SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
A.Y. 2022 - 23
Course: Machine Learning

Project Report

Program	MBA-Tech AI	
Semester	4th Semester	
Name of the Project:	Health prediction centre	
Details of Project Members		
Batch	Roll No.	Name
B2	R036	Daivik Jayan:-
B2	R038	Jhalak Mishra: Github
B2	R044	Tanya Rathod-
B2	R054	Archisha Sinha- GitHub
Date of Submission: 6th April,2023.		

Contribution of each project Members:

Roll No.	Name:	Contribution
R036	Daivik Jayan	Coding of Random Forest algorithm and Report.
R038	Jhalak Mishra	Data preprocessing and visualization, Report and Diabetes Predictor.
R044	Tanya Rathod	Coding of SVM and Report.
R054	Archisha Sinha	Graphical User Interface using HTML, Report, Maternal HeathCare Risk Predictor.

Project Report

Heart Stroke Prediction

by

Daivik Jayan, Roll number: R036

Jhalak Mishra, Roll number: R038

Tanya Rathod, Roll number: R044

Archisha Sinha, Roll number: R054

Course: Machine Learning

AY: 2022-23

Table of Contents

I. Project Idea and applications

Heart Stroke Prediction

Cardiovascular Diseases (CVDs) are the most common cause of death globally, accounting for 32% of all global deaths. Two most common CVDs are heart attack and heart stroke, which are caused by blockage of oxygen or blood supply to the heart muscle. Risk factors include unhealthy diet, tobacco use, diabetes, sedentary lifestyle, unhealthy use of alcohol, high blood pressure and family history.

Machine learning is a form of artificial intelligence that can analyze data, identify patterns and predict the outcome with minimal human intervention. This proposed model predicts heart stroke prediction of several individuals using two machine learning algorithms namely: Random Forest and Support Vector Machine(SVM).

This project can be used to detect whether they are at risk of getting a heart stroke. They need to enter multiple inputs such as their age, sex, do they have chest pain,their Blood pressure ,Cholesterol etc .Depending on these following inputs our model will predict whether the person is at a risk of getting a heart stroke/disease or not. The model boasts an accuracy of 90 percent or so.

Some of the advantages of our model and the website created for this model is as follows:

- Early detection: A website that predicts heart diseases can help identify potential heart problems at an early stage, allowing individuals to seek medical attention and treatment before the condition worsens.

- Convenience: Websites can be accessed from anywhere with an internet connection, making it easier for people to get information and assess their risk of heart disease without having to visit a doctor's office.

- Cost-effective: Websites are often free or low-cost, making it an affordable option for people who may not have access to medical care or want to save money on expensive medical tests.

- Personalized recommendations: A website that predicts heart diseases can provide personalized recommendations based on an individual's health history, lifestyle factors, and other relevant information.

- Empowerment: By providing people with information about their risk of heart disease, a website can empower them to make positive lifestyle changes, such as exercising more, eating a healthy diet, and quitting smoking, to reduce their risk of heart disease.

- Improved health outcomes: Early detection and prevention of heart disease can lead to improved health outcomes, including reduced mortality rates and better quality of life for those living with the condition.

Diabetes Prediction

Diabetes is a chronic metabolic disease characterized by high blood glucose levels, resulting from the body's inability to produce or use insulin effectively. It is a significant health concern worldwide, with approximately 463 million adults diagnosed with diabetes in 2019, and this number is expected to increase to 700 million by 2045.

Early detection and management of diabetes can prevent or delay complications, including cardiovascular disease, neuropathy, retinopathy, and kidney failure. Machine learning has emerged as a promising tool to predict and diagnose diabetes, allowing healthcare providers to identify individuals at high risk of developing the disease and initiate early interventions to prevent its onset.

Using machine learning algorithms, healthcare providers can analyze large datasets containing patient information, including demographic data, medical history, lifestyle factors, and laboratory results, to develop predictive models for diabetes. These models can then be used to identify patients at high risk of developing diabetes, allowing healthcare providers to initiate appropriate interventions, including lifestyle modifications, medication, or referral to specialist care.

Overall, the use of machine learning in diabetes prediction and diagnosis has the potential to improve the accuracy of risk assessment, enhance early detection, and improve patient outcomes by initiating timely and appropriate interventions.

Maternal Health Risk Prediction

Maternal health risk prediction models in machine learning (ML) can help identify pregnant women who are at high risk of complications during pregnancy, childbirth, or postpartum.

The use of such models can improve maternal and child health outcomes by enabling early interventions and targeted care.

To develop a maternal health risk prediction model in ML, the first step is to collect relevant data. This may include medical history, lifestyle, demographic factors, and data related to pregnancy outcomes and complications. The collected data needs to be cleaned and preprocessed to handle missing values, outliers, and categorical variables.

The performance of the trained model is then evaluated using metrics such as accuracy, precision, recall, F1-score, and area under the curve (AUC). If the model's performance is not satisfactory, hyperparameters need to be tuned to improve its accuracy.

Finally, the trained model is deployed in a user-friendly interface that can be used by healthcare providers to predict maternal health risk. The interface should provide clear and concise information about the risk factors and the interventions required to mitigate the risk.

In conclusion, maternal health risk prediction models in ML can help improve maternal and child health outcomes by enabling early interventions and targeted care. Developing a maternal health risk prediction model in ML involves collecting and preprocessing data, selecting important features, choosing an appropriate ML algorithm, training and evaluating the model, tuning hyperparameters, and deploying the model.

II.Dataset details

Heart Stroke Prediction

This dataset has been acquired from Kaggle and contains 12 features. Our model is trained on these features and predicts the changes of a person having a stroke or no. The features of our dataset are as follows:

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic] (Here Angina refers to chest pain.)
4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol[mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No](Was chest pain caused due to exertion or exercise)
10. Oldpeak: oldpeak = ST [Numeric value measured in depression] (this value tells us if there has been any prior damage to the heart, for example a minor heart attack.)
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping] (this means The ST segment shift relative to exercise-induced increments in heart rate, the ST/heart rate slope (ST/HR slope), has been proposed as a more accurate ECG criterion for diagnosing significant coronary artery disease (CAD).
12. HeartDisease: output class [1: heart disease, 0: Normal]

Diabetes Prediction

This data was compiled by the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict whether a patient has diabetes based on diagnostic measurements.

The selection of these instances from a larger database was subject to a number of restrictions. All female patients here are at least 21 years old and of Pima Indian descent.

1. Pregnancies: the number of pregnancies
2. Plasma glucose concentration after two hours of an oral glucose tolerance test
3. The diastolic blood pressure (mm Hg)
4. SkinThickness: girth of the triceps skin fold (mm)
5. Insulin: 2-Hour serum insulin (mu U/ml)
6. Body mass index (measured in kilogrammes per metre squared)
7. DiabetesPaternity: diabetes paternity function
8. Age: Age (years)
9. Outcome: Class variable (0 or 1)

Maternal Health Risk Prediction

This data was compiled by the Kaggle . The objective is to predict whether a patient who is basically an expected mother is at high low or mid risk of complexities at the moment.

The selection of these instances from a larger database was subject to a number of restrictions. All female patients here are at least 21 years old and above.

The dataset contains the following information:

1. Age
2. Systolic Blood Pressure
3. Diastolic Blood Pressure
4. Blood Sugar Level
5. Body Temperature
6. Heart Rate
7. Risk Level

III. Preprocessing and Visualization

Heart Stroke Prediction

Data Preprocessing

The preprocessing in our project included how to load, preprocess, and analyze a dataset in Python using various libraries such as pandas, matplotlib, seaborn, and scikit-learn.

The dataset used in this code is 'heart.csv', which contains 918 instances and 12 features . The features in this as mentioned above are 'Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol', 'FastingBS', 'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope', 'HeartDisease'.

First, we loaded the dataset using the pandas library, which is a widely used Python library for data manipulation and analysis. The 'pd.read_csv()' function reads the csv file and stores it in a pandas dataframe 'df'.

We then explored the dataset by printing the first five rows using 'df.head()', which provides a quick overview of the data. The shape and size of the dataset are then determined using 'df.shape' and 'df.size', respectively. The shape of the dataset shows that it has 918 instances and 12 features, while the size of the dataset gives the total number of cells.

To further understand the dataset, we used 'df.info()', which provides information about the dataset such as the number of non-null values in each column, the data type of each column, and the memory usage of the dataset.

To check if there are any missing values in the dataset, we used 'df.isnull()', which returns a boolean data frame indicating whether each cell in the dataset is null or not. The sum of the missing values in each column is then determined using 'df.isnull().sum()', which shows us that there are no missing values in the dataset. There were no null/missing values in the dataset as per the output of the code.

Next,we performed data preprocessing on the categorical variables in the dataset using the 'LabelEncoder' function from scikit-learn's preprocessing module. The 'LabelEncoder' function is used to transform categorical variables into numeric variables, which is a requirement for many machine learning algorithms. The code encodes the 'Sex', 'ChestPainType', 'RestingECG', 'ExerciseAngina', and 'ST_Slope' variables using 'le.fit_transform()' function and assigns the encoded variables to their respective columns in the dataframe 'df'.

After encoding the categorical variables, we split the dataset into a feature matrix 'X' and a target vector 'y'. The feature matrix 'X' contains all features except for the target variable 'HeartDisease', while the target vector 'y' contains only the target variable.

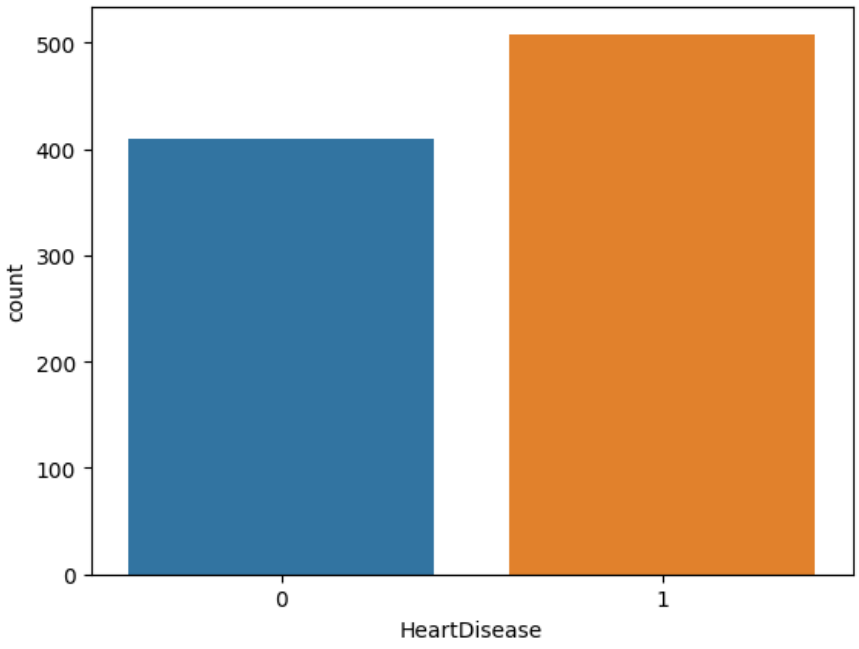
Finally, we used a for loop and 'np.unique()' function to print the unique values for each column of the dataframe 'df'. This provides a better understanding of the dataset and helps in identifying any outliers or inconsistencies in the data.

Data Visualization

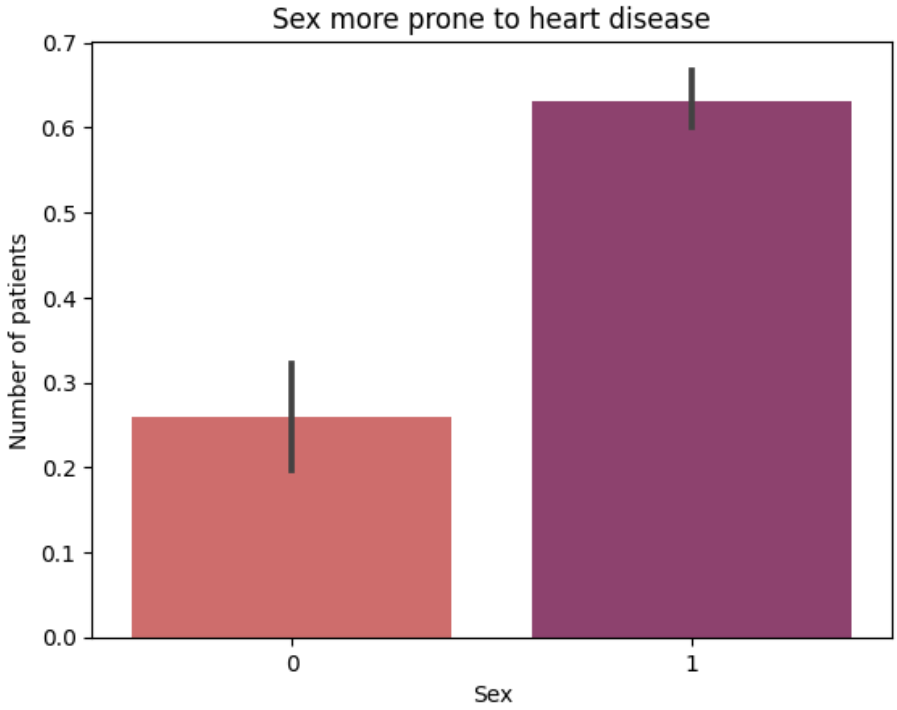
Data visualization techniques can be used to provide a better understanding of the underlying structure and relationships between the different variables in the dataset. One powerful tool for data visualization is the Seaborn library, which provides a range of functions and tools for creating different types of plots and charts.

