Individual Report

- Sanjana S Godolkar

- ## Dataset description

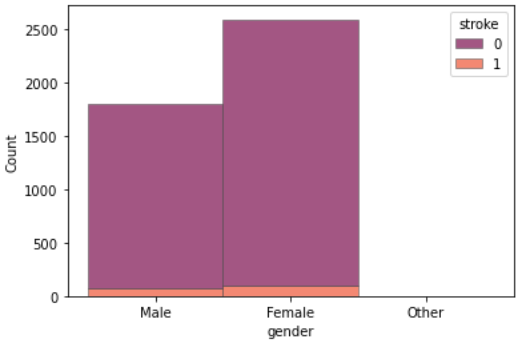
  The "healthcare-dataset-stroke-data" is a stroke prediction dataset from Kaggle that contains 5,110 observations (rows) with 12 attributes (columns). Each observation corresponds to one patient, and the attributes represent variables related to the health status of each patient.

  Dataset Description:

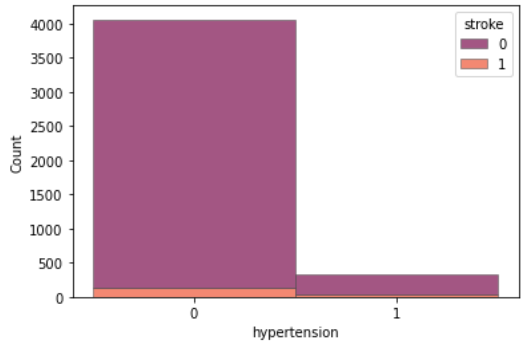
  1. ID: Patient identifier
  2. Gender: Patient's gender (Male, Female, or Other)
  3. Age: Patient's age (quantitative variable)
  4. Hypertension: Presence of hypertension (0 = no, 1 = yes; categorical variable)
  5. Heart Disease: Presence of heart disease (0 = no, 1 = yes; categorical variable)
  6. Ever Married: Marital status (categorical variable)
  7. Work Type: Patient's occupation (children, government job, never worked, private, self-employed; categorical variable)
  8. Residence Type: Patient's residence (urban, rural; categorical variable)
  9. Avg. Glucose Level: Average glucose level in the blood (quantitative variable)
  10. BMI: Body Mass Index (quantitative variable)
  11. Smoking Status: Smoking habits (formerly smoked, never smoked, smokes; categorical variable)
  12. Stroke: Stroke history (0 = no stroke risk, 1 = stroke risk; target variable)

- ## EDA

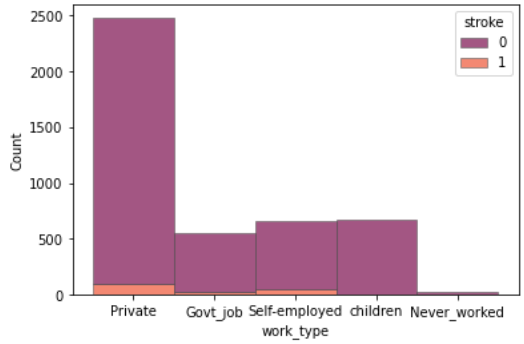
  Stacked Histogram of Gender and Stroke



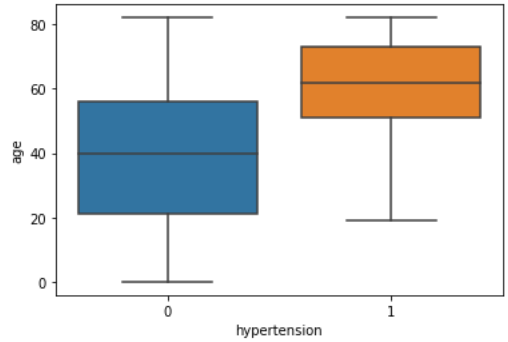
Stacked Histogram of hypertension and Stroke



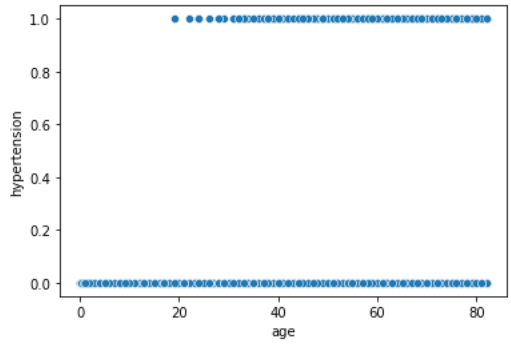
Stacked Histogram of Work type and Stroke



