HEART FAILURE ANALYSIS

GROUP MEMBER:

Sai Srija Achukolu

Lovely Professional University

12110309

CONTENTS

**Topics**  **Page No.**

1)Acknowledgement 3

2)Project Objective 4

3)Project Scope 5

4)Data Description 6-7

5)Data Pre-Processing 8-19

6)Model Building 20-43

7)Test Dataset 44

8)Codes 45-66

9)Future Scope of Improvements 67-69

10)Certificates 70

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my faculty, Prof.Arnab Chakraborty for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.I am obliged to my project team members for the valuable information provided by them intheir respective fields. I am grateful for their cooperation during the period of my assignment.

I also appreciate the time and effort you invest in creating a positive and inclusive learning environment. Your patience, understanding, and willingness to answer questions have fostered an atmosphere where students feel comfortable and encouraged to participate actively.

Sai Srija Achukolu

PROJECT OBJECTIVE

Heart failure is a critical medical condition that affects numerous individuals worldwide. By leveraging the power of artificial intelligence (AI) and machine learning (ML), we can develop predictive models to identify the risk factors associated with heart failure. In this project, we will focus on the factors of age, sex, and previous diseases undergone by individuals to predict the likelihood of heart failure.

To achieve this, we will gather a dataset that includes relevant features such as age, sex, previous diseases (such as hypertension, diabetes, etc.), and the presence or absence of heart failure. This dataset will serve as the basis for training and evaluating our models. For the heart failure prediction, we will implement three different ML algorithms: logistic regression, decision tree classifier, and random forest classifier. Each algorithm offers unique advantages in terms of interpretability, complexity, and accuracy. By comparing their performance, we can determine which model yields the best results for this particular task.

Finally, we will interpret the results, examining the importance of each feature in predicting heart failure. This analysis will provide insights into the role of age, sex, and previous diseases in influencing the likelihood of heart failure.

Overall, this project aims to develop an AI and ML-based heart failure analysis system that can provide valuable predictions and insights based on patient data. By leveraging logistic regression, decision tree classifier, and random forest classifier models, we seek to enhance our understanding of the factors contributing to heart failure and facilitate early detection and prevention strategies.

PROJECT SCOPE

The scope of this project encompasses the development and evaluation of an AI and ML-based heart failure analysis system. The project will focus on analyzing the relationship between age, sex, previous diseases undergone, and the likelihood of heart failure. The following aspects are included within the project scope:

* Data Collection: The project will involve gathering a dataset that includes relevant information such as age, sex, previous diseases, and the presence or absence of heart failure. The dataset will be obtained from reliable sources or existing databases related to heart health.
* Data Preprocessing: Prior to training the models, the dataset will undergo preprocessing steps to handle missing values, encode categorical variables, and perform feature scaling if necessary. This step ensures the data is prepared for training and evaluation.
* Exploratory Data Analysis: Exploratory data analysis will be conducted to gain insights into the relationships between age, sex, previous diseases, and heart failure. This analysis will help identify any patterns, trends, or correlations present in the data.
* Model Development: The project will involve implementing three machine learning algorithms: logistic regression, decision tree classifier, and random forest classifier. These algorithms will be used to develop predictive models based on the provided dataset.
* Model Training and Evaluation: The dataset will be split into training and testing subsets. The training data will be used to train the models, while the testing data will be used to evaluate their performance.

**DATA DESCRIPTION**

Data Description: The given train dataset has 299 rows and 12 columns.

|  |  |  |
| --- | --- | --- |
| Columns | Attribute name | Type |
| age | age | continous |
| anemia | anemia | categorical |
| diabetes | diabetes | categorical |
| Sex | sex | categorical |
| High bp | high bp | categorical |
| smoking | smoking | categorical |
| death | death | categorical |
| platelets | platelets | continous |

NOTE: The other columns which are not mentioned in the column row are not used in the code

Now,we will process the data.The methodology followed is given below:

**1) Split the data into features and target variable**

**2) Split the data into training and testing sets**

**3)Initialize models**

**4) Train models**

**5)Make predictions**

**6)Calculate accuracy score**

**7)Create a bar graph to compare accuracies**

Note: Checking for null values. If null values are present, we will fill them or drop the row containing the null value based on the dataset.

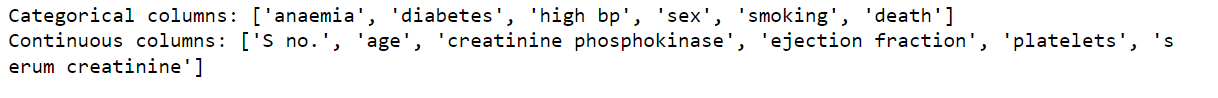
**Data pre-processing**

Segregating categorical and continuous variables is important in data analysis and modeling because these two types of variables require different treatment and analysis techniques. Here are some reasons why it is beneficial to segregate them:

Data Understanding: Separating categorical and continuous variables helps in gaining a better understanding of the nature of the data. Categorical variables represent distinct categories or groups, while continuous variables represent numerical values within a range. Identifying and categorizing variables based on their type allows for a more intuitive interpretation of the data.

Summary Statistics: Categorical and continuous variables require different summary statistics. For categorical variables, we typically use frequency counts and percentages to summarize the distribution of categories. In contrast, for continuous variables, summary statistics like mean, median, standard deviation, and quartiles provide insights into the central tendency, dispersion, and shape of the data distribution.

Visualization: Segregating variables enables appropriate visualization techniques for each type. Categorical variables can be visualized using bar charts, pie charts, or stacked column charts to display the distribution of categories. Continuous variables, on the other hand, can be visualized using histograms, box plots, or scatter plots to understand the distribution, variability, and relationships with other variables.



**This is the output when categorical and continuous type of variables are segregated.**

**Calculating null values**

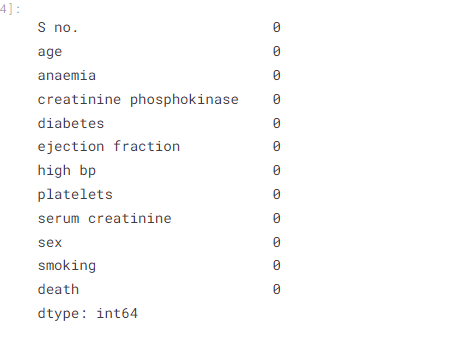
Calculating null values in a dataset is an essential step in data analysis and preprocessing. Understanding the importance of calculating null values can help ensure the integrity and quality of the data being analyzed. Here are some key reasons why calculating null values is significant:

Data Integrity: Null values represent missing or unknown information in a dataset. By identifying and calculating the null values, data analysts can gain insights into the completeness and accuracy of the dataset. This information is crucial for making informed decisions about handling missing data and ensuring the overall integrity of the dataset.

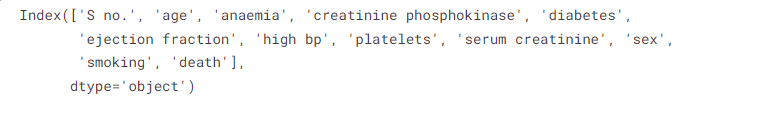
Data Quality Assessment: Null values can indicate potential data quality issues. High numbers of null values in certain variables or columns might suggest data collection or entry errors. Identifying and quantifying these null values allows data analysts to assess the quality of the dataset, identify problematic areas, and take appropriate steps to improve data quality.

Data Preprocessing: Null values need to be handled appropriately during data preprocessing to avoid bias or errors in subsequent analyses or model development. Calculating the null values helps in determining the extent of missing data and choosing suitable strategies for handling them, such as imputation (replacing missing values with estimated values) or removing rows or columns with excessive null values.

Statistical Analysis: Null values can affect statistical analyses and modeling outcomes. Calculating the null values helps analysts understand the impact of missing data on statistical measures such as means, standard deviations, correlations, and regression coefficients. By quantifying the extent of missingness, analysts can make informed decisions about the suitability and validity of statistical analyses conducted on the dataset.



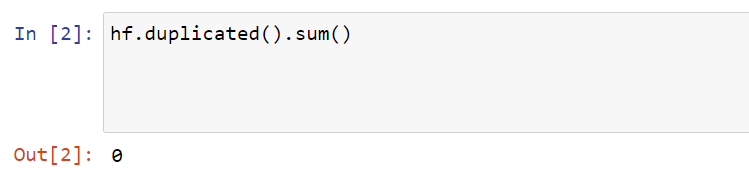
**The above output is the count of null values**

****

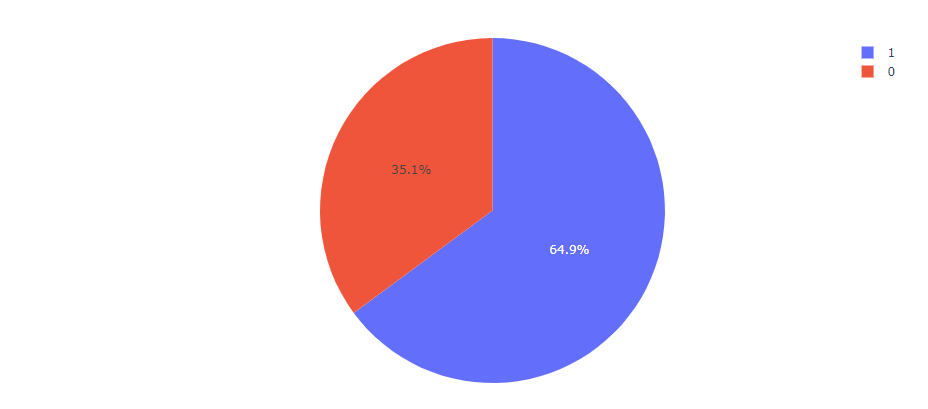
**The above output is the number of variables used**

****

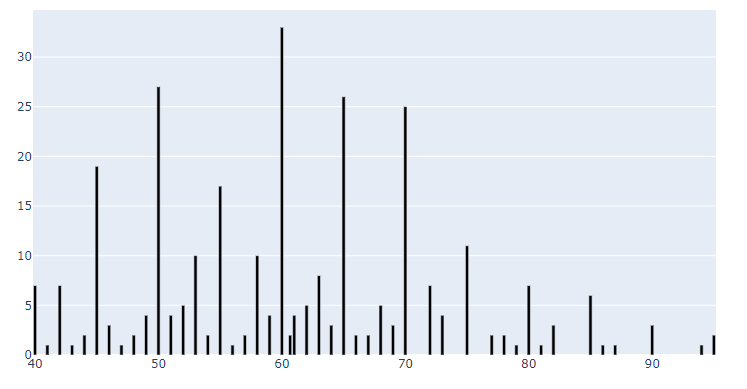
**Here 299,12 is the shape of the dataset which means it has 299 rows and 12 columns**



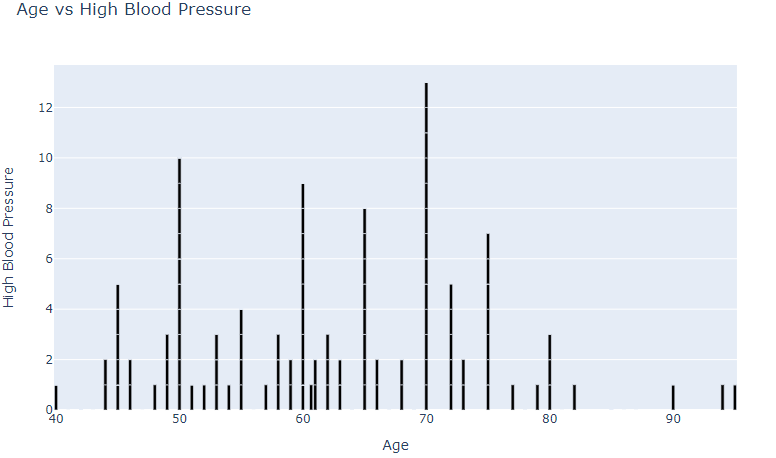
**The above output is 0 as there are no duplicates**



