.grid(True)
plt.show()
```



Build Support vector machine model for a given dataset

| import number                       | DATE:  | PAGE:       |
|-------------------------------------|--|-------------|
| Emport manpy as up                  | September 198  |             |
| impact matplotteb. pyplot as petple | a succes   |             |
| dass sym:                           | Marine !   |             |
| dep last was to                     | Total Control of the   | 45 45       |
| det last bet leaving rate on        | frobdomol, 610i  | aum=0.01,   |
|                                     |  |             |
| self beleauning sale                | 1,   |             |
| self lambla params lambda -1        | baraw.   |             |
| Self n their on their               | 1 10   |             |
| Self-we None                        | 3 0003   |             |
| self p=none                         | 13   | 113         |
| Fridgens Entelposition Has the      |  |             |
| def texcell xins:                   | and a const  |             |
| y= npowhere (y coso, -1, 1)         | STATE OF THE STATE | COLUMN TOWN |
| n-samples, n-jenhues = x. s         |  |             |
| self. winp-zerasin-jeatan           | cas  | - Carlos    |
| self.b=0.                           | B. Brancowski  |             |
| 611 11 3013 E                       | A STATE OF THE STA |             |
| per - Po mangelself. n-Pters        | ):   |             |
| pu Pdx, 7 - 8 Pa enuine             |  |             |
| condition = ylldx)"(                |  |             |
| sely-bl>=1                          |  | 0.00        |
| *L (oughton;                        |  |             |
| self.wo == self                     |  |             |
|                                     | sey-w)   |             |
|                                     | 4 - 1 3 - 1 - 1 - 1 - 1 - 1  |             |
| self: w-i self.                     |  |             |
|                                     | ot(x2, y28dx)  |             |
|                                     |  | ,           |
| self . p += self                    | y Aciaxi   |             |

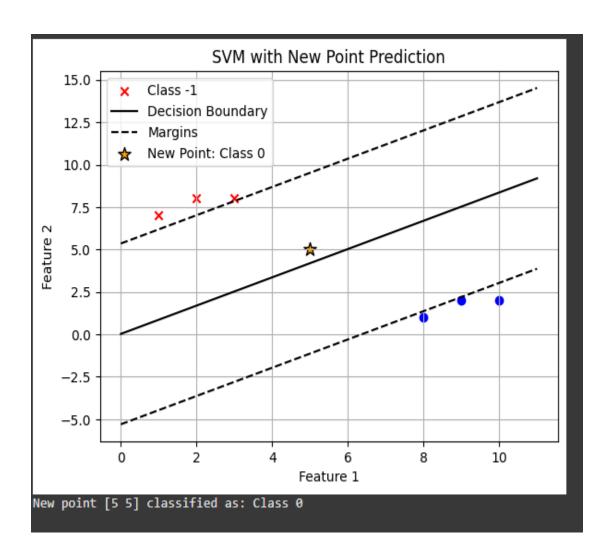
des predictions ( , x): approx = np-dot(x, self - wo) + self - b. retur operagolappiox) dy visualejetseff, x,y, new poent = none, predecteon = none) dop get hyperplane (x, 10, b, 64,80+): Ciscol (teapporate " Cosco-) number kig = plt requiels. ax= 10g.add-subplot(1,1,1) for i, sample in enumerate(x): e1 4 (6)==1: plt. xatter (sample(0), sample(1) markens 101, colors blue, label=class +14 q 9==0 (1) - e (xe " ") else: ple. scatter (sample (0), sample (1), mariencolon='red', label='(lass-1' PL == 20 else 11 11) x0=np.linspace(npmin(x1:0))-1,np.max(x1:0)+1, x1=get-hyperplanelxo, sey. w, secf. b, 0) XI m = get hyperplane (xo, sey, w, sey, b,-1) VI p=get-hyperplane(xo, self-10, self-1) ax. plot(x0, x1, 1k-1, label = 1 opicisson Boundary) on splot (x'o, x1-m, k-- 1, labets 'margins') ox apport (x 0, x1-p, 12-1) of new-point is not None: color= 'green' ? predection == 1 else 'arange'