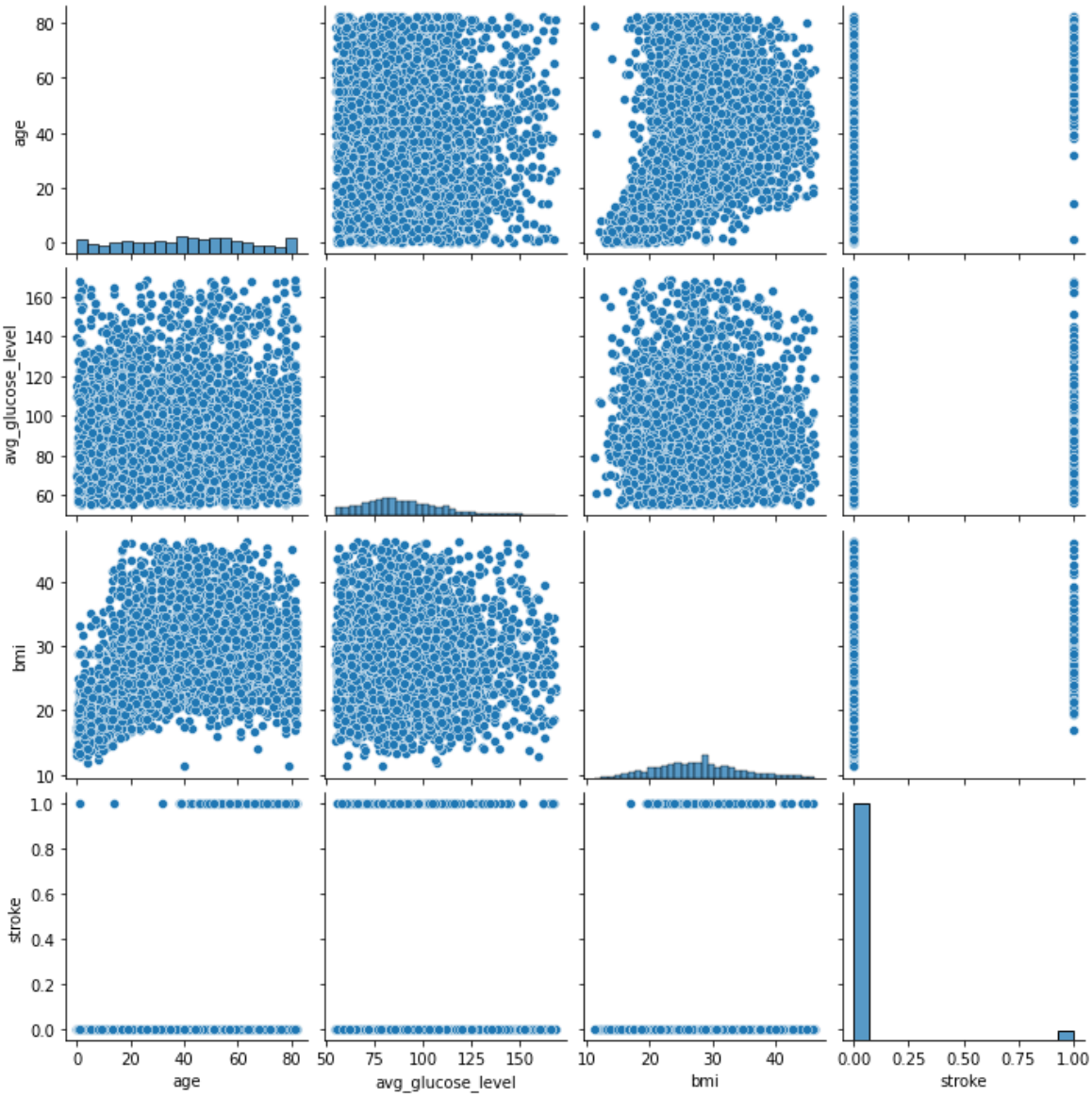
Box Plot of hypertension vs age



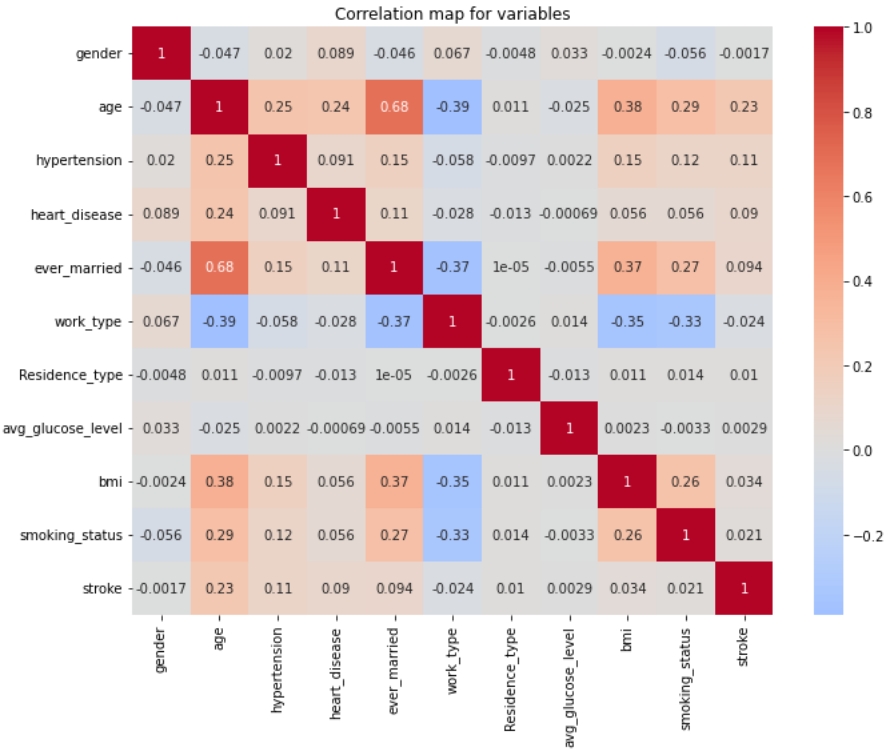
Scatter plot of Hypertension vs age



Pair plot for age, avg glucose level, bmi and stroke


age
avg_glucose_level
bmi
stroke
0
20
40
60
80
50
75
100
125
150
10
20
30
40
0.00
0.25
0.50
0.75
1.00
0.0
0.2
0.4
0.6
0.8
1.0
120
140
160

Heat Map



Feature Engineering

Based on the heat map above, obtained by encoding we decided to go ahead with the following features:

['age', 'hypertension_0', 'hypertension_1', 'heart_disease_0', 'heart_disease_1', 'stroke']]

Models:

Logistic Regression

The logistic function, also known as the sigmoid function, is an S-shaped curve that maps any real-valued number to a value between 0 and 1. It is given by the formula:

f(x) = 1 / (1 + e^(-x))

