# KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE – 641 049

**Department of Information Science and Engineering**



**U18ISI6204 – Machine Learning Techniques**

**Laboratory Manual**

**2020-2024 Batch**

List of Experiments

1. Download a simple dataset from https://archive.ics.uci.edu/ml/index.php and do the imputation process using basic statistics
2. Write a Program to implement Linear Regression (Scikit) involving single variable and multiple variables and analyze the house price prediction.
3. Write a Program to implement Logistic Regression by plotting the decision boundary and use it to classify spam mail
4. Use a sample dataset and with help of Support Vector Machine, classify the subject whether it has cancer or not
5. Implement KNN algorithm using the balanced iris data set for multiclass classification and predict the flower species
6. Using Principal component Analysis as Dimensionality reduction component implement Logistic Regression for detecting credit card frauds
7. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
8. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs
9. Write a program to implement k-means clustering algorithm for iris dataset

# Experiment No: 1

**INTRODUCTION**

In this experiment, we have to perform EDA on the titanic dataset. Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Specific statistical functions and techniques you can perform with EDA tools include:

* + Clustering and dimension reduction techniques, which help create graphical displays of high-dimensional data containing many variables.
  + Univariate visualization of each field in the raw dataset, with summary statistics.
  + Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you’re looking at.
  + Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
  + K-means Clustering is a clustering method in unsupervised learning where data points are assigned into K groups, i.e. the number of clusters, based on the distance from each group’s centroid. The data points closest to a particular centroid will be clustered under the same category. K-means Clustering is commonly used in market segmentation, pattern recognition, and image compression.
  + Predictive models, such as linear regression, use statistics and data to predict outcomes.

# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform Exploratory Data Analysis on the given dataset, which includes performing basics statistics like mean, median, quartile, standard deviation etc... and to handle missing values.

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries.

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib and seaborn – for data visualization

**STEP-3:** Loading the dataset using read\_csv method in pandas’ module.

**STEP-4:** Analyze the dataset using info method, which gives its data types and number of non- null values in each column.

**STEP-5:** Perform basic statistic operation using describe () method.

**STEP-6:** To handle missing value,

* 1. Too many missing values in the column then drop the column.
  2. Less number of missing values then look for any relationship with another columns and then take either mean or medium of the column, fill it.

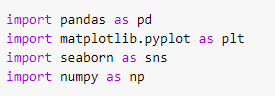
**STEP-7:** To find the relationship with other columns use heatmap and correlation matrix of the entire dataset.

**STEP-8:** If found any columns relation eg: for age column in titanic dataset is related with Pclass and sex column, then fill up the missing values with median of data which is related data between columns (fill the missing value of age for female in pclass 1 with median of data belonging to pclass 1 of female sex).

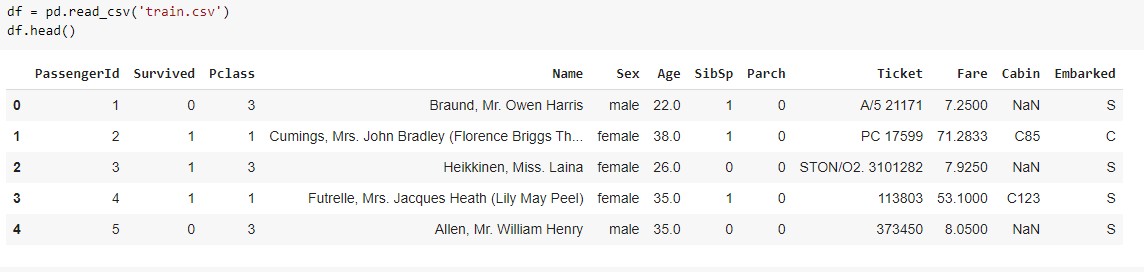
**STEP-9:** Stop the program.

