**P.R.POTE (PATIL) EDUCATION & WELFARE**

**TRUST’S GROUP OF INSTITUTIONS,**

**COLLEGE OF ENGINEERING & MANAGEMENT,**

**AMRAVATI.**



# Department Name:

**COMPUTER SCIENCE & ENGINEERING**

## Year : III Semester: V

**SUBJECT**: ET-LAB

**Name of Lab :- ET-Lab I**



## Department of Computer Science and Engineering

**Certificate**

This is to certify that Ms.**Vaishnavi Vijayrao Deshmukh** of **V** Semester of Bachelor of Engineering in **Computer Science Engineering** of P.R.Pote (Patil) College of Engineering & Management, Amravati, has completed the term work satisfactory in subject **ET-LAB** for the academic year 2021-2022 as prescribed in the curriculum.

## Place:- Amravati PRN No:……

**Date:- 6/01/2022 Roll No: 367**

**Subject Teacher Head of the Department**

### P. R. Pote (Patil) College of Engineering & Management, Amravati

**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Name of the Program: B. E. CSE** | **Academic Year: 2021-22** |
| **Class: 3rd Year** | **Semester: V** |
| **Section: A** | **Course Code:** |
| **Course/Subject: ET- Lab** | **Course Owner: Prof.T.G.Ghongade** |

|  |  |
| --- | --- |
| **SN** | **List of Practical(s) / Experiment(s)** |
| 1 | Write a program to study Exploratory Data Analysis.  a) Identify the nature of data and observe the various statistical parameters. b)Uploading of dataset in python |
| 2 | Write a program for supervised machine learning. |
| 3 | Write a Program for unsupervised machine learning.  a)Identify Variance, accuracy, bias. |
| 4 | Write a program for simple linear regression.  a)Estimating the accuracy of model and coefficient of estimate. |
| 5 | Write a program KNN. |
| 6 | Write a program for logistic regression. |
| 7 | Write a program for Decision tree. |
| 8 | Write a program for K-means Clustering |

**LIST OF (PRACTICALS / EXPERIMENTS) & PROGRESSIVE ASSESSMENT FOR TERM WORK**

Academic Year : - 2021-2022 Course :- ET-Lab

Subject & Code :- **ET- Lab** Semester:-V Name of Faculty: - Prof. T.G.Ghongade

Name of Student: - Vaishnavi Vijayrao Deshmukh PRN No………… Roll No 367

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SN.** | **Title of the Practical / Experiment** | **Page No** | **Date of Performance** | **Date of Submission** | **Assessment Marks (15)** | **Sign of Teacher and**  **Remarks** |
| **1** | Write a program to study Exploratory Data Analysis.   1. Identify the nature of data and observe the various statistical parameters. 2. Uploading of dataset in python |  |  |  |  |  |
| **2** | Write a program for supervised machine learning. |  |  |  |  |  |
| **3** | Write a Program for unsupervised machine learning.  a)Identify Variance, accuracy,  bias |  |  |  |  |  |
| **4** | Write a program for simple linear regression. a)Estimating the accuracy of model and coefficient of  estimate. |  |  |  |  |  |
| **5** | Write a program KNN. |  |  |  |  |  |
| **6** | Write a program for logistic regression. |  |  |  |  |  |
| **7** | Write a program for Decision tree. |  |  |  |  |  |
| **8** | Write a program for K-means Clustering |  |  |  |  |  |

**Signature of Faculty**

# PRACTICAL NO: 01

### AIM OF PRACTICAL / EXPERIMENT:

Write a program to study Exploratory Data Analysis.

a) Identify the nature of data and observe the various statistical parameters.b) Uploading of dataset in python

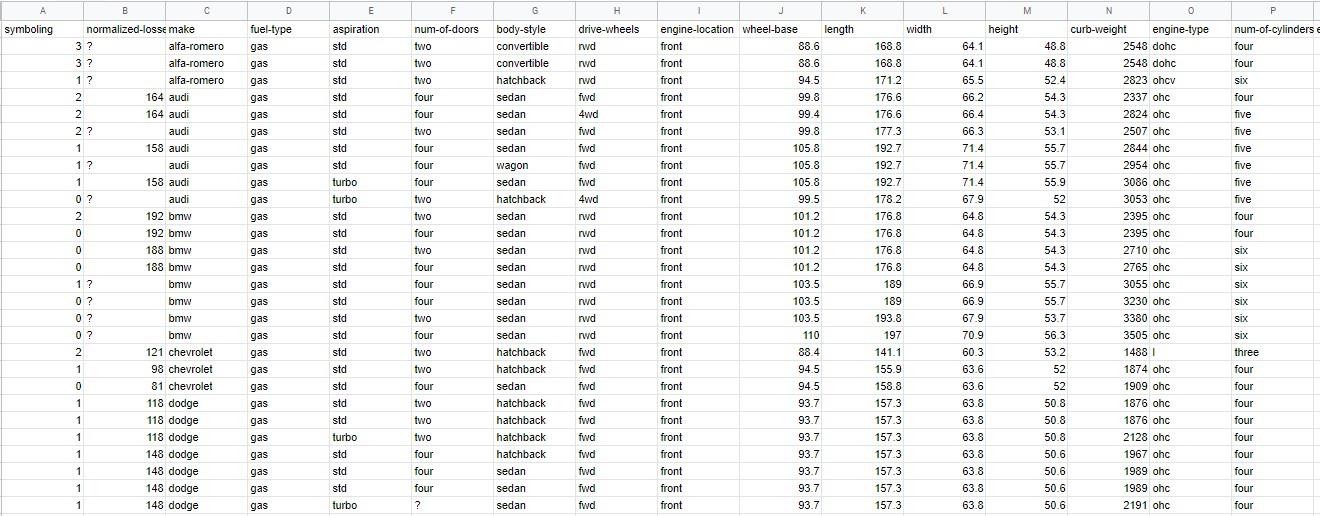
### SOFTWARE REQUIRED / EQUIPMENT DETAILS: -

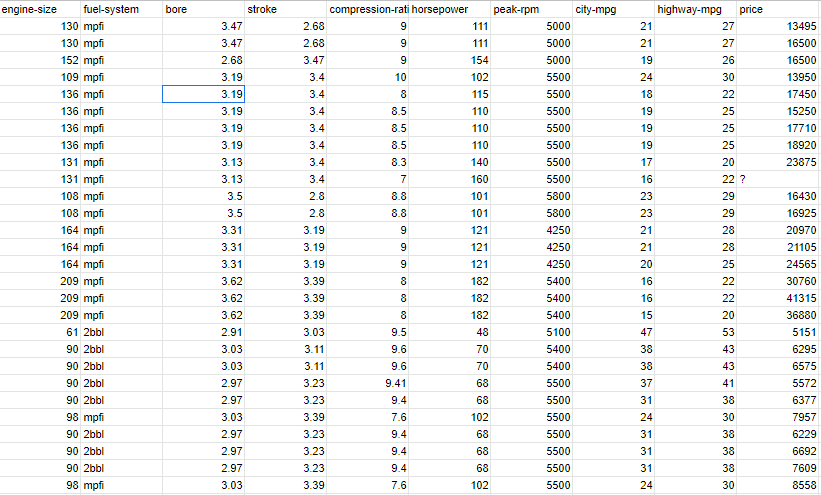
|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | NA |
| **2.** | Jupyter Notebook |  |  |

**THEORY:**

Exploratory Data Analysis, or EDA, is an important step in any Data Analysis or Data Science project. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset. EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better. In this article, we will understand EDA with thehelp of an example dataset. We will use Python language (Pandas library) for this purpose.

### Data Set:-





**Identification of Pattern of Data:**

It is a Continuous type of Data.

### Uploading data in python:

What is a CSV?

CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

For working CSV files in python, there is an inbuilt module called csv.

Python is a great language for doing data analysis, primarily because of the fantastic ecosystem of

data-centric python packages. Pandas is one of those packages and makes importing and analyzing data much easier.

Import Pandas

read\_csv is an important pandas function to read csv files and do operations on it. read\_csv is an important pandas function to read csv files and do operations on it.

### PROGRAM:

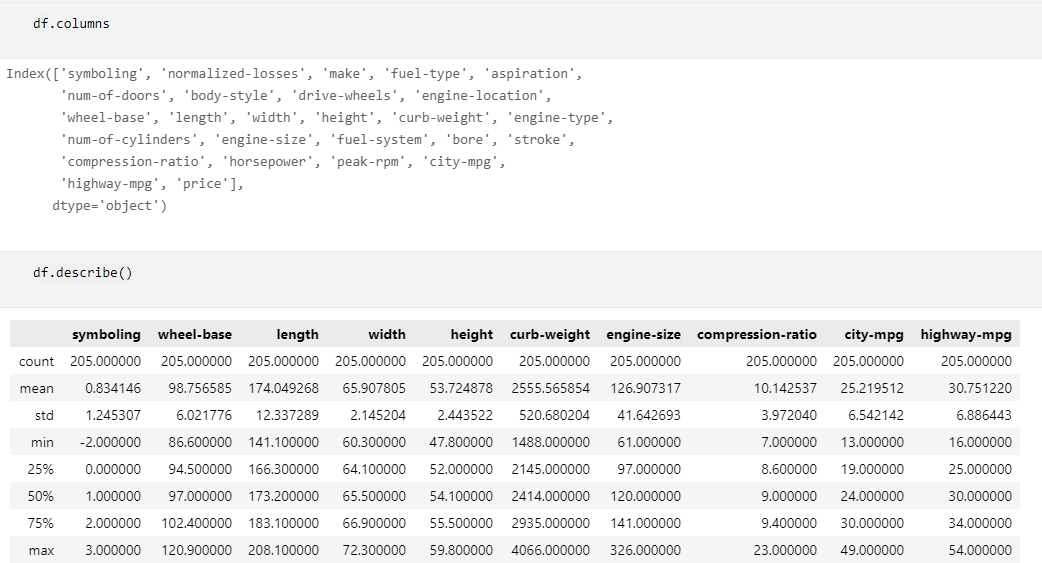
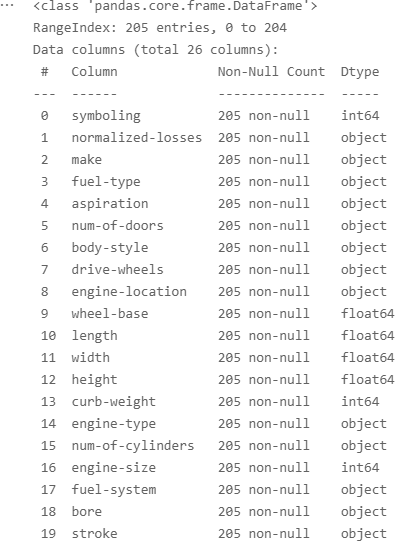
import numpy as np import pandas as pd