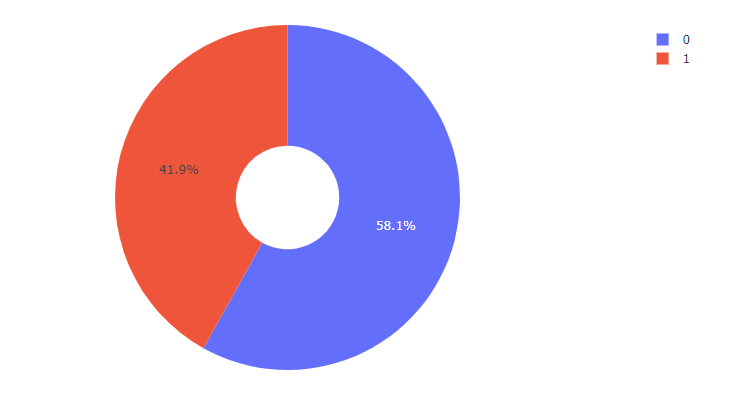
The above pie chart gives information about heart failure based on sex



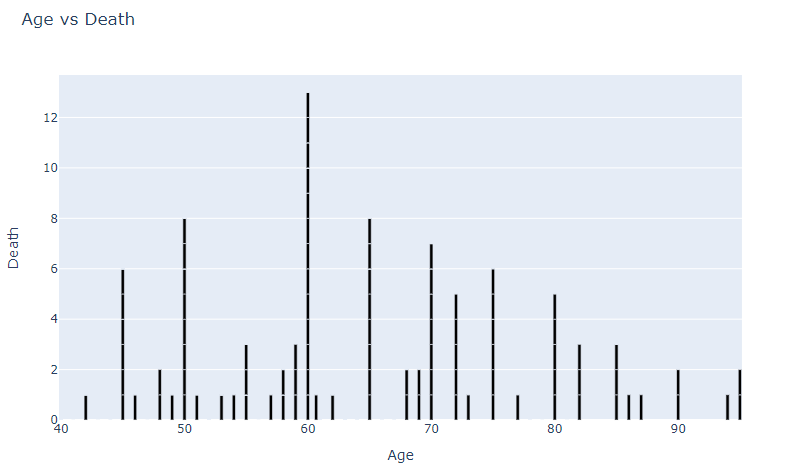
The above bar graph gives information about heart failure based on age



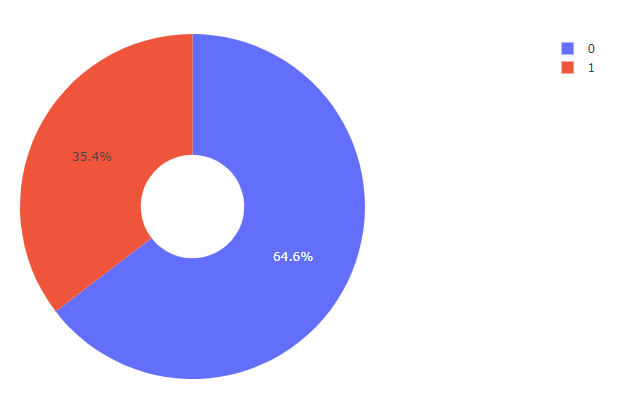
The above bar graph depicts age vs blood pressure



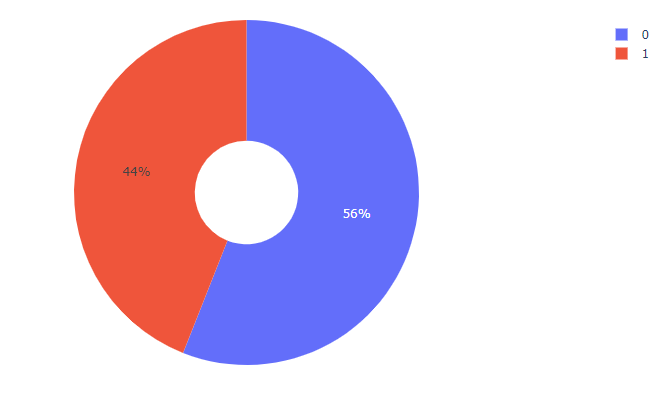
The above pie chart gives information about high bp based on sex

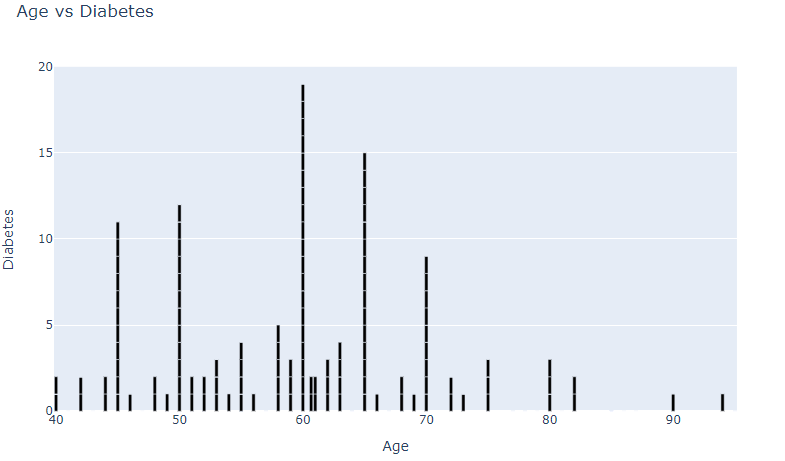


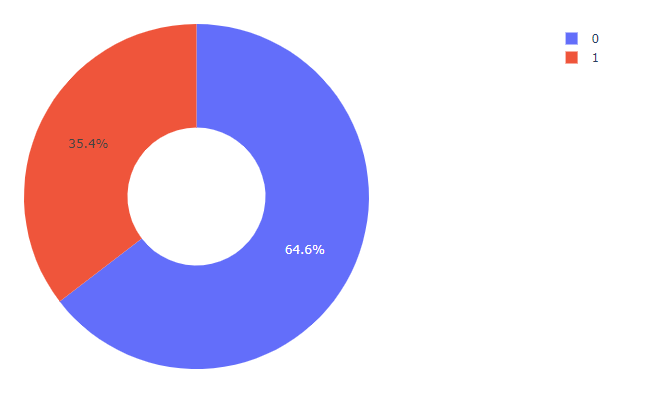
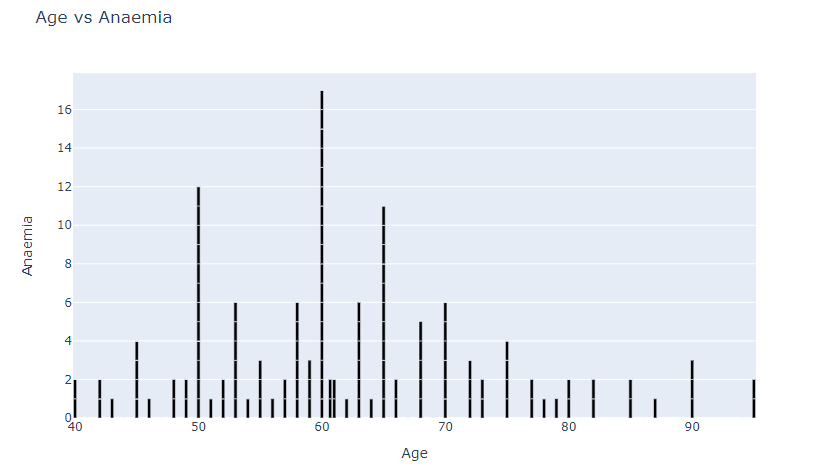
The above bar graph depicts age vs blood pressure



The above piechart depicts death based on sex

The above pie chart depicts diabetes based on sex



The above pie chart depicts anemia based on sex

The above bar graph depicts age vs anemia

Bar graphs and pie charts are widely used in data visualization for different purposes, and understanding their importance can help in effectively communicating data insights. Here are the key reasons why bar graphs and pie charts are significant:

Bar Graphs:

Comparison: Bar graphs are excellent for comparing and displaying the values of different categories or groups. They provide a visual representation of the magnitude or frequency of each category, making it easy to identify patterns, trends, or differences between the groups.

Categorical Data: Bar graphs are particularly useful for visualizing categorical data, where each category represents a distinct group or variable. They allow you to present the distribution or frequency of each category in a clear and straightforward manner.

Quantitative Data: Bar graphs can also represent quantitative data by discretizing the data into categories or ranges. For example, you can use a bar graph to display the number of observations falling within specific numerical intervals.

Comparison over Time: Bar graphs can be used to show changes or comparisons over time by utilizing grouped or stacked bar charts. This makes it easy to observe trends or variations in different categories across different time periods.

Pie Charts:

Composition: Pie charts are ideal for illustrating the composition or proportions of different categories within a whole. They provide a visual representation of how the parts relate to the whole and help in understanding the relative contribution or distribution of each category.

Percentage Representation: Pie charts automatically display the proportion of each category as a percentage of the whole, making it easy to compare the relative sizes of different categories at a glance.

Limited Categories: Pie charts are best suited for representing a small number of categories (typically less than six or seven) to maintain clarity and avoid visual clutter. They work well when you have a few distinct categories to represent.

Visual Appeal: Pie charts are visually appealing and can be an effective way to engage the audience. The circular shape and the division of the pie into slices can make the data presentation more visually interesting and memorable.

Both bar graphs and pie charts have their own advantages and are suitable for different scenarios. Choosing the appropriate visualization technique depends on the nature of the data, the purpose of the visualization, and the message you want to convey.

MODEL BUILDING

**Splitting data for training and testing purpose**

We split the given train dataset into two parts for training and testing purpose. The split ratio we used is 0.75 which indicates we used 80% data for training purpose and 20% data for testing purpose. We will be using the same split ratio for all the models trained.

Splitting data into training and testing sets is a critical step in machine learning and model development. It helps in evaluating the performance and generalization ability of a model. Here are the key reasons why splitting data into training and testing sets is important:

Model Evaluation: The primary purpose of a testing set is to evaluate the performance of a trained model. By using unseen data, you can assess how well the model generalizes to new, unseen examples. This evaluation is crucial in determining whether the model is performing accurately or if it suffers from overfitting (performing well on training data but poorly on unseen data) or underfitting (performing poorly on both training and unseen data).

Bias and Variance Assessment: The training and testing sets help in analyzing the bias and variance of a model. Bias refers to the error introduced by approximating real-world problems with a simplified model, while variance refers to the model's sensitivity to small fluctuations in the training data. Splitting the data allows you to identify whether your model suffers from high bias (underfitting) or high variance (overfitting) based on the training and testing performance.

Hyperparameter Tuning: Splitting data into training and testing sets is essential for tuning model hyperparameters. Hyperparameters are settings that are not learned from the data but are specified by the user. By assessing the model's performance on the testing set, you can adjust hyperparameters to improve the model's performance and generalize better.

**Random Forest Classifier Model**

A random forest classifier is a machine learning algorithm that belongs to the family of ensemble methods. It is a supervised learning algorithm used for classification tasks, where the goal is to predict the class or category of an input based on a set of features.

Random forest classifiers are called "random forests" because they are composed of multiple decision trees, and each tree is built using a random subset of the training data and a random subset of the features. This randomness helps to reduce overfitting and improve generalization.

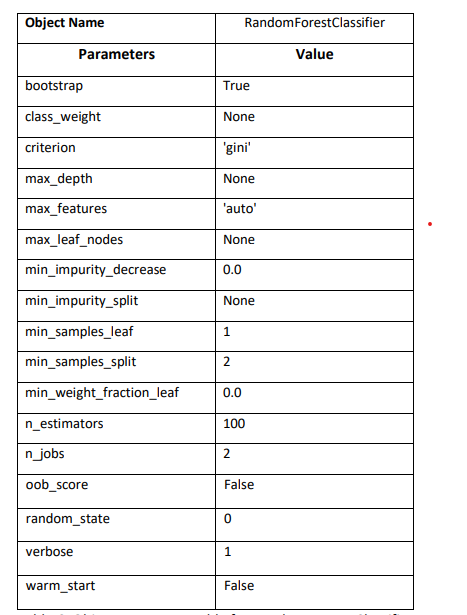
The basic idea behind a random forest classifier is to combine the predictions of multiple decision trees to make a final prediction. Each decision tree in the random forest is trained independently on a different subset of the training data. During the training process, at each node of a decision tree, a random subset of features is considered for splitting, and the best split is chosen based on certain criteria (such as Gini impurity or information gain).