# PROGRAM:

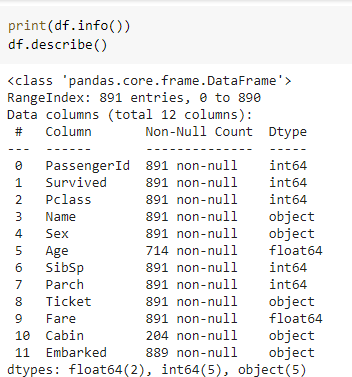
**Importing libraries**



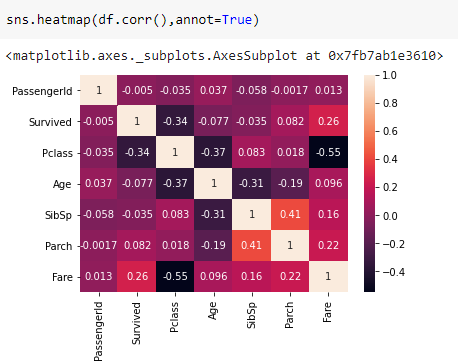
# Loading dataset



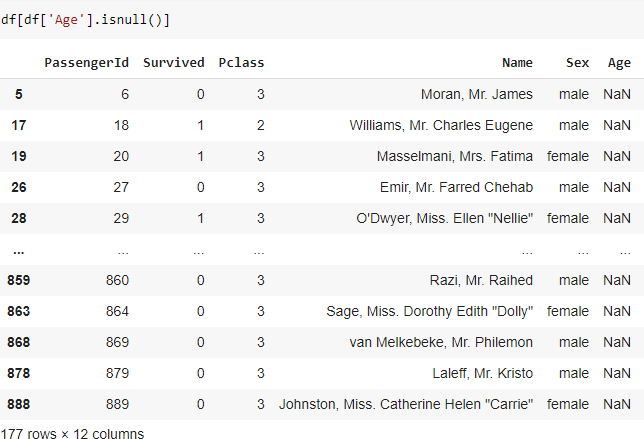
**Basic statistics operations**



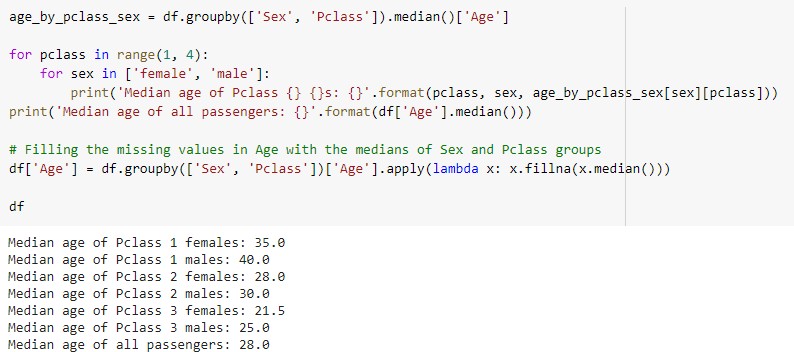
# Correlation between columns



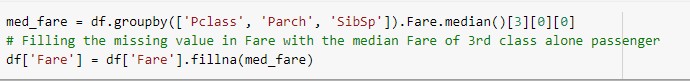
**Missing rows for the values in age.**



# Filling the missing values in age with median of corresponding data values in pclass and sex.



**Filling missing values in fare with median of datas corresponding to pclass Sibsp and Parch columns.**



# Experiment No: 2

**INTRODUCTION**

In this experiment, we have to perform Linear regression in one variable and multiple variable on the Real estate dataset.

**Simple Linear regression (SLR):** Linear regression assumes a linear or straight line relationship between the input variables (X) and the single output variable (y) is called simple linear regression.

1 y = b0 + b1 \* x

# For estimating the coefficient,

B1 = sum((x(i) - mean(x)) \* (y(i) - mean(y))) / sum( (x(i) - mean(x))^2 ) B0 = mean(y) - B1 \* mean(x)

**Multiple linear regression (MLR):**

also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the [linear relationship](https://www.investopedia.com/terms/l/linearrelationship.asp) between the explanatory (independent) variables and response (dependent) variable.

Formula and Calculation of Multiple Linear Regression

*yi*=*β*0+*β*1*xi*1+*β*2*xi*2+...+*βpxip*+*ϵ*

# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform Linear regression in single and multiple variables on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib and seaborn – for data visualization
4. Sklearn.model\_selection – train test data split
5. Sklearn.metrics – mean square error and r2 score.
6. Sklearn,linear\_model – for linear regression

**STEP-3:** Loading the dataset using read\_csv method in pandas module.

**STEP-4:** Analyze the dataset using info method, which gives its data types and number of non- null values in each columns.

**STEP-5:** Perform basic statistic operation using describe() method.

**STEP-6:** Use heatmaps, correlation matrix, regression plots and pairplots in seaborn to find the relationship between features.

**STEP-7:** Implement Simple Linear regression(singleLR) with only one variable

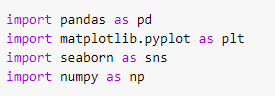
(X3 distance to the nearest MRT station) and calculate the MSE and R2 score for the singleLR model.

**STEP-8:** Implement Multiple Linear regression(multiLR) with selected variable (refined cols) which are pick out by analyzing the relationship between features and calculate the MSE and R2 score for the multiLR model.

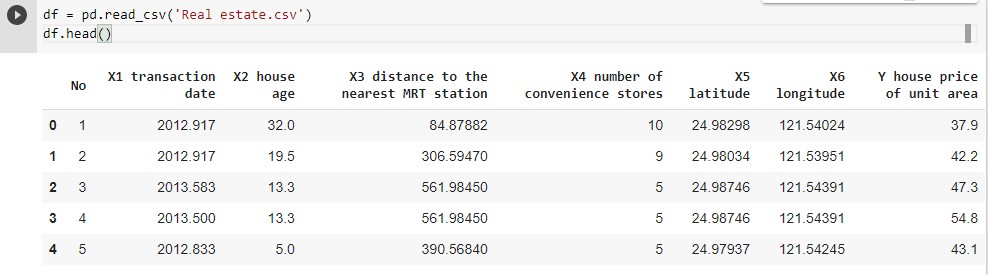
**STEP-9:** Stop the program.

# PROGRAM:

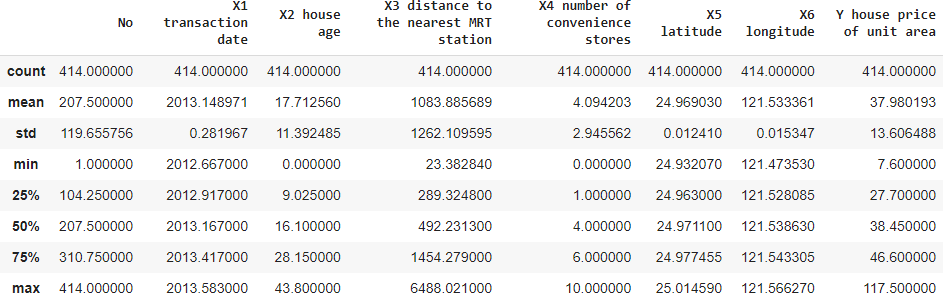
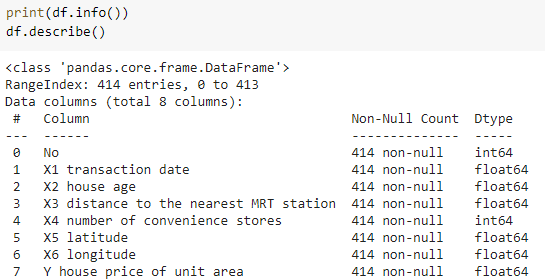
**Importing libraries**