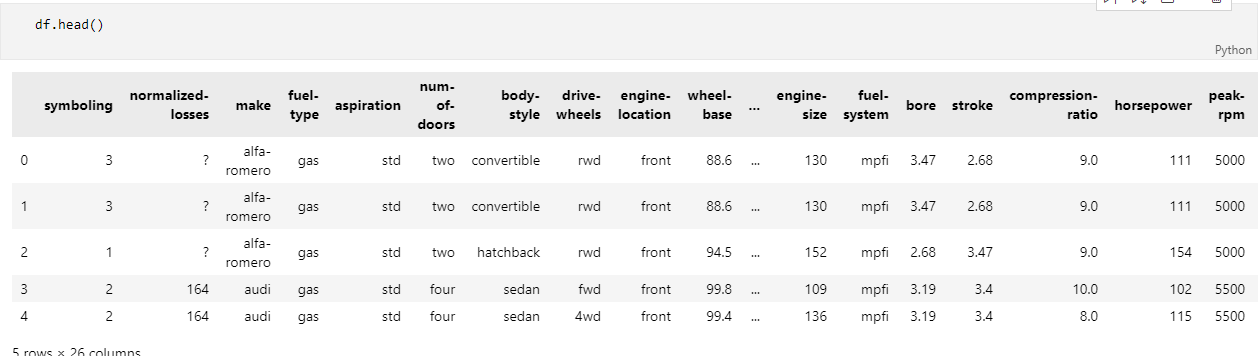
from sklearn.linear\_model import LinearRegression df=pd.read\_csv('autos\_dataset.csv')df

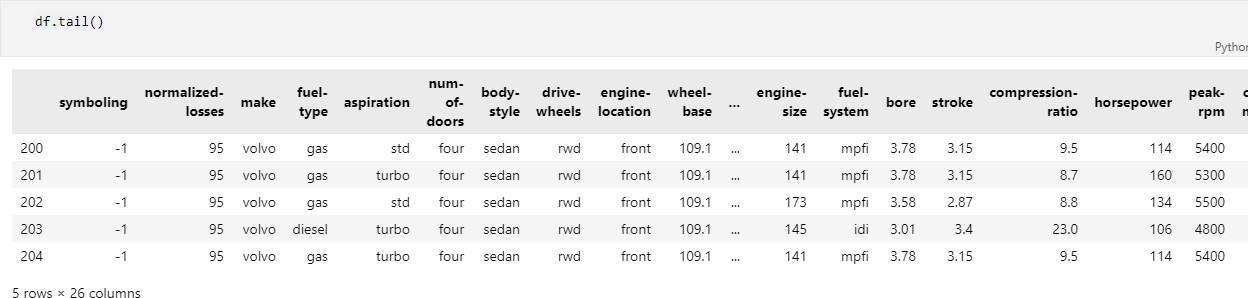
df.info() df.describe() df.head()

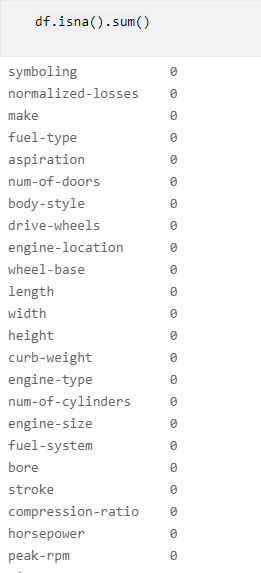
df.tail() df.isna().sum()

**Output:**









### CONCLUSION:

Thus we have studied and perform the practical for Exploratory Data Analysis.

**Signature of Faculty**

# PRACTICAL NO: 02

### AIM OF PRACTICAL / EXPERIMENT :

Write a program for supervised machine learning.

### SOFTWARE REQUIRED / EQUIPMENT DETAILS:-

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | NA |
| **2.** | Jupyter Notebook |  |  |

**THEORY:**

### What is machine learning?

Machine learning (ML) is a type of artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning [algorithms](https://whatis.techtarget.com/definition/algorithm) use historical data as input to predict new output values.

[Recommendation engines](https://whatis.techtarget.com/definition/recommendation-engine) are a common use case for machine learning. Other popular uses include frauddetection, spam filtering, malware threat detection, [business process automation](https://searchcio.techtarget.com/definition/business-process-automation) (BPA) and [predictive maintenance.](https://whatis.techtarget.com/definition/predictive-maintenance-PdM) Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: [supervised](https://www.techtarget.com/searchenterpriseai/definition/supervised-learning) learning, [unsupervised](https://whatis.techtarget.com/definition/unsupervised-learning) learning, semi- supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

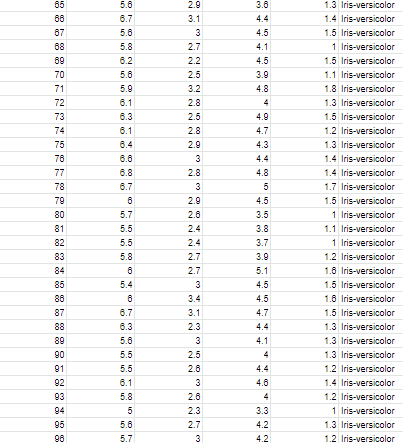
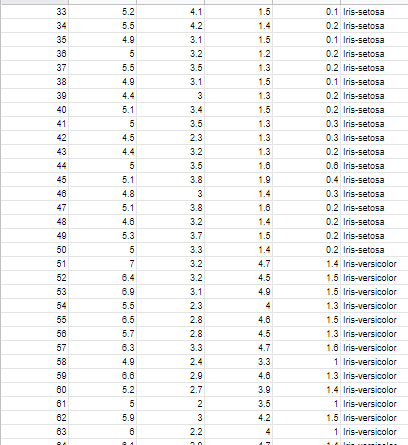
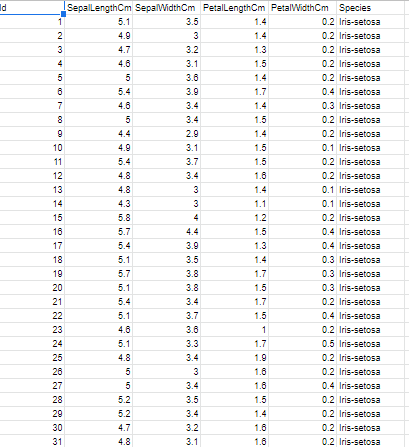
* **Supervised learning:** In this type of machine learning, [data scientists](https://www.techtarget.com/searchenterpriseai/definition/data-scientist) supply algorithmswith labeled training data and define the variables they want the algorithm to assess forcorrelations. Both the input and the output of the algorithm is specified.
* **Unsupervised learning:** This type of machine learning involves algorithms that train onunlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
* **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled [training data,](https://www.techtarget.com/searchenterpriseai/feature/Using-small-data-sets-for-machine-learning-models-sees-growth) but the model is free to explore the data on its own and develop its own understanding of thedata set.
* **Reinforcement learning**: Data scientists typically use [reinforcement learning](https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning) to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues asit works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

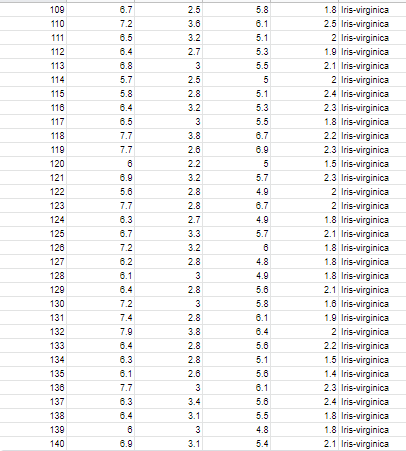
### Supervised learning:

Supervised machine learning requires the [data scientist](https://searchbusinessanalytics.techtarget.com/feature/Key-differences-of-a-data-scientist-vs-data-engineer) to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

* + Binary classification: Dividing data into two categories.
  + Multi-class classification: Choosing between more than two types of answers.
  + Regression modeling: Predicting continuous values.
  + Ensembling: Combining the predictions of multiple machine learning models to produce an accurate prediction.

### Data Set:





**Program Statement :**

To predict SepalLengthCm values from other independent variables Dependent Variable: SepalLengthCm

InDependent Variable >> SepalWidthCm, PetalLengthCm, PetalWidthCm, Species

**Code:-**

import pandas as pd import numpy as np

from sklearn.linear\_model import LinearRegression from sklearn.model\_selection import train\_test\_split

### # Model Evaluation

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error,r2\_score df.columns

df.axes df.info()

df['Id'].nunique() df['Species'].value\_counts()

df['Species'].replace({'Iris-setosa':0,'Iris-versicolor':1,'Iris-virginica':2},inplace= True) df.info() x = df.drop('SepalLengthCm',axis = 1)y =

df['SepalLengthCm'] y

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=1)y\_train

lr\_model=LinearRegression() lr\_model.fit(x\_train,y\_train)

### # Test Data Accuracy

y\_pred = lr\_model.predict(x\_test)

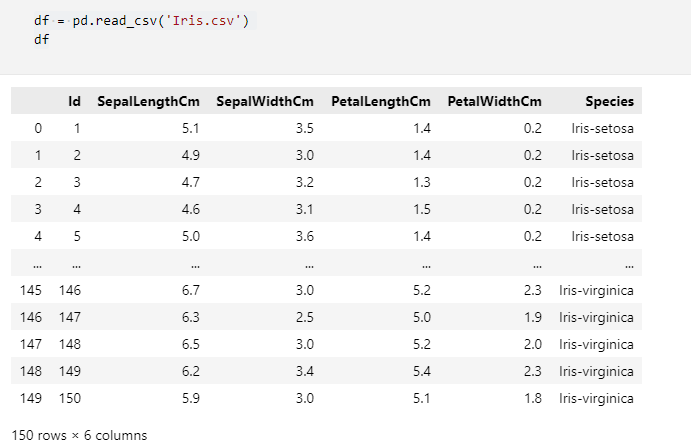
r2\_value = r2\_score(y\_test, y\_pred) print('R squared value is :',r2\_value)