When making predictions with a random forest classifier, each decision tree in the forest independently classifies the input, and the class that receives the majority of votes from the trees is considered as the final predicted class.

Random forest classifiers are known for their robustness, ability to handle high-dimensional data, and resistance to overfitting. They can be used for both binary and multi-class classification problems and have been successfully applied to various domains, such as finance, healthcare, and image recognition.

Ensemble Learning: Random forest classifiers belong to the ensemble learning family of algorithms. Ensemble learning involves combining multiple machine learning models to obtain better predictive performance than using a single model. Random forests combine the predictions of individual decision trees to make the final prediction.

The object description of the Random Forest Classifier used is given below:



DECISIONTREE

A decision tree is a supervised machine learning algorithm that is commonly used for both classification and regression tasks. It is a flowchart-like structure where each internal node represents a test on a feature, each branch represents the outcome of the test, and each leaf node represents a class label or a value.

1. Decision Nodes: Internal nodes in a decision tree represent decisions or tests based on the feature values. The test at each node splits the data into two or more branches, leading to different paths in the tree based on the outcomes of the test.

2. Leaf Nodes: Leaf nodes in a decision tree represent the final predicted class label or the predicted value for a given input. They are the endpoints of the decision process.

3. Feature Selection: Decision trees determine the best feature to split the data at each internal node based on certain criteria such as Gini impurity, entropy, or information gain. These measures assess the homogeneity or purity of the target variable within the resulting subsets after the split.

4. Recursive Partitioning: Decision trees employ a top-down approach known as recursive partitioning. Starting from the root node, the tree recursively splits the data based on the selected features until a stopping criterion is met, such as reaching a maximum depth, achieving a minimum number of samples in a node, or obtaining homogeneous subsets.

5. Handling Categorical and Numerical Features: Decision trees can handle both categorical and numerical features. For categorical features, the tree splits the data based on the categories, creating separate branches for each category. For numerical features, different splitting strategies, such as binary splits or multiway splits, can be used to divide the data.

6. Pruning: Decision trees are prone to overfitting, which means they may become too complex and capture noise or irrelevant details in the training data. To address this, pruning techniques are applied to simplify the tree by removing or merging certain nodes, leading to improved generalization.

7. Interpretability: Decision trees are highly interpretable models, as their flowchart-like structure allows for clear visualization and understanding of the decision-making process. Decision trees can provide insights into the important features and their relative importance for making predictions.

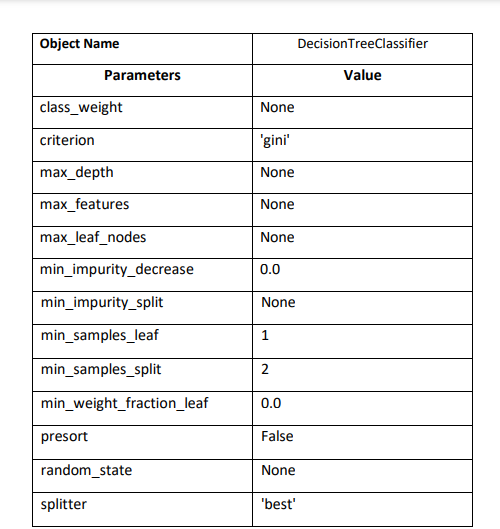
8. Ensemble Methods: Decision trees are often used as building blocks in ensemble methods such as random forests and gradient boosting. These methods combine multiple decision trees to improve predictive accuracy and reduce overfitting.

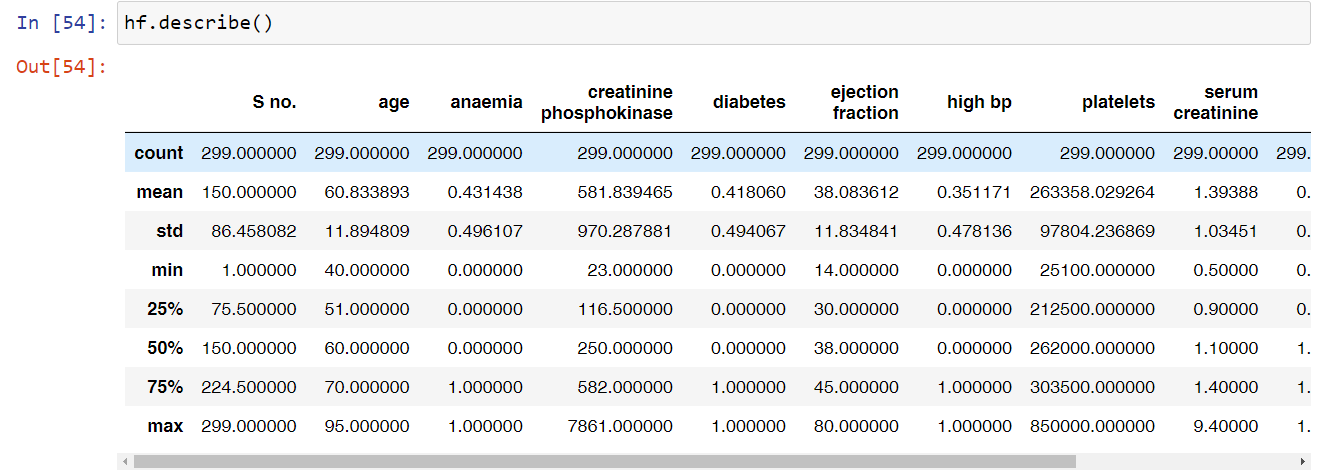
Decision trees are popular due to their simplicity, interpretability, and ability to handle both categorical and numerical features. However, they can be sensitive to small changes in the data and prone to overfitting on complex problems. Techniques like pruning, tuning hyperparameters, and using ensemble methods help overcome these limitations and enhance the performance of decision tree models.

Splitting Criteria: Decision trees use different criteria to evaluate the quality of a split. Some commonly used measures include Gini impurity and entropy. Gini impurity measures the probability of misclassifying a randomly selected element, while entropy calculates the impurity or disorder of a set. These measures help determine the optimal feature and splitting point at each node.

Handling Missing Values: Decision trees can handle missing values in the data. When encountering missing values during the training or prediction process, the algorithm can either assign the majority class/value, propagate the majority class/value down the tree, or consider missing values as a separate category during the split.

**The object description of the Decision Tree Classifier used is given below:**

****



The describe( ) option is used to obtain a statistical summary of a dataset or a specific DataFrame in pandas. It provides descriptive statistics that give insights into the central tendency, dispersion, and shape of the distribution of the data.

When applied to a DataFrame, describe() generates a summary of the numerical columns by default, including the count, mean, standard deviation, minimum value, quartiles (25th, 50th, and 75th percentiles), and maximum value. It can be helpful for getting a quick overview of the data, identifying any outliers, understanding the data ranges, and detecting potential issues such as missing values.

LINEAR REGRESSION

Linear regression is a statistical modeling technique used to establish a relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the variables and aims to find the best-fit line that represents the relationship.

In linear regression, the dependent variable (also known as the target variable or response variable) is a continuous numerical variable that we want to predict or explain. The independent variables (also called predictors or features) are one or more variables that are used to predict or explain the dependent variable.

The goal of linear regression is to find the equation of a straight line that minimizes the sum of the squared differences between the actual values of the dependent variable and the predicted values by the linear model. This process is known as "fitting the line" or "training the model."

The equation of a simple linear regression model with one independent variable can be represented as:

Y = β0 + β1\*X

where:

- Y represents the dependent variable

- X represents the independent variable

- β0 is the y-intercept (the value of Y when X is 0)

- β1 is the slope (the change in Y per unit change in X)

The coefficients (β0 and β1) are estimated using various optimization algorithms, such as ordinary least squares (OLS), which minimize the sum of squared residuals (the differences between the actual and predicted values).

Linear regression is widely used for various purposes, such as:

- Predictive modeling: Predicting future values of the dependent variable based on the values of the independent variables.

- Relationship analysis: Assessing the strength and direction of the relationship between variables.

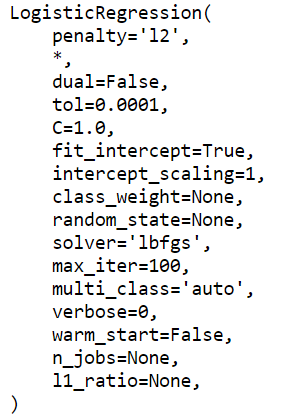
- Variable selection: Identifying the most influential predictors among a set of independent variables.

- Estimating impact: Understanding how changes in the independent variables affect the dependent variable.

Extensions to linear regression include multiple linear regression (with more than one independent variable), polynomial regression (allowing for curved relationships), and generalized linear regression (for non-normal distributions or categorical dependent variables).

Linear regression has assumptions that need to be checked, such as linearity, independence of errors, homoscedasticity (constant variance), and normality of errors. Violations of these assumptions may affect the reliability and interpretation of the results.

**The object description of the Decision Tree Classifier used is given below:**



INTIALIZING MODELS

Initializing models is a crucial step in the machine learning workflow. Here are a few reasons why we need to initialize models:

1.Model Configuration: Initialization allows us to define and configure the model's architecture, hyperparameters, and settings. Different models have various parameters that need to be set before training. For example, in a neural network, we need to specify the number of layers, the number of neurons in each layer, activation functions, and learning rate. Initialization provides an opportunity to set these parameters appropriately for the task at hand.

2. Memory Allocation: Initializing a model allocates memory for the model's parameters and internal variables. By allocating memory upfront, the model can efficiently store and process the required information during training or prediction. Without proper initialization, memory allocation may lead to errors or inefficient memory usage.

3. Random Initialization: In some models, such as neural networks, initializing the parameters randomly can help break symmetry and improve the convergence of the learning algorithm. Random initialization allows the model to start with different initial values, reducing the chances of getting stuck in suboptimal solutions.

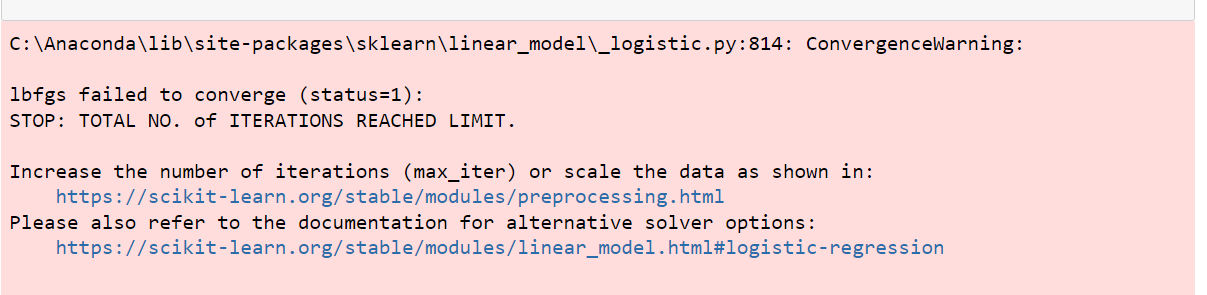
4. Reusability: Initializing a model makes it ready for training and prediction. By initializing the model at the beginning of each training session or when needed, we can reuse the model object and ensure it is in a clean state, ready to learn from new data.

5. Consistency and Reproducibility: Initializing a model ensures consistent behavior across different runs. By specifying the initial configuration and settings, we can reproduce the same results if we use the same initialization parameters. This is crucial for experiment reproducibility and debugging.

6. Customization and Fine-tuning: Initialization allows for model customization and fine-tuning. We can set specific initialization techniques or strategies based on the nature of the problem, the data, or domain-specific knowledge. Customization options can include setting initial weights, biases, or other model-specific parameters.

In summary, initializing models is essential to configure the model, allocate memory, set random initial values, enable reusability, ensure consistency, and customize the model for specific tasks. Proper initialization is a critical step to ensure the model's effective learning and performance.

WE MIGHT GET ERROR IN THE LOGISTIC REGRESSION AS FOLLOWS:



The error can be avoided by increasing the maximum iterations. Logistic Regression has maximum iterations as 100 as shown in the object parameters. This 100 can be increased,say 200. We use the function max\_iter(200) function to avoid this error.