```
DATE
                                                PAGE
    label - 1 "Now New potraticion 5"1" ?} pudecien = 1
   plt. Natter (new-positio), new-positio, carolar, 5=100,
        ragicalar = black |, tabel - label, markens " ")
ar legendo
plt. Mabel ("Feature")
pit. Ylabel ("Feature 2")
plt take ("com wan new potat predaction")
pit. show()
if -none -- " main -- ":
    ((CE, 00, ES, P), EE, 87, E8, 87, E8, 97, E6, 1) 3) parage x
    y=0p.anay((0,0,0,1,1,1))
new-posit = np. auny (CCS, 577).
SUM = SVMC)
SUM. 18+ (X,4)
prediction = sum. predict (new-point)[0]
sun visualize (X, y, new point enow point (a), prediction:
puntly " were point spew-point (0) & classified as: 5' class 1"
```

```
import numpy as np
import matplotlib.pyplot as plt
class SVM:
    def __init__(self, learning_rate=0.001, lambda_param=0.01, n_iters=1000):
        self.lr = learning_rate
        self.lambda_param = lambda_param
        self.n_iters = n_iters
```

```
self.w = None
  self.b = None
def fit(self, X, y):
  y = np.where(y \le 0, -1, 1)
  n_samples, n_features = X.shape
  self.w = np.zeros(n_features)
  self.b = 0
  for in range(self.n iters):
     for idx, x_i in enumerate(X):
        condition = y[idx] * (np.dot(x_i, self.w) + self.b) >= 1
        if condition:
          self.w -= self.lr * (2 * self.lambda_param * self.w)
        else:
          self.w -= self.lr * (2 * self.lambda_param * self.w - np.dot(x_i, y[idx]))
          self.b += self.lr * y[idx]
def predict(self, X):
  approx = np.dot(X, self.w) + self.b
  return np.sign(approx)
def visualize(self, X, y, new_point=None, prediction=None):
  def get_hyperplane(x, w, b, offset):
     return (-w[0] * x + b + offset) / w[1]
  fig = plt.figure()
  ax = fig.add subplot(1, 1, 1)
  for i, sample in enumerate(X):
     if y[i] == 1:
        plt.scatter(sample[0], sample[1], marker='o', color='blue', label='Class + 1' if i == 0 else "")
     else:
        plt.scatter(sample[0], sample[1], marker='x', color='red', label='Class -1' if i == 0 else "")
  x0 = \text{np.linspace}(\text{np.min}(X[:, 0]) - 1, \text{np.max}(X[:, 0]) + 1, 100)
  x1 = get hyperplane(x0, self.w, self.b, 0)
  x1_m = get_hyperplane(x0, self.w, self.b, -1)
  x1_p = get_hyperplane(x0, self.w, self.b, 1)
  ax.plot(x0, x1, 'k-', label='Decision Boundary')
  ax.plot(x0, x1_m, 'k--', label='Margins')
  ax.plot(x0, x1_p, 'k--')
  if new point is not None:
     color = 'green' if prediction == 1 else 'orange'
     label = f'New Point: Class {"1" if prediction == 1 else "0"}'
     plt.scatter(new_point[0], new_point[1], c=color, s=100, edgecolors='black', label=label, marker='*')
  ax.legend()
  plt.xlabel("Feature 1")
  plt.ylabel("Feature 2")
```