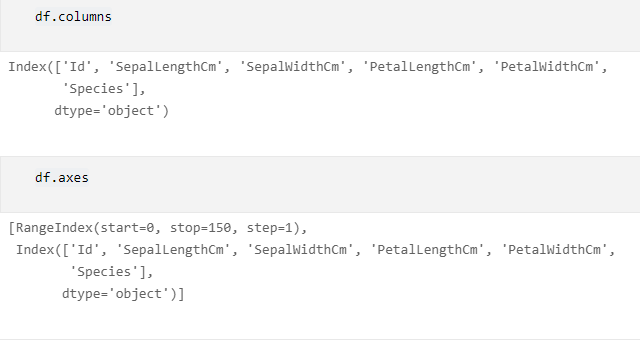
### # Train Data Accuracy

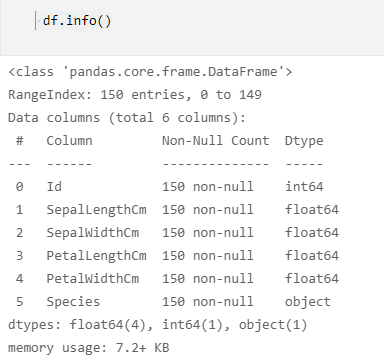
y\_pred\_train = lr\_model.predict(x\_train)

r2\_value = r2\_score(y\_train, y\_pred\_train)print('R squared value is :',r2\_value)

**Output:-**

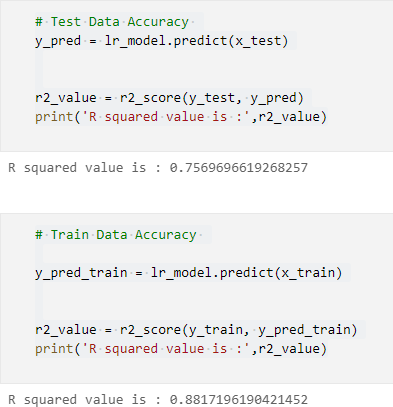












**Conclusion:** Thus we have studied and write the program for supervised machine learning.

**Signature of Faculty**

# PRACTICAL NO: 03

### AIM OF PRACTICAL / EXPERIMENT:

Write a Program for unsupervised machine learning.a) Identify Variance, accuracy, bias.

### SOFTWARE REQUIRED /EQUIPMENT DETAILS :-

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | **NA** |
| **2.** | Jupyter Notebook |  |  |

**THEORY:**

### What is Unsupervised Learning?

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

### The unsupervised learning algorithm can be further categorized into two types of problems:

* **Clustering:** Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.
* **Association**: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

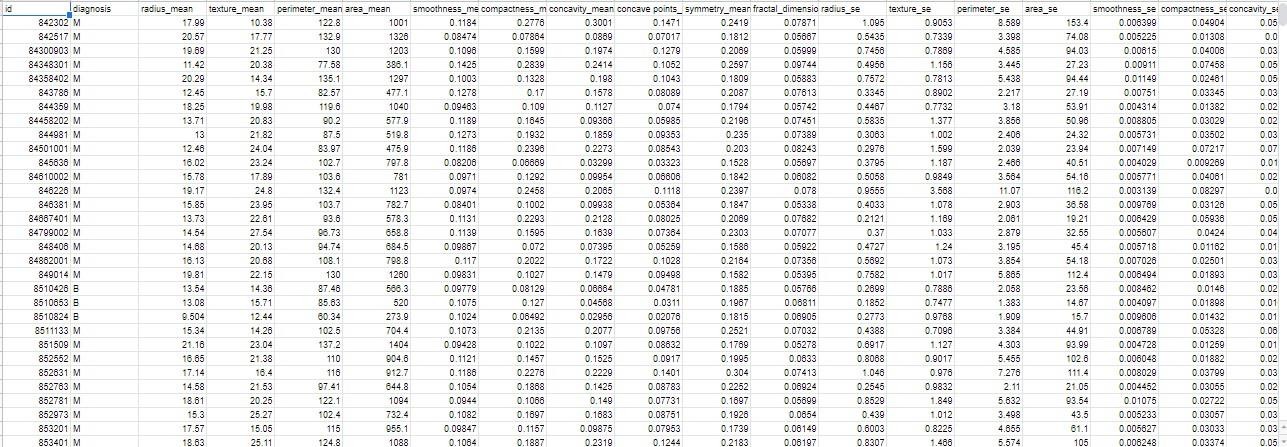
### Advantages of Unsupervised Learning

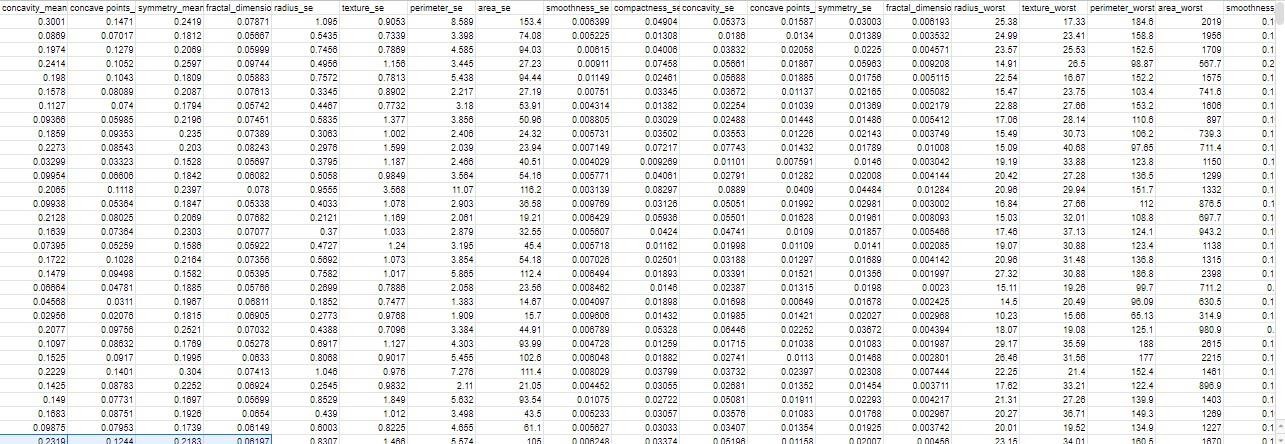
* Unsupervised learning is used for more complex tasks as compared to supervised learningbecause, in unsupervised learning, we don't have labeled input data.
* Unsupervised learning is preferable as it is easy to get unlabeled data in comparison to labeled data.

### Disadvantages of Unsupervised Learning

* Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output.
* The result of the unsupervised learning algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

**Data Set:-**





### Code:-

**# Importing the necessary libraries** import warnings warnings.filterwarnings('ignore')

import numpy as np import pandas as pd

from sklearn.preprocessing import StandardScaler from sklearn.decomposition import IncrementalPCA from sklearn.cluster import KMeans

from sklearn import metrics

import matplotlib.pyplot as plt

### # reading the Cancer Wisconsin (Diagnostic) Data Setdf =

pd.read\_csv('data.csv')

### # displaying top 5 instances of the dataset

print(df.head())

### # the unwanted columns, 'id' and 'Unnamed: 32' are dropped

df.drop(['id', 'Unnamed: 32'], axis = 1, inplace = True)

### # label-encoding of the target label, 'diagnosis' such that B(Benign) -> 0 and M (Malignant) -> 1

df['diagnosis'] = df['diagnosis'].map({'B':0, 'M':1})

### # spliting into X (features) and y (target label)

X = df.iloc[:, 1:] y = df['diagnosis']

### # feature scaling

X\_scaled = StandardScaler().fit\_transform(X)

### # Incremental Principal Component Analysis to select 2 features such that they explainas much variance as possible

pca = IncrementalPCA(n\_components = 2) X\_pca = pca.fit\_transform(X\_scaled)

### # Scatter Plot of the 2 Principal Components with labels indicated by colors

plt.scatter(X\_pca[:,0], X\_pca[:,1], c = y, cmap = 'plasma') plt.xlabel('1st Principal Component')

plt.ylabel('2nd Principal Component') plt.title('Scatter Plot of the 2 Principal Component') plt.show()

**# k-Means Clustering with 2 clusters as there are 2 labels** model = KMeans(n\_clusters = 2, random\_state=1234).fit(X\_pca) y\_cluster = model.predict(X\_pca)

**# Getting the Accuracy of the k-Means Clustering Model** print('Accuracy of the Model: ', metrics.accuracy\_score(y, y\_cluster)) print()

**# Getting the Precision of the k-Means Clustering Model** print('Precision of the Model: ', metrics.precision\_score(y, y\_cluster)) print()

**# Getting the Recall of the k-Means Clustering Model** print('Recall of the Model: ', metrics.recall\_score(y, y\_cluster)) print()

**# Getting the F1-Score of the k-Means Clustering Model** print('F1-Score of the Model: ', metrics.f1\_score(y, y\_cluster)) print()

### # plotting the decision boundary in the scatter plot of the 2 Principal Components with labels indicated by colors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x\_min, | x\_max = X\_pca[:, 0].min() | - 1, | X\_pca[:, 0].max() | + 1 |
| y\_min, | y\_max = X\_pca[:, 1].min() | - 1, | X\_pca[:, 1].max() | + 1 |

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.1),

np.arange(y\_min, y\_max, 0.1))

Z = model.predict(np.c\_[xx.ravel(), yy.ravel()])Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, cmap = 'plasma')

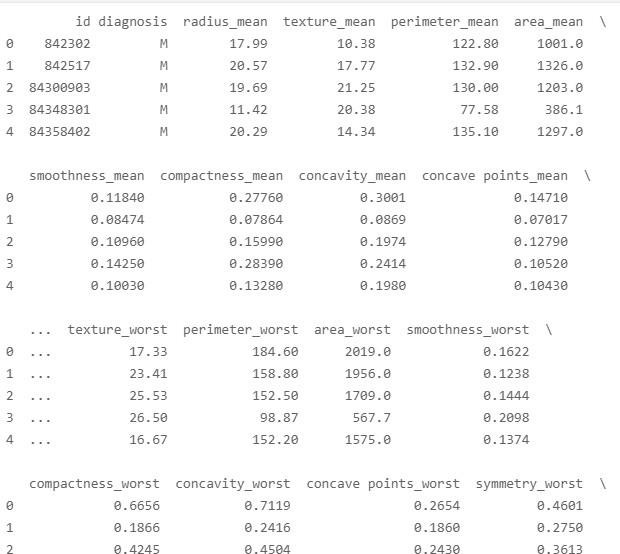
plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c = y, s = 30, edgecolor = 'k', cmap = 'plasma')plt.xlabel('1st Principal Component')

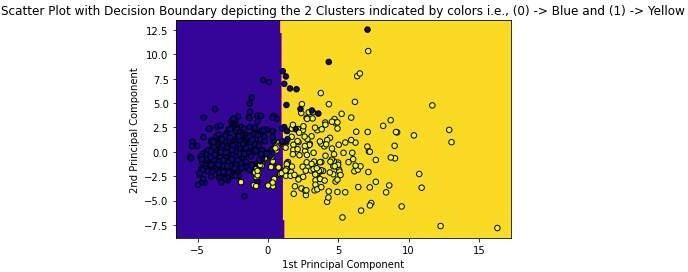
plt.ylabel('2nd Principal Component')

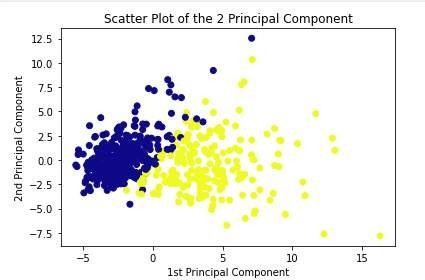
plt.title('Scatter Plot with Decision Boundary depicting the 2 Clusters indicated bycolors i.e., (0) -> Blue and (1) -> Yellow')

plt.show()

### Output:







**Conclusion:**

Thus we have studied and write a Program for unsupervised machine learning.

**Signature of Faculty**

# PRACTICAL NO: 04

### AIM OF PRACTICAL / EXPERIMENT :

Write a program for simple linear regression.

a) Estimating the accuracy of model and coefficient of estimate.

**SOFTWARE REQUIRED /EQUIPMENT DETAILS :-**

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | **NA** |
| **2.** | Jupyter Notebook |  |  |

**THRORY:-**

### Linear Regression:

Linear regression attempts to model the relationship between two variables by fitting a linear equation toobserved data. One variable is considered to be an explanatory variable, and the other is considered to bea dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Before attempting to fit a linear model to observed data, a modeler should first determine whether or notthere is a relationship between the variables of interest. This does not necessarily imply that one variable *causes* the other (for example, higher SAT scores do not *cause* higher college grades), but that there is some significant association between the two variables. If there appears to be no association between the proposed explanatory and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will notprovide a useful model. A valuable numerical measure of association between two variables is

the [correlation coefficient](http://www.stat.yale.edu/Courses/1997-98/101/correl.htm), which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables. A linear regression line has an equation of the form *Y = a + bX*, where *X* is the explanatory variable and *Y* is the dependent variable. The slope of the line is *b*, and *a* is the intercept (the value of *y* when *x* =0).

### The Four Assumptions of Linear Regression:

[Linear regression](https://www.statology.org/introduction-to-simple-linear-regression/) is a useful statistical method we can use to understand the relationship between two variables, x and y. However, before we conduct linear regression, we must first make sure that four assumptions are met:

1. **Linear relationship:** There exists a linear relationship between the independent variable, x, and thedependent variable, y.
2. **Independence:** The residuals are independent. In particular, there is no correlation between consecutiveresiduals in time series data.
3. **Homoscedasticity**: The residuals have constant variance at every level of x.
4. **Normality:** The residuals of the model are normally distributed.

If one or more of these assumptions are violated, then the results of our linear regression may be unreliable or even misleading.

Linear Regression is generally classified into two types:

* 1. Simple Linear Regression
  2. Multiple Linear Regression

### Simple Linear Regression

In Simple Linear Regression, we try to find the relationship between a single independent

variable (input) and a corresponding dependent variable (output). This can be expressed in the form ofa straight line.

The same equation of a line can be re-written as:

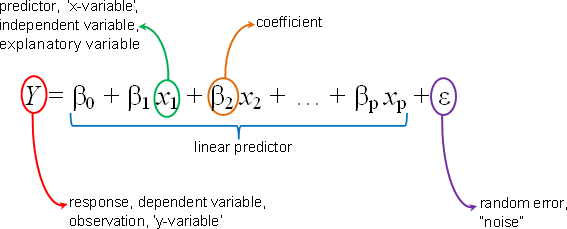


* + - **Y** represents the output or dependent variable.
    - **β0 and β1** are two unknown constants that represent the intercept and coefficient (slope)respectively.
    - **ε** (Epsilon) is the error term.

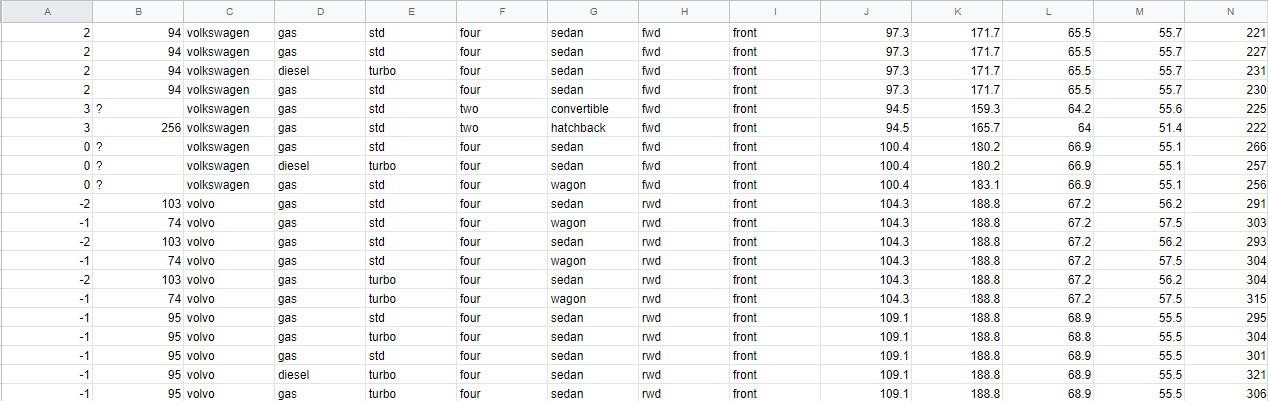
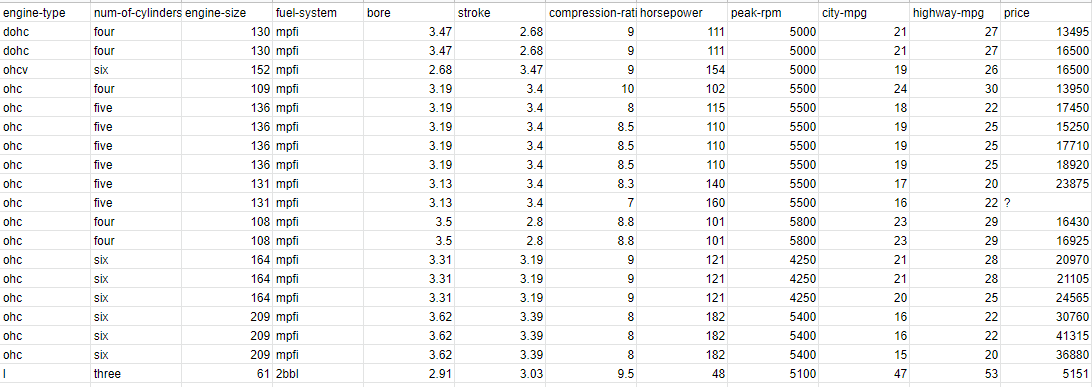
### Multiple Linear Regression

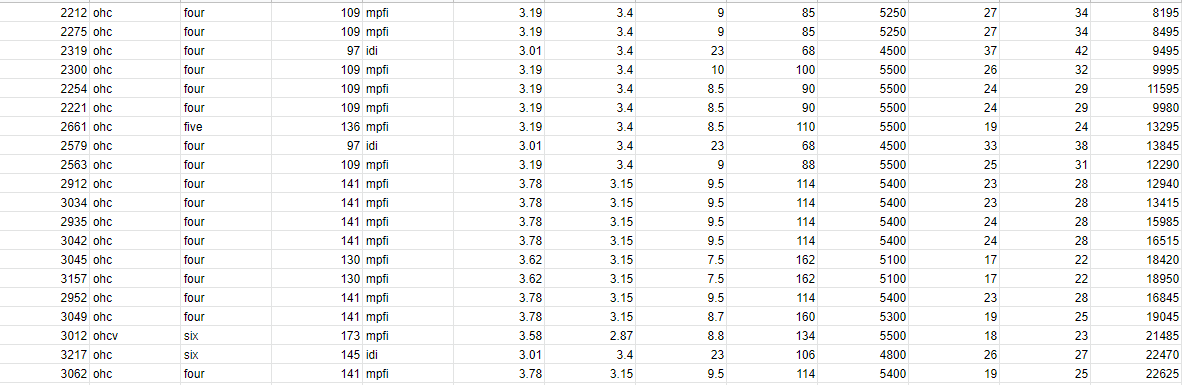
In Multiple Linear Regression, we try to find the relationship between 2 or more independent variables (inputs) and the corresponding dependent variable (output). The independent variables can be continuous or categorical.

The equation that describes how the predicted values of y is related to p independent variables is calledas Multiple Linear Regression equation :



### Data Set:





**PROBLEM STATEMENT :**

[link text](https://)Predict the price of cars using various variable

> Indented block

### Code:

import pandas as pd import numpy as np

from sklearn.linear\_model import LinearRegression from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score,mean\_absolute\_error,mean\_squared\_error import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv('autos\_dataset.csv') df.head().T

df.shape df.info()

### # df.isna().sum()

df.replace({'?':np.nan},inplace = True)

### #df.isna().sum()

df.info()

df.drop(['make','aspiration','fuel-type','num-of-doors','body-style','drive-wheels','engine-location','engine- type','fuel-system'],axis = 1, inplace=True)

### # df.info()

df.drop('normalized-losses',axis =1 , inplace =True)

### # df.isna().sum()

df['bore'].fillna(df['bore'].median(),inplace= True) df['stroke'].fillna(df['stroke'].median(),inplace= True) df['horsepower'].fillna(df['horsepower'].median(),inplace= True) df['peak-rpm'].fillna(df['peak-rpm'].median(),inplace= True) df['price'].fillna(df['price'].median(),inplace= True)

### # df.isna().sum()

df['bore'] = df['bore'].astype(float) df['stroke'] = df['stroke'].astype(float)

df['horsepower'] = df['horsepower'].astype(float) df['peak-rpm'] = df['peak-rpm'].astype(float) df['price'] = df['price'].astype(float)

df['num-of-cylinders'] = df['num-of-cylinders'].astype(float) df['num-of-cylinders'].unique()

df['num-of-cylinders'].replace({'four':4, 'six':6, 'five':5, 'three':3, 'twelve':12,'two':2, 'eight':8},inplace = True)

x = df.drop('price',axis = 1) y = df['price']

x\_train,x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=10)

### # x\_train

lr\_model = LinearRegression() lr\_model.fit(x\_train,y\_train)

1. Normality of Residual:
   1. Density plot (sns.kdeplot)
   2. qq plot
   3. Shapiro Test
   4. KS Test

5. Normality

### # scipy and statsmodel

1. Homoscedascticity : scatterplot

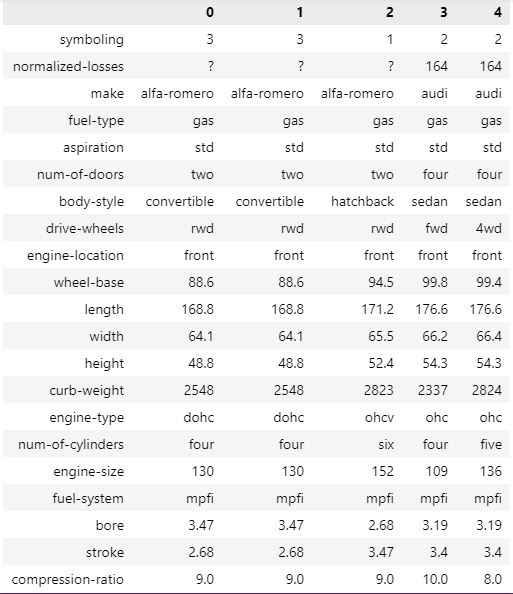
y\_pred = lr\_model.predict(x\_test) # y\_pred[:5]

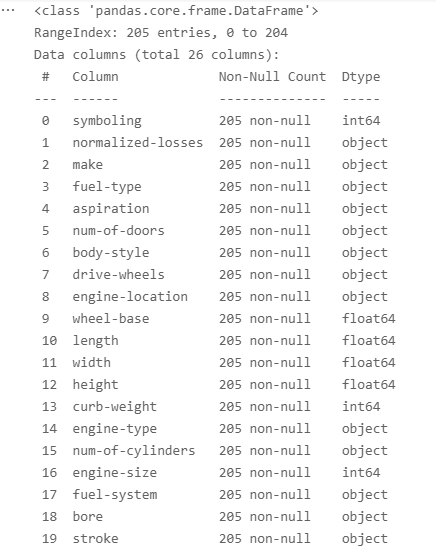
mse = mean\_squared\_error(y\_test, y\_pred) print("Mean squared value is ",mse)

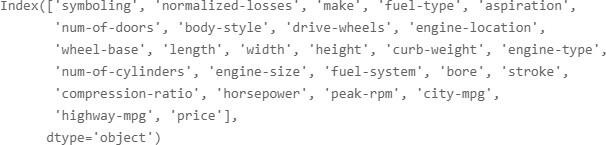
mae = mean\_absolute\_error(y\_test, y\_pred) print("Mean Absolute Error is ",mae)

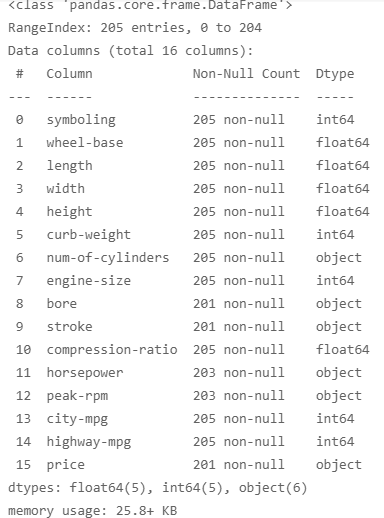
\_model.predlrict(x\_test) x\_test.iloc[0:2]

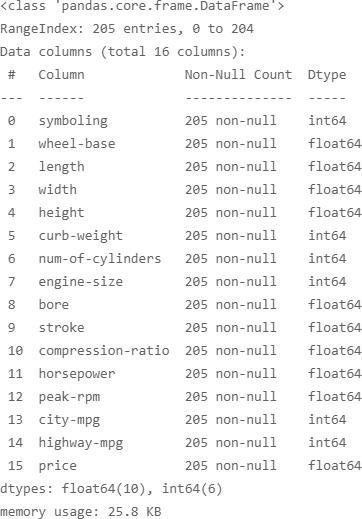
### Output:

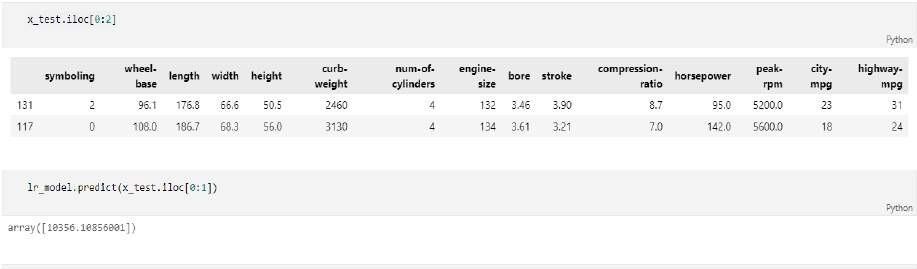












**Conclusion:**

Thus we have studied write the program for simple linear regression.

**Signature of Faculty**

# PRACTICAL NO: 05

### AIM OF PRACTICAL / EXPERIMENT :

Write a program for KNN.

### SOFTWARE REQUIRED /EQUIPMENT DETAILS :-

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items /**  **Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | **NA** |
| **2.** | Jupyter Notebook |  |  |

**THEORY:-**

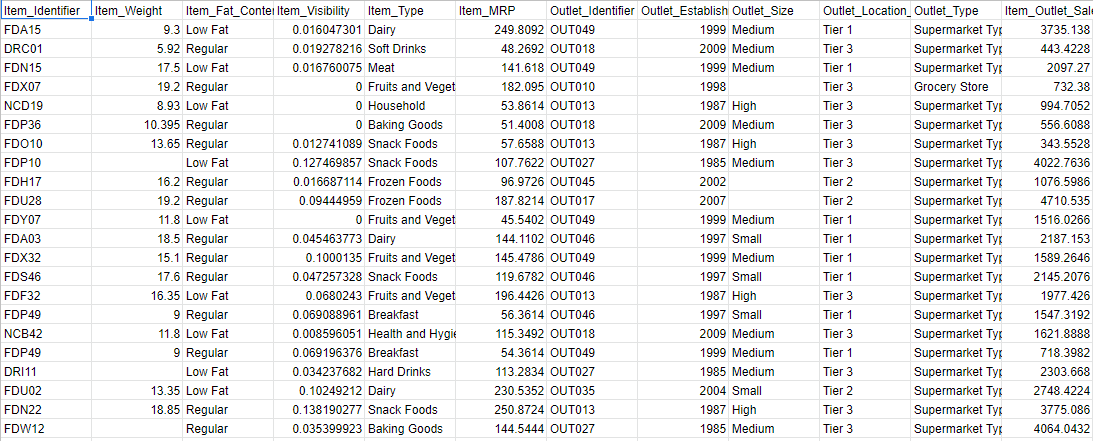
K-Nearest Neighbor(KNN) Algorithm for Machine Learning:

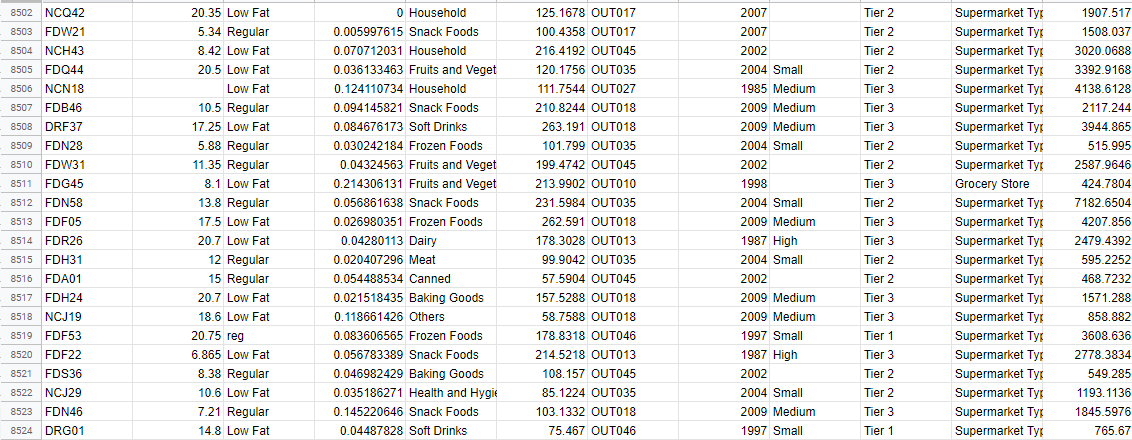
* K-NN algorithm stores all the available data and classifies a new data point based on thesimilarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

How does K-NN work?

* Step-1: Select the number K of the neighbors
* Step-2: Calculate the Euclidean distance of K number of neighbors
* Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
* Step-4: Among these k neighbors, count the number of the data points in each category.
* Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
* Step-6: Our model is ready.

### Data Set:-





**Code:-**

import pandas as pd import numpy as np

from sklearn.preprocessing import StandardScaler,MinMaxScaler from sklearn.neighbors import KNeighborsRegressor

from sklearn.model\_selection import train\_test\_split,GridSearchCV,RandomizedSearchCV

from sklearn.metrics import r2\_score,mean\_squared\_error, mean\_absolute\_error df = pd.read\_csv('Sales\_data.csv')

df

df.info() df.isna().sum()

df['Item\_Identifier'].value\_counts() df['Item\_Weight']

df['Item\_Weight'].fillna(df['Item\_Weight'].mean(),inplace=True) df['Item\_Fat\_Content'].value\_counts()

df['Item\_Fat\_Content'] = df['Item\_Fat\_Content'].apply(lambda x : x.lower()) df['Item\_Fat\_Content'].value\_counts()

df['Item\_Fat\_Content'].replace({'lf':0,'low fat':0,'regular':1,'reg':1},inplace = True)

df['Item\_Fat\_Content'].value\_counts() df['Item\_Visibility'] df['Item\_Type'].value\_counts()

df\_item\_type = pd.get\_dummies(df['Item\_Type'],prefix='Item\_Type') # df\_item\_type

df[['Item\_MRP']].describe() df['Outlet\_Identifier'].value\_counts()

df\_Outlet\_Identifier = pd.get\_dummies(df['Outlet\_Identifier'],prefix='Outlet\_Identifier') df\_Outlet\_Identifier

df['Outlet\_Establishment\_Year']

df['Outlet\_Establishment\_Year'] = 2021 - df['Outlet\_Establishment\_Year'] df['Outlet\_Establishment\_Year']

df['Outlet\_Size'].mode()

df['Outlet\_Size'].mode().loc[0] df['Outlet\_Size'].fillna('Medium',inplace = True)

df['Outlet\_Size'].replace({'Small':0,'Medium':1,'High':2},inplace = True) df['Outlet\_Location\_Type'].value\_counts() df['Outlet\_Location\_Type'].replace({'Tier 1':1,'Tier 2':2,'Tier 3':3},inplace = True) df['Outlet\_Type'].value\_counts()

df\_Outlet\_Type = pd.get\_dummies(df['Outlet\_Type'],prefix='Outlet\_Type') df\_Outlet\_Type

column\_list =['Item\_Identifier','Item\_Type','Outlet\_Identifier','Outlet\_Type'] df.drop(column\_list,axis = 1, inplace = True)

# df.drop('Outlet\_Est\_Year',axis = 1, inplace = True)

df\_list = [df,df\_item\_type,df\_Outlet\_Identifier,df\_Outlet\_Type] df = pd.concat(df\_list,axis = 1)

df

x = df.drop('Item\_Outlet\_Sales',axis = 1)

y = df['Item\_Outlet\_Sales']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=1)

knn\_model = KNeighborsRegressor() knn\_model.fit(x\_train,y\_train)

y\_pred = knn\_model.predict(x\_test)

mse = mean\_squared\_error(y\_pred,y\_test) print("Mean squared Error :",mse)

rmse = np.sqrt(mse)

print("Root Mean Squared Error :",rmse)

r2\_value = r2\_score(y\_test,y\_pred) print("R Squared Value is :",r2\_value)

y\_pred\_train = knn\_model.predict(x\_train)

mse = mean\_squared\_error(y\_train,y\_pred\_train) print("Mean squared Error :",mse)

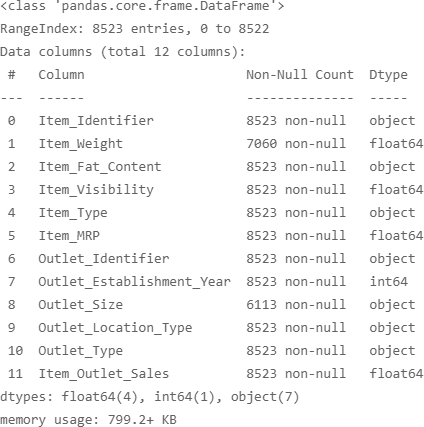
rmse = np.sqrt(mse)

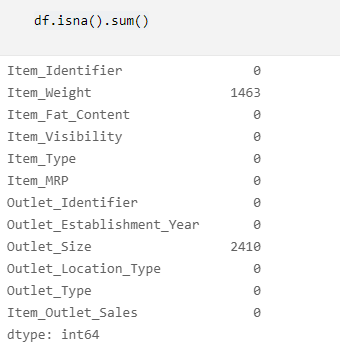
print("Root Mean Squared Error :",rmse)

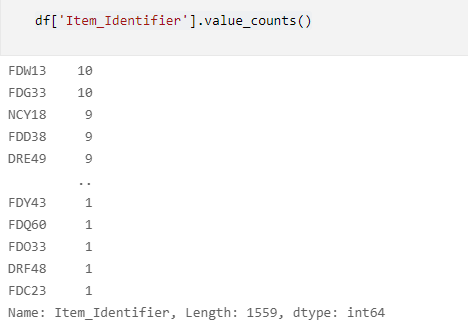
r2\_value = r2\_score(y\_train,y\_pred\_train) print("R Squared Value is :",r2\_value)

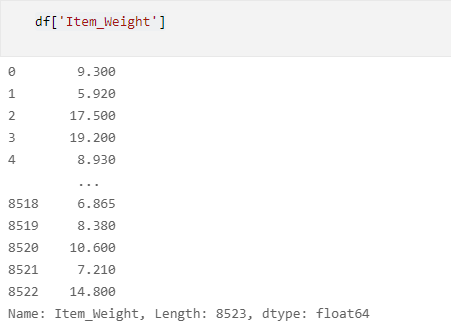
### Output:





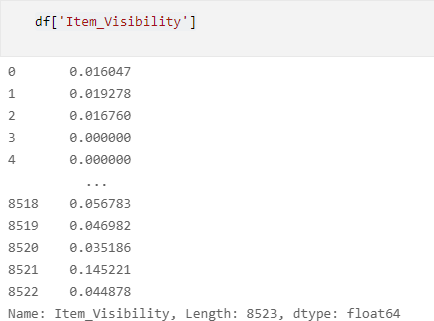


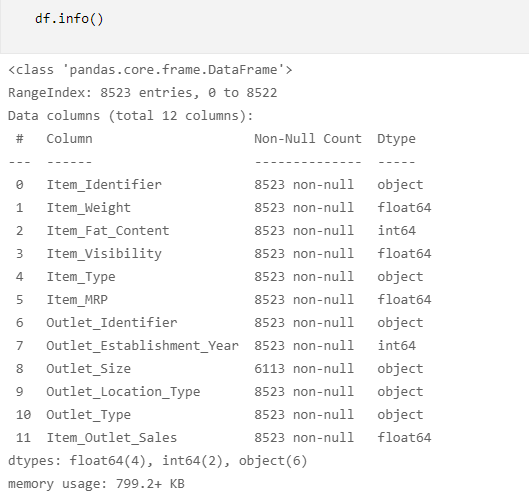


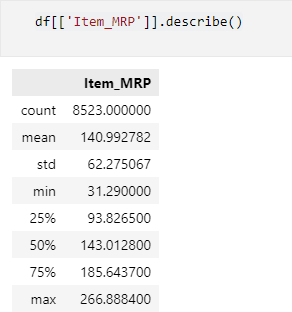




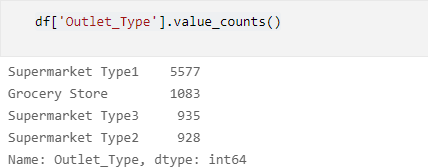


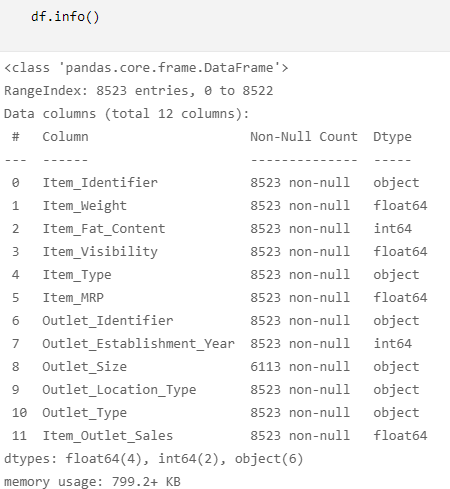


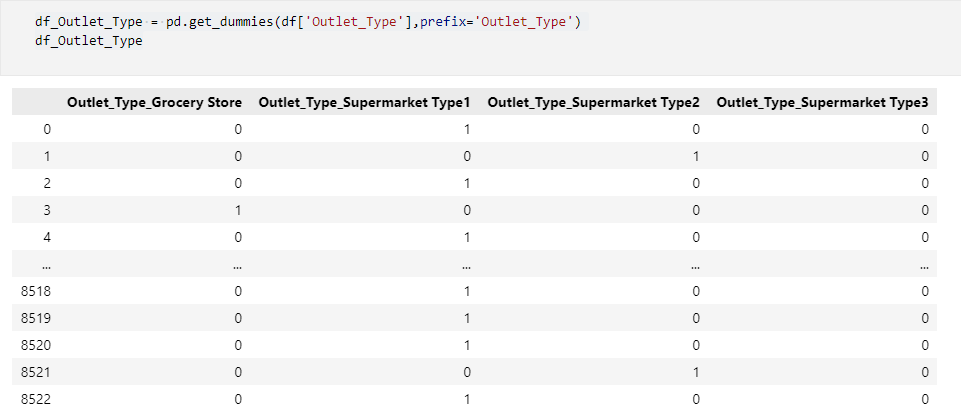


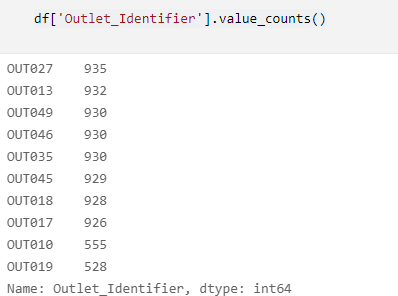


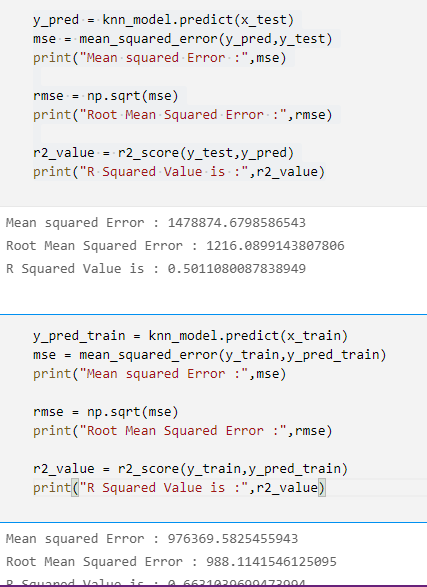












**Conclusion:**

Thus we have learn about KNN algorithm and write the program for it.

**Signature of Faculty**

# PRACTICAL NO: 06

### AIM OF PRACTICAL / EXPERIMENT :

Write a program for logistic regression.

### SOFTWARE REQUIRED / EQUIPMENTDETAILS:-

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | **NA** |
| **2.** | Jupyter Notebook |  |  |

**THEORY:**

This type of statistical analysis (also known as *logit model*) is often used for predictive analytics and modeling, and extends to applications in machine learning. In this analytics approach, the dependent variable is finite or categorical: either A or B (binary regression) or a range of finite options A, B, C or D (multinomial regression). It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.

This type of statistical analysis (also known as *logit model*) is often used for predictive analytics and modeling, and extends to applications in machine learning. In this analytics approach, the dependent variable is finite or categorical: either A or B (binary regression) or a range of finite options A, B, C or D (multinomial regression). It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.

### Why logistic regression is important?

Predictive models built using this approach can make a positive difference in your business or organization. Because these models help you understand relationships and predict outcomes, you can act to improve decision-making. For example, a manufacturer’s analytics team can use logistic regression analysis as part of a statistics software package to discover a probability between part failures in machines and the length of time those parts are held in inventory. With the information it receives from this analysis, the team can decide to adjust delivery schedules or installation times to eliminate future failures.In medicine, this analytics approach can be used to predict the likelihood of disease or illness fora given population, which means that preventative care can be put in place. Businesses can use this approach to uncover patterns that lead to higher employee retention or create more profitable products byanalyzing buyer behavior. In the business world, this type of analysis is applied by data scientists whose goal is clear: to analyze and interpret complex digital data.

### Data set:

**code:-**

import pandas as pd import numpy as np

from sklearn.linear\_model import LogisticRegression from sklearn.model\_selection import train\_test\_split#

### #Evaluation

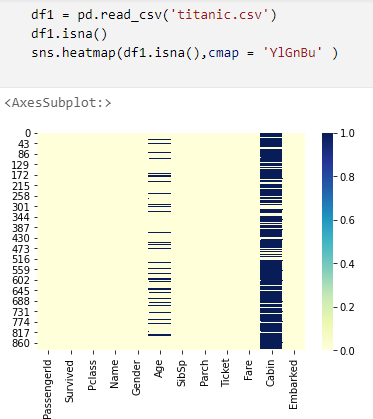
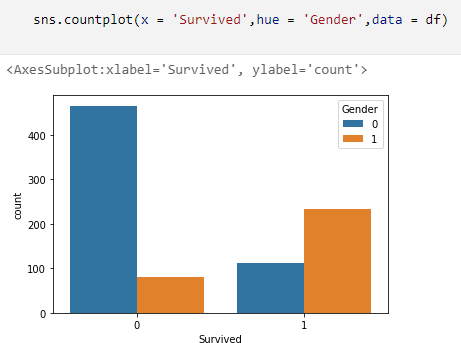
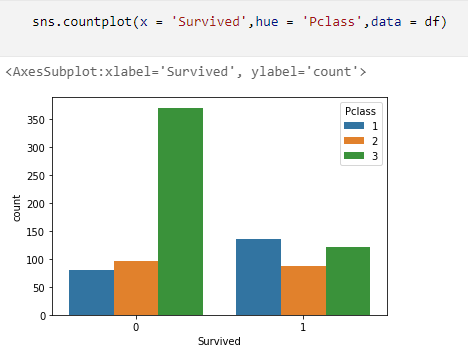
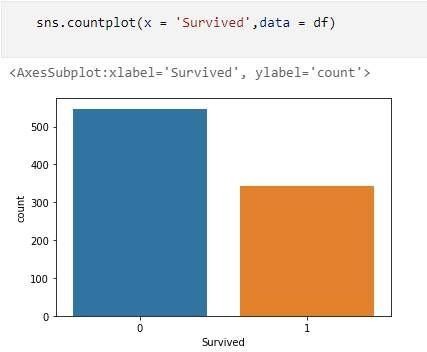
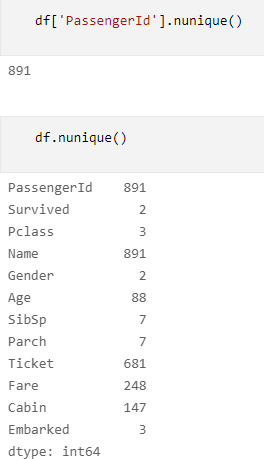
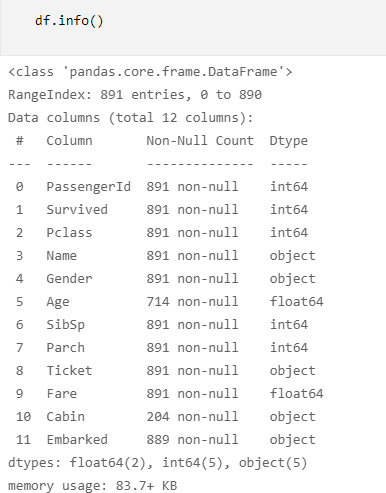
from sklearn.metrics import confusion\_matrix, multilabel\_confusion\_matrix from sklearn.metrics import classification\_report,accuracy\_score

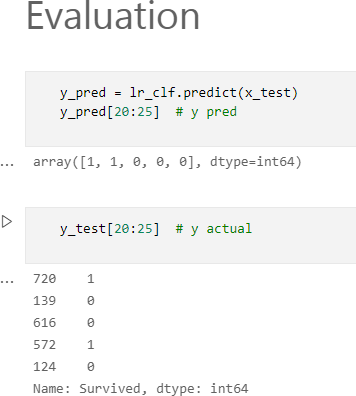
from sklearn.metrics import roc\_curve,roc\_auc\_score import seaborn as sns

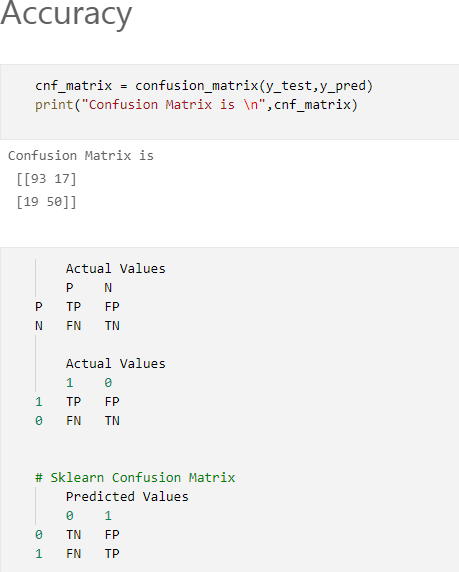
import matplotlib.pyplot as plt import warnings warnings.filterwarnings('ignore')

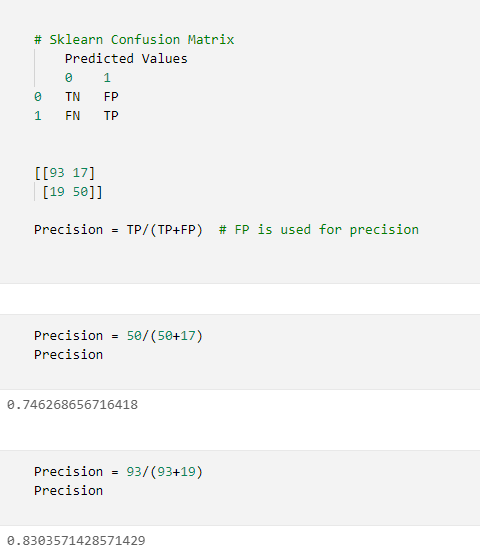
### Output:

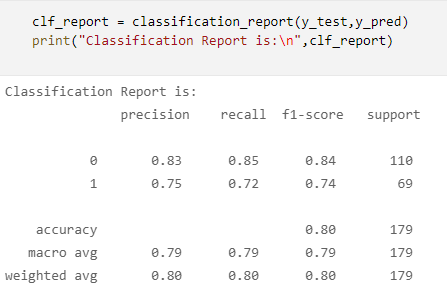












**Conclusion:**

Thus we have executed the program for Logistic regression.

**Signature of Faculty**

# PRACTICAL NO: 07

### AIM OF PRACTICAL / EXPERIMENT :

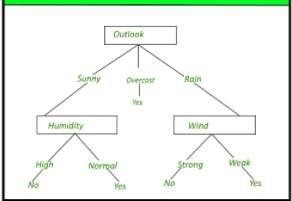
Write a program for Decision tree.

### SOFTWARE REQUIRED / EQUIPMENTDETAILS:-

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | **NA** |
| **2.** | Jupyter Notebook |  |  |

**THEORY:**

### Decision Tree :

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

### Construction of Decision Tree :

A tree can be *“learned”* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

### Decision Tree Representation :

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.

The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No). For example, the instance

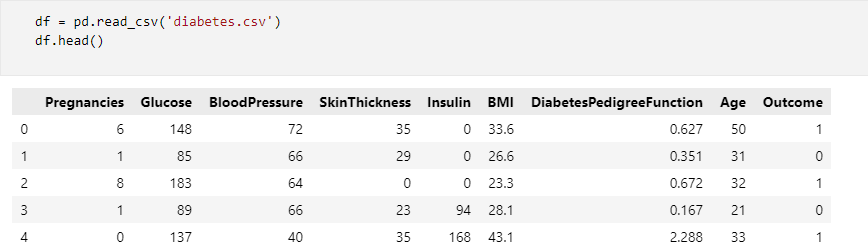
### Code:-

import pandas as pd import numpy as np

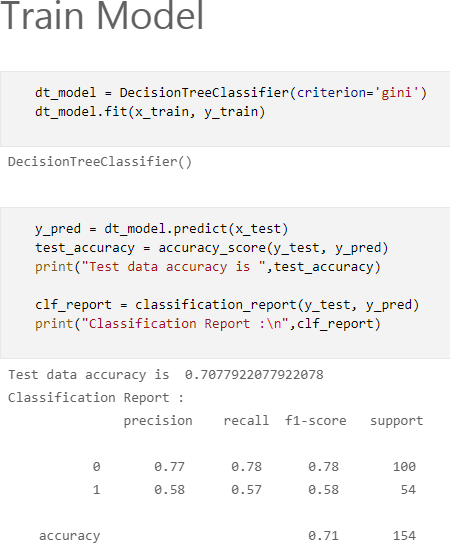
from sklearn.tree import DecisionTreeClassifier from sklearn.tree import plot\_tree

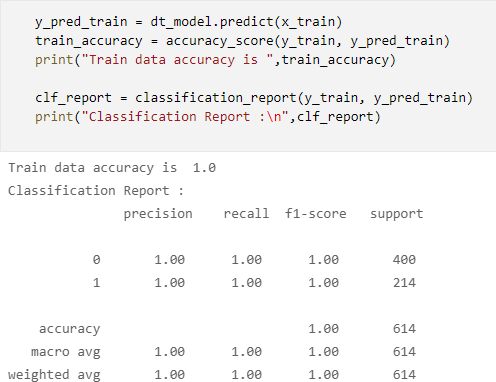
from sklearn.model\_selection import train\_test\_split,GridSearchCV,RandomizedSearchCV from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report import matplotlib.pyplot as plt

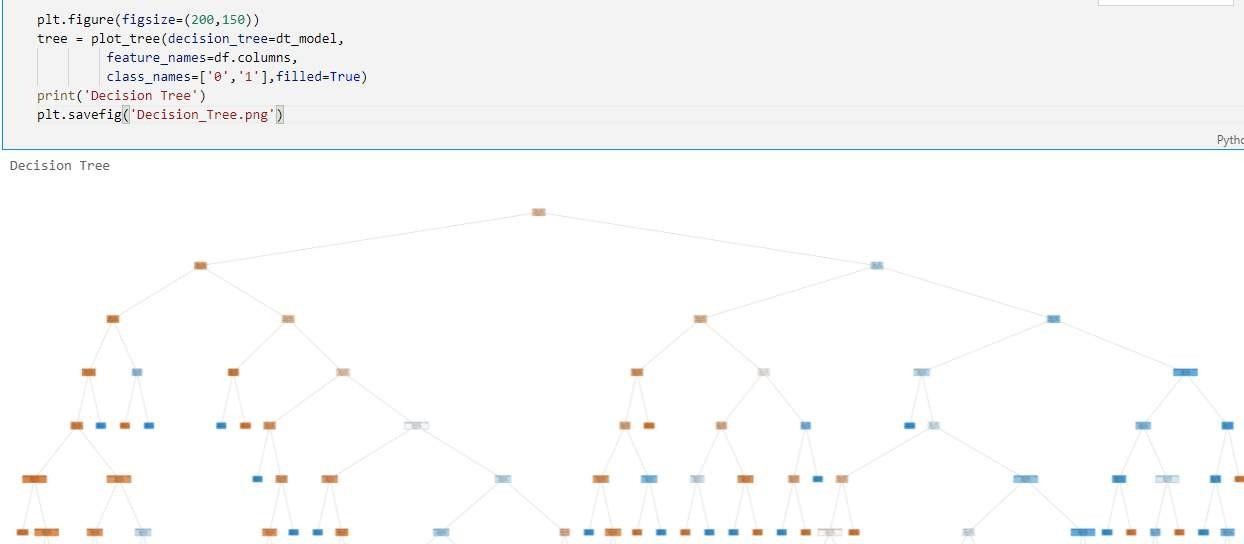
import seaborn as sns

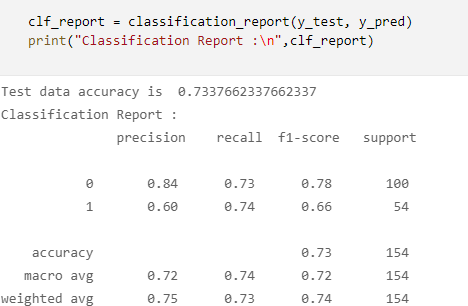


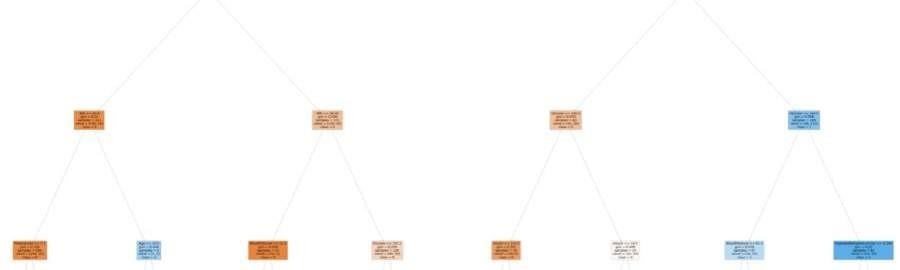
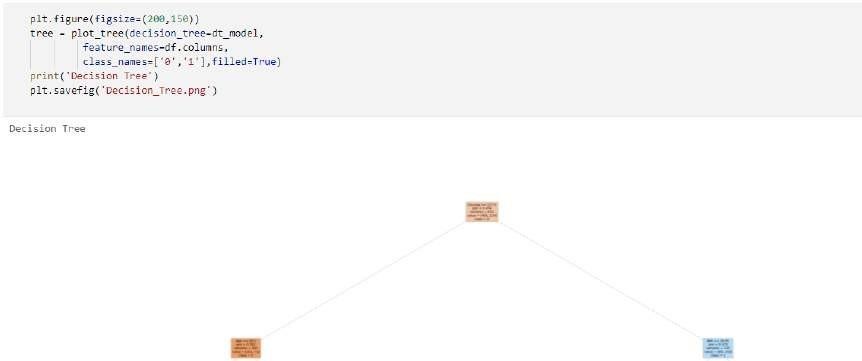












### Conclusion:

Thus we have executed the program for Decision tree.

**Signature of Faculty**

# PRACTICAL NO: 08

### AIM OF PRACTICAL / EXPERIMENT :

Write a program for k-means clustering

### SOFTWARE REQUIRED / EQUIPMENTDETAILS:-

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Name of Equipment / Items / Software Tool** | **Specification** | **Qty Required** |
| **1** | Anaconda(Python) | (python 3.7.9) | **NA** |
| **2.** | Jupyter Notebook |  |  |

**THEORY:**

### K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

### What is K-Means Algorithm?

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

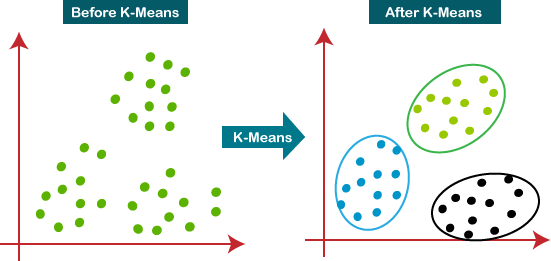
It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k- center, create a cluster.
* Hence each cluster has datapoints with some commonalities, and it is away from other clusters.



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

**Step-1:** Select the number K to decide the number of clusters.

**Step-2:** Select random K points or centroids. (It can be other from the input dataset).

**Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.

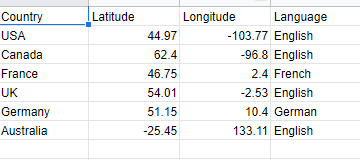
**Step-4:** Calculate the variance and place a new centroid of each cluster.

**Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

**Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.

**Step-7**: The model is ready

### Data set:-



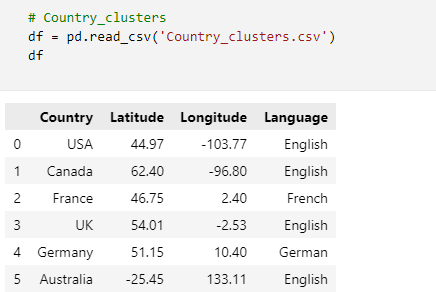
**Program & Output:**

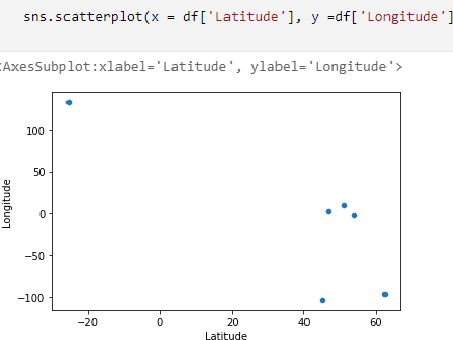
import pandas as pd import numpy as np

from sklearn.cluster import KMeans

from sklearn.preprocessing import MinMaxScaler,StandardScaler import matplotlib.pyplot as plt

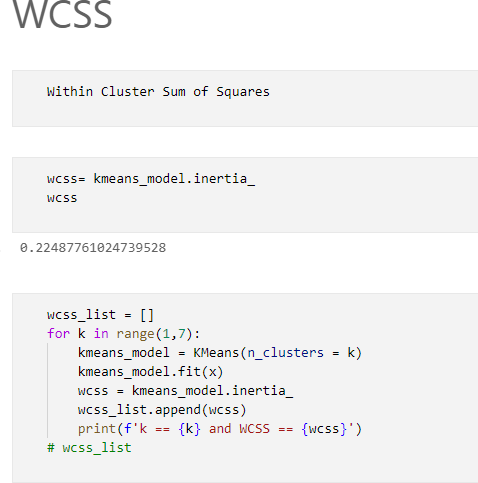
import seaborn as sns

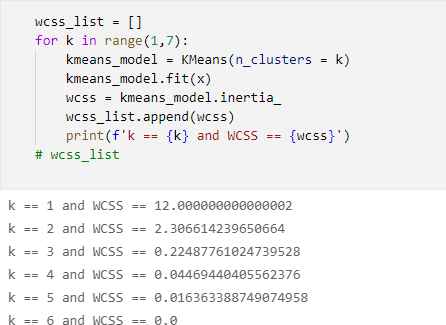


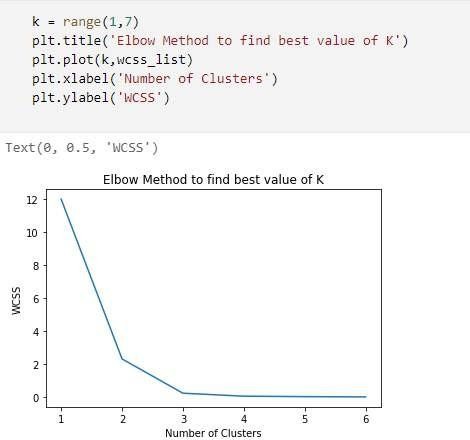












### Conclusion:-

Thus we have successfully executed the program for k-means clustering.

### Signature of Faculty