The max\_iter parameter is used to control the maximum number of iterations or epochs that an iterative algorithm will run when training a model. It specifies the maximum number of times the learning algorithm will iterate over the training data or update the model's parameters.

Logistic Regression: In logistic regression, max\_iter defines the maximum number of iterations for the optimization algorithm to converge.

TRAINING MODELS

We train models in machine learning to enable them to learn patterns and relationships within data and make accurate predictions or decisions on new, unseen data. Training a model involves presenting it with a labeled dataset (known as the training data) and allowing the model to adjust its internal parameters or weights based on the patterns in the data.

**Here are a few key reasons why we train models:**

1. Pattern Learning: Training allows a model to learn and capture patterns, relationships, and dependencies within the training data. By observing the input features and their corresponding labels or outcomes, the model adjusts its internal parameters to identify patterns that can be used to make predictions or classifications on new, unseen data.

2. Generalization: Training a model aims to achieve good generalization performance. Generalization refers to the model's ability to accurately predict or classify unseen data that it has not encountered during training. By learning from a diverse and representative training dataset, the model aims to generalize well to new examples and make reliable predictions.

3. Parameter Estimation: Many machine learning models have adjustable parameters that control the model's behavior and predictions. During training, the model adjusts these parameters to minimize a loss function or maximize a performance metric. The optimization process finds the optimal values of the parameters that best fit the training data and optimize the model's predictive capabilities.

4. Decision Boundaries: In classification tasks, training helps establish decision boundaries that separate different classes or categories. By learning from labeled examples, the model identifies regions in the feature space where different classes are likely to be found. This allows the model to make predictions or assign labels to unseen examples based on their feature values.

5. Model Evaluation and Improvement: Training a model provides an opportunity to evaluate its performance and identify areas for improvement. By analyzing metrics such as accuracy, precision, recall, or mean squared error on the training data, we can assess how well the model is learning and adjust its architecture, hyperparameters, or training process if necessary.

Ultimately, training a model is a crucial step in the machine learning workflow as it enables the model to acquire knowledge from the training data, make accurate predictions on new data, and continually improve its performance.

PREDICTIONS FOR MODELS

Make Predictions: Pass the preprocessed input data to the loaded model and use its prediction function or method to generate predictions. The specific function or method for making predictions depends on the model and the programming framework or library you are using. For example, in scikit-learn, you typically use the predict( ) method of the trained model to obtain predictions.

Interpret and Use Predictions: Once the predictions are generated, you can interpret and use them according to the context of your problem. For classification tasks, predictions may correspond to class labels or probabilities. In regression tasks, predictions typically represent numerical values. You can further analyze, visualize, or take actions based on the predictions.

Here's an example code snippet to illustrate the prediction process using scikit-learn:

import pandas as pd

from sklearn.linear\_model import LogisticRegression

# Assuming 'X\_train' and 'y\_train' are your training data

# Assuming 'X\_test' is the new data for prediction

# Preprocess the new data if necessary

# Load the trained model

model = LogisticRegression()

# Fit the model to the training data (optional if already trained)

# Make predictions on the new data

predictions = model.predict(X\_test)

# Interpret and use the predictions

print(predictions)

To make predictions with trained data models, you typically follow these steps:

Preprocess the Data: Ensure that the new data you want to make predictions on is preprocessed in the same way as the training data. This may involve scaling numerical features, encoding categorical variables, handling missing values, or any other necessary data transformations.

Load the Trained Model: Load the trained model that you want to use for making predictions. This could be a machine learning model such as a decision tree, logistic regression, neural network, or any other model you have trained previously.

CALCULATING ACCURACY SCORE FOR DIFFERENT DATA MODELS

The accuracy score is an essential evaluation metric for data models, particularly for classification tasks. It provides a measure of how well the model's predictions match the true labels or outcomes in the dataset. Here are the key reasons why accuracy score is important for data models:

1. Performance Assessment: Accuracy score allows you to assess the overall performance of the model in terms of correctly predicting the classes or labels. It gives you a single metric to understand how well the model is performing on the given task.

2. Model Comparison: Accuracy score enables you to compare different models or variations of a model to determine which one performs better. By comparing accuracy scores, you can choose the model that yields higher accuracy and thus higher predictive performance.

3. Decision-Making: Accuracy score helps in making informed decisions based on the model's performance. It can be used as a criterion to decide whether the model is accurate enough for a particular application or if further improvements are needed.

4. Benchmarking: Accuracy score serves as a benchmark to compare the model's performance against human-level performance or other existing systems. It helps determine how well the model is performing relative to established baselines or standards.

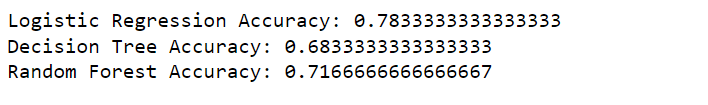
5. Business Impact: Accuracy score has direct implications on the business or real-world impact of the model's predictions. In applications such as fraud detection, medical diagnosis, or customer churn prediction, accurate predictions can significantly impact decision-making, resource allocation, and business outcomes.

6. Model Improvement: Monitoring accuracy scores over time can help identify potential issues or areas for model improvement. If accuracy decreases or fails to meet expectations, it can signal the need for further investigation, feature engineering, hyperparameter tuning, or exploring more advanced modeling techniques.

However, it is important to note that accuracy score alone may not be sufficient in all scenarios. Depending on the problem and data characteristics, other evaluation metrics such as precision, recall, F1 score, or area under the ROC curve (AUC-ROC) may provide a more comprehensive understanding of the model's performance, particularly in cases of imbalanced datasets or when different types of errors have different consequences.

Therefore, while accuracy score is a valuable metric, it is advisable to consider other evaluation metrics as well to gain a more complete assessment of the model's performance and suitability for the task at hand.

**The accuracy score calculated for the project is:**



COMPARING THE THREE DATA MODELS USED AND PLOT IT

Comparing different model accuracies and visualizing them in a bar graph can provide valuable insights and help in decision-making. Here are the reasons why it is necessary:

1. Model Selection: Comparing accuracies allows you to identify the model that performs the best on your task. By comparing the accuracy scores of different models, you can choose the one that achieves the highest accuracy, indicating better predictive performance. The bar graph provides a clear visual representation of the comparative accuracies, making it easier to determine the best-performing model.

2. Performance Evaluation: Comparing accuracies helps you evaluate and understand the relative performance of different models. It provides a quantitative measure to assess how well each model is capturing the underlying patterns and making accurate predictions. The bar graph allows for a quick and intuitive comparison of the models' performance, making it easier to interpret and draw conclusions.

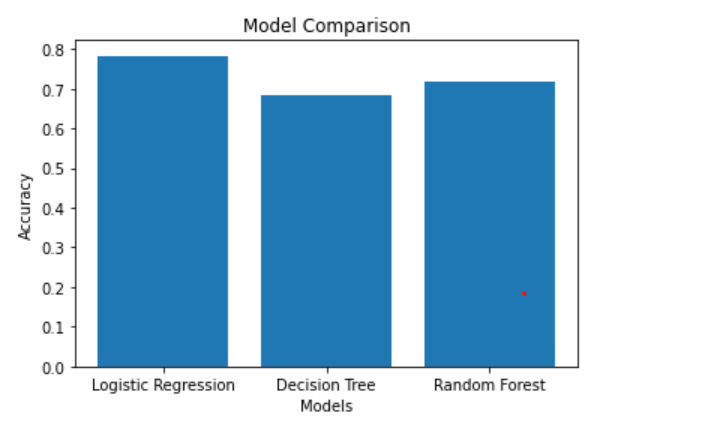
3. Model Improvement: Comparing accuracies can highlight areas for model improvement. If you observe significant differences in accuracy among the models, it indicates that certain modeling approaches or techniques may be more effective for the given task. This insight can guide further exploration, experimentation, or refinement of the models to achieve higher accuracy.

4. Communication and Visualization: Visualizing model accuracies in a bar graph simplifies the communication of results to stakeholders or non-technical audiences. The visual representation makes it easier to grasp and communicate the performance differences among the models. The bar graph provides a clear and concise summary, enabling effective storytelling and supporting decision-making processes.

5. Experimentation and Research: Comparing model accuracies can serve as a basis for experimentation and research. It helps researchers and practitioners understand the impact of different algorithms, feature engineering techniques, or hyperparameter settings on the model's accuracy. By plotting and comparing accuracies over multiple experiments, it becomes possible to identify patterns, trends, or relationships that can guide further investigations.

Overall, comparing model accuracies and visualizing them in a bar graph is crucial for informed decision-making, performance evaluation, model improvement, effective communication, and supporting further experimentation and research. It allows you to make data-driven choices and select the best-performing model for your specific problem.

The bar plot when three data model accuracies are compared is as follows:



TEST DATASET

**Testing the given Test Dataset**

We were given a test dataset for the heart failure analysis. We pre-process the given test dataset in similar way we pre-processed our train dataset. The methodology followed is given below:

1) Split the data into features and target variable

2) Split the data into training and testing sets

3)Initialize models

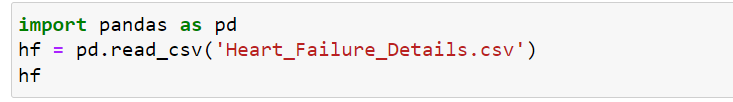
4) Train models

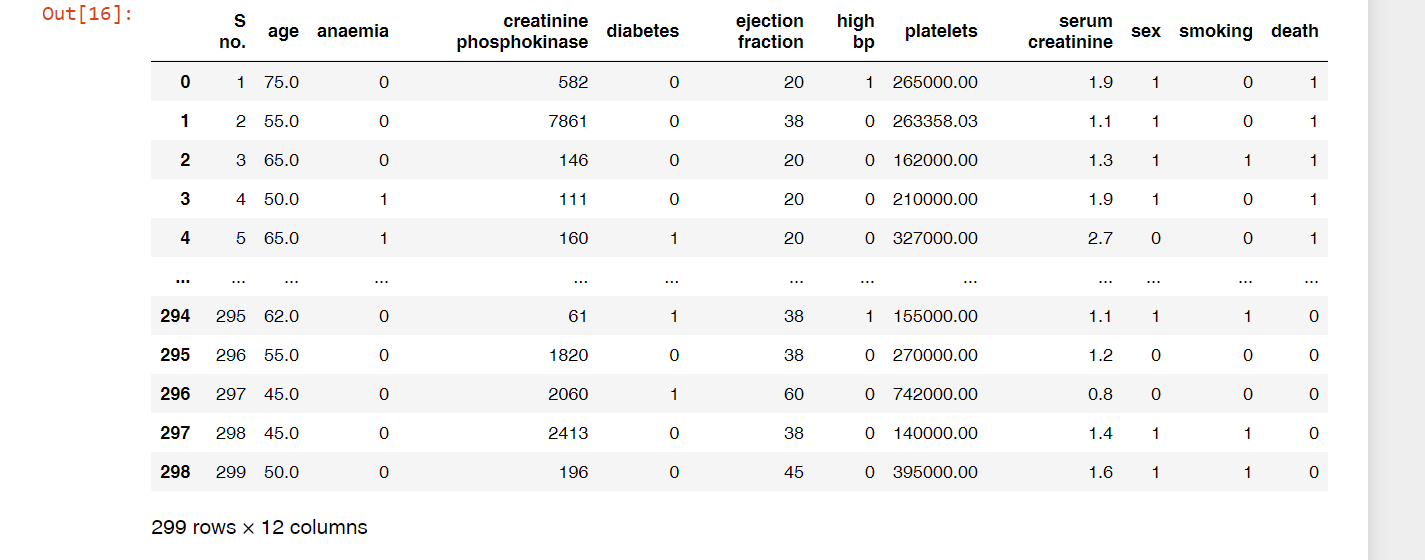
5)Make predictions

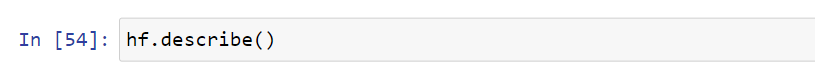
6)Calculate accuracy score

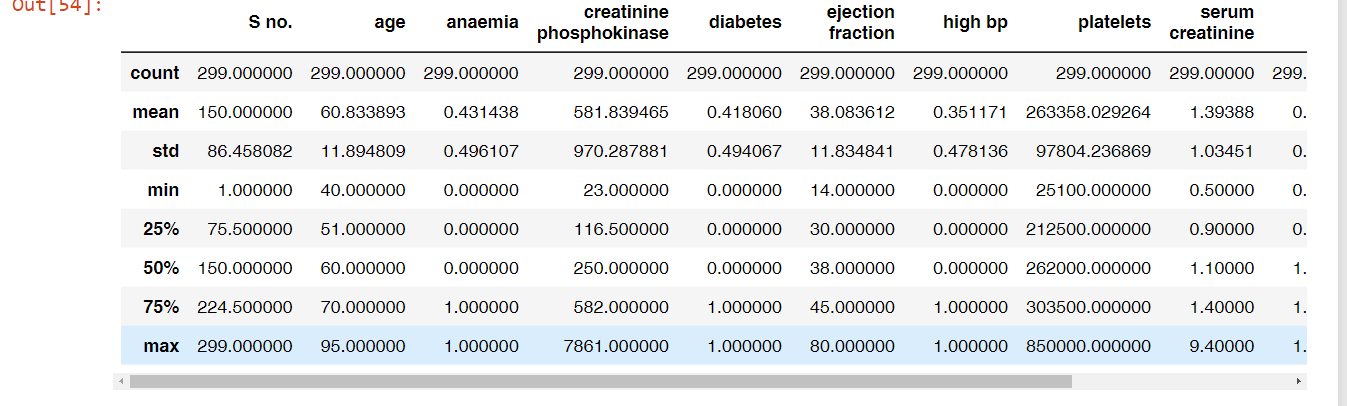
7)Create a bar graph to compare accuracies

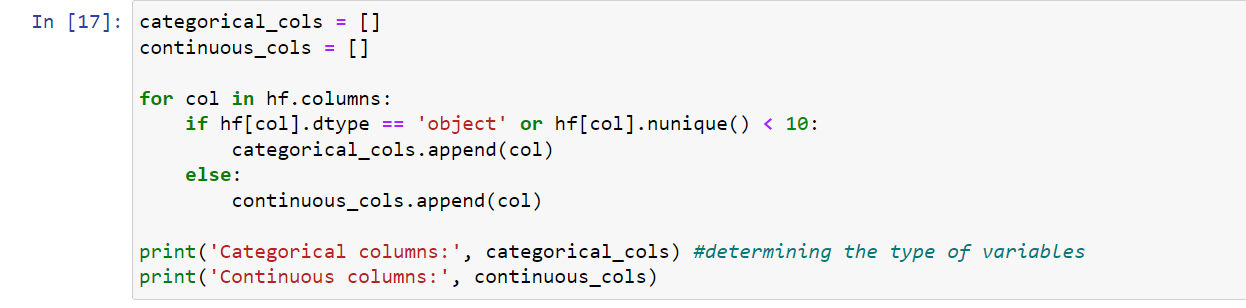
CODES



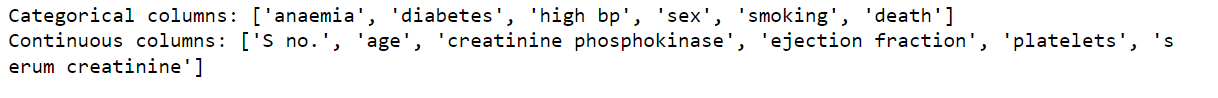


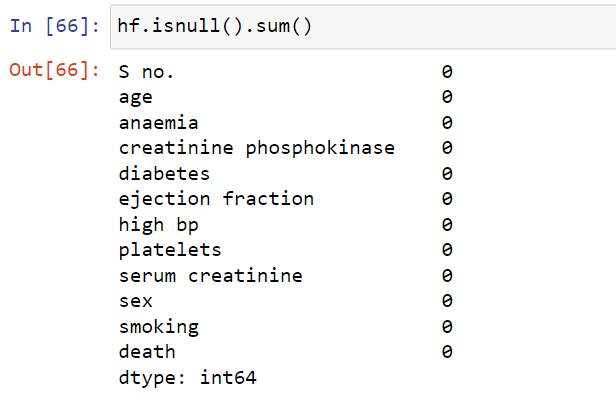


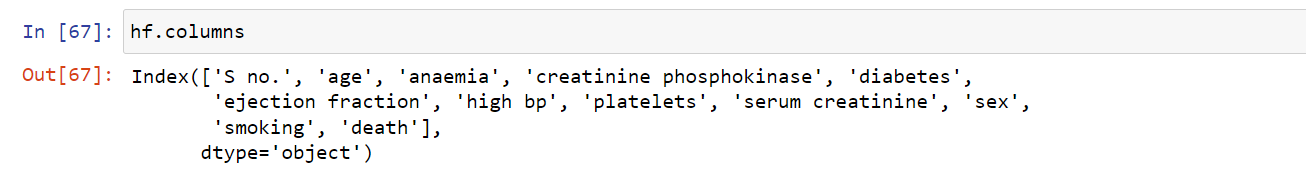
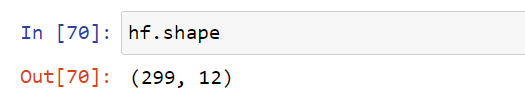




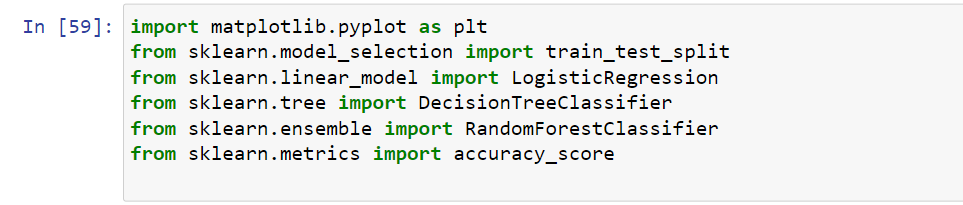
OUTPUT:

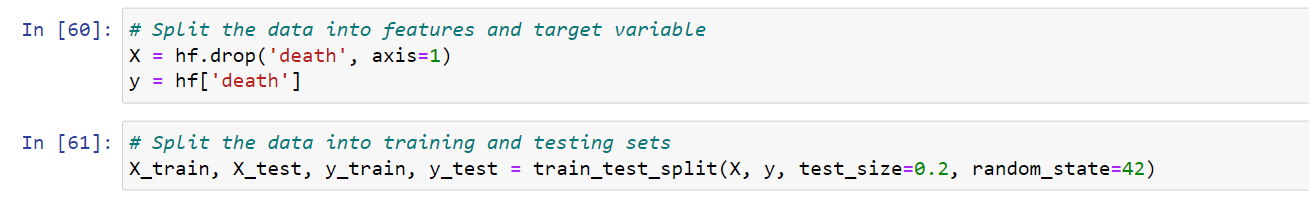


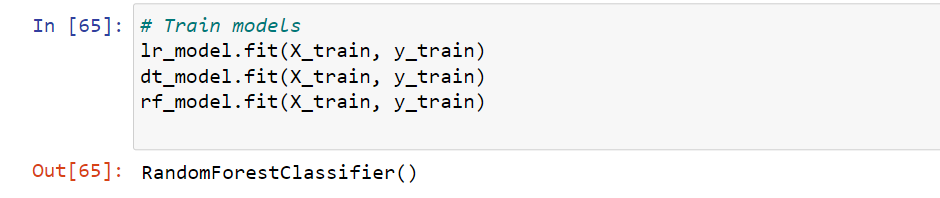


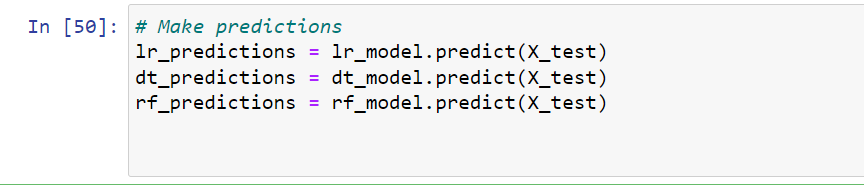


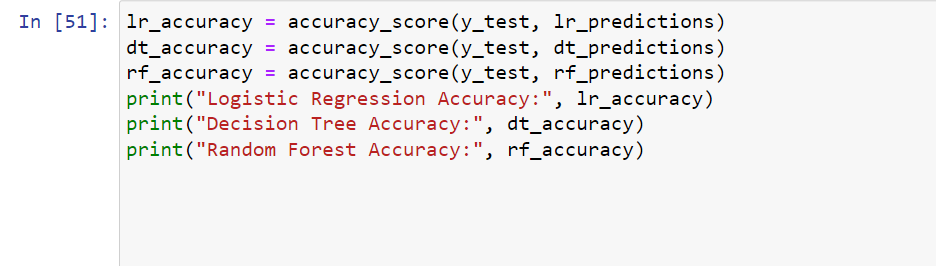


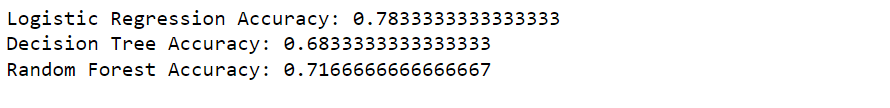
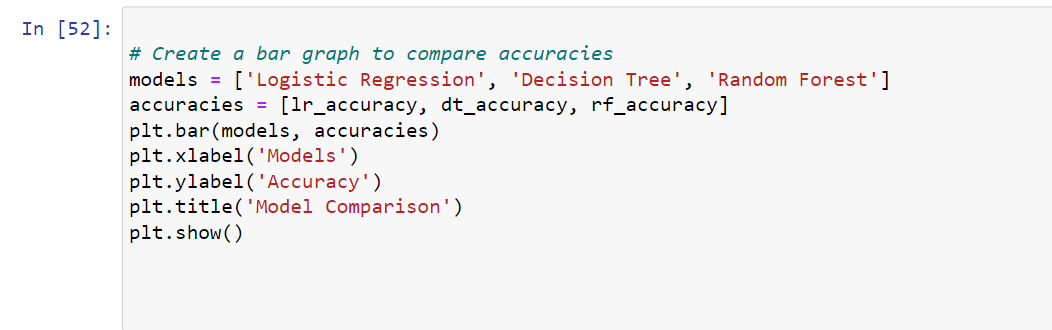


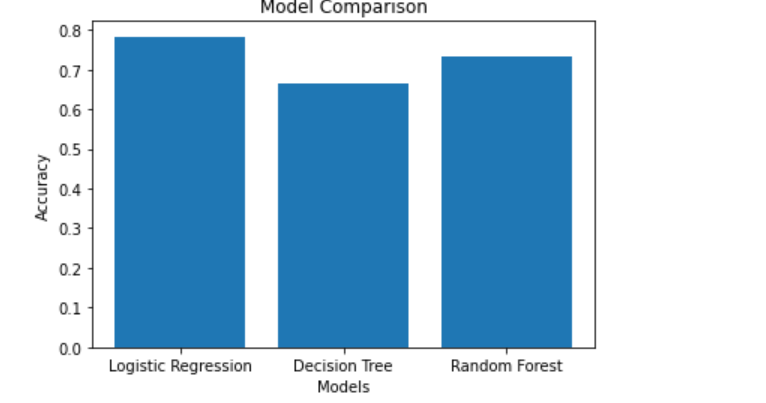
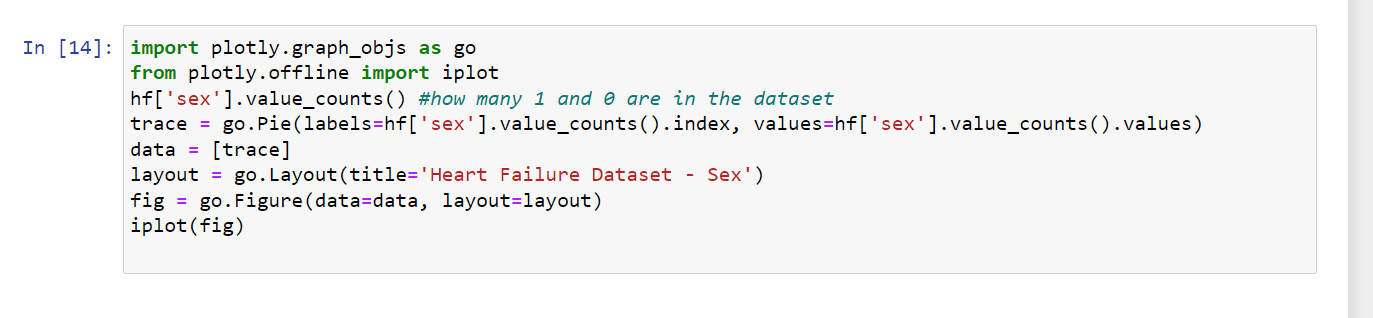




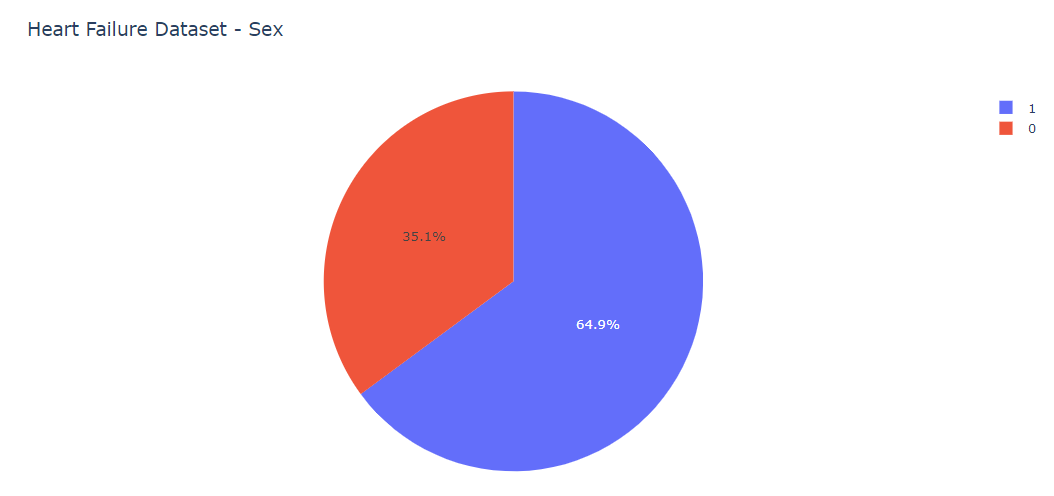
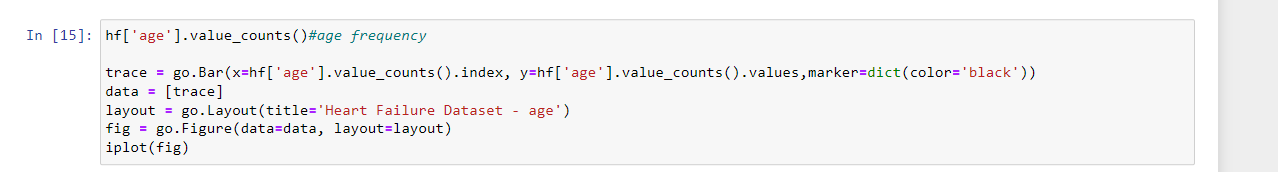


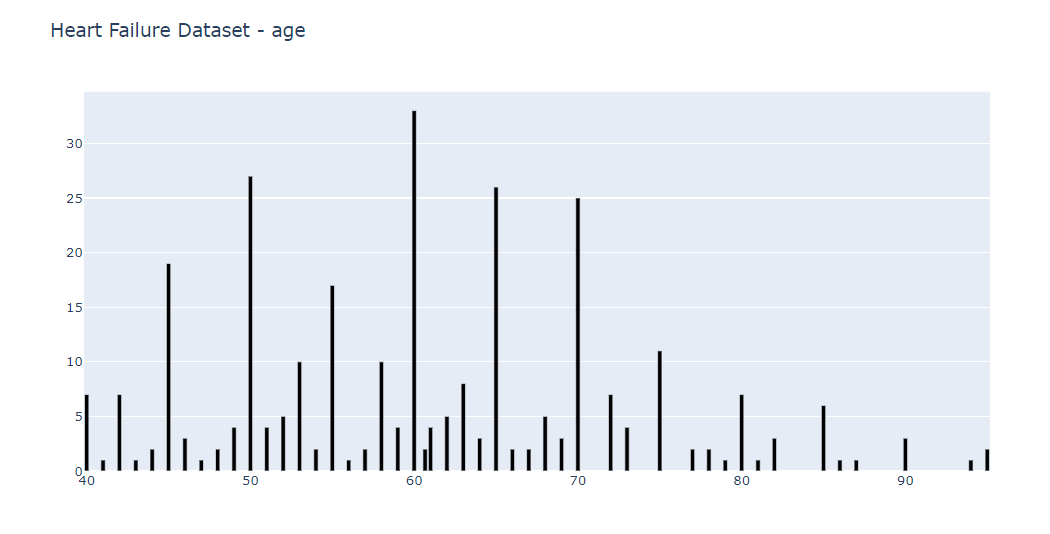
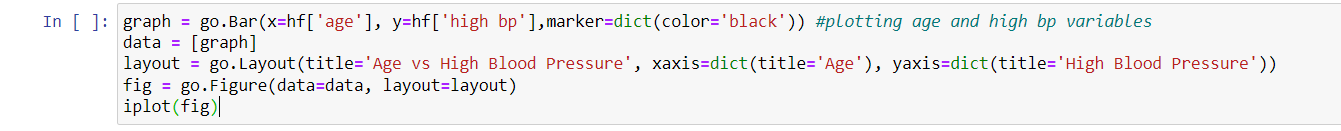


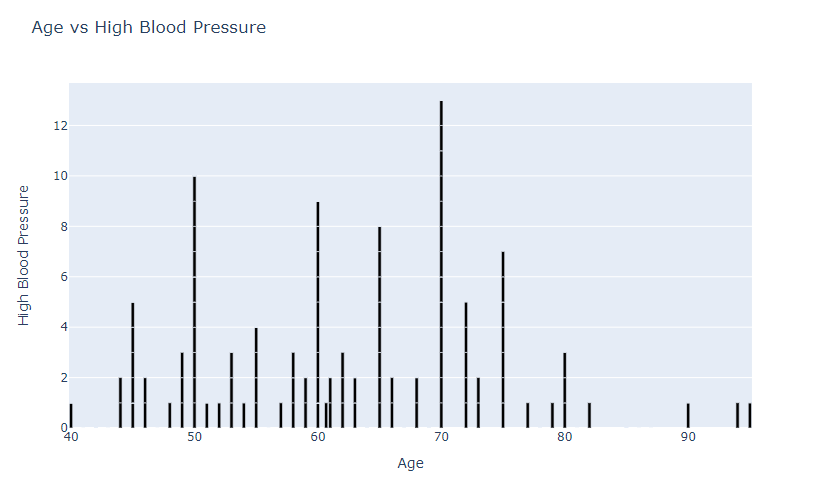


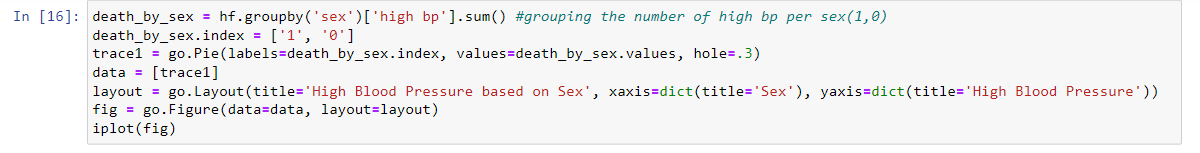


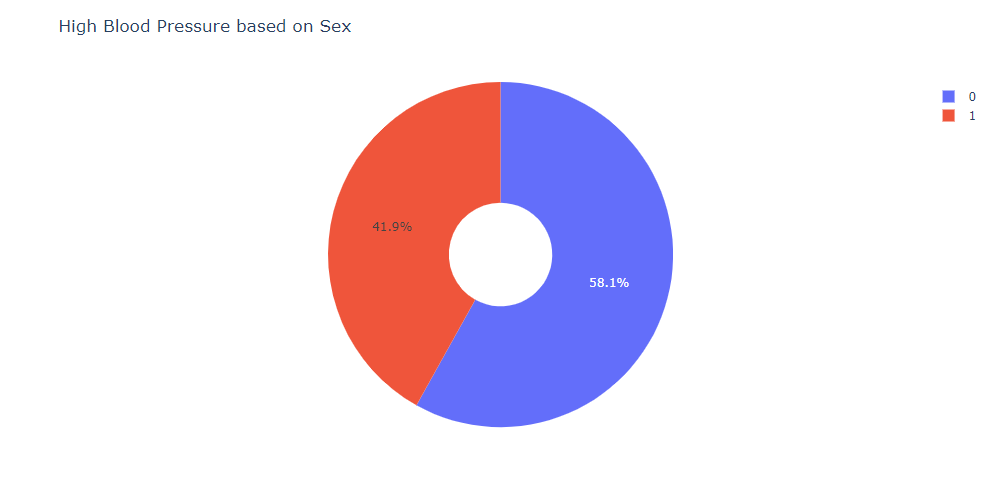
Output:

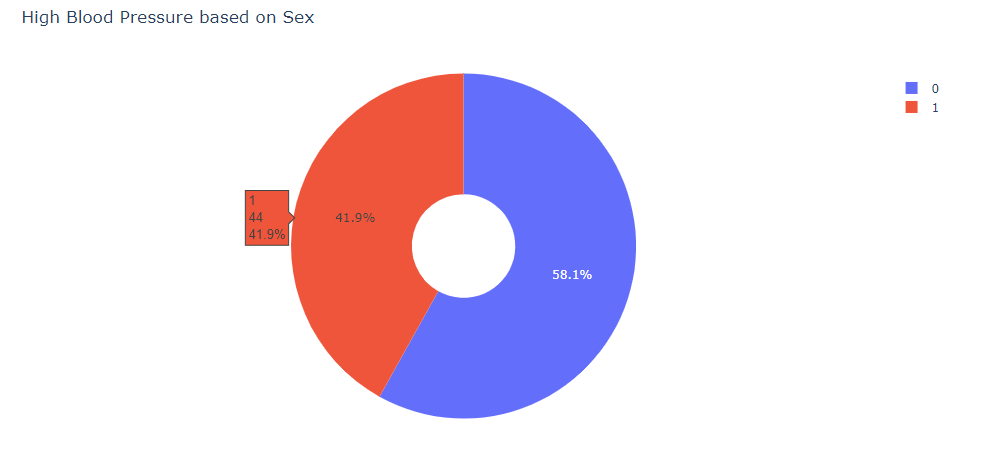


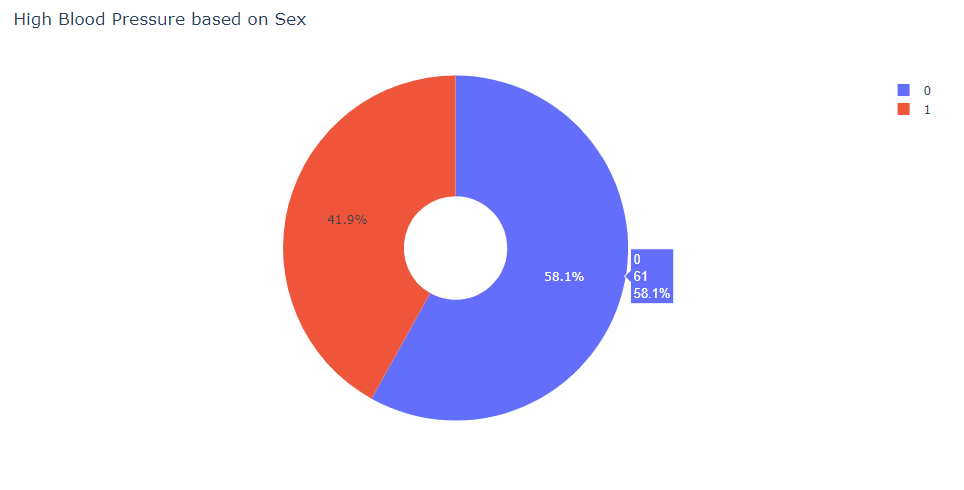




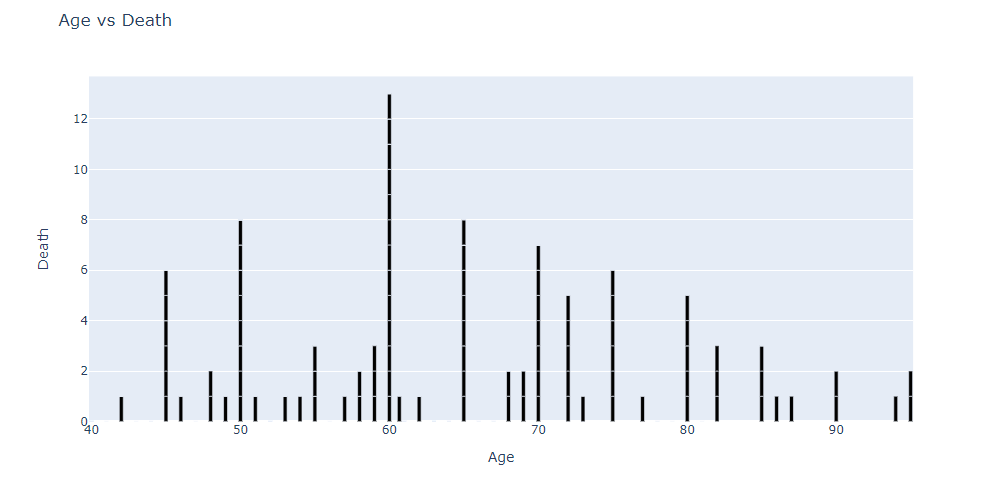
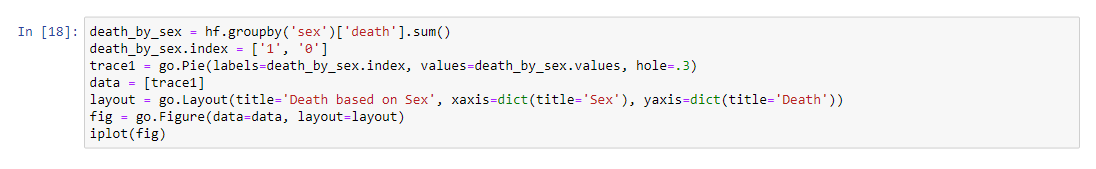


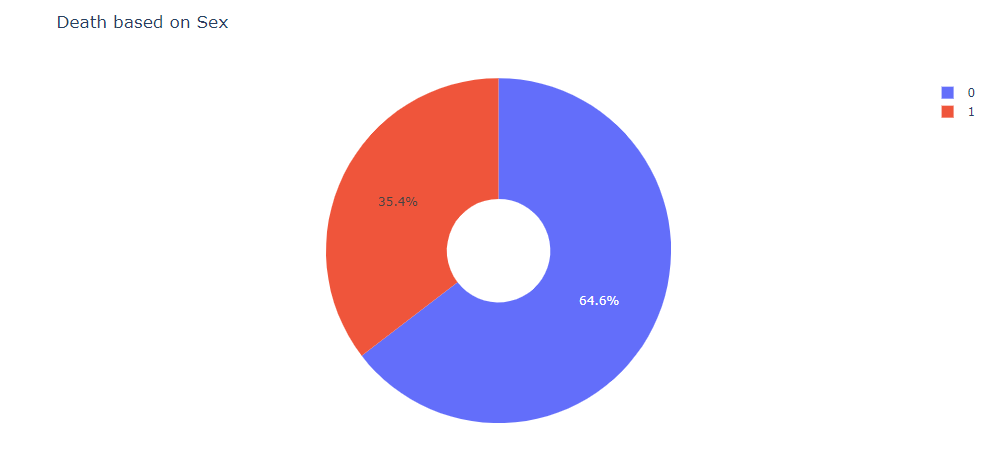


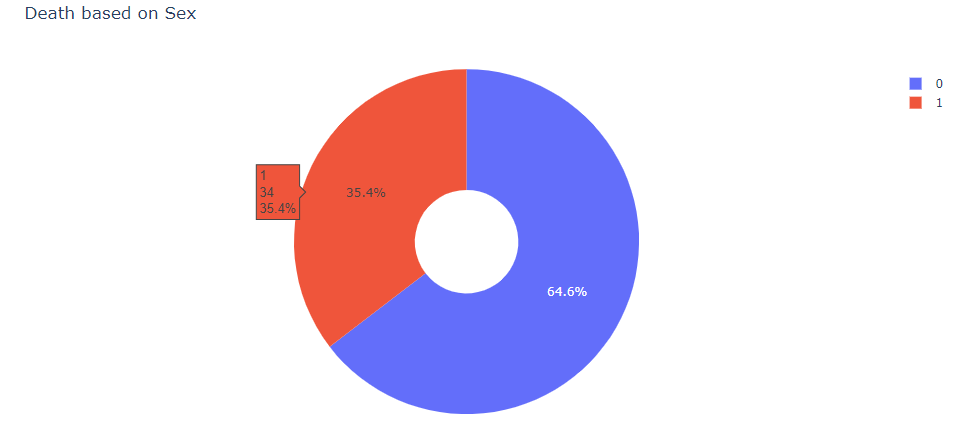


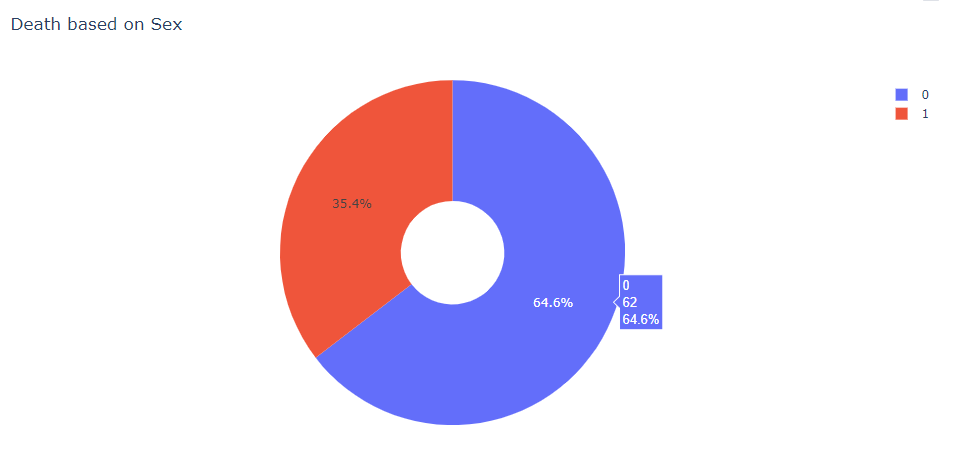
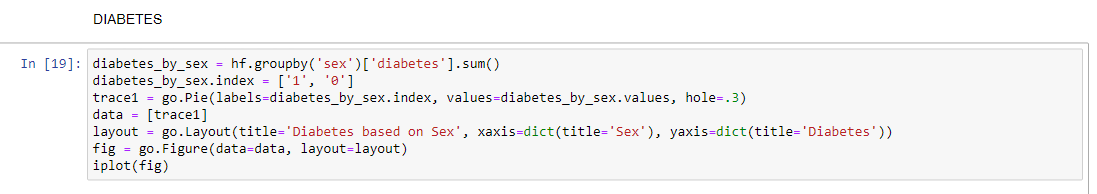


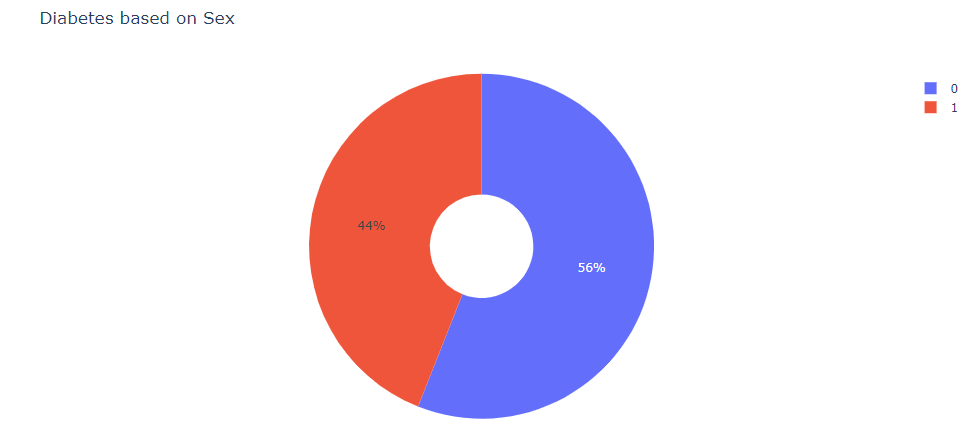


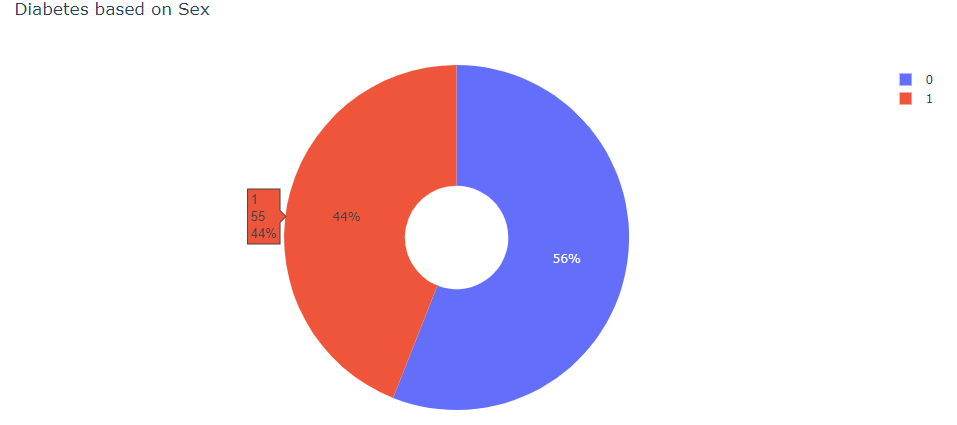


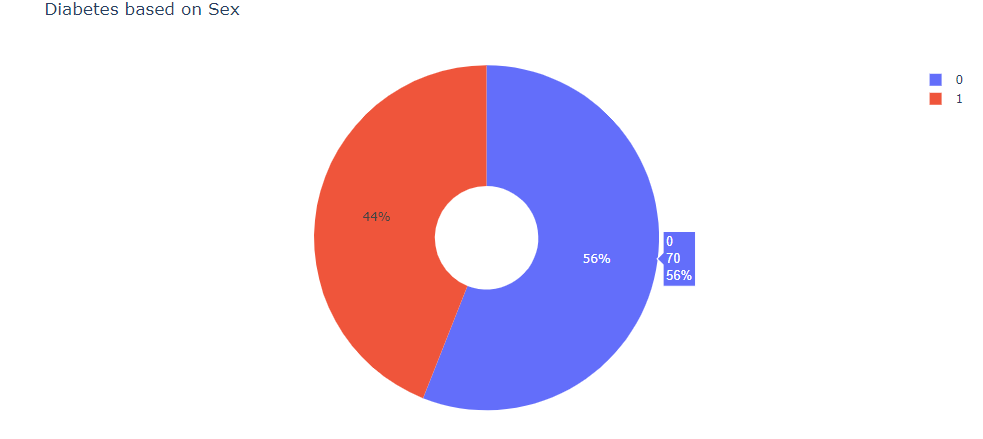


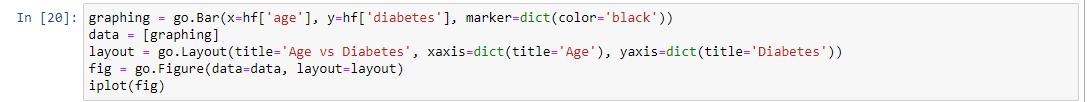


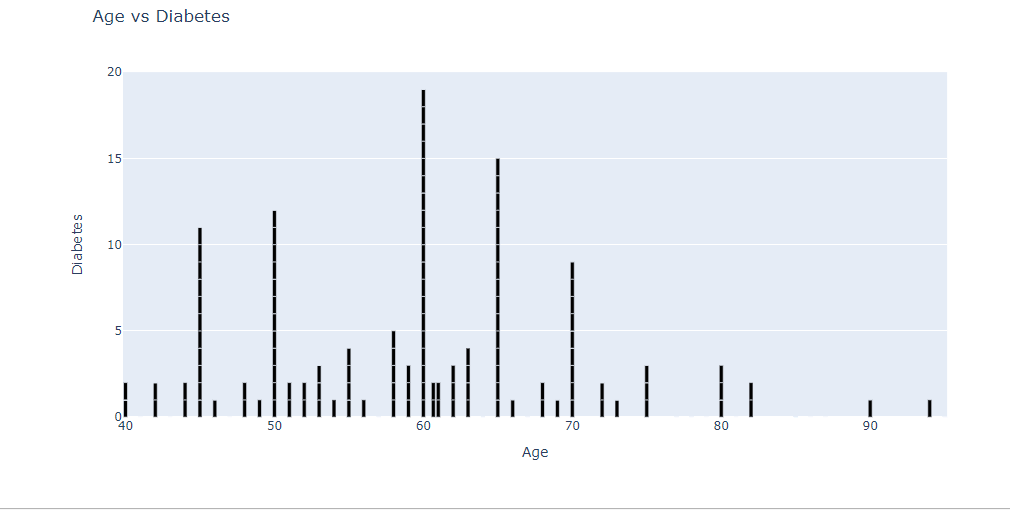
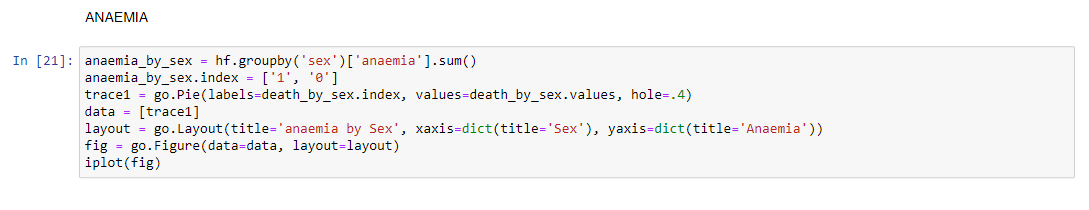


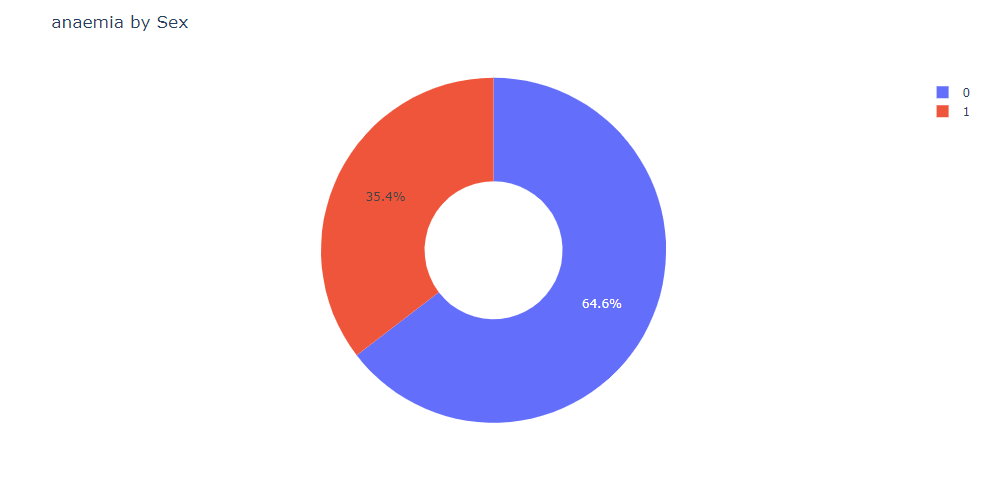


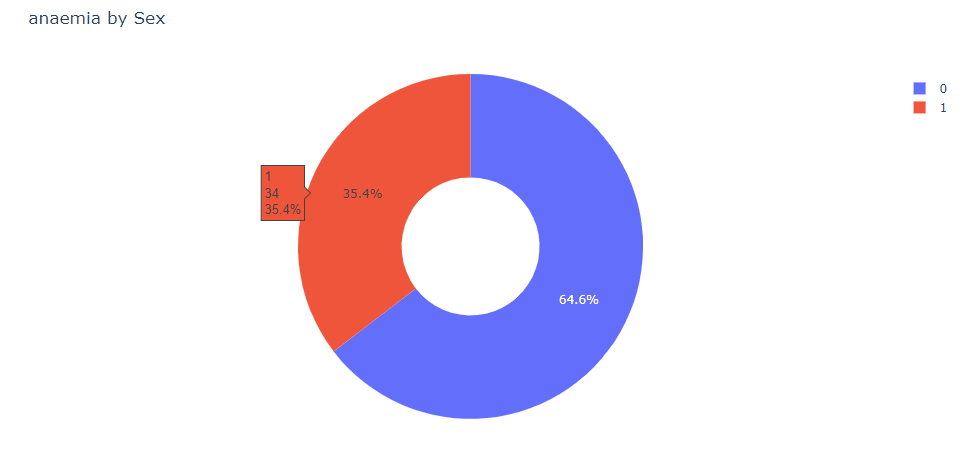


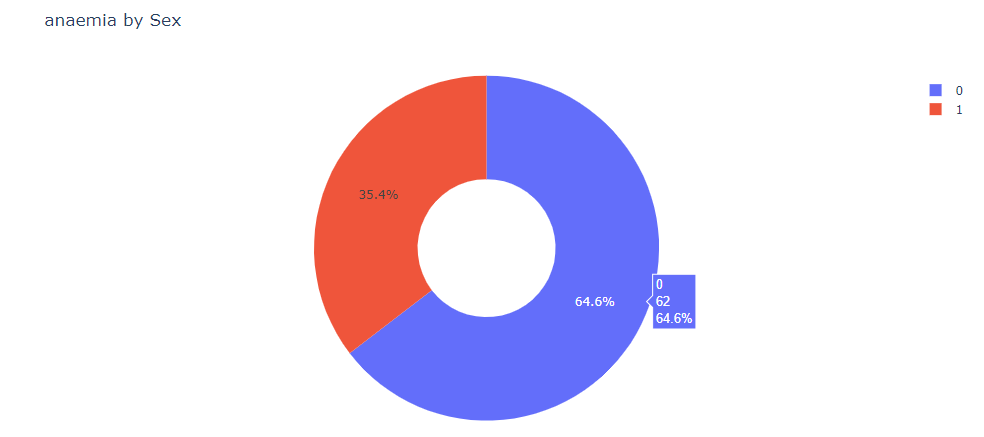


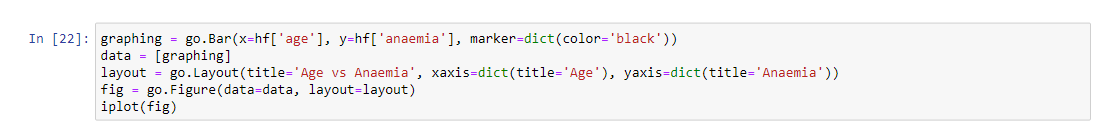


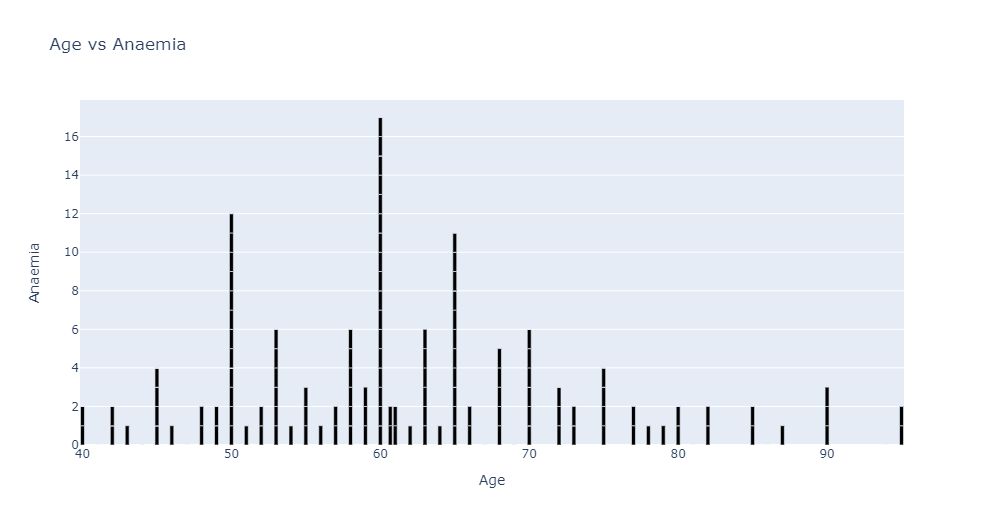


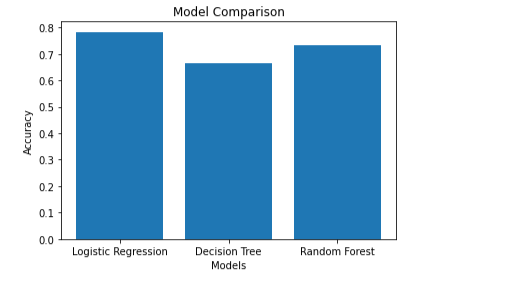










FINAL COMPARISION OF DATA MODELS

FUTURE SCOPE OF IMPROVEMENTS

Heart failure analysis is an important area of research and has several potential future scope of improvements. Here are a few areas that can be explored for enhancing heart failure analysis:

1. Advanced Machine Learning Techniques: Applying more advanced machine learning algorithms and techniques can improve the accuracy and predictive power of heart failure analysis. For example, deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), can be explored to extract more intricate patterns and temporal dependencies in heart failure data.

2. Feature Engineering and Selection: Developing innovative feature engineering and feature selection techniques can help identify the most relevant and informative features for heart failure analysis. This can involve incorporating domain knowledge, exploring new biomarkers or imaging features, or utilizing feature selection algorithms to automatically identify the most predictive features.

3. Personalized Risk Stratification: Tailoring heart failure analysis to individual patients can enable personalized risk stratification and treatment planning. Integrating patient-specific data, such as genetic profiles, electronic health records, lifestyle factors, and biomarkers, can improve risk prediction models and guide personalized interventions for prevention and management.

4. Multimodal Data Integration: Integrating diverse data sources and modalities can enhance heart failure analysis. Combining clinical data with genetic data, wearable sensor data, medical imaging, and patient-reported outcomes can provide a comprehensive view of the patient's health status. Novel data fusion techniques can be employed to extract meaningful information from multimodal data and improve predictive models.

5. Real-Time Monitoring and Alert Systems: Developing real-time monitoring systems that leverage wearable devices and Internet of Things (IoT) technologies can enable continuous tracking of heart failure patients. These systems can provide timely alerts, detect early signs of deterioration, and facilitate proactive interventions to prevent adverse events.

6. Explainability and Interpretability: Enhancing the interpretability and explainability of heart failure analysis models is important for gaining trust and understanding the reasoning behind predictions. Research on interpretable machine learning methods, visualization techniques, and model-agnostic explainability approaches can help clinicians and patients comprehend and trust the results of heart failure analysis models.

7. Data Sharing and Collaborative Research: Encouraging data sharing and collaborative research efforts can accelerate advancements in heart failure analysis. Large-scale, multi-center studies, consortia, and data repositories can facilitate the pooling of data resources and the development of robust and generalizable models.

These are just a few potential areas for future improvement in heart failure analysis. The interdisciplinary nature of heart failure research provides ample opportunities for collaboration among clinicians, data scientists, engineers, and researchers to work together towards improving early detection, risk stratification, treatment selection, and patient outcomes in heart failure management.

CERTIFICATE

This is to certify that MS. Sai Srija Achukolu of Lovely Professional University, registration number:12110309, has successfully completed a project on Heart Failure Analysis using Machine Learning with Python under the guidance of Prof. Arnab Chakraborty.

Prof. Arnab Chakraborty

Globsyn Finishing School