```
plt.title("SVM with New Point Prediction")
     plt.grid(True)
     plt.show()
if __name__ == "__main__":
  X = np.array([
    [1, 7],
    [2, 8],
    [3, 8],
    [8, 1],
    [9, 2],
    [10, 2]
  ])
  y = np.array([0, 0, 0, 1, 1, 1]) # 0 -> -1, 1 -> +1
  new_point = np.array([[5, 5]])
  svm = SVM()
  svm.fit(X, y)
  prediction = svm.predict(new_point)[0]
  svm.visualize(X, y, new_point=new_point[0], prediction=prediction)
  print(f"New point {new_point[0]} classified as: {'Class 1' if prediction == 1 else 'Class 0'}")
```



Implement Random forest ensemble method on a given dataset

| 5 | Labre  |
|---|--|
|   | Random facest ensemble algaershm.  |
|   | The second control of  |
|   | Empart pandas as pd.   |
|   | Empart pandas as pa  |
|   | from sklean ensemble Emport Randomtorest classifier  |
|   | from sklean metales emport accuracy score,   |
|   | at the land television   |
|   |  |
|   | unt = "https://www.gethubusercontext.com/jtrounderl  |
| I | Datasets master bling-indian-diapers actions   |
| I | columns = ["Pregnancies", "cylucose", "BloodPressure",   |
| ĺ | "SwenTheckness", "Ensuler", "BM3", "Deabetes Pedigsee Pune   |
| ı | "Age", "target"].  |
|   | aj = pd. read-carcurly names; columns).  |
|   | print (" pataret Prevers in", dj. heart ()).   |
|   | x = dj.dop("target", axes=1)   |
|   | y=01 (Hayer") (1000)   |
| - | X tueso X lot 11 to 11 to 12 t |
|   | X-train, X-test, y-train, y-test= train test-spire.  |
|   | (x, y, test seze = 0:2, nondom state = uz)   |
|   | af-madel = Random Forest clausiffer (n-estimatous =100,  |
|   | handom state cus)  |
|   | of model oft (x train, y train)  |
| - | g-pred = of -model. predict (x-text):  |
| Ц | per alla Accusació accordo   |
|   | The party has a variable at the contract of th |
| - | (g-test, by-pred)).  |

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = [
  "Pregnancies", "Glucose", "BloodPressure", "SkinThickness",
  "Insulin", "BMI", "DiabetesPedigreeFunction", "Age", "target"
df = pd.read_csv(url, names=columns)
print("Dataset Preview:\n", df.head())
X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)
print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

| Dataset Preview:    |                  |             |              |         |         |      |   |
|---------------------|------------------|-------------|--------------|---------|---------|------|---|
| Pregnancies G       | lucose Blo       | oodPressure | SkinThi      | ckness  | Insulin | BMI  | \ |
| 0 6                 | 148              | 72          |              | 35      | 0       | 33.6 |   |
| 1 1                 | 85               | 66          |              | 29      | 0       | 26.6 |   |
| 2 8                 | 183              | 64          |              | 0       | 0       | 23.3 |   |
| 3 1                 | 89               | 66          |              | 23      | 94      | 28.1 |   |
| 4 0                 | 137              | 40          |              | 35      | 168     | 43.1 |   |
|                     |                  |             |              |         |         |      |   |
| DiabetesPedigree    | <b>eFunction</b> | Age targe   | t            |         |         |      |   |
| 9                   | 0.627            | 50          | 1            |         |         |      |   |
| 1                   | 0.351            | 31          | 3            |         |         |      |   |
| 2                   | 0.672            | 32          | 1            |         |         |      |   |
| 3                   | 0.167            | 21          | Э            |         |         |      |   |
| 4                   | 2.288            | 33          | 1            |         |         |      |   |
|                     |                  |             |              |         |         |      |   |
| Accuracy: 0.7532467 | 7532467533       |             |              |         |         |      |   |
|                     |                  |             |              |         |         |      |   |
| Classification Repo | ort:             |             |              |         |         |      |   |
| pred                | cision           | recall f1-  | score s      | support |         |      |   |
|                     |                  |             |              |         |         |      |   |
| 0                   | 0.82             | 0.80        | 9.81         | 151     |         |      |   |
| 1                   | 0.64             | 0.66        | <b>9.6</b> 5 | 80      |         |      |   |
|                     |                  |             |              |         |         |      |   |
| accuracy            |                  |             | a.75         | 231     |         |      |   |
|                     | 0.73             | 0.73        | <b>3.</b> 73 | 231     |         |      |   |
| weighted avg        | 0.76             | 0.75        | <b>3.7</b> 5 | 231     |         |      |   |

Implement Boosting ensemble method on a given dataset

| 7 | DATE: PAGE:   |
|---|---|
| - | Booshing.   |
|   | Empart pandas as pd.  |
| _ | from sklean, madel selection Emporer transfer spice                 |
|   | from skleaus, ensomble Pompout AdaBoost-clausepess                  |
|   | from sklean, metules emport accuracy score,                         |
|   | Claus pecakon report  |
| 1 | est="https://aav.gethubusercontext.com/jbiownlee/patarets/          |
|   | master/pena-endeans-deabetes data csy"                              |
|   | column: F"Pregnancies", "ejhucuse", "Bloodflessure", "scents        |
|   | new", " Insulan", "BM3:, "Orabetes Pedigues Punction", "age"        |
|   | "Lauget")   |
|   | dj = pd. sead_csv (ux 1, nones = columns).                          |
|   | print ("Dataset Previous In", of shead )                            |