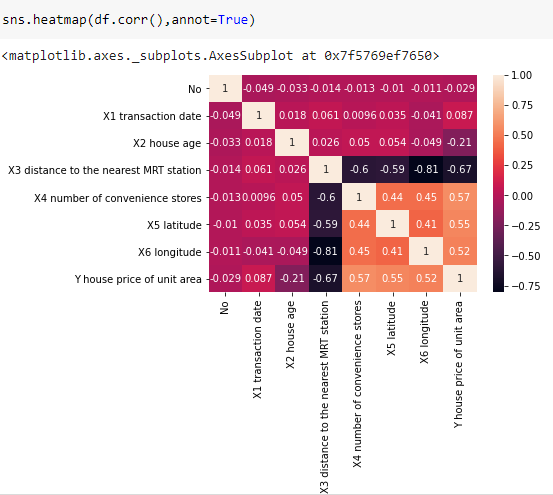
# Loading dataset



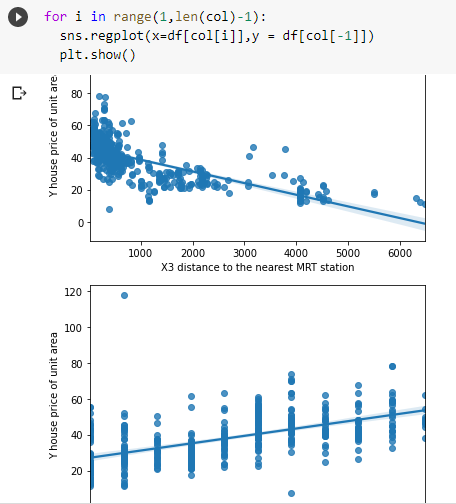
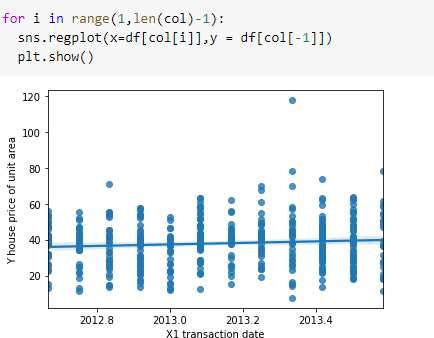
**Basic statistics operations**

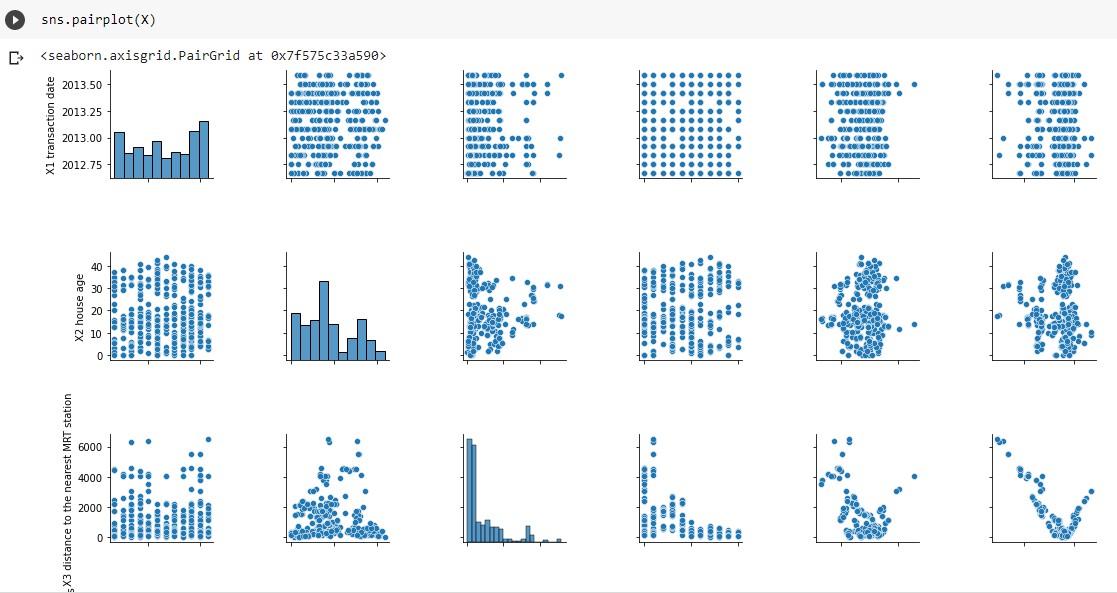


# Correlation between columns



**Regression plots and pairplots.**

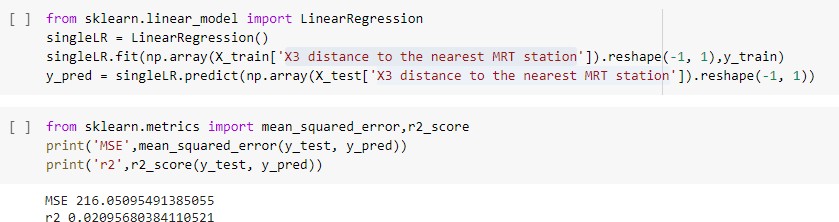




# Train test split.



**Linear regression in one variable:**



# Multiple Linear regression.

**Experiment No: 3**

# INTRODUCTION

In this experiment, we have to perform Logistic regression on the covid dataset.

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) that in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binar](https://en.wikipedia.org/wiki/Binary_variable)y [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), although many more complex [extensions](https://en.wikipedia.org/wiki/Logistic_regression#Extensions) exist. In [regression](https://en.wikipedia.org/wiki/Regression_analysis) [analysis](https://en.wikipedia.org/wiki/Regression_analysis), **logistic regression** (or **logit regression**) is [estimating](https://en.wikipedia.org/wiki/Estimation_theory) the parameters of a logistic model (a form of [binary regression](https://en.wikipedia.org/wiki/Binary_regression)).

# Linear Regression Equation:



Where, y is dependent variable and x1, x2 ... and Xn are explanatory variables.

# Sigmoid Function:

**Apply Sigmoid function on linear regression:**



# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform Logistic regression on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib and seaborn – for data visualization
4. Sklearn.model\_selection – train test data split
5. Sklearn.metrics – f1 score.
6. Sklearn,linear\_model – for logistic regression

**STEP-3:** Loading the dataset using read\_csv method in pandas module.

**STEP-4:** Analyze the dataset using info method, which gives its data types and number of non- null values in each columns.

**STEP-5:** Perform basic statistic operation using describe() method.

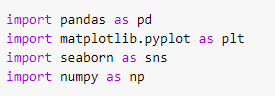
**STEP-6:** Use heatmaps, correlation matrix, regression plots and pairplots in seaborn to find the relationship between features.

**STEP-6:** Implement Logistics regression(logreg) with all variable and calculate the f1 score.

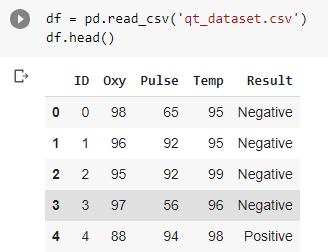
**STEP-8:** Stop the program.