For the given dataset, which contains information about patients with suspected heart disease, many different types of visualizations can be used to explore the data. The following are the visualizations we used to create different graphs to understand our data:

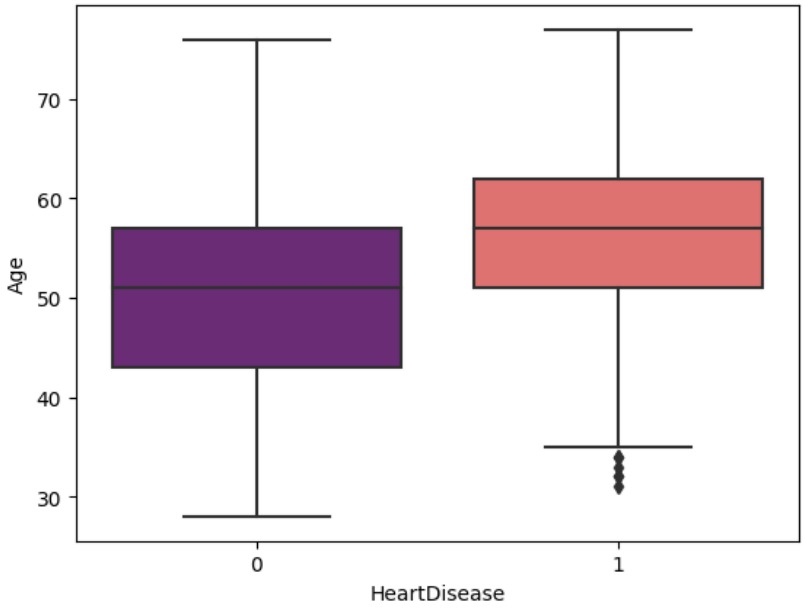
1. The below counterplot/bar plot depicts the number of patients in our dataset which are prone to heart disease given the features and which are not. The inference from this bar plot is that out of 918 instances, approximately 400 people are not prone to heart disease while the other 500 people(Approximately) had.



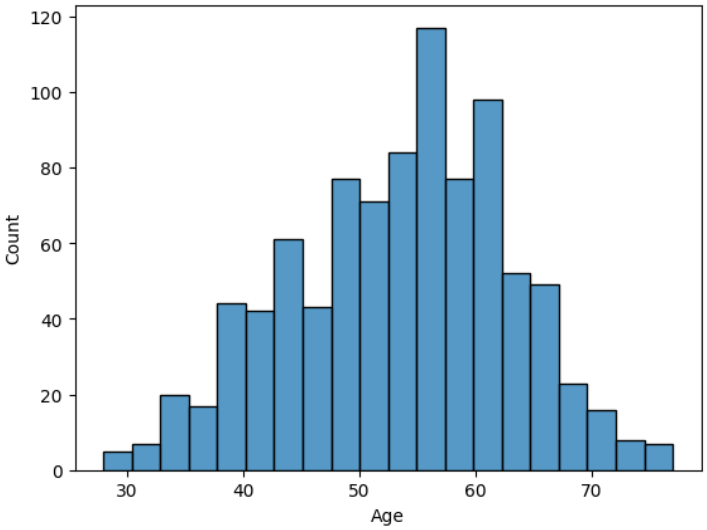
2. Given that there are more male patients with heart illness than female patients, it may be concluded from this image that men are more likely than women to have heart disease. The fact that the standard for men with heart disease is greater than the bar for women with heart disease serves as the foundation for this assumption.



3. This boxplot suggests that people with heart disease(1) are often older than those without heart disease. The distribution of ages for patients with heart disease is more skewed towards older ages than the distribution for patients without heart disease, and this inference is based on the facts that the median (represented by the line inside the box) for patients with heart disease is higher than the median for patients without heart disease.

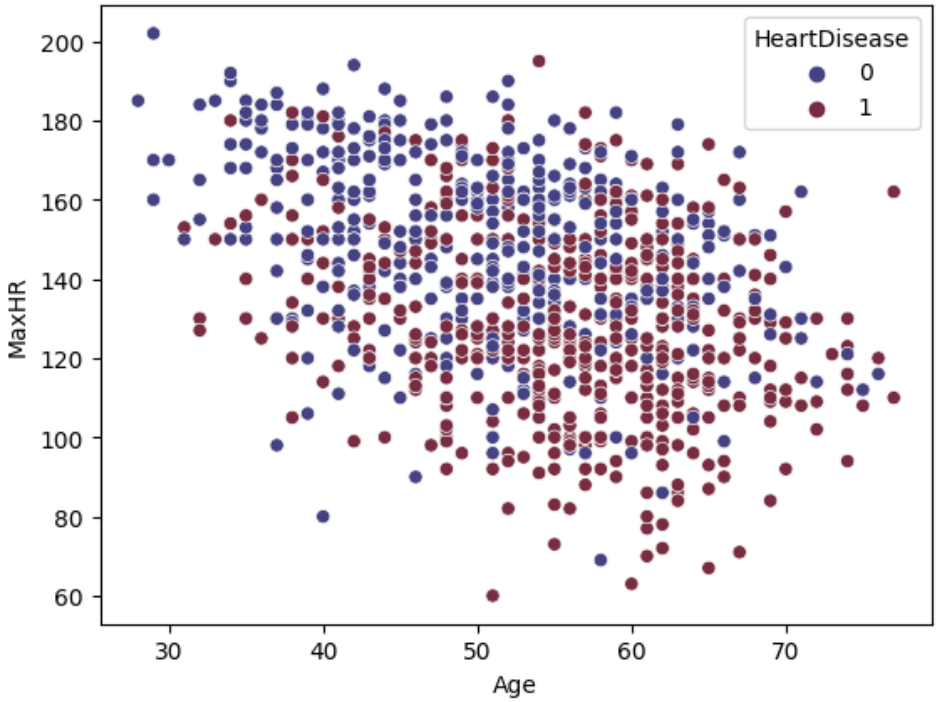


4. The distribution of ages in the dataset is inferred from this representation to be about normally distributed, with a peak occurring between the ages of 55 and 65. This conclusion is based on the histogram's resemblance to a bell curve.
   Another conclusion drawn from this visualization is that there aren't many patients in the dataset who are under the age of 40. This is supported by the fact that there are few patients in the youngest age bins of the histogram (i.e., ages less than 40). This can be as a result of the lower prevalence of cardiac disease in younger people or a dataset restriction (i.e., it may not include many younger patients).

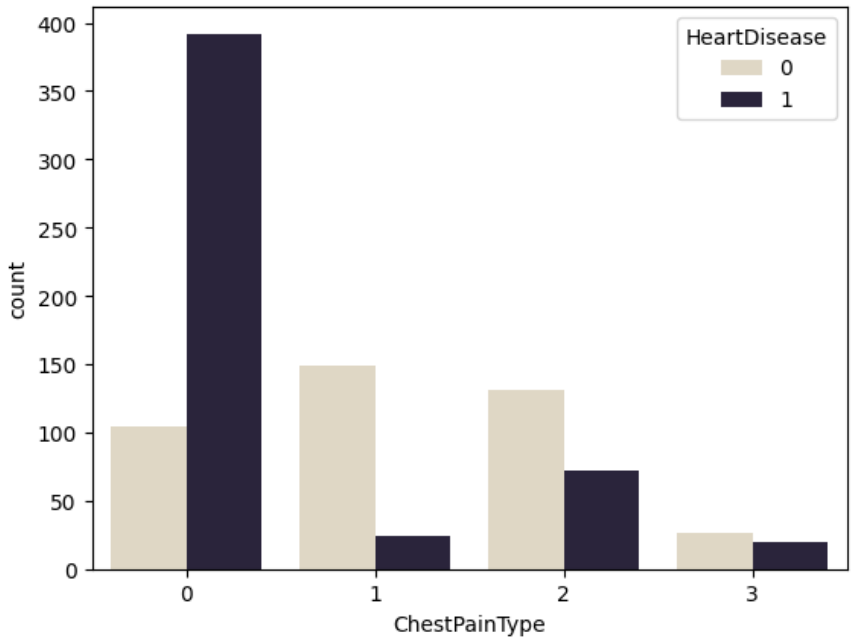


5. This graph suggests that individuals with heart illness often have lower maximal heart rates than those without heart disease. This is due to the fact that people with heart disease tend to cluster towards the lower end of the maximum heart rate range, while healthy individuals are more evenly distributed across the maximum heart rate range.

   Another conclusion drawn from this visualization is that, whether or not patients have cardiac disease, there is a universally negative correlation between age and maximal heart rate. Therefore maximal heart rate tends to decline as age rises.



6. This graph suggests that people with type 0 and type 1 chest pain are more likely to suffer heart disease than those with type 2 and type 3 chest pain. This is based on the observation that patients with chest pain kinds 0 and 1 are more likely to have heart disease than those without, but patients with chest pain types 2 and 3 are more likely to have heart disease.
   Another conclusion drawn from this visualization is that regardless of whether a patient has cardiac disease, type 3 chest pain is the most prevalent form of chest pain reported by patients in the sample. This is based on the fact that the plot's y-axis shows that chest pain type 3 has the largest count.

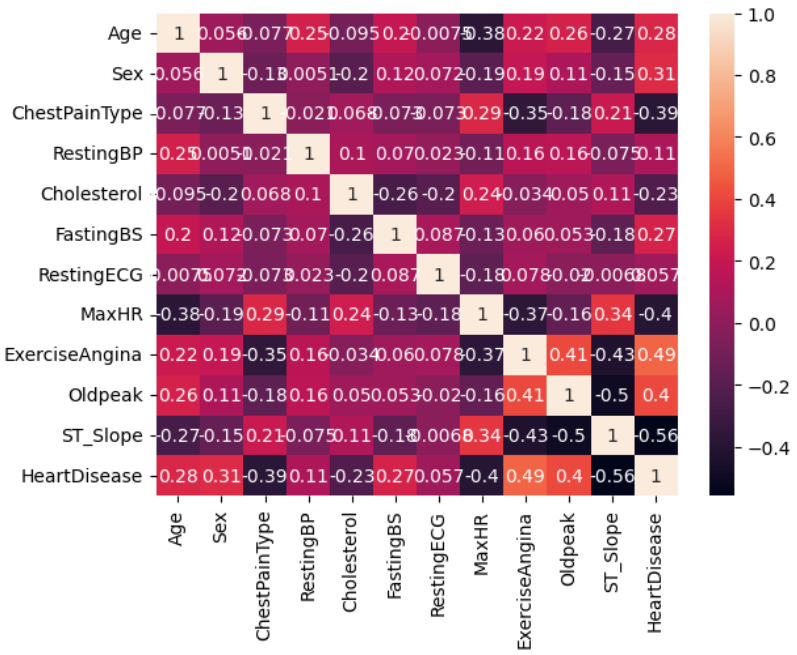


7. The heatmap is an effective tool for showing the strength and direction of the linear connection between variable pairs in the dataset. A correlation coefficient that is positive shows a positive association between two variables, while a correlation value that is negative suggests a negative relationship. A correlation value of 0 shows that there is no linear association.

   Age, resting blood pressure (RestingBP), and cholesterol levels are favorably connected with one another, but age, RestingBP, and cholesterol levels are negatively correlated with maximum heart rate obtained during exercise (MaxHR).

   This shows that older individuals with greater RestingBP and cholesterol levels may have a higher risk of developing heart disease, but those with a higher maximum heart rate obtained during exercise may have a lower risk of developing heart disease.

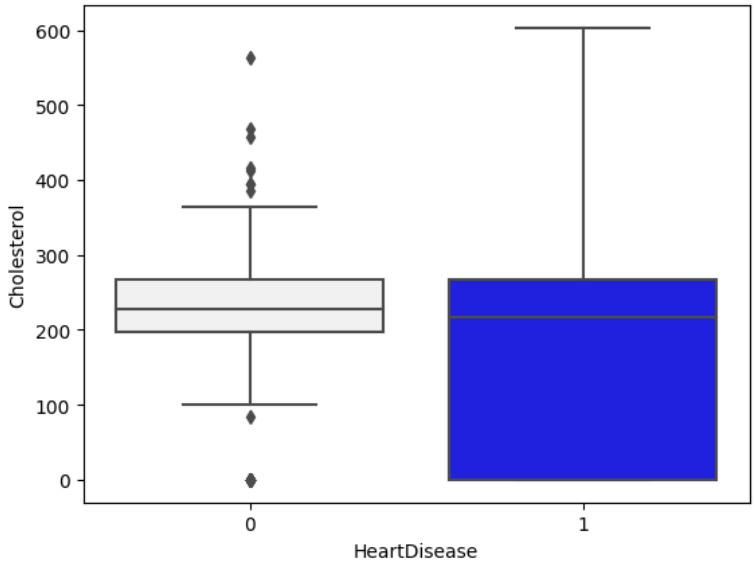
   There is a somewhat significant link between chest pain type and heart disease, which suggests that people with particular forms of chest pain may be more likely to develop heart disease. In addition, there is a modest negative connection between the slope of the ST segment during peak exertion (ST Slope) and heart disease, suggesting that individuals with a more downwardly sloped ST segment during exercise may be less likely to develop heart disease.