|   | X = dj-drop ('target', ands=1)                                      |
|   | y= of ('taiget')  |
|   | X-train, X-test, y-train, y-test= train-test-splet (x, y,           |
|   | test size = 0.2, sandom-state= 40)                                  |
|   | boost model = AdaBoostclassifier (n-& Rinatoss=100, leauning-10     |
|   | 1.0, dandom states us se  |
|   | boost model - per (x-train, y-train).                               |
|   | ty pied = boost model predict (x text).                             |
|   | but ( Accuracy " accuracy scorely test, y pred))                    |
|   | prent ("classeperation Report: In", classe peration report (y test, |
|   | ( pred)).   |
|   | B. Touch as well a to the broken war. My                            |

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score, classification_report
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = [
  "Pregnancies", "Glucose", "BloodPressure", "SkinThickness",
  "Insulin", "BMI", "DiabetesPedigreeFunction", "Age", "target"
df = pd.read_csv(url, names=columns)
print("Dataset Head:\n", df.head())
X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=40)
boost_model = AdaBoostClassifier(n_estimators=100, learning_rate=1.0, random_state=40)
boost_model.fit(X_train, y_train)
y_pred = boost_model.predict(X_test)
print("\n--- AdaBoost Results ---")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

| Dataset Head:  |              |           |       |               |         |      |   |
|----------------|--------------|-----------|-------|---------------|---------|------|---|
| Pregnancie     | s Glucose    | BloodPres | sure  | SkinThickness | Insulin | BMI  | \ |
| 9 6            | 148          |           | 72    | 35            | 0       | 33.6 |   |
| 1 1            | 85           |           | 66    | 29            | 0       | 26.6 |   |
| 2 8            | 183          |           | 64    | 9             | 0       | 23.3 |   |
| 3 1            | 89           |           | 66    | 23            | 94      | 28.1 |   |
| 4 0            | 137          |           | 40    | 35            | 168     | 43.1 |   |
|                |              |           |       |               |         |      |   |
| DiabetesPed:   | igreeFunctio | n Age t   | arget |               |         |      |   |
| 0              | 0.62         | 7 50      | 1     |               |         |      |   |
| 1              | 0.35         | 1 31      | 0     |               |         |      |   |
| 2              | 0.67         | 2 32      | 1     |               |         |      |   |
| 3              | 0.16         | 7 21      | 0     |               |         |      |   |
| 4              | 2.28         | 8 33      | 1     |               |         |      |   |
|                |              |           |       |               |         |      |   |
| AdaBoost Re    | esults       |           |       |               |         |      |   |
| Accuracy: 0.74 | 391774891774 | 89        |       |               |         |      |   |
| Classification | Report:      |           |       |               |         |      |   |
|                | precision    | recall    | f1-sc | ore support   |         |      |   |
|                |              |           |       |               |         |      |   |
| 9              | 0.75         | 0.88      | 0.    | 81 142        |         |      |   |
| 1              | 0.74         | 0.54      | 0.    | 62 89         |         |      |   |
|                |              |           |       |               |         |      |   |
| accuracy       |              |           | 0.    | 75 231        |         |      |   |
| macro avg      | 0.75         | 0.71      | 0.    | 72 231        |         |      |   |
| weighted avg   | 0.75         | 0.75      | 0.    | 74 231        |         |      |   |

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

| - | Emeans clustering.   |
|---|--|
| 4 | Impart parlas aspol  |
| 4 | Emport matphotos byptot as plt.                                |
| 4 | from externo cluster emport lemeans.                           |
| 4 | from sklearn preprocessing emport standardscales               |
| - | son sklean, decomposition impart PCA.                          |
| 7 | el : "https://www.g?thutsercontent.com/gbroidnlee/ Darasons    |
| 1 | master/pena-indians-deabetes, data, CAV".                      |
| 1 | columns = ["Pregarancees", "ejlucose", " BloodPressure",       |
| 4 | "Skarthickney, traduling, "Braz", "Drabetes pediguer Function" |
| 1 | "Age", "tauget".   |
|   | Af = Pdo sead_cov(us), names = columns);                       |
|   | pullet (" bataset previously", dj head())                      |
|   | X=df.d.ep('target',axes=1).                                    |
|   | Scales Standardscales ()                                       |
|   | ecaled data = scales, fit tuenspain (x)                        |
|   | k=3  |
| 1 | emeans = Kmeansla-clusters=k, uandomstate = u2)                |
| 1 | Knears per (scaled data)                                       |
| 1 | 4) (1) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1                       |
| 1 | prent ("Inclustred pala Previous In", dj. head()).             |
| 1 | pease P(A(n, components=2)                                     |
|   | reduced data = pca 184 1                                       |
| 1 | pt figure ( popsige = ( 5,6))                                  |
|   | plt matterfreduced + 1 = 2                                     |
| - | (2 Knear-labels, Inner ) reduced durals, 13,                   |
|   | plt. Scatter (knows as a comp = 'veridas')                     |
|   | · Clubton c-1  |
|   | (enters. C:,13, 5=300, c='red' q trancer = 1 x',               |