# PROGRAM:

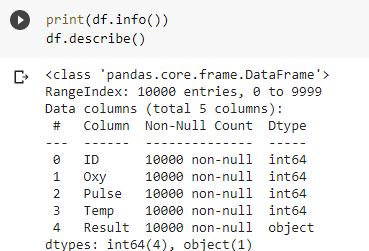
**Importing libraries**

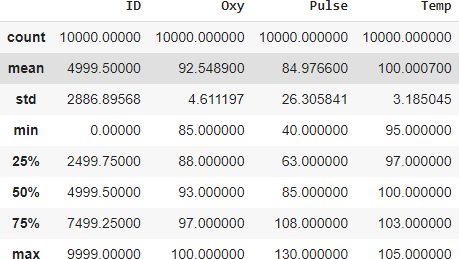


# Loading dataset

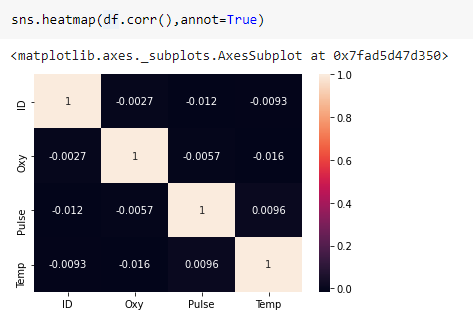


**Basic statistics operations**

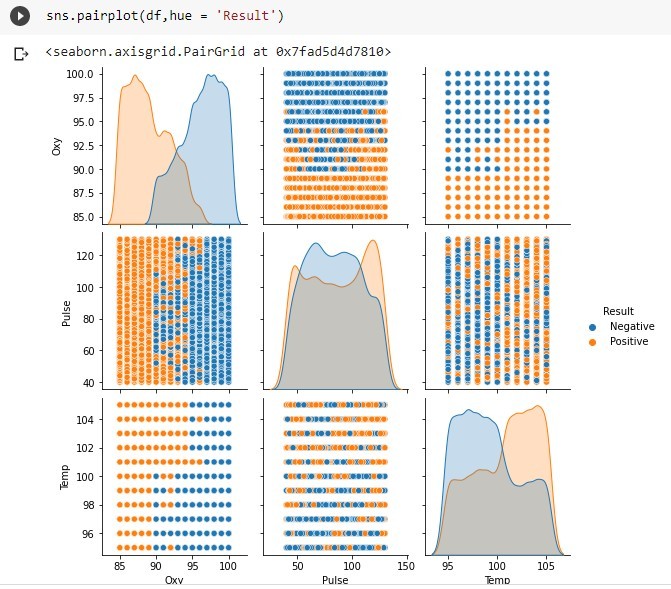




# Correlation between columns



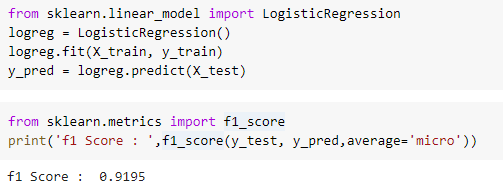
**Pairplots.**



# Train test split.



**Logistic regression :**



# Experiment No: 4

**INTRODUCTION**

In this experiment, we have to perform Support vector machine on the cancer dataset.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

# Types of SVM

**SVM can be of two types:**

* **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non- linear data and classifier used is called as Non-linear SVM classifier.

# Support Vectors:

* The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform Support vector Machine on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib and seaborn – for data visualization
4. Sklearn.model\_selection – train test data split
5. Sklearn.metrics –confusion matrix and classification report.
6. Sklearn,svm– for support vector regression
7. Sklearn.decomposition – for PCA
8. Sklearn.preprocessing – for Normalisation

**STEP-3:** Loading the dataset using read\_csv method in pandas module.

**STEP-4:** Analyze the dataset using info method, which gives its data types and number of non- null values in each columns.

**STEP-5:** Perform basic statistic operation using describe() method.

**STEP-6:** Use heatmaps, correlation matrix, regression plots and pairplots in seaborn to find the relationship between features.

**STEP-7**: Normalize the data points

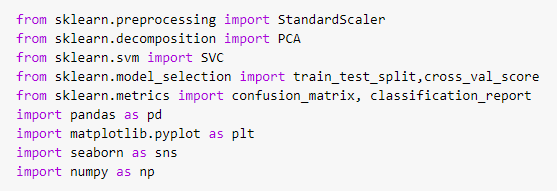
**STEP-8**: Using selective feature, perform PCA in order to reduce number of feature from 30 to 11.

**STEP-9:** Implement SVM with 11 PCA variable and calculate classification report and confusion matrix.

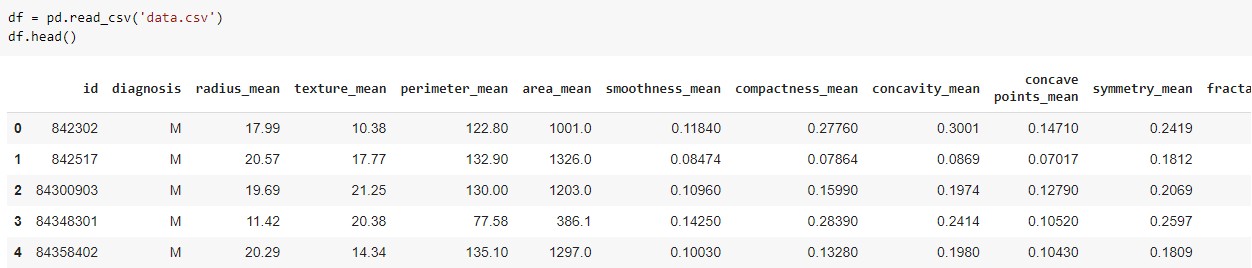
**STEP-10:** Stop the program.