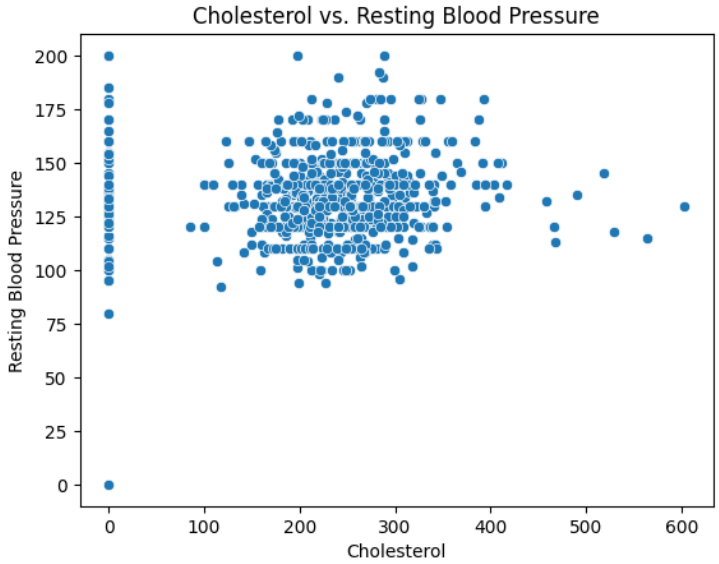
8. The x-axis indicates the prevalence or absence of cardiac disease, whilst the y-axis indicates cholesterol levels.

   The box plot reveals that the median cholesterol levels of individuals with heart disease are somewhat higher than those of people without heart disease. In addition, the box plot reveals that there are a few heart disease patients with abnormally high or low cholesterol levels (outliers).

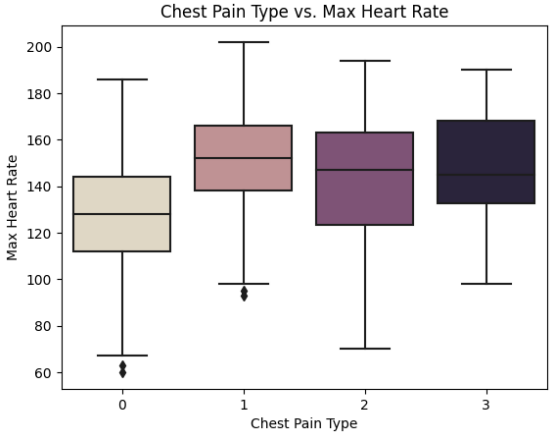
   Consequently, we may deduce that high cholesterol levels may be a risk factor for heart disease, and additional research is required to determine the nature of the association between cholesterol levels and heart disease.



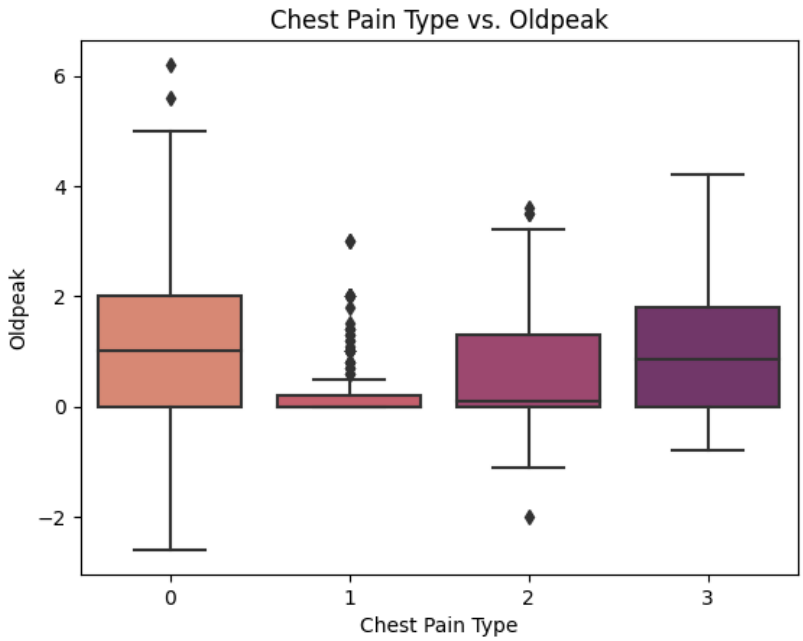
9. There is no apparent linear association between cholesterol levels and resting blood pressure, as seen by the scatter figure. Yet, we can observe that there is a concentration of data points with elevated cholesterol and blood pressure during rest.
   This implies that increased cholesterol levels may be related with greater resting blood pressure, although other variables also influence resting blood pressure. Consequently, further research is required to comprehend the association between cholesterol levels and resting blood pressure, as well as the function of other possible heart disease risk factors.



10. Patients with chest pain types 0 and 1 tend to have a greater maximal heart rate than those with chest pain types 2 and 3. This shows that the kind of chest discomfort may be associated with the highest heart rate reached during exercise.
    Yet, there is a substantial overlap between the various forms of chest pain, suggesting that chest pain type alone may not be a reliable predictor of the highest heart rate obtained during exercise. Consequently, further research is required to comprehend the association between chest pain kind and maximal heart rate obtained during exercise, as well as the function of other possible heart disease risk factors.

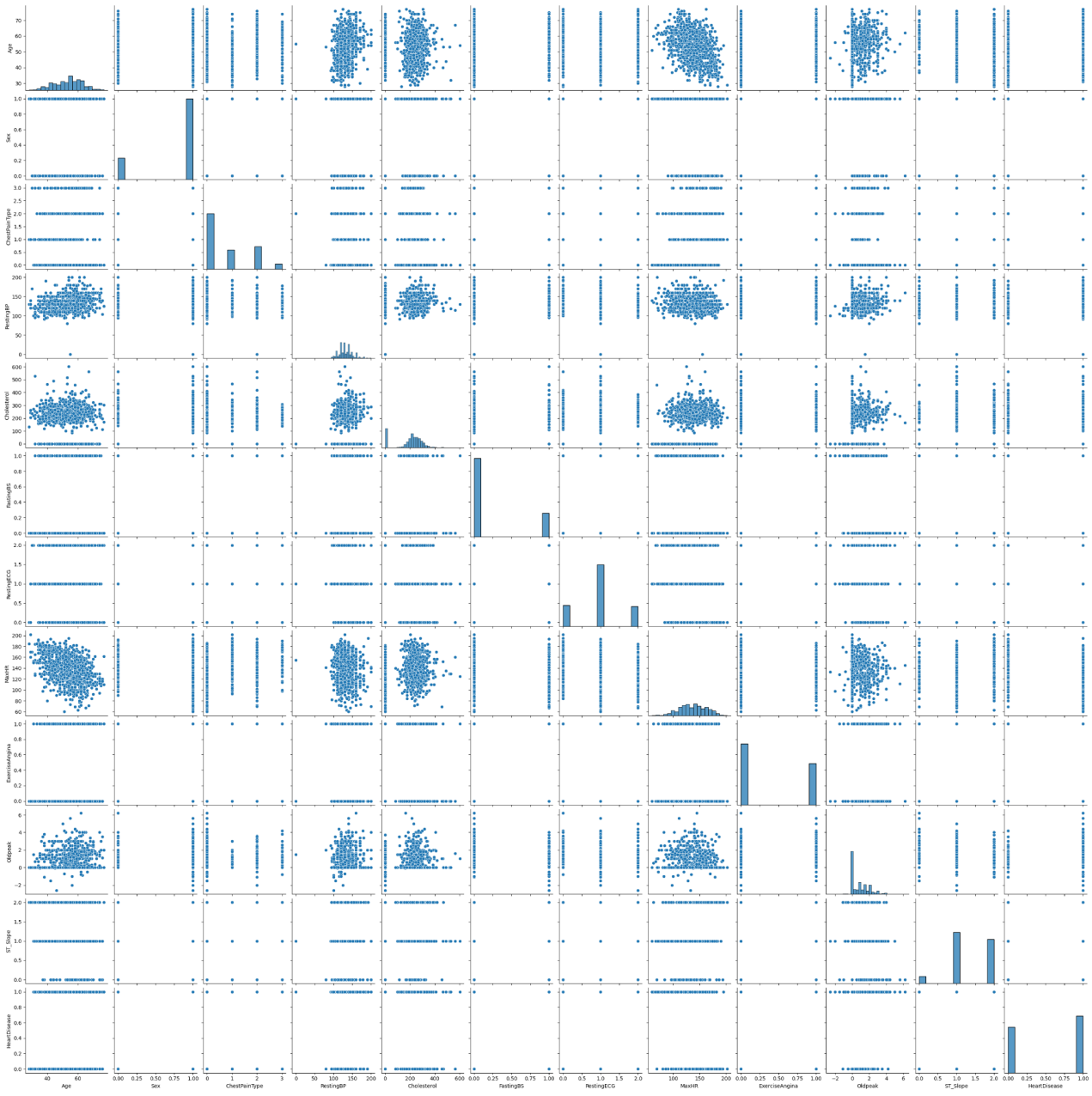


11. The boxplot illustrates the association between the variables Chest Pain Type and Oldpeak. Oldpeak indicates the ST depression generated by activity compared to rest and is hence a significant predictor of heart disease. Patients with Chest Pain Type 0 and 1 had a greater median Oldpeak value compared to patients with Chest Pain Types 2 and 3. This indicates that people with Chest Pain Types 0 and 1 may be more susceptible to cardiac disease than those with Chest Pain Types 2 and 3. Yet, further research and statistical tests are need to validate this association.



12. From this pairplot, the following conclusions may be drawn:
    a. Age correlates positively with RestingBP and Cholesterol.
    b. Age and RestingBP have a negative connection with MaxHR.
    c. Age, RestingBP, Cholesterol, MaxHR, and Oldpeak seem to be distributed differently across people with and without heart disease.
    d. There is no obvious linear connection between the variables, indicating that a nonlinear model may be required for forecasting.
    e. The association between ChestPainType and the other variables in the dataset is unclear.
    f. The relationship between Oldpeak and MaxHR is negative.
    g. There seems to be a link between resting blood pressure and cholesterol.

    Overall, the pairplot may help us uncover any potential correlations between the variables and is a valuable tool for selecting features in machine learning models.



Diabetes Prediction

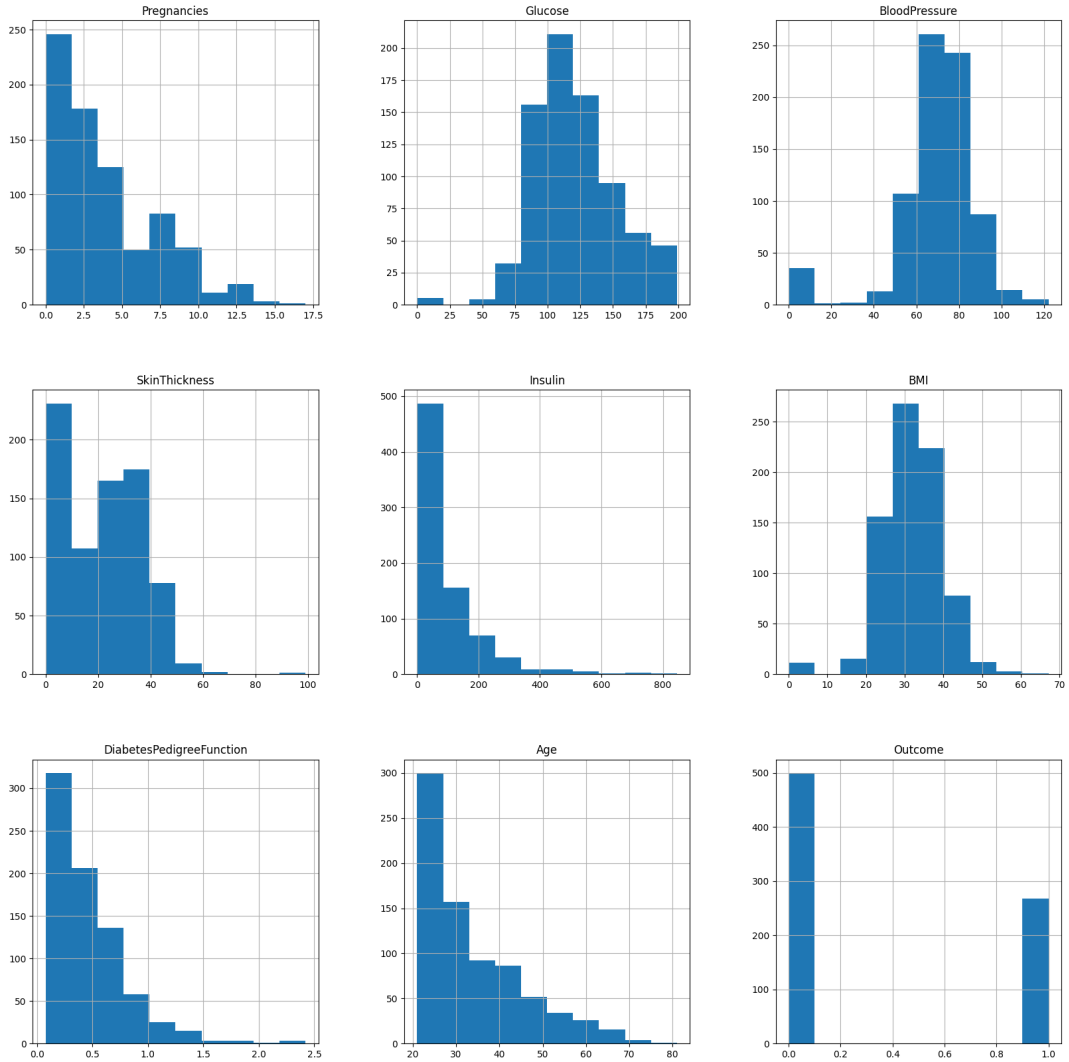
Data Preprocessing

1. Data Collection: Collecting the relevant data from various sources, including electronic health records, surveys, and other medical data sources.
2. Data Cleaning: Checking the data for missing values, duplicates, and outliers. Handle missing values by either imputing them with mean, median, mode, or using advanced techniques such as K-Nearest Neighbors imputation or interpolation. Outliers can be handled by capping or replacing them with mean/median values or removing them altogether.

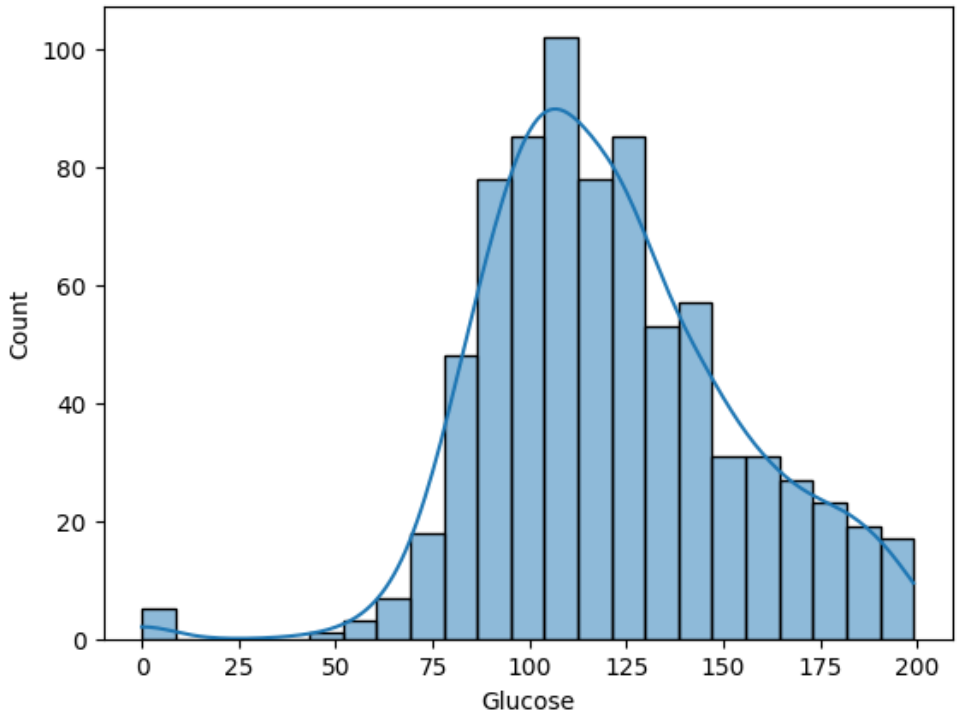
3. Data Transformation: Scaling or normalizing numerical features to ensure that they have a similar scale.
4. Train-Test Split: Splitting the data into training and testing sets to evaluate the performance of the model.

Data Visualization

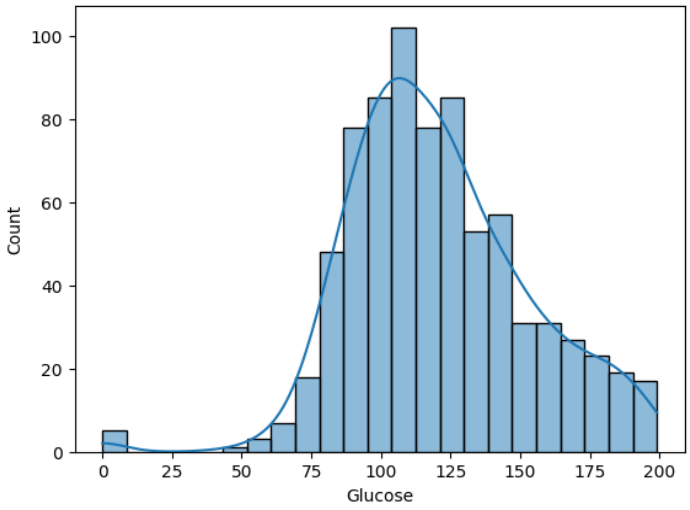
1. This command can be useful for quickly visualizing the distribution of each numeric feature in the dataset. For example, it can help identify any features that have a large number of values in a particular range or features that have many missing values. Additionally, it can be useful for identifying potential outliers or skewness in the distribution of a feature, which can be important considerations when building a predictive model.



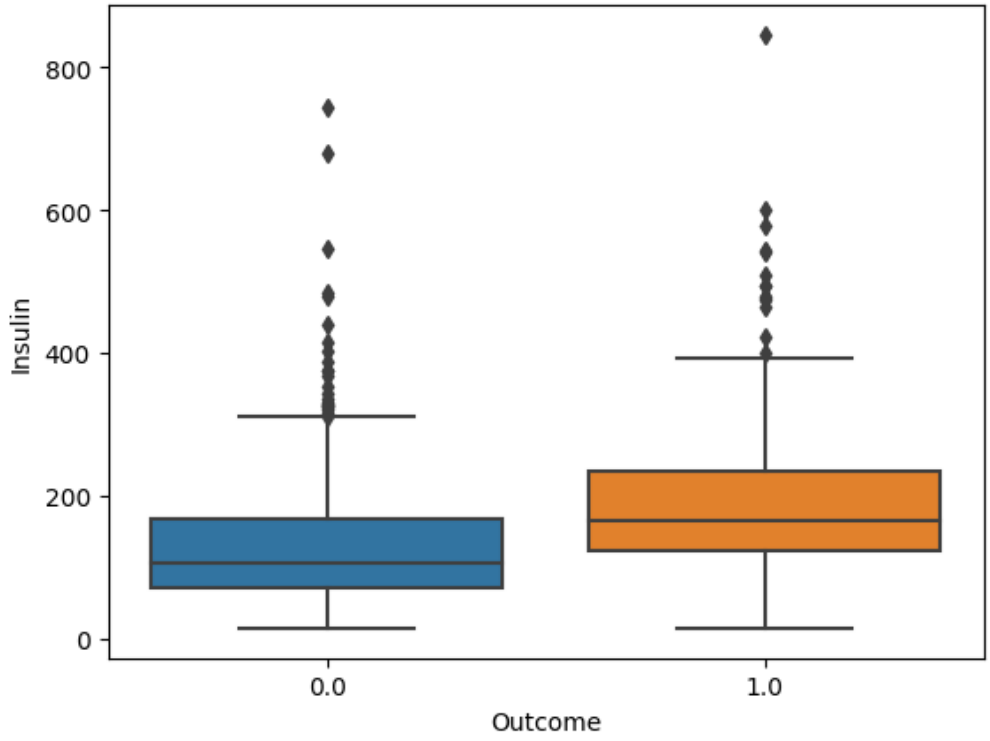
2. The histogram can give insights into the range of glucose values in the dataset and the number of observations in each range. The kernel density estimation plot can provide additional information about the shape of the distribution of the feature, indicating whether it is normally distributed or skewed, which can be an important consideration when selecting appropriate statistical methods for data analysis.



3. The scatter plot can show any patterns or trends in the data, such as a positive or negative correlation between the two features, as well as how these patterns relate to the target variable.



4. The box in the boxplot represents the interquartile range (IQR) of the data, with the lower and upper hinges representing the first and third quartiles, respectively. The horizontal line inside the box represents the median insulin level, while the whiskers extending from the box represent the range of the data, excluding outliers.
   The plot shows that the median insulin level is higher for diabetic patients (Outcome = 1) compared to non-diabetic patients (Outcome = 0). There are also some outliers for both classes, with diabetic patients having more extreme values. Overall, this suggests that insulin levels may be a useful feature for predicting diabetes in this dataset.



Maternal Health Risk Prediction

Data Preprocessing

Data preprocessing is an essential step in machine learning (ML) that involves cleaning, transforming, and organizing raw data into a format suitable for analysis. The goal of data preprocessing is to ensure that the data is consistent, accurate, and complete, and to eliminate errors, inconsistencies, and irrelevant information.

The following are the key steps involved in data preprocessing in ML:

Data cleaning: This involves handling missing data, removing duplicates, and dealing with outliers. Missing data can be handled by either deleting the rows or filling in the missing values using techniques such as mean imputation, median imputation, or regression

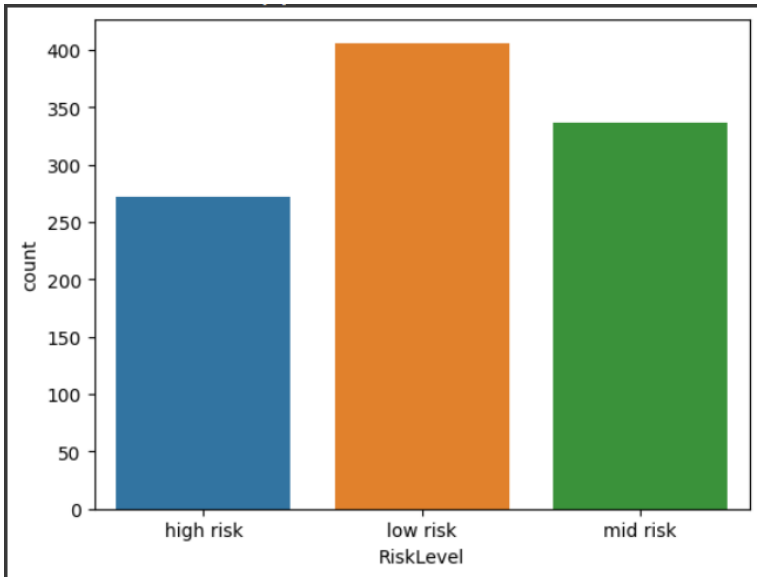
imputation. Duplicates can be removed by identifying and dropping identical rows. Outliers can be handled by identifying them using statistical techniques such as z-score or interquartile range (IQR) and either removing or correcting them.
Since our dataset was completely clean dataset, we did not have to clean it.

Data transformation: This involves converting data into a form that can be analyzed by ML algorithms. Some common data transformation techniques include feature scaling, normalization, and encoding categorical variables. Feature scaling involves scaling features to the same range, while normalization involves transforming data into a standard normal distribution. Categorical variables can be encoded into numerical values using techniques such as one-hot encoding or label encoding.
We did Lable Encoding on the Risk Level feature as it had a Categorical Value.

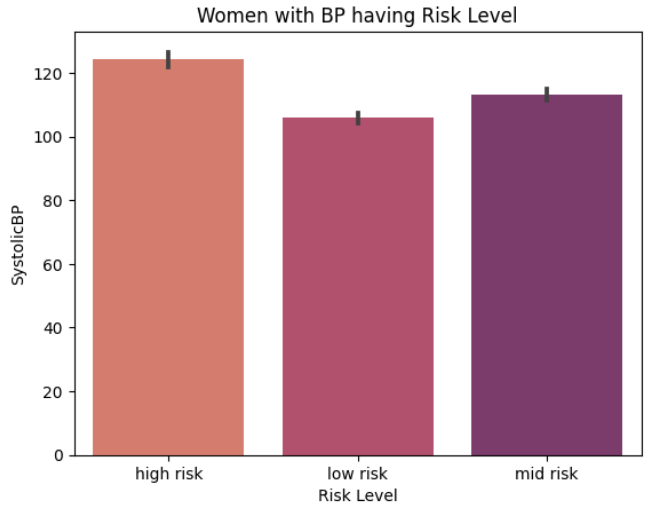
In summary, data preprocessing in ML is a critical step in preparing data for analysis. It involves cleaning, transforming, organizing, integrating, reducing, and discretizing data to ensure that it is consistent, accurate, and complete. Proper data preprocessing can improve the accuracy and reliability of ML models and enable better insights and decision-making.