```
pit. 1816 ("Emeans Clustering with Pin")

pit. xlabel ("Principal component")

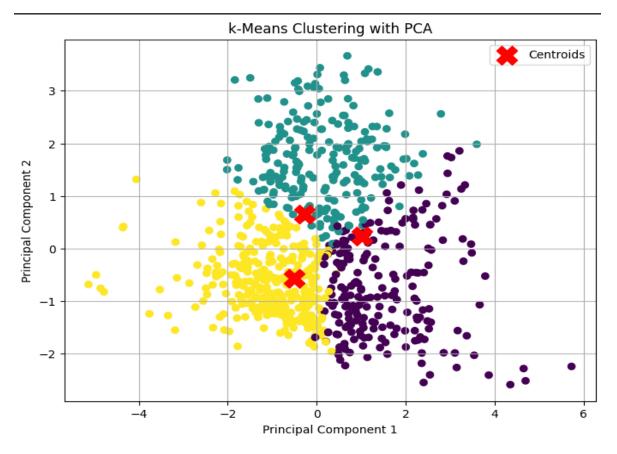
pit. cylabel ("Principal component 2")

pit. logende)

pit. showe)
```

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = [
  "Pregnancies", "Glucose", "BloodPressure", "SkinThickness",
  "Insulin", "BMI", "DiabetesPedigreeFunction", "Age", "target"
df = pd.read_csv(url, names=columns)
print("Dataset Preview:\n", df.head())
X = df.drop('target', axis=1)
scaler = StandardScaler()
scaled_data = scaler.fit_transform(X)
k = 3
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(scaled_data)
df['Cluster'] = kmeans.labels_
print("\nClustered Data Preview:\n", df.head())
pca = PCA(n_components=2)
reduced_data = pca.fit_transform(scaled_data)
plt.figure(figsize=(8, 6))
plt.scatter(reduced_data[:, 0], reduced_data[:, 1], c=kmeans.labels_, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', marker='X',
label='Centroids')
plt.title("k-Means Clustering with PCA")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend()
plt.grid(True)
plt.show()
```

| Da  | taset Preview: |            |         |          |           |         |         |              |          |
|-----|----------------|------------|---------|----------|-----------|---------|---------|--------------|----------|
|     | Pregnancies    | Glucose    | BloodPr | essure   | SkinThick | ness    | Insulin | BMI          | <b>\</b> |
| 0   | 6              | 148        |         | 72       |           | 35      | 9       | 33.6         |          |
| 1   | 1              | 85         |         | 66       |           | 29      | 0       | 26.6         |          |
| 2   | 8              | 183        |         | 64       |           | 0       | 0       | 23.3         |          |
| 3   | 1              | 89         |         | 66       |           | 23      | 94      | 28.1         |          |
| 4   | 0              | 137        |         | 40       |           | 35      | 168     | 43.1         |          |
|     |                |            |         |          |           |         |         |              |          |
|     | DiabetesPedig  | reeFunctio | n Age   | target   |           |         |         |              |          |
| 0   |                | 0.62       | 7 50    | 1        |           |         |         |              |          |
| 1   |                | 0.35       |         | 0        |           |         |         |              |          |
| 2   |                |            | 2 32    | 1        |           |         |         |              |          |
| 3   |                |            | 7 21    | 9        |           |         |         |              |          |
| 4   |                | 2.28       | 8 33    | 1        |           |         |         |              |          |
| -01 |                |            |         |          |           |         |         |              |          |
| CI  | ustered Data P |            | n1 In   |          | Chi Thia  |         | *1      | DUT          |          |
| _   | Pregnancies    |            | RTOOGEL |          | SkinThick |         | Insulin | BMI          | `        |
| 0   | 6<br>1         | 148<br>85  |         | 72<br>66 |           | 35      | 9<br>9  | 33.6<br>26.6 |          |
|     |                |            |         |          |           | 29      |         |              |          |
| 2   | 8<br>1         | 183<br>89  |         | 64<br>66 |           | 0<br>23 | 9<br>94 | 23.3<br>28.1 |          |
| 4   | 9              | 137        |         | 40       |           | 35      | 168     | 43.1         |          |
| 4   | О              | 13/        |         | 40       |           | 35      | 108     | 43.1         |          |
|     | DiabetesPedig  | reeFunctio | n Age   | target   | Cluster   |         |         |              |          |
| 0   | Diabetesreaig  | 0.62       | _       | 1        | 1         |         |         |              |          |
| 1   |                | 0.35       |         | 9        | 2         |         |         |              |          |
| 2   |                | 0.67       |         | 1        | 1         |         |         |              |          |
| 3   |                | 0.16       |         | 9        | 2         |         |         |              |          |
| 4   |                | 2.28       |         | 1        | 9         |         |         |              |          |