# PROGRAM:

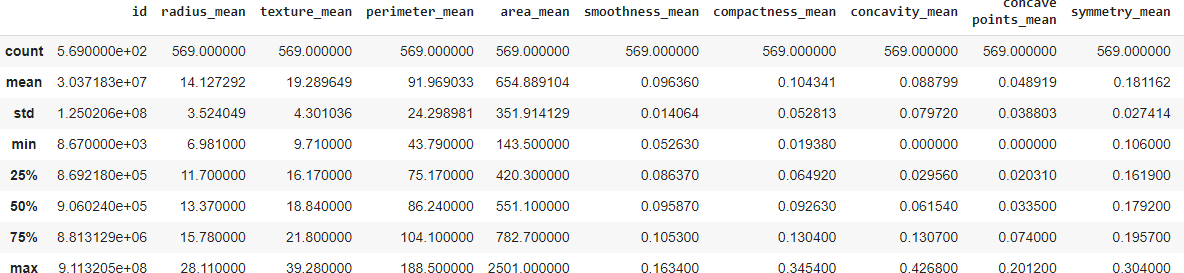
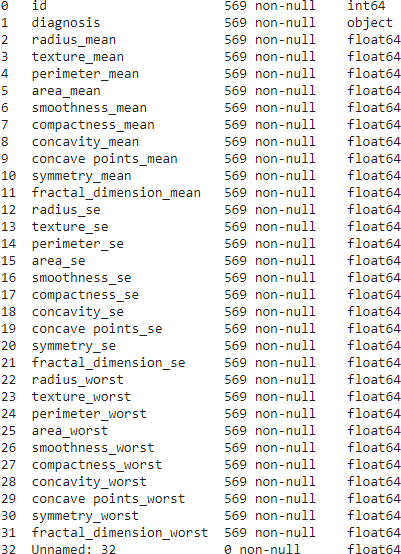
**Importing libraries**

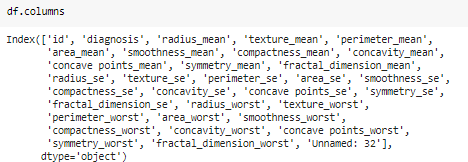


# Loading dataset

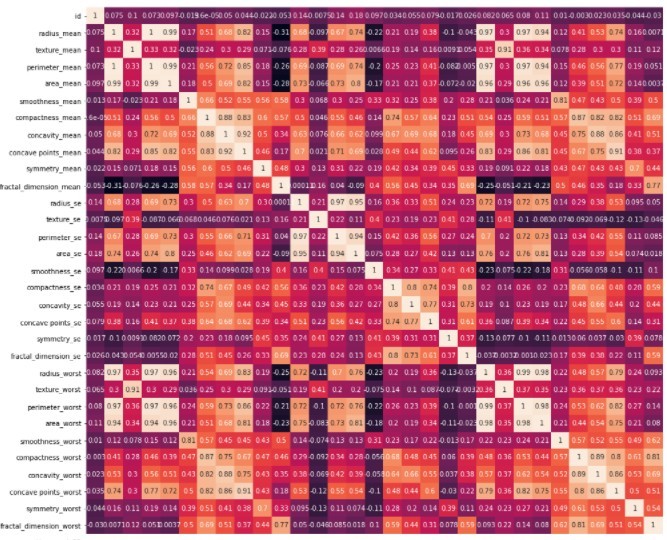


**Basic statistics operations**

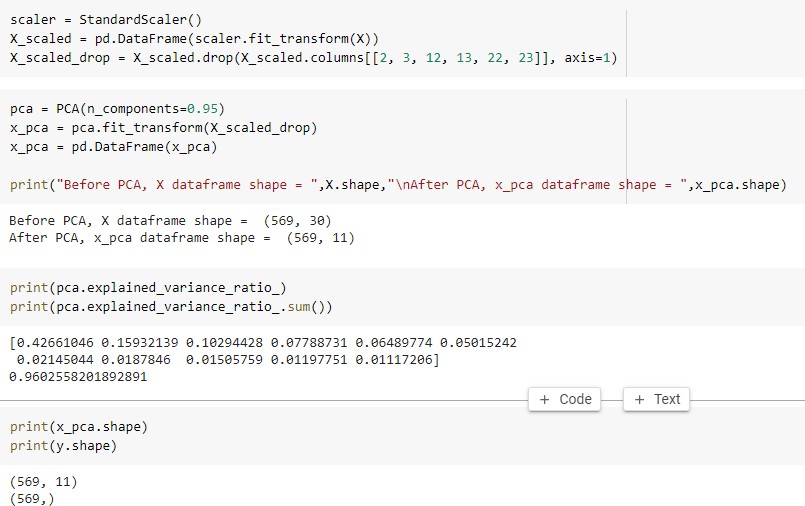




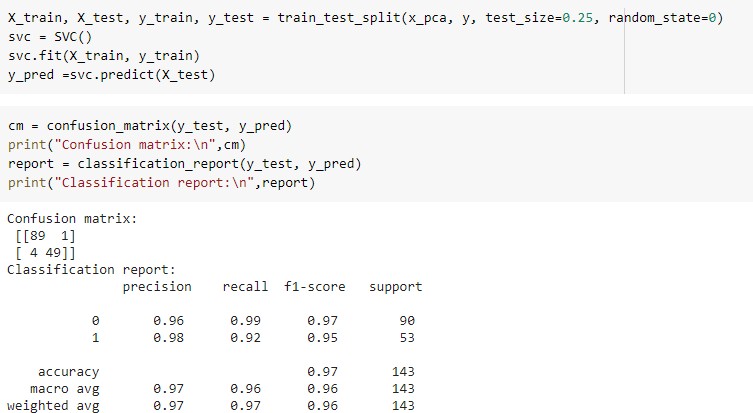
# Correlation between columns



**Normalization and PCA.**



# Train test split, SVM model and model evaluation:



**Experiment No: 5**

# INTRODUCTION

In this experiment, we have to perform k nearest neighbor on the iris dataset. The K-NN working can be explained on the basis of the below algorithm:

* **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.

# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform K- nearest neighbor on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib and seaborn – for data visualization
4. Sklearn.model\_selection – train test data split and cross\_val\_score
5. Sklearn.metrics – model evaluation.
6. Sklearn.datasets – For iris dataset
7. Sklearn.neighbor – For KNeighborsClassifier

**STEP-3:** Loading the dataset using load\_iris method in sklearn.datasets module.

**STEP-4:** Analyze the dataset using info method, which gives its data types and number of non- null values in each columns.

**STEP-5:** Perform basic statistic operation using describe() method.

**STEP-6:** Use heatmaps, correlation matrix, regression plots and pairplots in seaborn to find the relationship between features.

**STEP-7:** Implement KNeighborClassifier with k value ranging from 1 to 25 and save the accuracy score of test dataset for each k value in a score list.