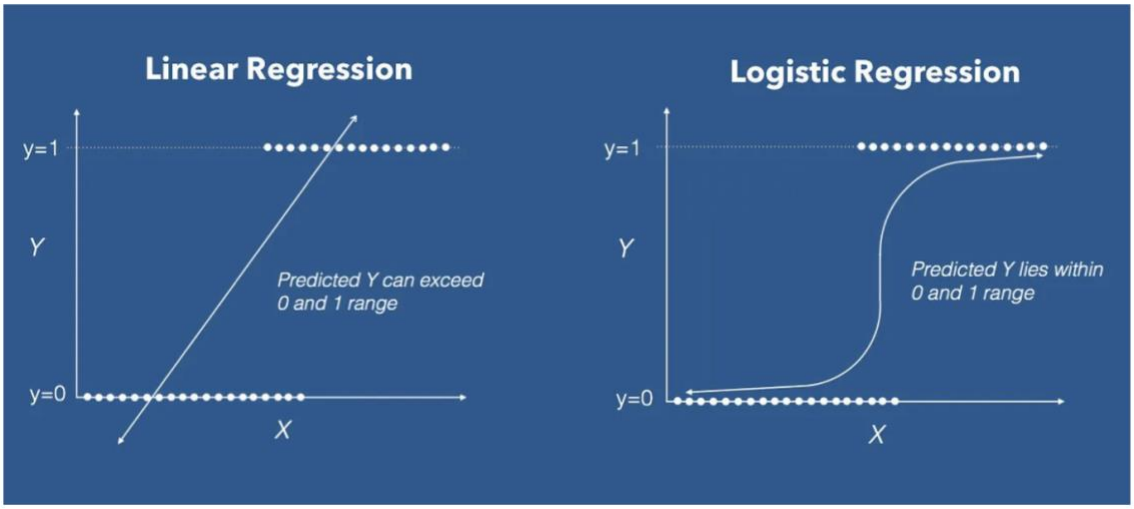
where x is the input, e is the base of the natural logarithm, and f(x) is the output of the logistic function.

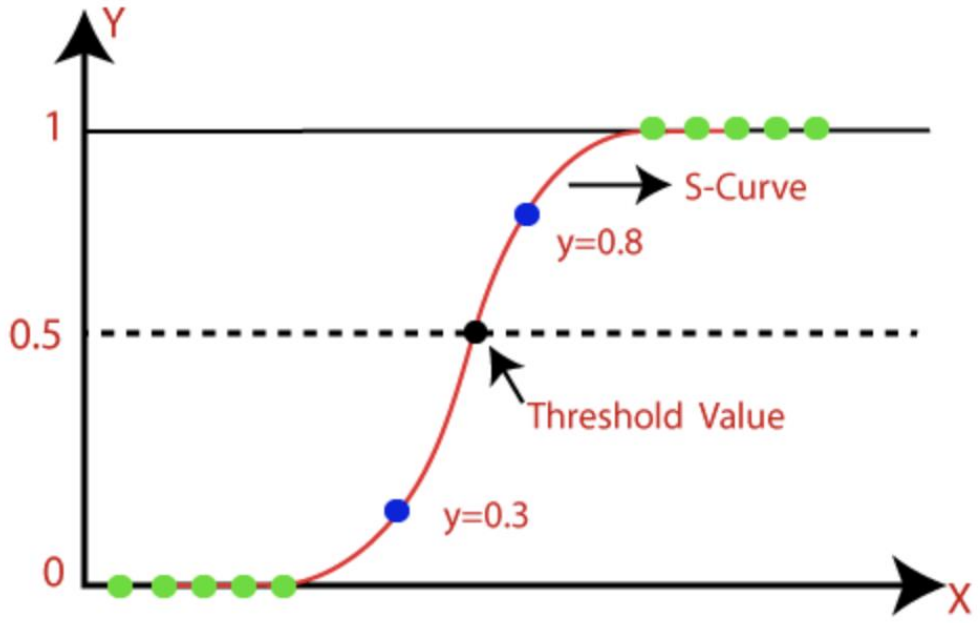
The logistic function is used in logistic regression to model the relationship between the independent variables and the probability of a certain outcome. The output of the logistic function represents the probability that an observation belongs to a particular class (e.g., 1 or 0, Yes or No, True or False).

In logistic regression, the linear combination of input features (independent variables) is passed through the logistic function to produce a probability value. The equation can be written as:

p(y=1) = 1 / (1 + e^(-(b0 + b1 * x1 + b2 * x2 + ... + bn * xn)))

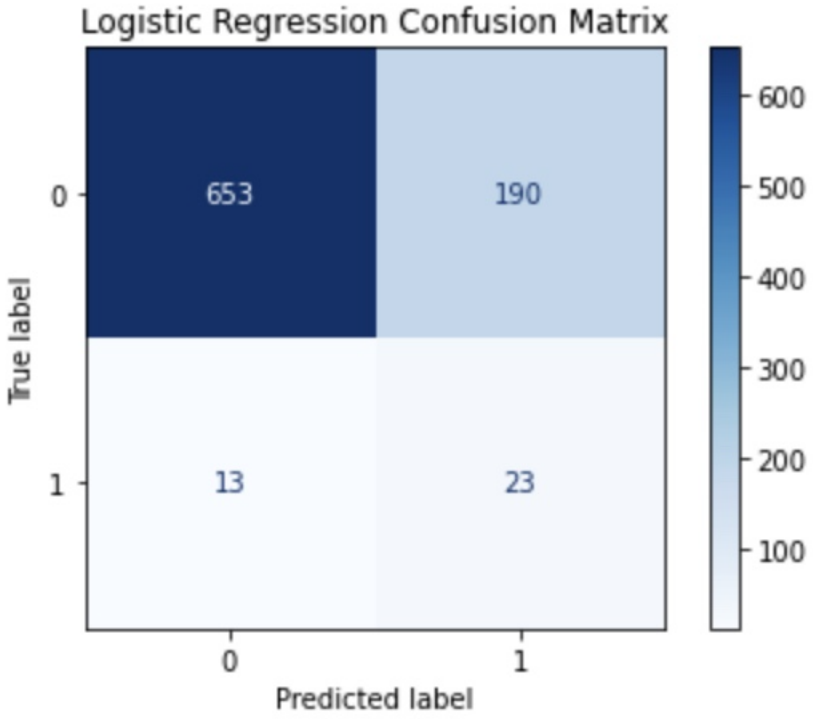
where p(y=1) is the probability of the dependent variable being equal to 1, b0 is the bias term or the y-intercept, b1, b2, ..., bn are the coefficients of the independent variables x1, x2, ..., xn and e is the base of the natural logarithm.
The logistic regression model calculates the best-fitting coefficients (b0, b1, b2, ..., bn) using maximum likelihood estimation, which aims to maximize the likelihood that the observed data comes from the modeled distribution.
By applying a threshold to the output probabilities, logistic regression can be used to classify observations into two classes. For example, if the probability output is greater than 0.5, the observation can be classified as class 1; otherwise, it is classified as class 0.
Logistic regression is widely used in various applications, such as spam detection, medical diagnosis, and marketing campaign optimization. Its ability to provide probabilities and classify new data using continuous and discrete datasets makes it an essential machine-learning algorithm.

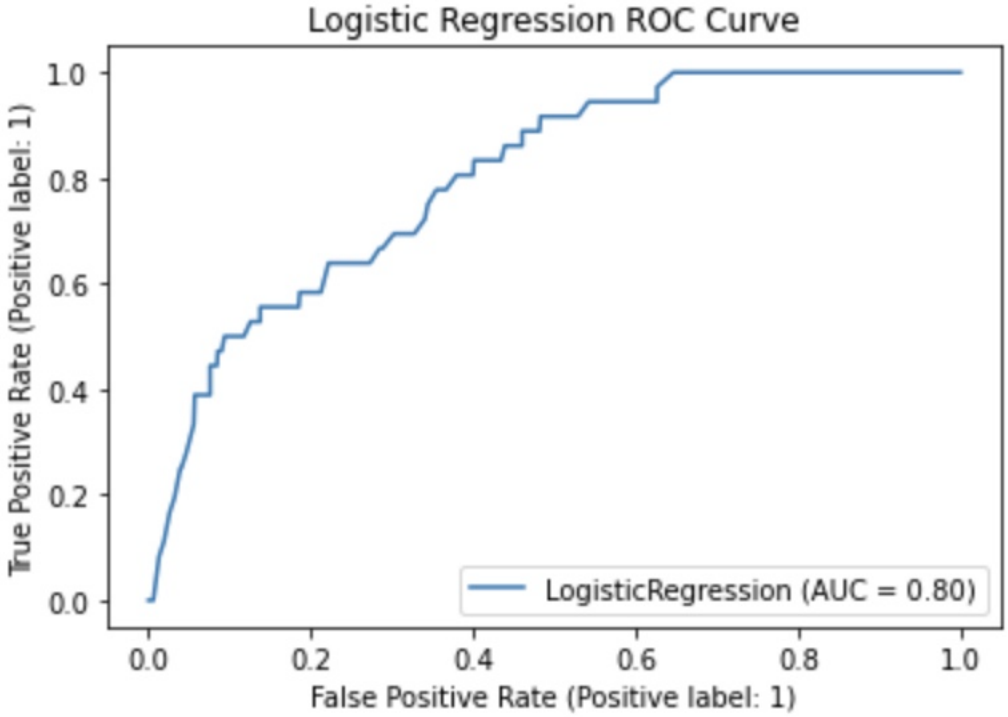




The result of our project for Logistic Regression

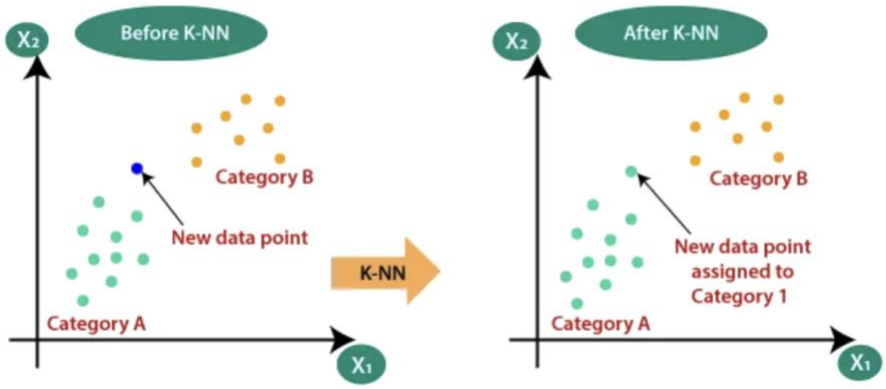
```
Logistic Regression Accuracy: 0.7690557451649602
[[653 190]
 [ 13  23]]
              precision    recall  f1-score   support

           0       0.98      0.77      0.87       843
           1       0.11      0.64      0.18        36

    accuracy                           0.77       879
   macro avg       0.54      0.71      0.53       879
weighted avg       0.94      0.77      0.84       879
```


Logistic Regression Confusion Matrix
653
190
13
23
0
1
True label
0
1
Predicted label
600
500
400
300
200
100

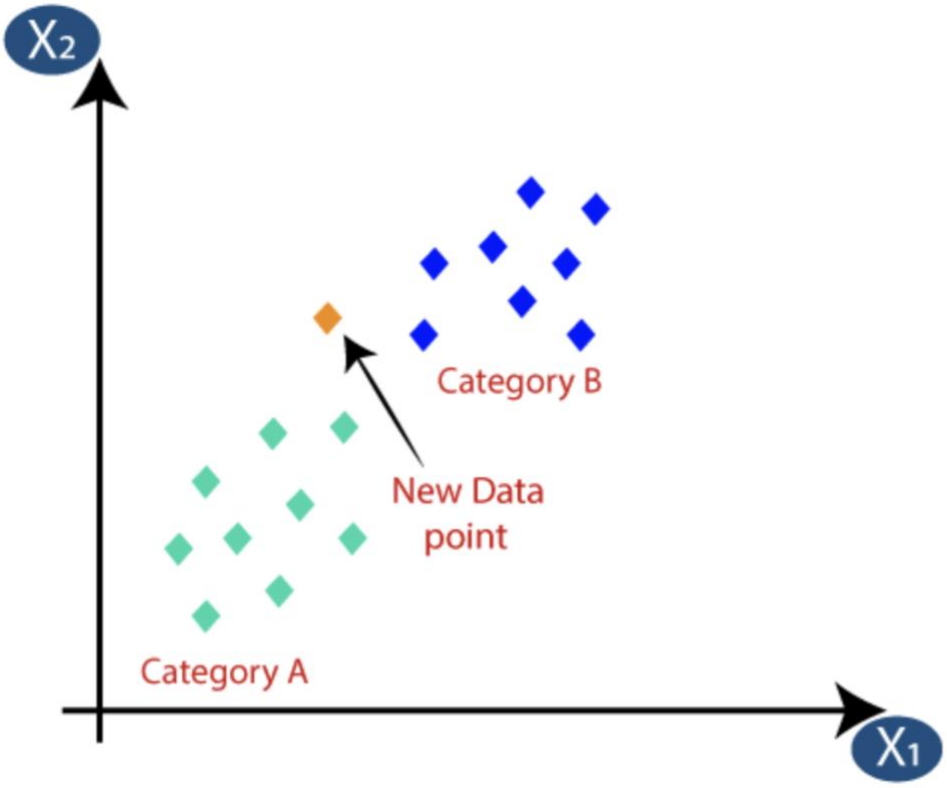

Logistic Regression ROC Curve
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
LogisticRegression (AUC = 0.80)
0.0
0.2
0.4
0.6
0.8
1.0

K-Nearest Neighbors

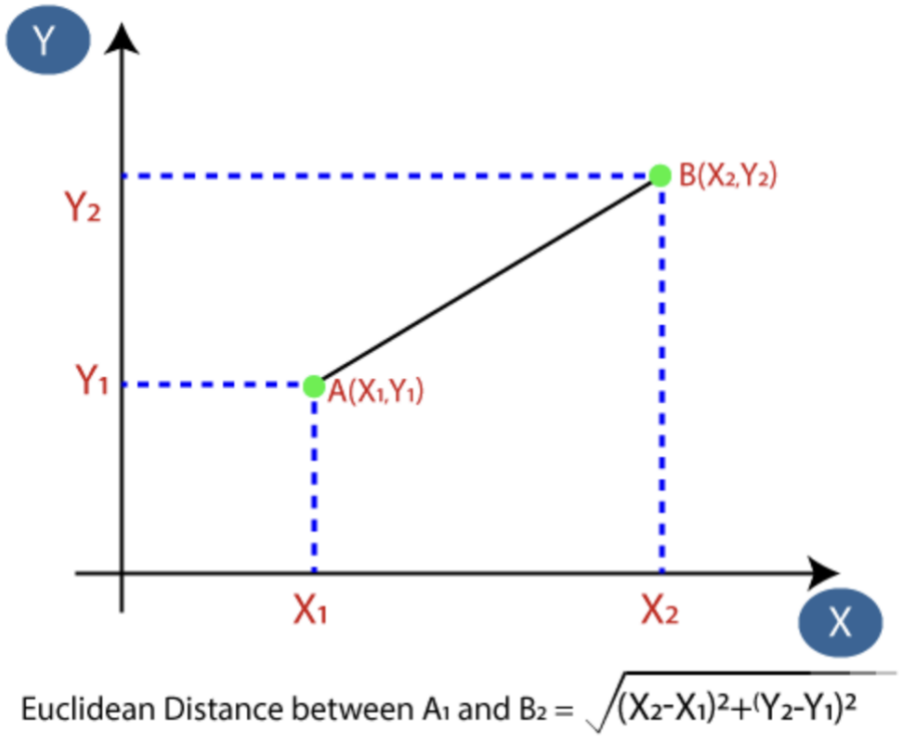
K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points which is closest to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and the class that holds the highest probability will be selected. In the case of regression, the value is the mean of the 'K' selected training points.Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories? To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



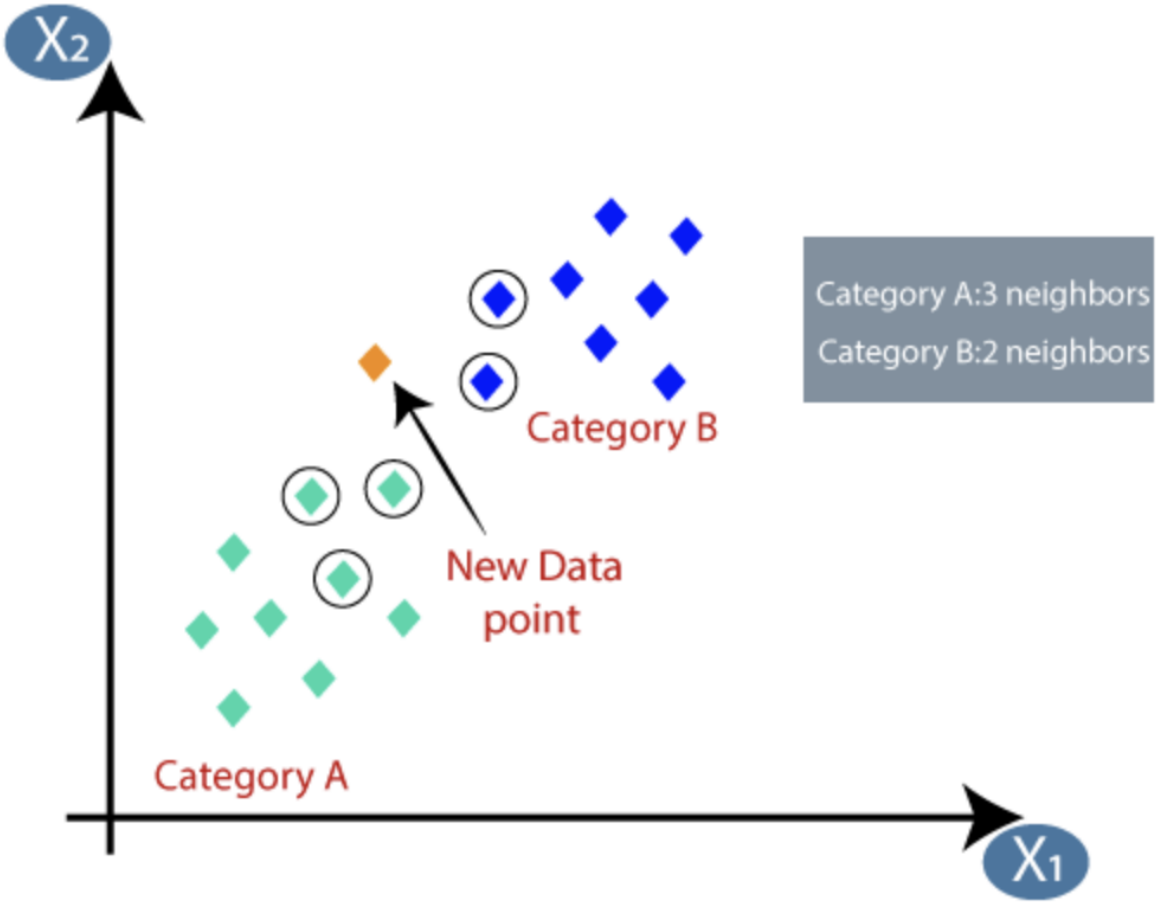
Suppose we have a new data point, and we need to put it in the required category. Consider the below image:



Firstly, we will choose the number of neighbors, so we will choose the k=5.
Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



By calculating the Euclidean distance, we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



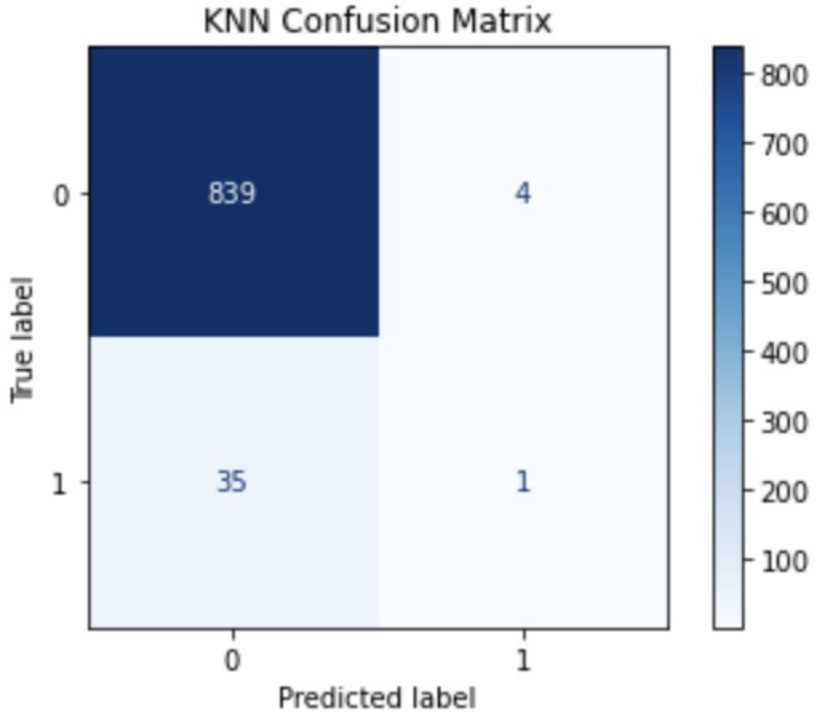
As we can see the 3 nearest neighbors are from Category A, hence this new data point must belong to Category A.

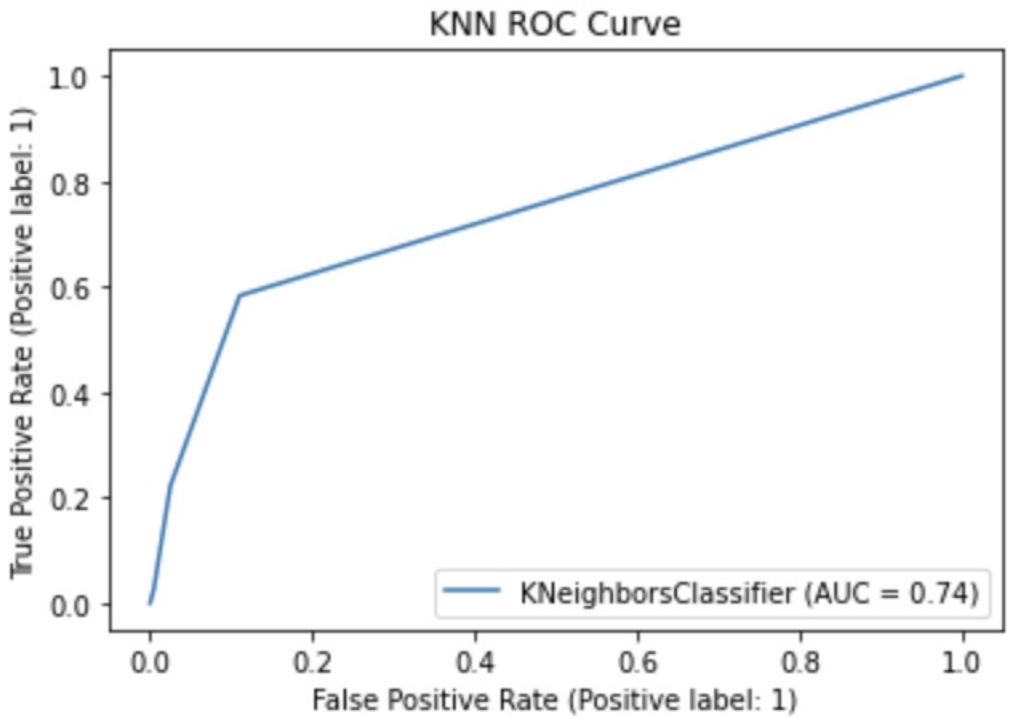
Selecting the value of K in the K-NN algorithm:

1. There is no way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
2. A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model. Large values for K are good.

The result of our project for KNN

```
K-Nearest Neighbors Accuracy: 0.9556313993174061
[[839   4]
 [ 35   1]]
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       843
           1       0.20      0.03      0.05        36

    accuracy                           0.96       879
   macro avg       0.58      0.51      0.51       879
weighted avg       0.93      0.96      0.94       879
```





4.6. Support vector Machine

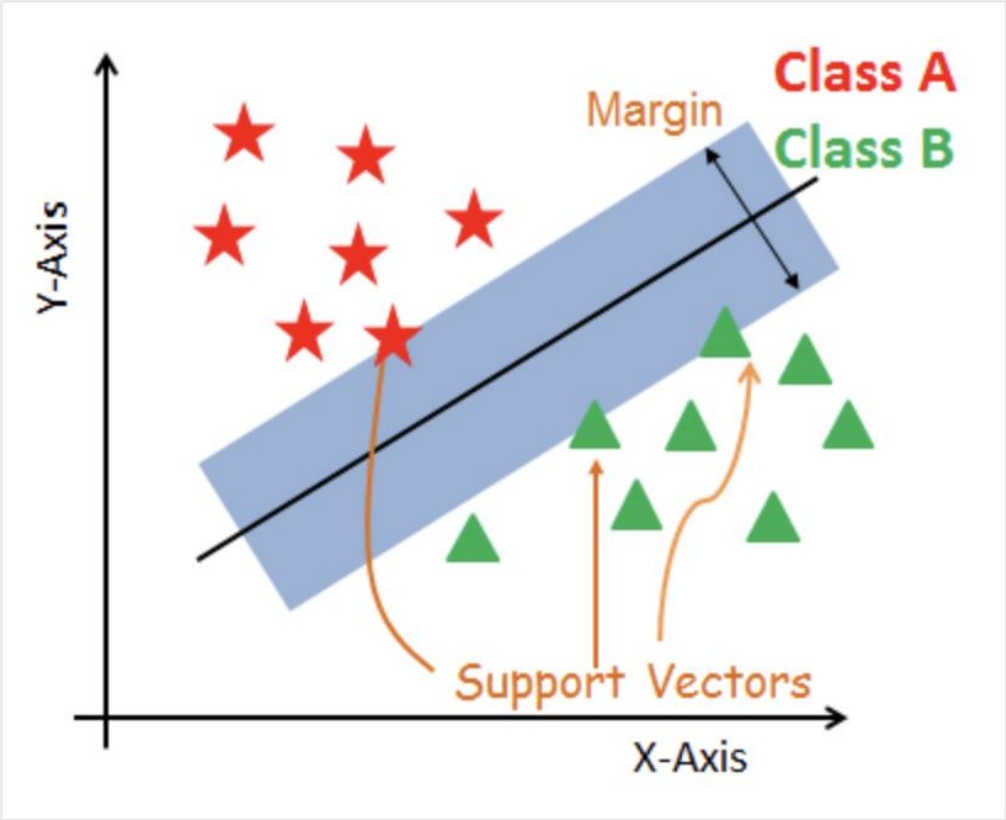
Support Vector Machine (SVM) is a widely used supervised learning algorithm, predominantly employed for classification tasks, but also applicable to regression problems in machine learning. The objective of the SVM algorithm is to determine the optimal decision boundary or hyperplane that separates an n-dimensional space into distinct classes. This enables the accurate categorization of new data points in the future.

SVM selects the most critical points or vectors that facilitate the construction of the hyperplane. These critical points are referred to as support vectors, which give the algorithm its name, Support Vector Machine.

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

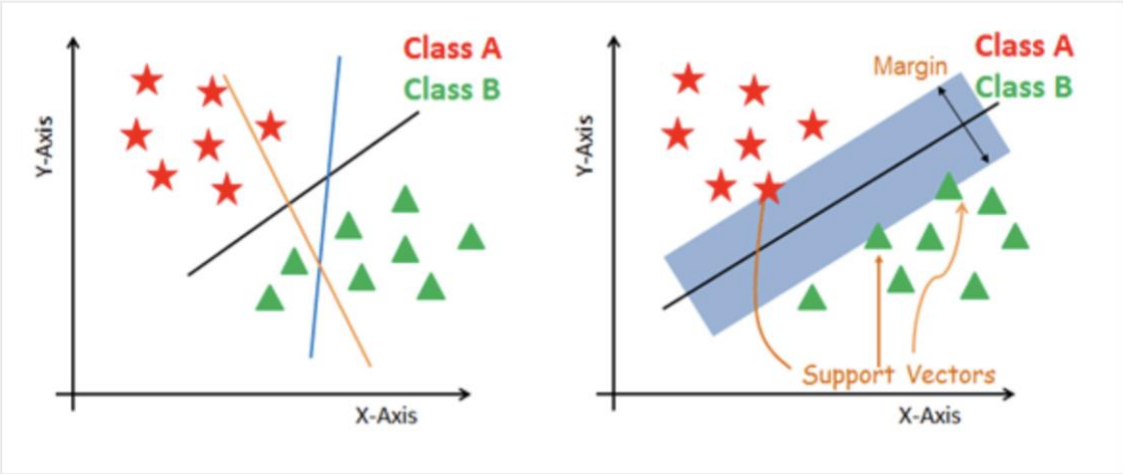
A hyperplane is a decision plane that separates a set of objects having different class memberships.

A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.



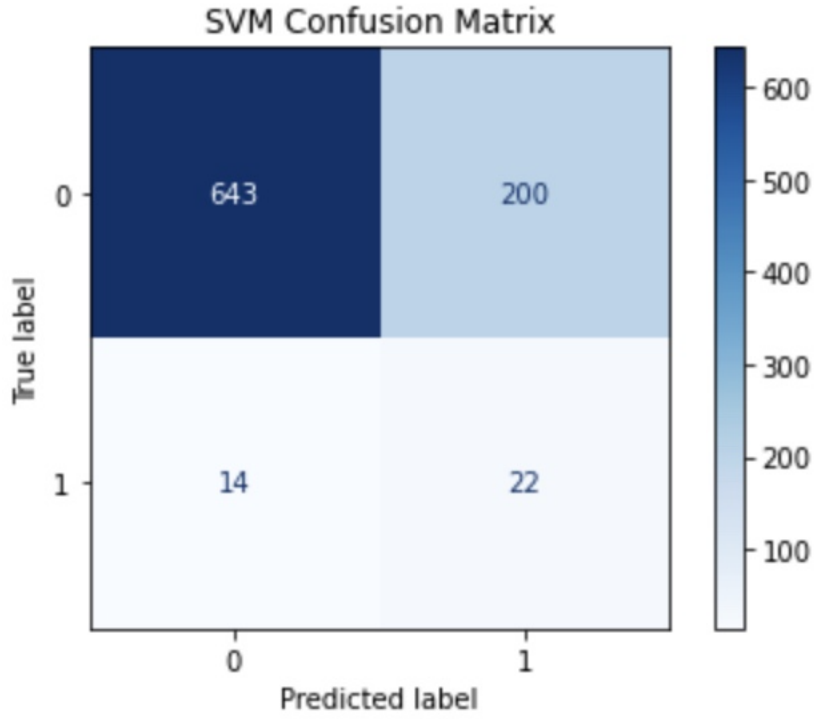
The primary goal of SVM is to optimally separate the given dataset. The margin is defined as the distance between the nearest points of each class. The objective is to choose a hyperplane that maximizes the margin between support vectors in the dataset. SVM identifies the maximum margin hyperplane through the following steps:

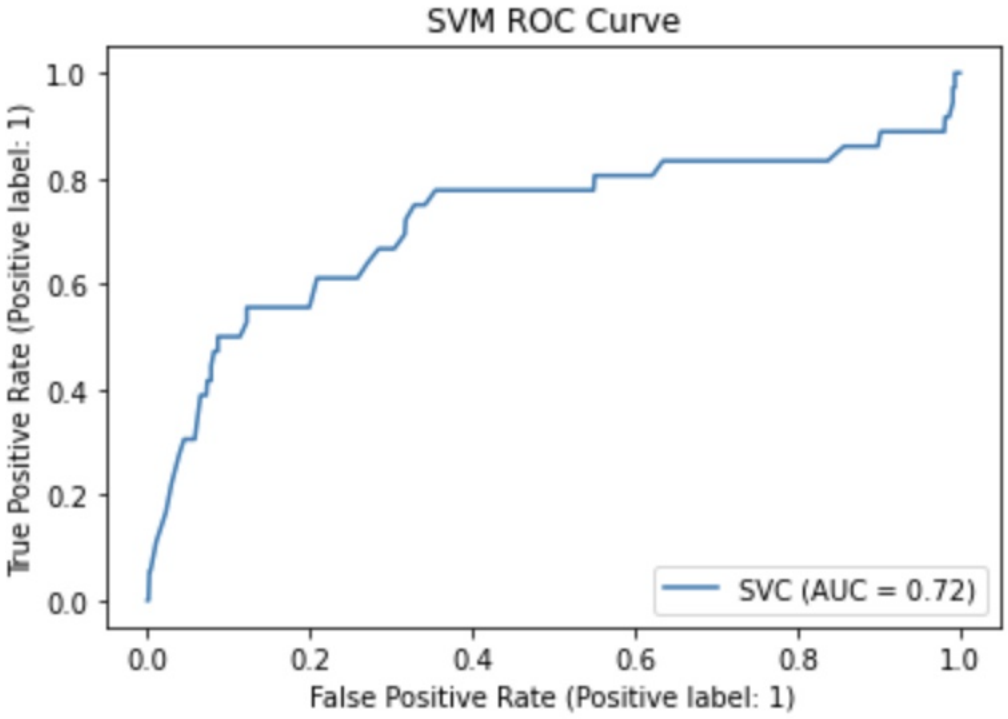
1. Generate hyperplanes that best separate the classes. The left-hand side figure illustrates three hyperplanes (black, blue, and orange). The blue and orange hyperplanes have higher classification errors, while the black hyperplane correctly separates the two classes.
2. Choose the hyperplane that maximizes the distance from the nearest data points of each class, as depicted in the right-hand side figure.



The result of our project for SVM

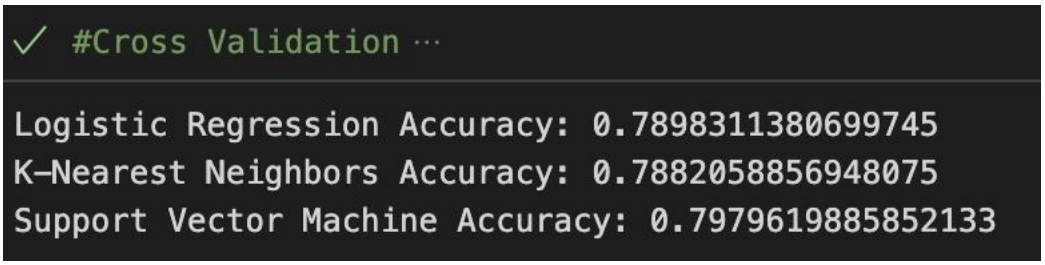
```
Support Vector Machine Accuracy: 0.7565415244596132
[[643 200]
 [ 14  22]]
              precision    recall  f1-score   support

           0       0.98      0.76      0.86       843
           1       0.10      0.61      0.17        36

    accuracy                           0.76       879
   macro avg       0.54      0.69      0.51       879
weighted avg       0.94      0.76      0.83       879
```





Cross Validation Scores:

Applied cross Validation for all the models mentioned above



Code Percentage: 42%

References:

https://chat.openai.com/

https://www.kaggle.com/code/frankmollard/develop-a-stroke-risk-web-app-using-streamlit

https://www.kaggle.com/code/mechatronixs/prediction-with-7-classification-models-roc-auc

https://www.kaggle.com/code/sreelakshmi20bce1464/stroke-prediction-using-5-models

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9268898/