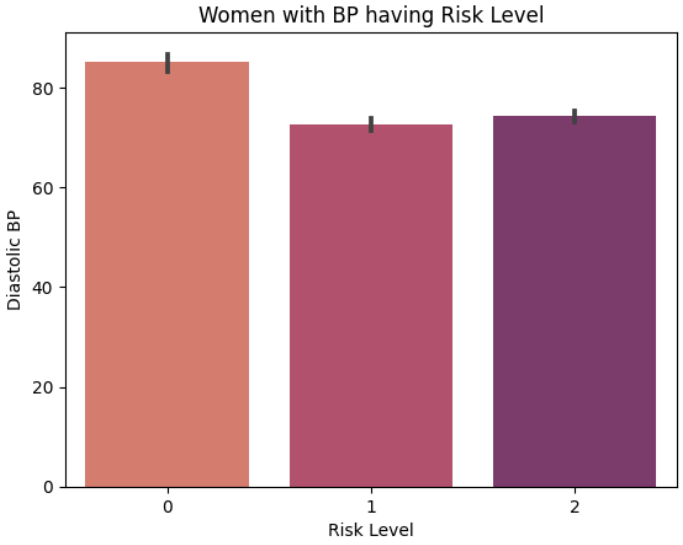
Data Visualization

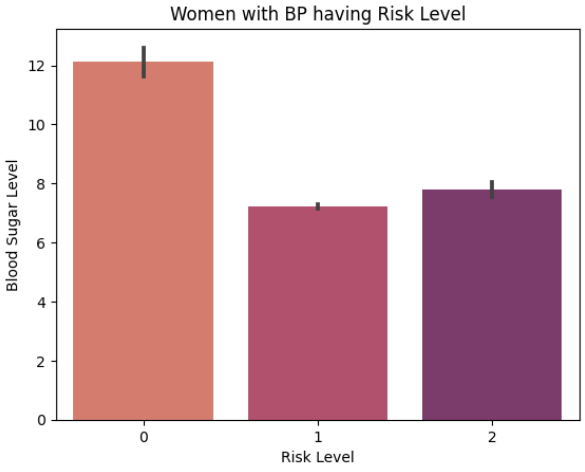
A countplot to check how many expected mothers are at high, low, or mid risk of health. Over here, approximately 270 expected mothers are at High risk and need uttmost care at the moment, around 400 expected mpthers are at Low risk which is the maximum and around 350 mothers are at Mid risk and need some care towards them.

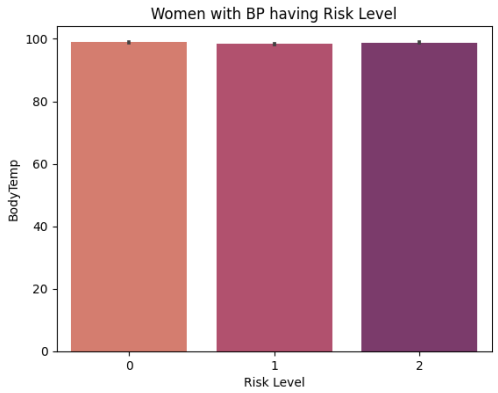


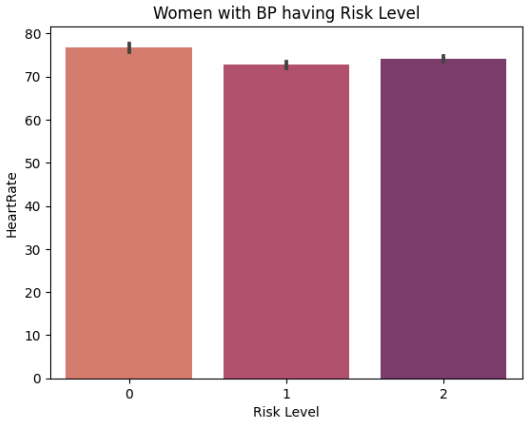
This is a pairplot, i.e. all the features against all the other features. it basically shows us the relationship between the two features.

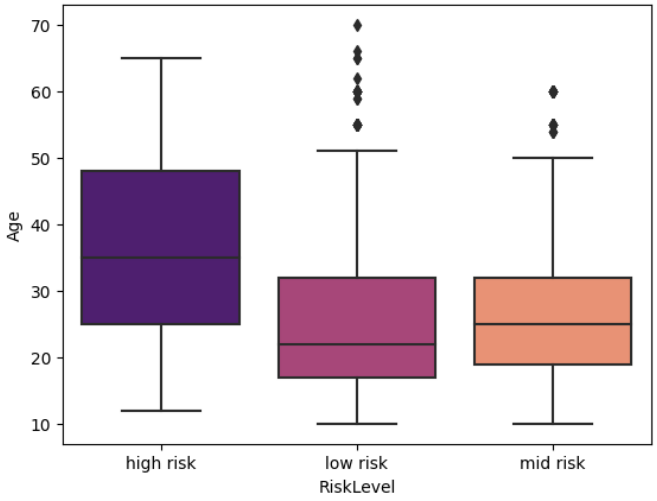

Women with BP having Risk Level
SystolicBP
0
20
40
60
80
100
120
high risk
low risk
mid risk
Risk Level

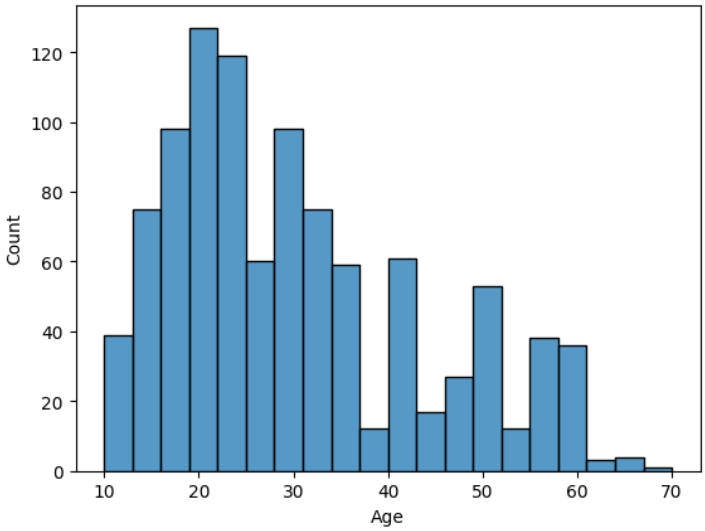

Women with BP having Risk Level
Diastolic BP
0
20
40
60
80
0
1
2
Risk Level

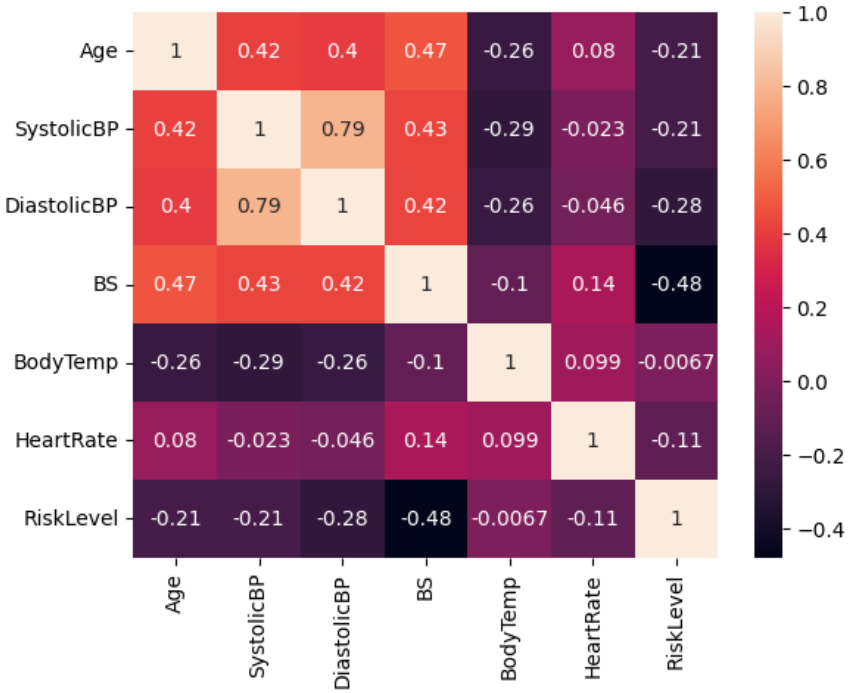

Women with BP having Risk Level
Blood Sugar Level
0
2
4
6
8
10
12
0
1
2
Risk Level


Women with BP having Risk Level
BodyTemp
Risk Level
0
20
40
60
80
100
0
1
2


Women with BP having Risk Level
HeartRate
Risk Level
0
10
20
30
40
50
60
70
80
0
1
2


Age
RiskLevel
10
20
30
40
50
60
70
high risk
low risk
mid risk


Count
Age
0
20
40
60
80
100
120
10
20
30
40
50
60
70


Age 1 0.42 0.4 0.47 -0.26 0.08 -0.21
SystolicBP 0.42 1 0.79 0.43 -0.29 -0.023 -0.21
DiastolicBP 0.4 0.79 1 0.42 -0.26 -0.046 -0.28
BS 0.47 0.43 0.42 1 -0.1 0.14 -0.48
BodyTemp -0.26 -0.29 -0.26 -0.1 1 0.099 -0.0067
HeartRate 0.08 -0.023 -0.046 0.14 0.099 1 -0.11
RiskLevel -0.21 -0.21 -0.28 -0.48 -0.0067 -0.11 1
Age
SystolicBP
DiastolicBP
BS
BodyTemp
HeartRate
RiskLevel
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

IV.Model Creation

Heart Stroke Prediction

For model creation we used two Algorithms: SVM and Random Forest.

1. Support Vector Machine:

   SVM is a supervised machine learning algorithm that can be used for both classification and regression. The goal is to find a hyperplane that maximizes the margin between the different classes of data. If the data cannot be separated by a linear hyperplane, SVM can use a kernel trick to map the data to a higher-dimensional space. It is a powerful and widely used algorithm, but can be computationally expensive when dealing with large datasets and requires careful selection of hyperparameters to achieve good performance.

   Support Vector Machines (SVMs) are powerful tools for classification tasks, such as predicting heart stroke. They work by finding the optimal boundary between two classes of data points, which is the distance between the decision boundary and the closest data points from each class. SVMs are effective for predicting heart stroke, with high accuracy rates and the ability to handle large datasets.

   SVM is trained on a set of features that are associated with an increased risk of stroke, such as age, blood pressure, cholesterol levels, and smoking status. The SVM then uses these features to classify new individuals as either being at high or low risk for stroke.

   Train test size was set as 70-30.

   For implementing the SVM algorithm we used the sklearn library and classified it as “Linear”, as in the data set all the entities are linearly plotted and not non-linearly.

2. Random Forest :

   Random Forest is used for supervised learning tasks including classification, regression, and others. It is an ensemble learning technique that integrates various decision trees to provide predictions that are more reliable and accurate.

   A Random Forest model consists of a collection of decision trees, each of which is trained using a portion of the data that is chosen at random. In order to lessen overfitting and boost generalized performance, the trees are built using a random subset of features.

Each decision tree in the forest separately categorizes or predicts the outcome based on the input data during the prediction phase, and the ultimate outcome is decided by taking the majority vote or averaging the outputs of all the trees.

High accuracy, resistance to overfitting, and handling huge datasets with numerous variables are some of the reasons we chose to go with random forest algorithm over others.

Train test size was set as 70-30.

Other hyperparameters were tuned and the most accurate results were obtained with Number of decision trees equal to 100.

Diabetes Prediction

Random Forest Algorithm:

We first import the necessary modules for evaluation and the RandomForestClassifier class from the sklearn.ensemble module.

Next, we initialize the RandomForestClassifier object by setting the number of trees to 300 and the random state to 10. We then fit the classifier to the training data using the fit() method, which takes the training features X_train and training labels y_train as inputs.

After training the classifier, we make predictions on the testing data using the predict() method with the testing features X_test as input. The predicted labels are stored in the pred variable.

The RandomForestClassifier is a type of ensemble learning algorithm that builds multiple decision trees and combines their predictions to improve the accuracy and robustness of the model. It randomly selects a subset of features and data points at each node of each decision tree to ensure that each tree is unique and diverse. The final prediction of the random forest is determined by taking the mode (most frequent) of the predictions of all the individual decision trees.

Random forests are known to be highly accurate and robust to noise and outliers in the data, making them a popular choice for classification problems. In our case, we are using a random forest classifier to predict whether a patient has diabetes or not based on their clinical and demographic features.

Maternal Health Risk Prediction

Support Vector Machine

SVM is a supervised machine learning algorithm that can be used for both classification and regression. The goal is to find a hyperplane that maximizes the margin between the different classes of data. If the data cannot be separated by a linear hyperplane, SVM can use a kernel trick to map the data to a higher-dimensional space. It is a powerful and widely used algorithm, but can be computationally expensive when dealing with large datasets and requires careful selection of hyperparameters to achieve good performance.

Support Vector Machines (SVMs) are powerful tools for classification tasks, such as predicting heart stroke. They work by finding the optimal boundary between two classes of data points, which is the distance between the decision boundary and the closest data points from each class. SVMs are effective for predicting heart stroke, with high accuracy rates and the ability to handle large datasets.

SVM is trained on a set of features that are associated with an increased risk of stroke, such as age, blood pressure, cholesterol levels, and smoking status. The SVM then uses these features to classify new individuals as either being at high or low risk for stroke.

Train test size was set as 70-30.

For implementing the SVM algorithm we used the sklearn library and classified it as “Linear”, as in the data set all the entities are linearly plotted and not non-linearly.

3. Random Forest :

Random Forest is used for supervised learning tasks including classification, regression, and others. It is an ensemble learning technique that integrates various decision trees to provide predictions that are more reliable and accurate.

A Random Forest model consists of a collection of decision trees, each of which is trained using a portion of the data that is chosen at random. In order to lessen overfitting and boost generalized performance, the trees are built using a random subset of features.

Each decision tree in the forest separately categorizes or predicts the outcome based on the input data during the prediction phase, and the ultimate outcome is decided by taking the majority vote or averaging the outputs of all the trees.

High accuracy, resistance to overfitting, and handling huge datasets with numerous variables are some of the reasons we chose to go with random forest algorithm over others.

Train test size was set as 70-30.

Other hyperparameters were tuned and the most accurate results were obtained with Number of decision trees equal to 100.

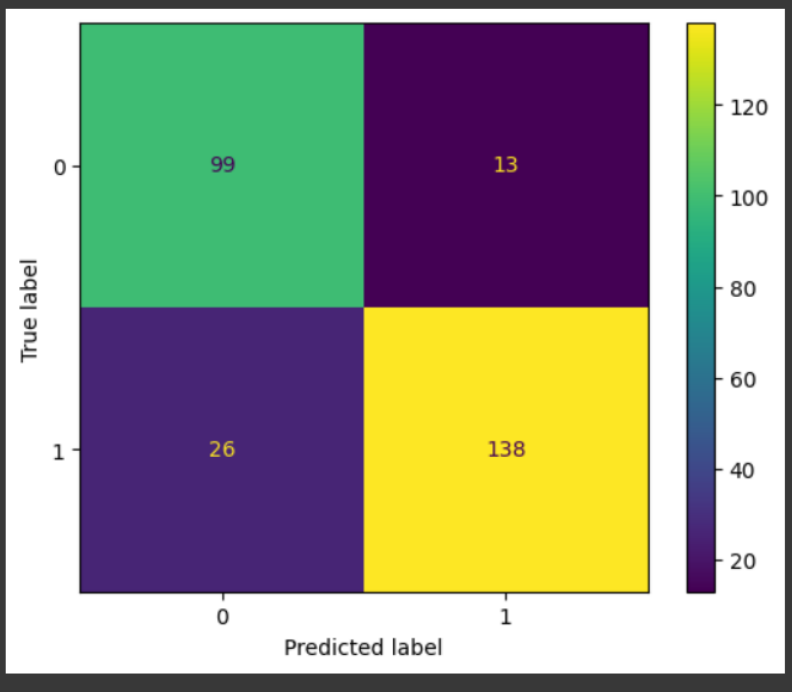
V. Model Evaluation

Heart Stroke Prediction

- Confusion Matrix:
  A confusion matrix is a tool used to evaluate the performance of a classification model. It summarizes the model's predictions and actual outcomes in a tabular format and provides metrics such as accuracy, precision, recall, and F1 score. The benefits of using a confusion matrix include identifying errors, evaluating model performance, and creating visualizations to communicate results. By using a confusion matrix, you can better understand your model's performance and identify areas for improvement.

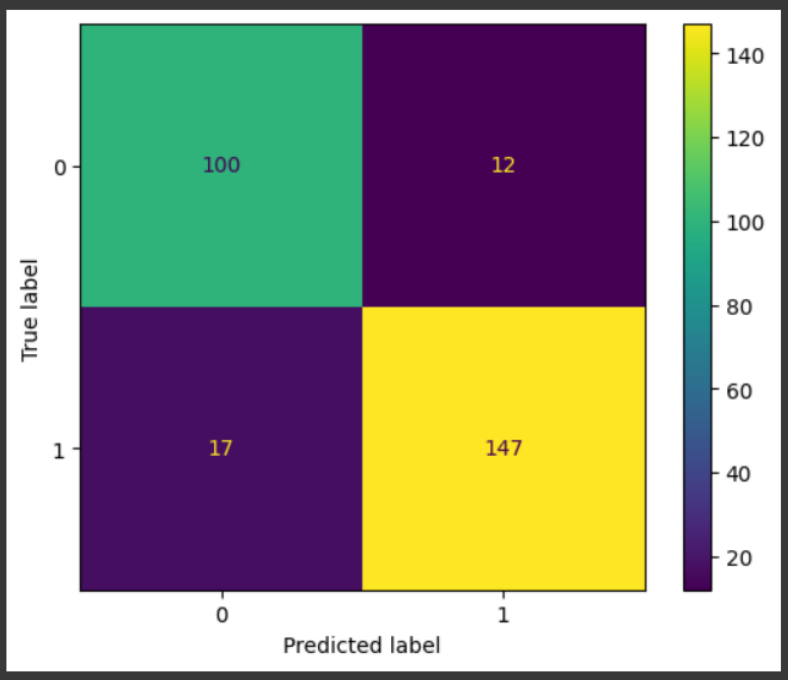
  Confusion Matrix of SVM:

```
labels = [0,1]
confusion_matrix_svm= confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix_svm, display_labels=labels)
disp.plot();
```



Confusion Matrix of Random Forest:

```
labels = [0,1]
confusion_matrix_rf= confusion_matrix(y_test,pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=confusion_matrix_rf, display_labels=labels)
disp.plot();
```

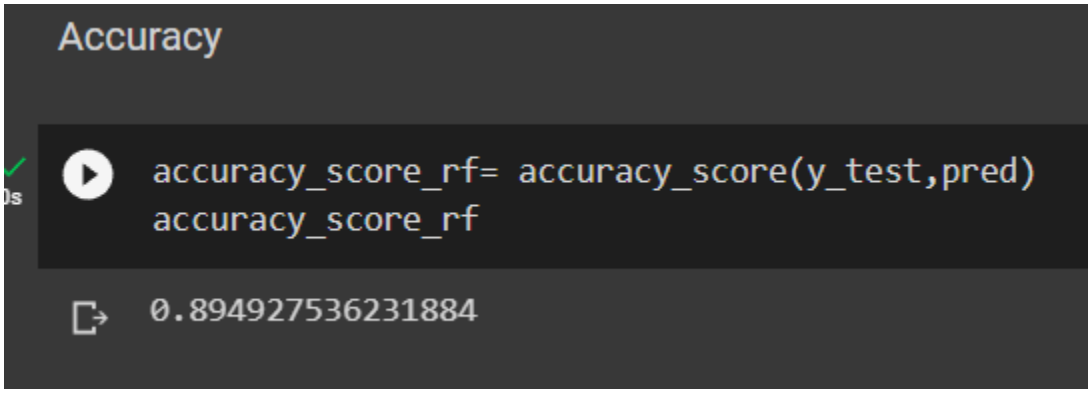


- Accuracy Score:
  Determining a model's accuracy score is crucial since it enables us to assess how successfully the model is carrying out the task. The accuracy score is a measure that expresses how many of the model's predictions were accurate.
  We can also decide how to improve the model and eventually ensure that it is successful for its intended use by periodically analyzing the accuracy score of our model during the development and testing phases.

Accuracy of model created using SVM:

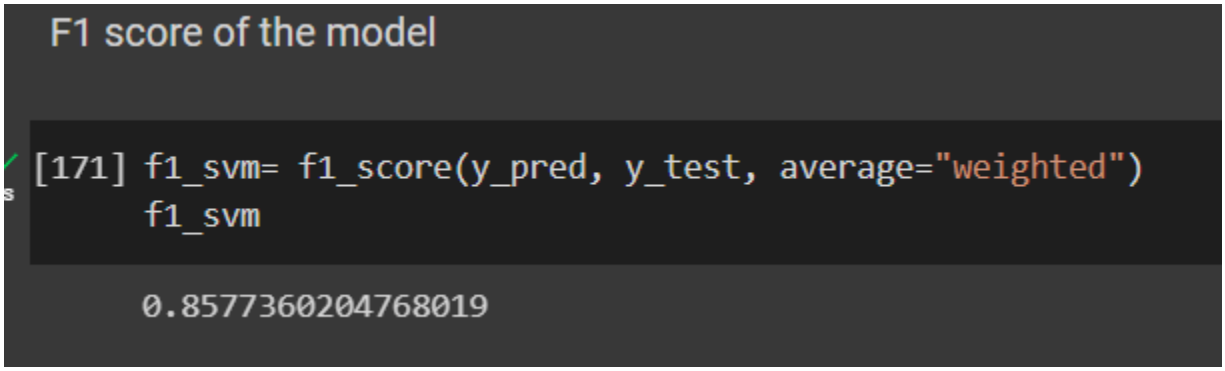
```
[170] accuracy_score_svm= accuracy_score(y_test, y_pred)
      accuracy_score_svm

      0.8586956521739131
```

Accuracy of model created using Random Forest:

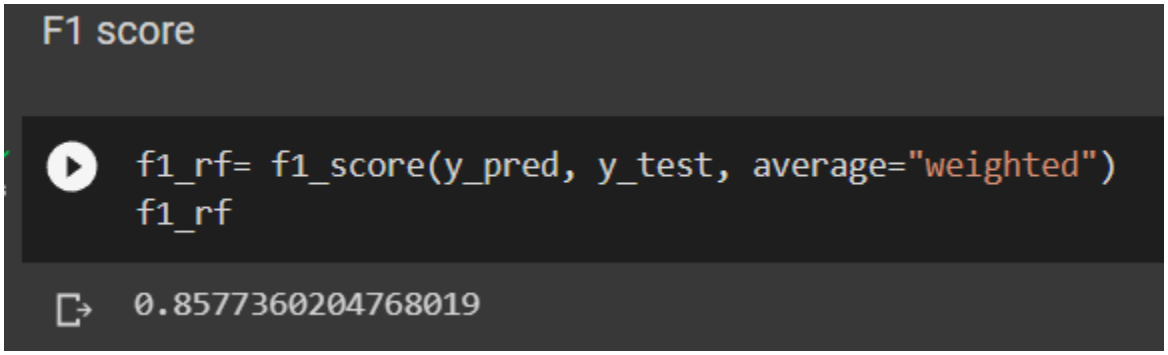


- F1 score:
  The F1 score is a useful performance metric in machine learning that takes both precision and recall into account, making it a better judge of overall performance whereas accuracy can sometimes be misleading.

  F1 score of SVM:



  F1 score of Random Forest:

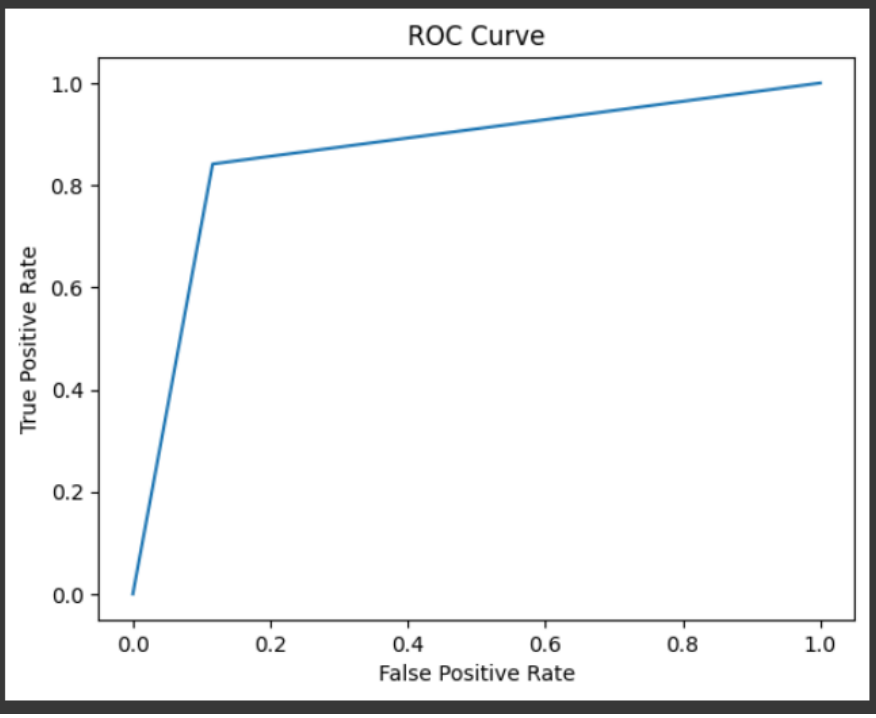


- ROC Curve and Area under the ROC curve:

  A ROC curve is a graph that shows the performance of a binary classification model by plotting the True Positive Rate against the False Positive Rate at different thresholds. The area under the ROC curve (AUC-ROC) is a measure of the classifier's ability to distinguish between positive and negative classes. Using a ROC curve and AUC-ROC helps in evaluating model performance, identifying the optimal threshold, and visualizing the results. They are useful tools for selecting the best classification model and understanding its performance

  ROC curve for SVM:

```
[106] fpr_svm, tpr_svm, _ = roc_curve(y_test, y_pred)

      #create ROC curve
      plt.plot(fpr_svm,tpr_svm)
      plt.title("ROC Curve")
      plt.ylabel('True Positive Rate')
      plt.xlabel('False Positive Rate')
      plt.show()
```

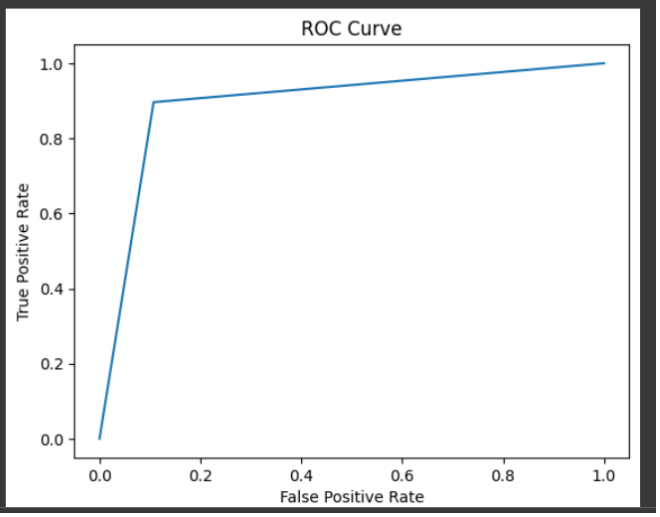


  Area under the curve:

```
[107] score_svm= roc_auc_score(y_test, y_pred)
      score_svm

      0.8626959930313588
```

ROC Curve for Random Forest:

```
fpr_rf, tpr_rf, _ = roc_curve(y_test, pred)

#create ROC curve
plt.plot(fpr_rf,tpr_rf)
plt.title("ROC Curve")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



Area under the curve:

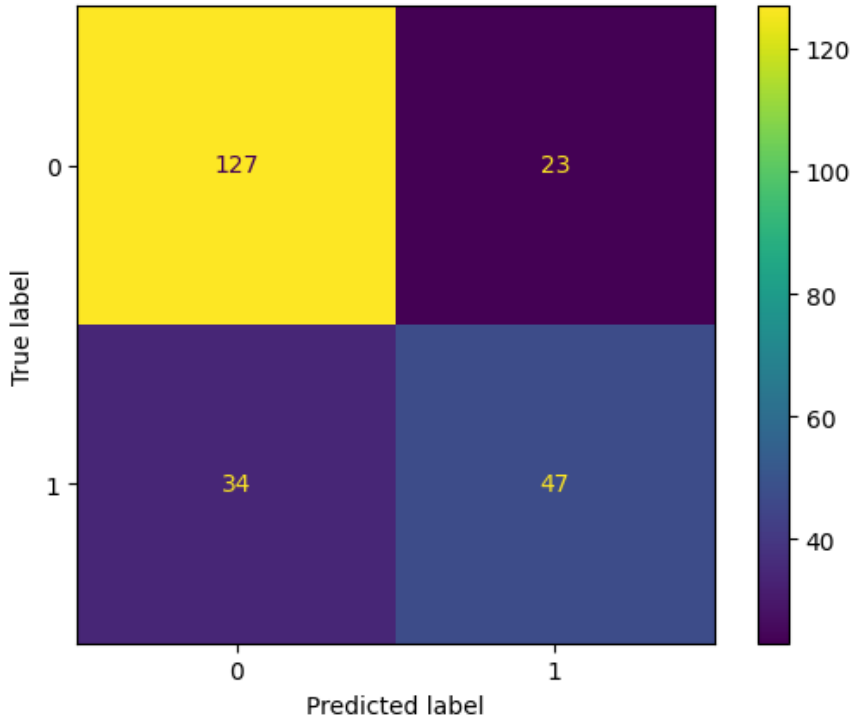
```
[117] score_rf= roc_auc_score(y_test, pred)
      score_rf

      0.8945993031358885
```

Inference:

From the performance metrics applied on both models we can see that Random Forest Algorithm works better.The margin is negligible but our topic necessitates maximum accuracy possible as it predicts if you are at risk of getting a stroke.

Diabetes Prediction

1. Confusion Matrix



2. Accuracy Score

Accuracy

```
[89] accuracy_score_rf= accuracy_score(y_test,pred)
     accuracy_score_rf
```

0s

```
0.7532467532467533
```

3. F1 score

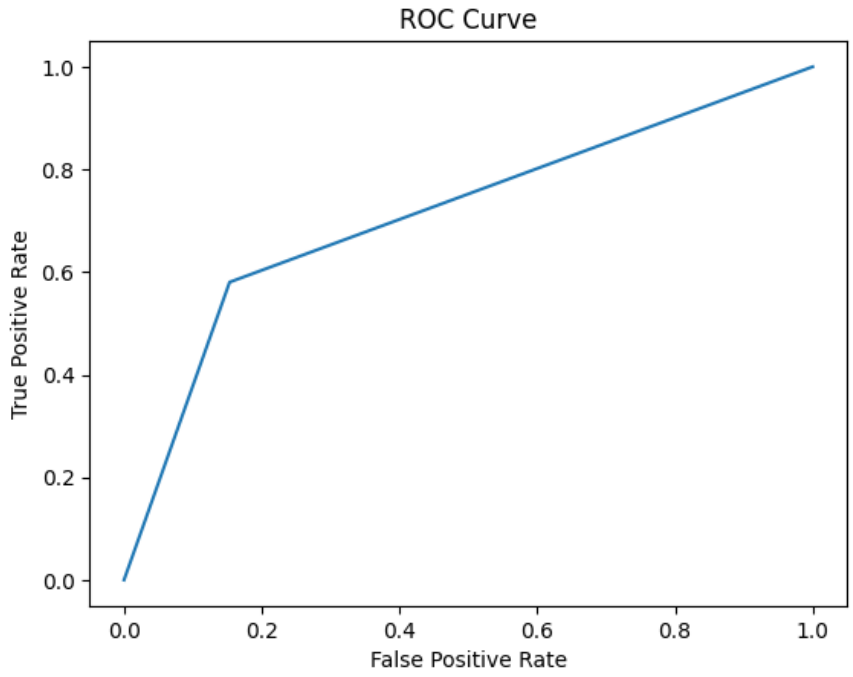
F1 score

```
[90] f1_rf= f1_score(pred, y_test, average="weighted")
     f1_rf
```

0s

```
0.7578706508882612
```

4. ROC Curve and Area under the ROC curve:



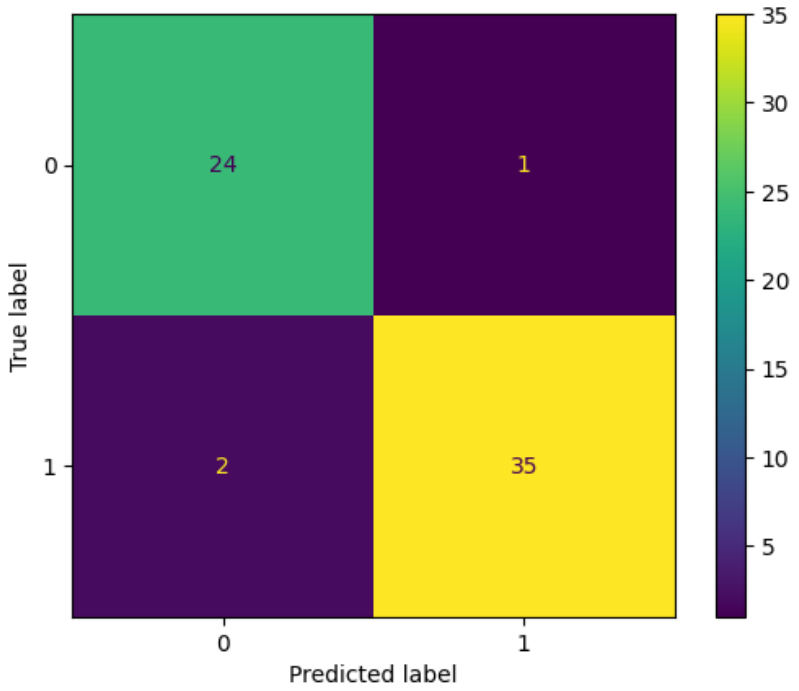

```
[92] score_rf= roc_auc_score(y_test, pred)
     score_rf

     0.7134567901234568
```

Inference:
By using random forest algorithm we get accuracy as 75%(Approx) and F1 score as 75%(Approx). The margin is negligible but our topic necessitates maximum accuracy possible as per the dataset as it predicts if you are at risk of getting diabetes or not.

Maternal Health Risk Prediction

Support Vector Machine

Confusion Matrix:



Accuracy Score:

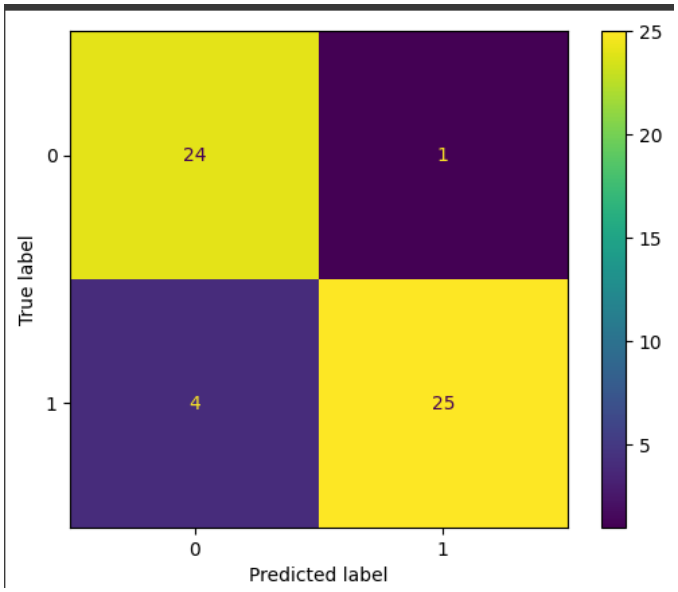
```
Accuracy of the model

[36] accuracy_score_svm= accuracy_score(y_test, y_pred)
     accuracy_score_svm

     0.7254901960784313
```

F1 Score:

```
F1 score of the model

     f1_svm= f1_score(y_pred, y_test, average="weighted")
     f1_svm

     0.7430885544475808
```

<u>Random Forest Classifier</u>

Confusion Matrix:



Accuracy Score:

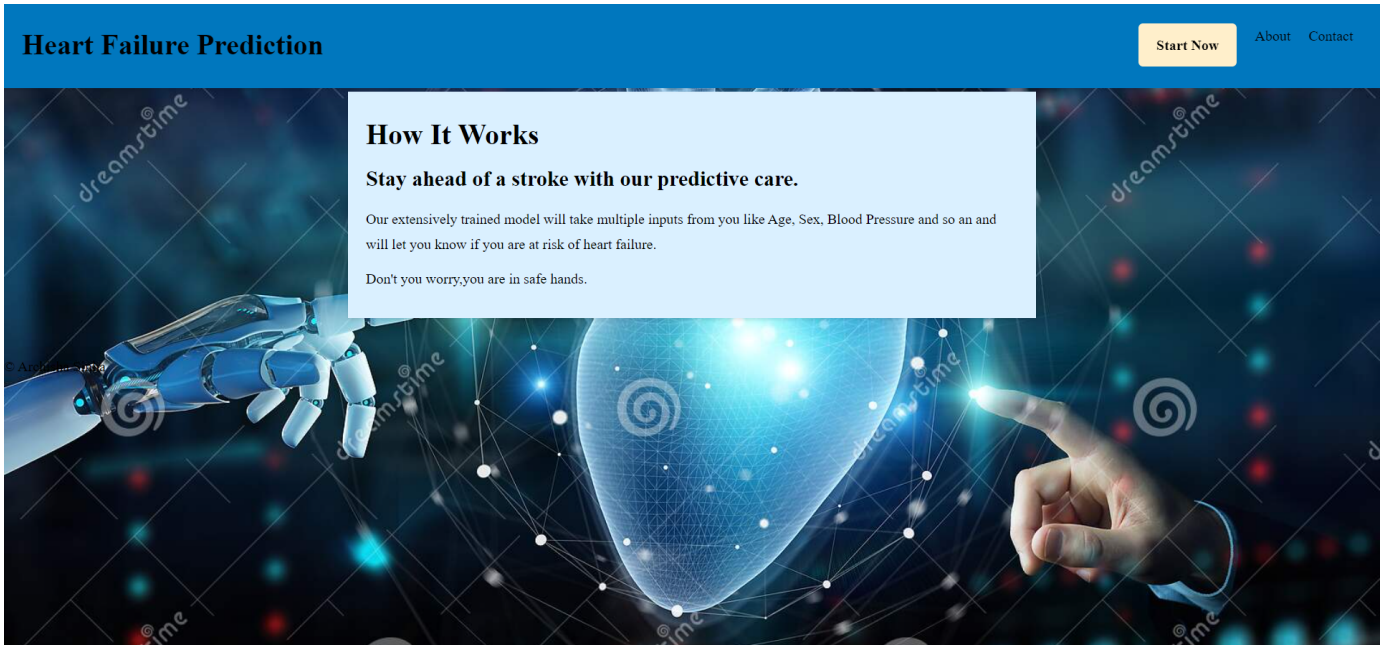
```
Accuracy

accuracy_score_rf= accuracy_score(y_test,pred)
accuracy_score_rf

0.7941176470588235
```

F1 Score:

```
F1 score

[41] f1_rf= f1_score(y_pred, y_test, average="weighted")
     f1_rf

     0.7430885544475808
```

VI. User Interface-Website

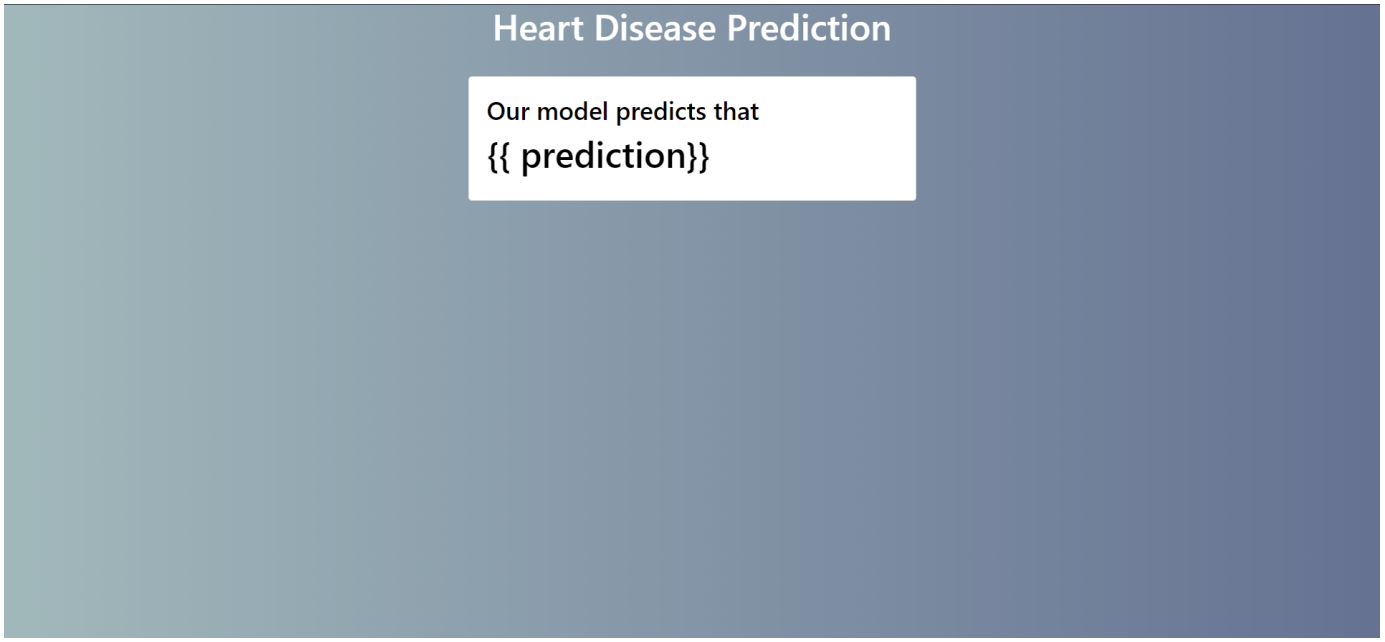
⇒ **A home page is created with HTML which briefs about our Heart Stroke Prediction Model.**



.

⇒ **After clicking on the "Start Now" button, it redirects to a page where the user has to input some of the asked information which will help the machine to predict with accuracy that the user is at risk of getting a heart stroke or not.**

⇒ After the user has entered all the necessary details, our prediction model starts its prediction and shows the results on a new page, made as follows from html.



Some of the advantages of our prediction model is as follows:

1. Early detection: Early detection of heart diseases is critical as it can allow for prompt treatment, which can prevent the condition from worsening and causing further damage.
2. Convenience: A website that predicts heart diseases can be accessed from anywhere and at any time.
3. Cost-effective: Using a website to predict heart diseases can be a cost-effective alternative to traditional medical check-ups.

VII. Learning from the Project

Heart Stroke Prediction

This project has been an extremely enriching experience for us as it provided us with hands-on experience in implementing various machine learning algorithms and creating visualizations to gain insights from data. We learned about the importance of domain knowledge and how it can help us in selecting relevant features for our model.

During the course of this project, we learned about two popular classification algorithms, Support Vector Machine (SVM) and Random Forest, and implemented them to predict heart disease in patients. We also explored various data preprocessing techniques like handling missing values and feature scaling to improve the performance of our models. Additionally, we gained domain knowledge about the various features that can indicate the presence of heart disease in patients.

Of the many new things we learned during this project one was on how to create a website to make our project more user-friendly and accessible to the general public. Our website takes several input patient data that covers a wide range like heart rate and BP and so on and using our model gives a prediction about whether or not they are at risk of heart disease.Our model boasts an accuracy of 90 percent or so. This added an extra dimension to our project, making it more accessible and engaging for users.

Another important aspect of our project was data visualization. We used Python's Seaborn library to create various visualizations of the data, such as scatterplots, bar plots, and box plots. These visualizations helped us gain insights and understand patterns in the data, such as the correlation between age and heart disease, or the relationship between cholesterol and resting blood pressure.

We also learned about heatmaps and how they can be used to visualize correlations between different features in the data. And lastly, we learnt about how to make a repository on github.

In addition to the technical skills we acquired, this project also taught us the importance of collaboration and communication. We worked together as a team and divided the tasks according to our strengths and weaknesses. We held regular meetings to discuss our progress and any challenges we faced. This helped us to stay organized and on track with our project goals. We also learned how to effectively communicate our findings and inferences to a layman who may not have a technical background.

Furthermore, this project allowed us to apply our knowledge to a real-world problem. Heart disease is a major health issue that affects millions of people worldwide. By developing a machine learning model to predict heart disease, we felt that we were contributing to a larger goal of improving public health. This project helped us to see the potential of machine learning and data science in creating positive social impact.

Diabetes Prediction

In this project, we utilized the Random Forest algorithm to predict diabetes using the Pima Indian Diabetes dataset. The Random Forest algorithm is a powerful and widely-used machine learning algorithm that works well with complex datasets. Through the implementation of the Random Forest algorithm, we were able to predict diabetes with a high degree of accuracy. This demonstrates the potential of machine learning algorithms in healthcare and medicine.

Preprocessing and visualization techniques are essential in preparing data for machine learning models. We utilized techniques such as data cleaning, handling missing values, and data visualization to ensure that the data is in the correct format for our model. By visualizing the data, we were able to gain insights into the relationships between the various features and the target variable. These insights were then used to further refine the data and improve the accuracy of our model.

Model evaluation is an essential component of any machine learning project. In this project, we utilized confusion matrices to evaluate the performance of our model. Confusion matrices allowed us to visualize the true positive, true negative, false positive, and false negative predictions of our model. Through this evaluation, we were able to identify areas of improvement in our model and fine-tune it for better performance.

Overall, this project serves as a valuable example of the power of machine learning in healthcare and medicine.

Maternal Health Risk Prediction

Developing a maternal health risk prediction model in machine learning (ML) involves working with sensitive and complex data related to pregnancy outcomes and complications. Here are some of the key learnings from a maternal health risk prediction ML project:

Data collection and preprocessing are critical: Collecting high-quality data that is clean, accurate, and complete is essential for building a robust maternal health risk prediction model. Preprocessing data is equally important as it helps to eliminate errors, inconsistencies, and irrelevant information.

Feature selection is important: The selection of relevant features can have a significant impact on the performance of the maternal health risk prediction model. It is important to choose the most important features that are most likely to be associated with the outcome of interest.

Choosing an appropriate ML algorithm is crucial: There are several ML algorithms available for classification problems, and choosing the right one is crucial for building an accurate and robust maternal health risk prediction model. The selected algorithm should be able to handle the type and size of the data and should have good performance metrics.

Model performance evaluation is key: Evaluating the performance of the maternal health risk prediction model using metrics such as accuracy, precision, recall, F1-score, and area under the curve (AUC) is critical. Model performance should be assessed on an independent test dataset to avoid overfitting.

VIII. Challenges Faced

Heart Stroke Prediction

The first challenge was choosing an appropriate dataset and gaining knowledge about different features. Many datasets had features which were quite irrelevant and some had quite less data for prediction.

Another challenge we faced was selecting the appropriate machine learning algorithm for our project. We had to consider various algorithms, such as SVM and Random Forest, before selecting the best one that could predict heart disease with optimal accuracy. This required a deep understanding of the strengths and weaknesses of each algorithm and their suitability for our data.

Furthermore, we encountered some difficulties in creating a website for our project. As we were making a website for the first time, we had to learn the basics of programming and user interface design, which took some time and effort. Additionally, we had to ensure that our website was user-friendly and visually appealing, which required attention to detail and good design skills.

Lastly, while connecting our project to a webpage, we had to face the challenge of web development. This required knowledge of HTML and basics of web development which we had to learn. Finally we lacked knowledge about creating a repository on Github. We learned it through some tutorials and reliable sources.

Diabetes Prediction

Data integrity is one of the greatest obstacles in any data-related endeavour. Inaccurate, insufficient, or inconsistent data can result in skewed or inaccurate model predictions. We must ensure that the data in this endeavour is accurate, comprehensive, and pertinent to the task at hand.

Classification tasks can be significantly hampered by imbalanced datasets. In this endeavour, the dataset is imbalanced, meaning that one class (i.e., patients without diabetes) contains substantially more instances than the other (i.e., patients with diabetes). This can result in erroneous model predictions, so we must employ techniques such as oversampling or undersampling to resolve this issue.

While machine learning models can generate accurate predictions, they are often opaque, making it difficult to interpret the results. In this endeavour, we must ensure that the model is interpretable and can provide insight into the contributing factors to the prediction.

Such healthcare-related initiatives may entail sensitive data, and it is crucial to ensure that the data is handled with the uttermost integrity and confidentiality. Consequently, we must work on this endeavour with data security and access restrictions in mind.

IX. Conclusion

The key takeaways of this project are:

1. Importance of domain knowledge: The project highlighted the importance of having a good understanding of the domain when working on a machine learning project. In this case, it was important to have knowledge of heart disease, diabetes and its different features in order to select appropriate variables and algorithms.

2. Preprocessing is crucial: Preprocessing plays a crucial role in the success of a machine learning project. This project involved handling missing values, standardizing features, and encoding categorical variables. This helped to ensure that the data was in a format that the algorithms could work with and produced better results.

3. Data visualization is important: Data visualization can be used to identify trends and relationships in the data, which can help in making informed decisions about feature selection and algorithm selection. The project used various types of graphs such as histograms, scatter plots, and box plots to visualize the data.

4. Algorithm selection is important: The project used two different algorithms, SVM and Random Forest, to predict heart disease. Both algorithms produced good results, but Random Forest performed slightly better. This highlights the importance of selecting the right algorithm for the problem at hand.

5. GUI can make the project more accessible: The project created a GUI for the prediction model, making it more user-friendly and accessible to a wider audience. This can be a useful skill for data scientists to have, especially if they want to create tools for non-technical users.

6. Web development skills can enhance the project: Connecting the project to a webpage can make it more accessible and increase its visibility. This can involve learning skills such as HTML, CSS, and JavaScript, which can be valuable for data scientists who want to create interactive data visualizations or deploy machine learning models on the web.

Overall, this project was a great learning experience for us. We gained hands-on experience in machine learning, data preprocessing, data visualization, and GUI creation. We are confident that the skills and knowledge we gained during this project will be valuable in our future endeavors in data science and machine learning.

X. References

Heart Failure Prediction Dataset | Kaggle

Diabetes Dataset. (n.d.). Diabetes Dataset | Kaggle. https:///datasets/mathchi/diabetes-data-set

Maternal Health Risk Prediction Dataset