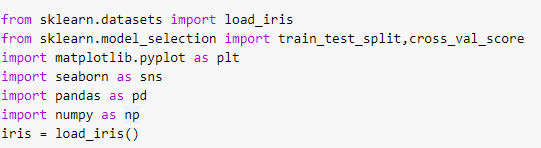
**STEP-8:** Plot the accuracy\_score in y axis and k value in x axis, find out the k value which gives high accuracy on test data.

**STEP-9:** Do the step 7 and 8 for 10-fold validation set.

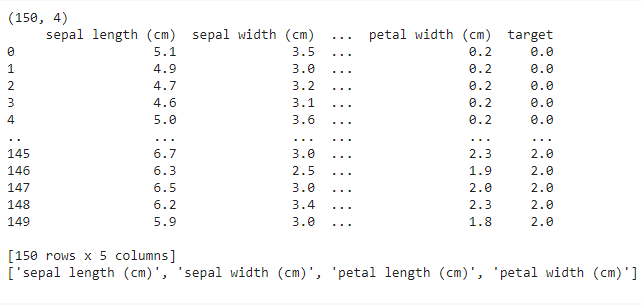
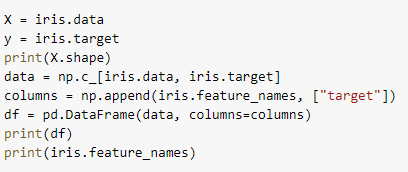
**STEP-10:** Conclude the best k value which works good in both test and validation set. **STEP-11:** Use that K value to build the final KNN model and print the accuracy\_score. **STEP-12:** Stop the Program.

# PROGRAM:

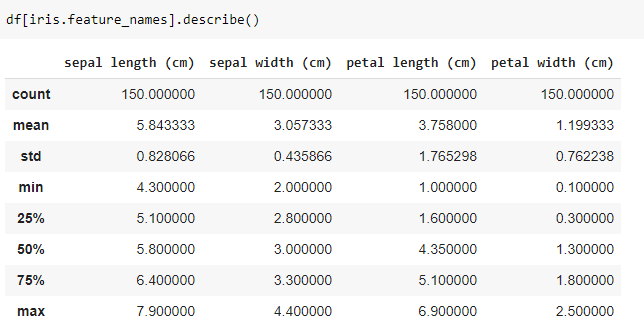
**Importing libraries and load the data**



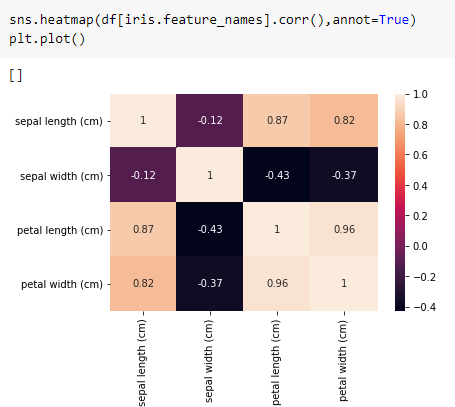
# Split the data into independent and dependent variable



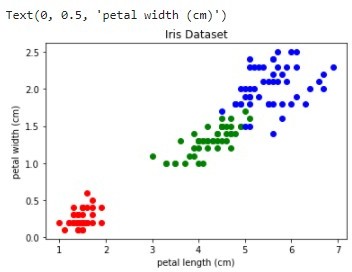
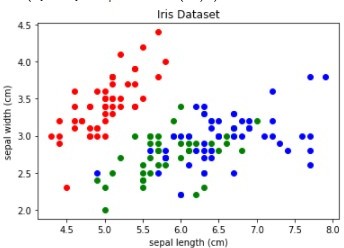
**Basic statistics operations**

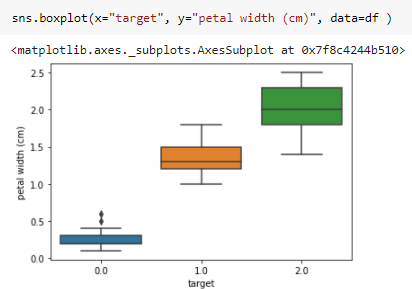
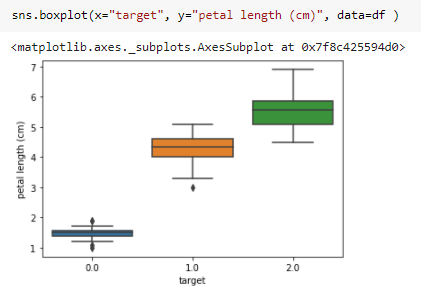


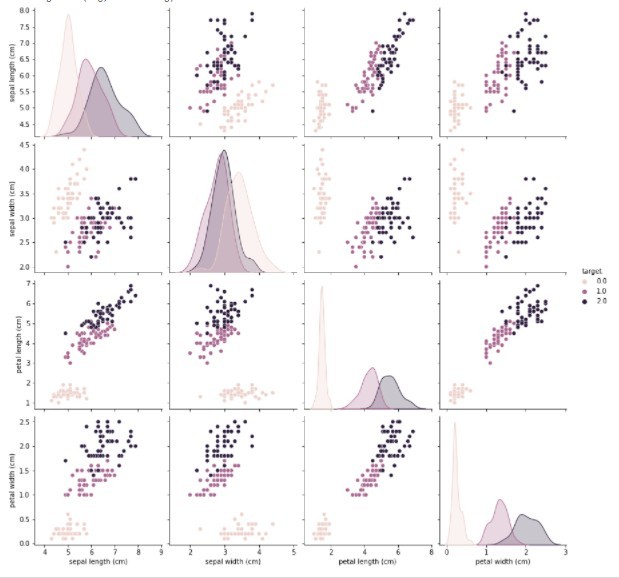
# Correlation between columns



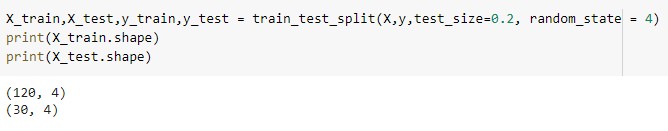
**Scatter plots, Boxplot and pairplots.**



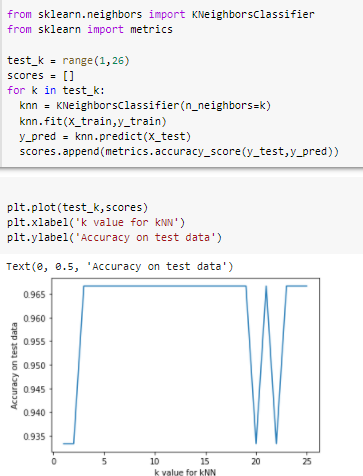




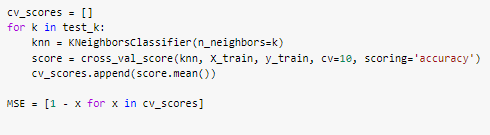
# Train test split.