Implement Dimensionality reduction using Principal Component Analysis (PCA) method

| u.  | PcA.  |
|-----|---|
|     | Emport pardas as pd.  |
|     | from skleasn. decomposition impact PCA                                |
|     | from steleasn. Preprocessing imposite Standard Scales,                |
|     | Emport matplotist. pythat as plt.                                     |
|     | Unt: "https://www.gsthubuseccontent.com/jbuoconter/patousets/         |
| -   | master/pina- indians-deabetes data csv".                              |
|     | columns: ["Pregnancees", "ejemose", "BloodPressure", "Skentheckne     |
|     | _15', "Ensulen". "PM 3°, "DiabetesPedigue Function", "Age". "+auget"] |
|     | at = pd. sead. (sv (un), names = columns).                            |
|     | puent ("ourgenal Datasto", of head ().                                |
|     | rales standard sales.   |
|     | scaled data - scales, 18t transform (xaled data)                      |
|     | pead = pd. parations (data = pea sexut, column = (Pc1', 'Pc2'))       |
|     | print("10 PLA Reduced Datas In", praid head ().                       |
|     | pr+ fegure (fegripe x(8,6)).  |
|     | pit. scatter (pearly ("Pet), pearly ("C") · alpha=0,7).               |
|     | plt. Xlabel ("Publishal (Omponed))                                    |
|     | pH. Yaber (Paintipal components).                                     |
|     | pit, title (IPIA - Dimonsionally Reduction')                          |
|     |   |
| 100 | DATE:   |
|     |   |
| 7   | pet gred truce  |
|     | August 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1                         |
|     | pit show()  |
|     | Cho vo  |
|     | No. or  |
|     | Carrottera  |

```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = [
  "Pregnancies", "Glucose", "BloodPressure", "SkinThickness",
  "Insulin", "BMI", "DiabetesPedigreeFunction", "Age", "target"
df = pd.read_csv(url, names=columns)
print("Original Data:\n", df.head())
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_data)
pca_df = pd.DataFrame(data=pca_result, columns=['PC1', 'PC2'])
print("\nPCA Reduced Data:\n", pca_df.head())
plt.figure(figsize=(8, 6))
plt.scatter(pca_df['PC1'], pca_df['PC2'], alpha=0.7)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA - Dimensionality Reduction')
plt.grid(True)
plt.show()
```

```
Original Data:
     Pregnancies
                     Glucose BloodPressure SkinThickness Insulin
                                                                                  BMI
                         148
                                                                                33.6
0
1
2
3
4
                          85
                                                                29
                1
                                             66
                                                                            0
                                                                                26.6
                8
                         183
                                             64
                                                                0
                                                                            0 23.3
                          89
                                             66
                                                                           94 28.1
                0
                                             40
                                                                          168 43.1
   DiabetesPedigreeFunction
                                           target
0
                           0.627
1
2
3
4
                           0.351
                                     31
                                                 0
                           0.672
                                     32
                                                 1
                           0.167
                                     21
                                                 0
                           2.288
                                                 1
PCA Reduced Data:
           PC1
  1.756947 1.111743
-1.507421 -0.559406
0.650822 1.929576
-1.587398 -1.065075
   2.483374 -2.359563
```