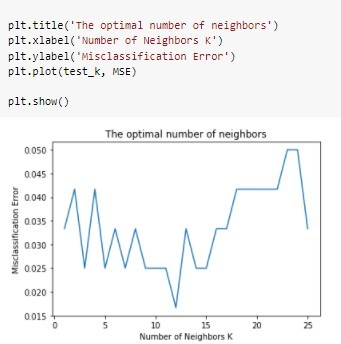


**Finding best k value for test dataset:**

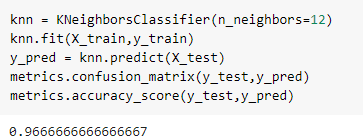


# Finding best k value for 10-fold validation dataset.





**Build the KNN model with best k value:**



# Experiment No: 6

**INTRODUCTION**

In this experiment, we have to perform logistic regression along with PCA on the cancer dataset.

**Principal Component Analyis** is basically a statistical procedure to convert a set of observation of possibly correlated variables into a set of values of linearly uncorrelated variables.

Each of the principal components is chosen in such a way so that it would describe most of the still available variance and all these principal components are orthogonal to each other. In all principal components first principal component has maximum variance.

**Uses of PCA:**

* It is used to find inter-relation between variables in the data.
* It is used to interpret and visualize data.
* As number of variables are decreasing it makes further analysis simpler.
* It’s often used to visualize genetic distance and relatedness between populations.

These are basically performed on square symmetric matrix. It can be a pure sums of squares and cross products matrix or Covariance matrix or Correlation matrix. A correlation matrix is used if the individual variance differs much.

# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform logistic regression along with PCA on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib and seaborn – for data visualization
4. Sklearn.model\_selection – train test data split
5. Sklearn.metrics –f1 score.
6. Sklearn,linear\_model– for logistic regression
7. Sklearn.decomposition – for PCA
8. Sklearn.preprocessing – for Normalisation

**STEP-3:** Loading the dataset using read\_csv method in pandas module.

**STEP-4:** Analyze the dataset using info method, which gives its data types and number of non- null values in each columns.

**STEP-5:** Perform basic statistic operation using describe() method.

**STEP-6:** Use heatmaps, correlation matrix, regression plots and pairplots in seaborn to find the relationship between features.

**STEP-7**: Normalize the data points

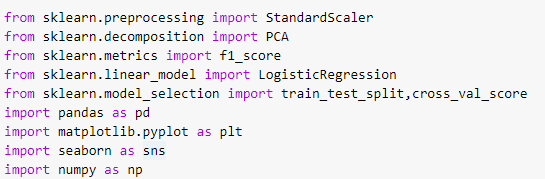
**STEP-8**: Using selective feature, perform PCA in order to reduce number of feature from 30 to 11.

**STEP-9:** Implement logistic regression with 11 PCA variable and calculate f1 score.

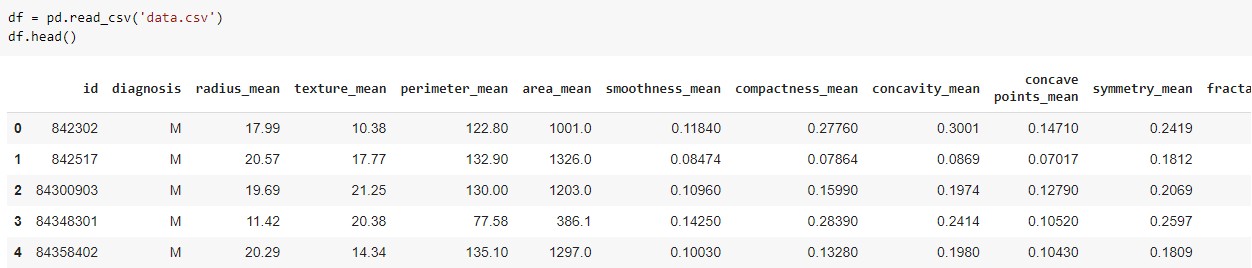
**STEP-10:** Stop the program.

# PROGRAM:

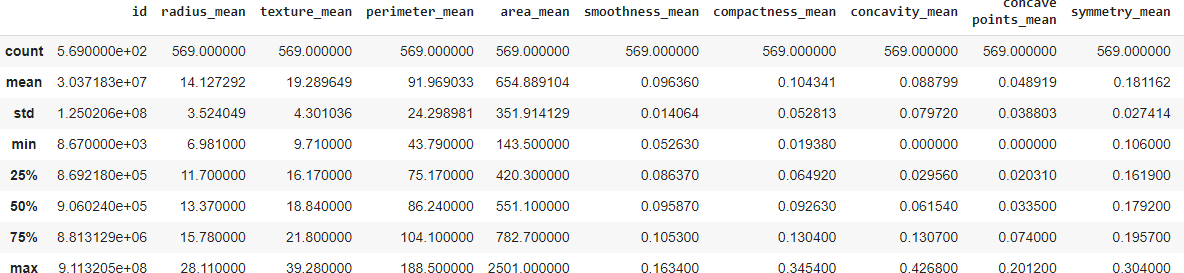
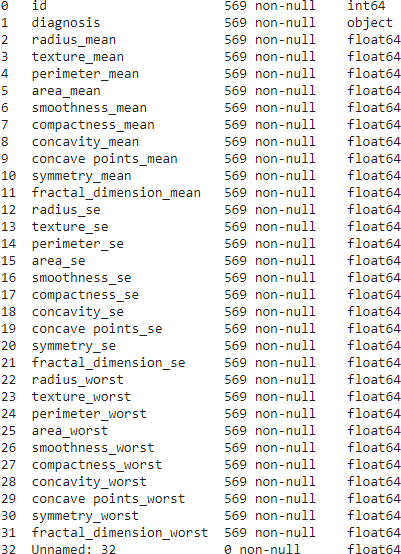
**Importing libraries**

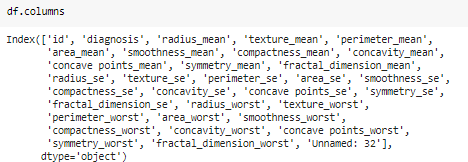


# Loading dataset

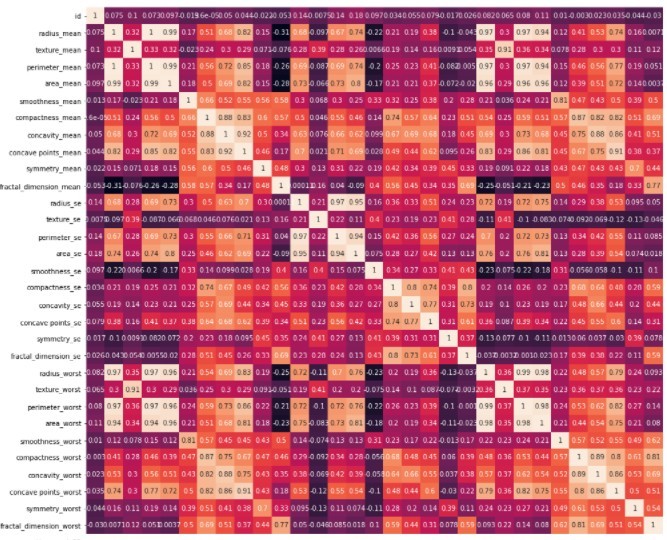


**Basic statistics operations**

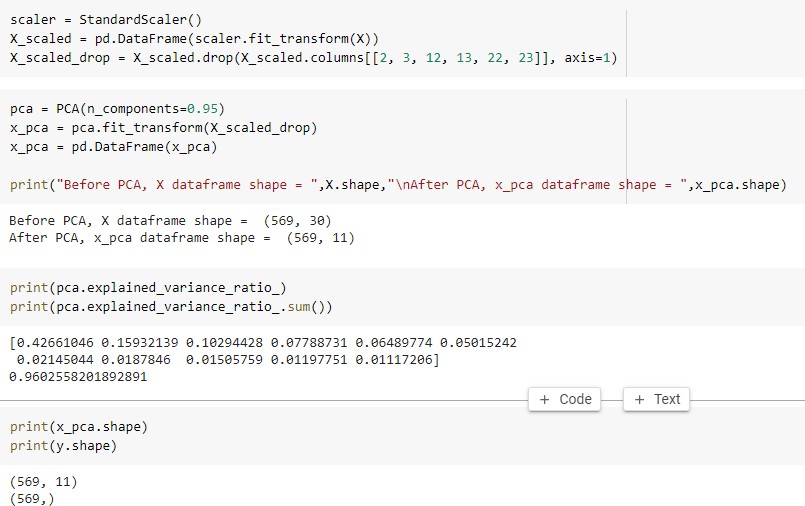




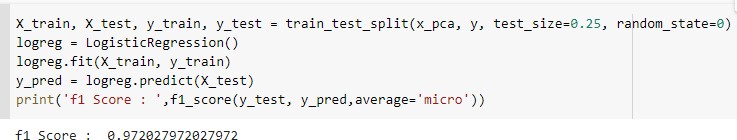
# Correlation between columns



**Normalization and PCA.**



# Train test split, logistic model and model evaluation:

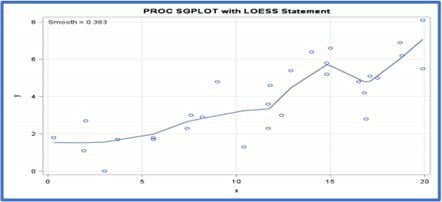


# Experiment No: 8

**INTRODUCTION**

In this experiment, we have to perform non parametric locally weighted regression on the restaurant dataset.

Loess regression is a nonparametric technique that uses local weighted regression to fit a smooth curve through points in a scatter plot.



# OBJECTIVE OF THE EXERCISE/EXPERIMENT

To perform non parametric locally weighted regression on the given dataset, using scikit library

# STEP 2: ACQUISITION PROCEDURE:

**STEP-1:** Start the program.

**STEP-2:** import all the necessary libraries

1. Numpy – array manipulation
2. Pandas – dataframe manipulation
3. Matplotlib– for data visualization

**STEP-3:** Loading the dataset using read\_csv method in pandas module.

**STEP-4:** Read the Given data Sample to X and the curve (linear or non linear) to Y

**STEP-5:** Set the value for Smoothening parameter or Free parameter say τ

**STEP-6:** Set the bias /Point of interest set x0 which is a subset of X

**STEP-7:** Determine the weight matrix using :



**STEP-8:** Determine the value of model term parameter β using:



**STEP-9:** Prediction = x0\*β

**STEP-10:** Stop the program.

# PROGRAM:

import matplotlib.pyplot as plt import pandas as pd

import numpy as np

def kernel(point, xmat, k): m,n = np.shape(xmat)

weights = np.mat(np.eye((m))) for j in range(m):

diff = point - X[j]

weights[j,j] = np.exp(diff\*diff.T/(-2.0\*k\*\*2)) return weights

def localWeight(point, xmat, ymat, k): wei = kernel(point,xmat,k)

W = (X.T\*(wei\*X)).I\*(X.T\*(wei\*ymat.T)) return W

def localWeightRegression(xmat, ymat, k): m,n = np.shape(xmat)

ypred = np.zeros(m) for i in range(m):

ypred[i] = xmat[i]\*localWeight(xmat[i],xmat,ymat,k) return ypred

data = pd.read\_csv('10-dataset.csv') bill = np.array(data.total\_bill)

tip = np.array(data.tip)

mbill = np.mat(bill) mtip = np.mat(tip)

m= np.shape(mbill)[1] one = np.mat(np.ones(m))

X = np.hstack((one.T,mbill.T))

ypred = localWeightRegression(X,mtip,0.5) SortIndex = X[:,1].argsort(0)

xsort = X[SortIndex][:,0]

fig = plt.figure()

ax = fig.add\_subplot(1,1,1) ax.scatter(bill,tip, color='green')

ax.plot(xsort[:,1],ypred[SortIndex], color = 'red', linewidth=5) plt.xlabel('Total bill')

plt.ylabel('Tip') plt.show();

# Output:

