Final Project

Executive Summary:

In this study, we aimed to assess the predictive performance of various classifiers for identifying disease outcomes based on symptom profiles and patient demographics.

Key questions addressed included the selection of appropriate classifiers and the evaluation of their accuracy and discriminative ability.

Results revealed varying performance among the classifiers. Logistic Regression exhibited moderate accuracy (52.6%) and discriminative ability (ROC AUC: 0.62), while Decision Tree achieved slightly higher accuracy (59.2%) but lower discriminative ability (ROC AUC: 0.59).

Notably, the Random Forest Classifier outperformed others with the highest accuracy (57.9%) and superior discriminative ability (ROC AUC: 0.70). However, the AdaBoost Classifier yielded suboptimal results.

Insights gained suggest the potential of ensemble methods like Random Forest for disease prediction tasks.

Recommendations for future research include exploring advanced ensemble techniques, optimizing model hyperparameters, and enhancing data quality and interpretability.

By addressing these recommendations, future research can advance disease prediction using data mining techniques, thereby improving healthcare decision-making and patient outcomes.

“What problems did you specifically address?”

In the data mining project, the specific problems addressed include:

Disease Prediction: The primary objective is to predict the likelihood of a positive or negative outcome for various diseases based on the given parameters, excluding the disease itself. This involves developing predictive models that utilize symptoms, demographic information, and health indicators to make accurate predictions.

Model Generalization: Ensuring that the predictive models generalize well to unseen data. This involves employing techniques such as cross-validation and model evaluation to assess the performance of the models and mitigate overfitting.

Data Preprocessing: Preparing the dataset for analysis by handling missing values, encoding categorical variables, and scaling numerical features. Proper data preprocessing is essential for building robust predictive models.

“What dataset did you use?”

The dataset used in the data mining project is called "Disease_symptom_and_patient_profile_dataset." It comprises columns representing various attributes related to disease symptoms, patient profiles, and outcome variables. The columns include Disease, Fever, Cough, Fatigue, Difficulty Breathing, Age, Gender, Blood Pressure, Cholesterol Level, and Outcome Variable.

	Disease	Fever	Cough	Fatigue	Difficulty Breathing	Age	Gender	Blood Pressure	Cholesterol Level	Outcome Variable
0	Influenza	Yes	No	Yes	Yes	19	Female	Low	Normal	Positive
1	Common Cold	No	Yes	Yes	No	25	Female	Normal	Normal	Negative
2	Eczema	No	Yes	Yes	No	25	Female	Normal	Normal	Negative
3	Asthma	Yes	Yes	No	Yes	25	Male	Normal	Normal	Positive
4	Asthma	Yes	Yes	No	Yes	25	Male	Normal	Normal	Positive

The attributes include:

Disease: The type of disease or condition the patient is diagnosed with.
Symptoms: Binary indicators (Yes/No) representing the presence or absence of symptoms such as fever, cough, fatigue, and difficulty breathing.
Demographic Information: Attributes such as age and gender provide insights into the patient's profile.
Health Indicators: Attributes like blood pressure and cholesterol level offer additional health-related information.
Outcome Variable: Indicates the outcome of the diagnostic test or assessment, typically binary (Positive/Negative) for disease presence.

“How did you preprocess the data?”

We pre-processed the data using the following steps:

1. Drop Irrelevant Column:

- We removed the 'Disease' column from the dataset as it was not required for the predictive modelling task.

```
data = data.drop(columns=['Disease'])
```

2. Label Encoding:
    - For categorical columns (columns with dtype 'object'), we applied label encoding to convert categorical values into numerical representations.
    - We iterated through each column in the dataset, checked if its dtype was 'object', and if so, applied label encoding using scikit-learn's LabelEncoder() class.
    - Label encoding assigns a unique integer to each categorical value within a column, thereby converting it into a numerical format suitable for machine learning algorithms.

```
label_encoders = {}
for column in data.columns:
    if data[column].dtype == 'object':
        label_encoders[column] = LabelEncoder()
        data[column] = label_encoders[column].fit_transform(data[column])
```

3. Convert Age to Integer:
    - We converted the 'Age' column to integer data type ('int32') to ensure consistency and compatibility with other numerical features in the dataset.

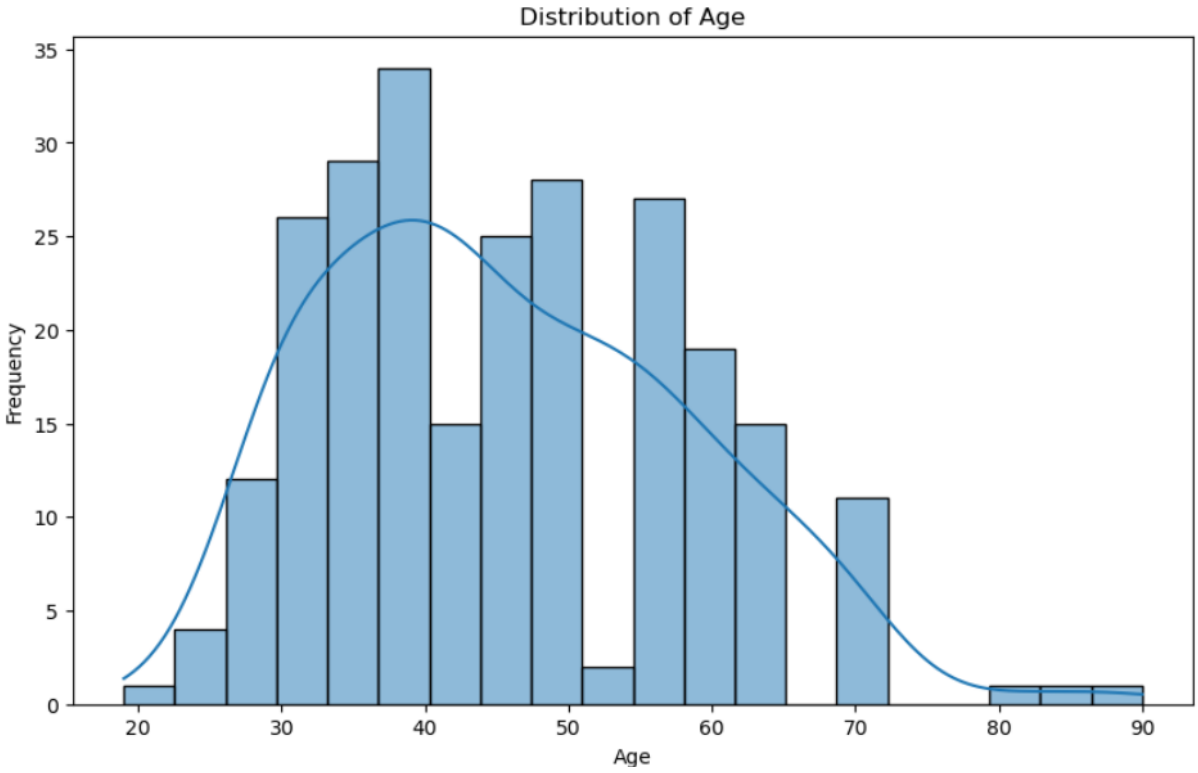
```
data['Age'] = data['Age'].astype('int32')
```

4. Remove Duplicates:
    - We removed duplicate records from the dataset, this step ensured that each patient record in the dataset was unique, preventing any redundancy or bias in the analysis. Removing duplicates helps improve the quality and reliability of the dataset for subsequent data mining tasks.

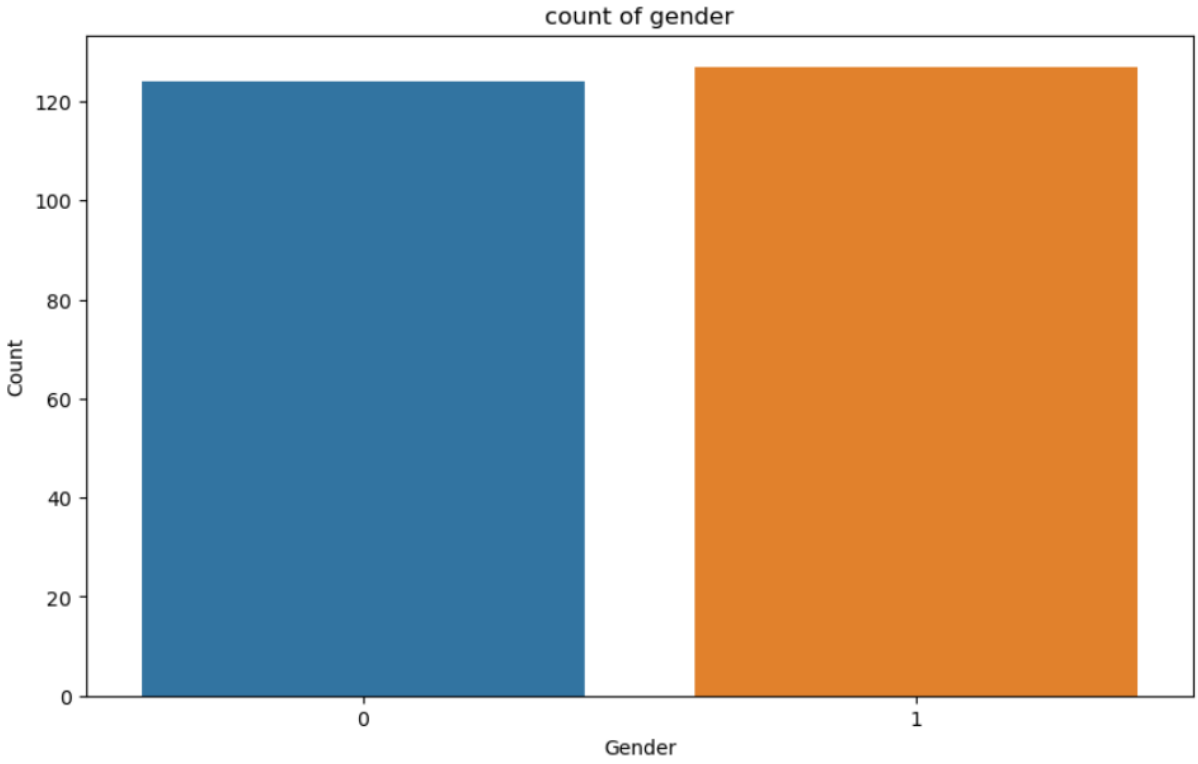
```
data = data.drop_duplicates()
```

We also performed basic EDA on the dataset using `.info()`, `.describe()`, and `.columns` methods

Additionally we have a age distribution graph,



And a gender encoded graph of male and female frequency



“How did you organize the data?”

We organized the data into features (X) and the target variable (y) as follows:

```
X = data.drop('Outcome Variable', axis=1)
y = data['Outcome Variable']
```

- **Features (X)**: We extracted all columns from the dataset except for the 'Outcome Variable' column, which contains the target variable to be predicted. These columns represent the input features used to train the predictive model.
- **Target Variable (y)**: We isolated the 'Outcome Variable' column, which serves as the target variable for prediction. This column contains the labels corresponding to the outcome we aim to predict.

After organizing the data, we split it into training and testing sets using the train_test_split function:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

- **Training and Testing Sets**: We divided the data into two sets: a training set (X_train, y_train) used to train the model and a testing set (X_test, y_test) used to evaluate its performance. The test_size parameter specifies the proportion of the dataset to include in the testing set, and random_state ensures reproducibility by seeding the random number generator.

```
X_train.head()
```

	Fever	Cough	Fatigue	Difficulty Breathing	Age	Gender	Blood Pressure	Cholesterol Level
253	0	1	1	1	55	0	2	1
37	0	0	1	0	30	0	0	0
169	0	0	0	0	45	0	1	2
197	1	1	1	1	45	1	0	0
203	0	0	1	0	48	1	2	0

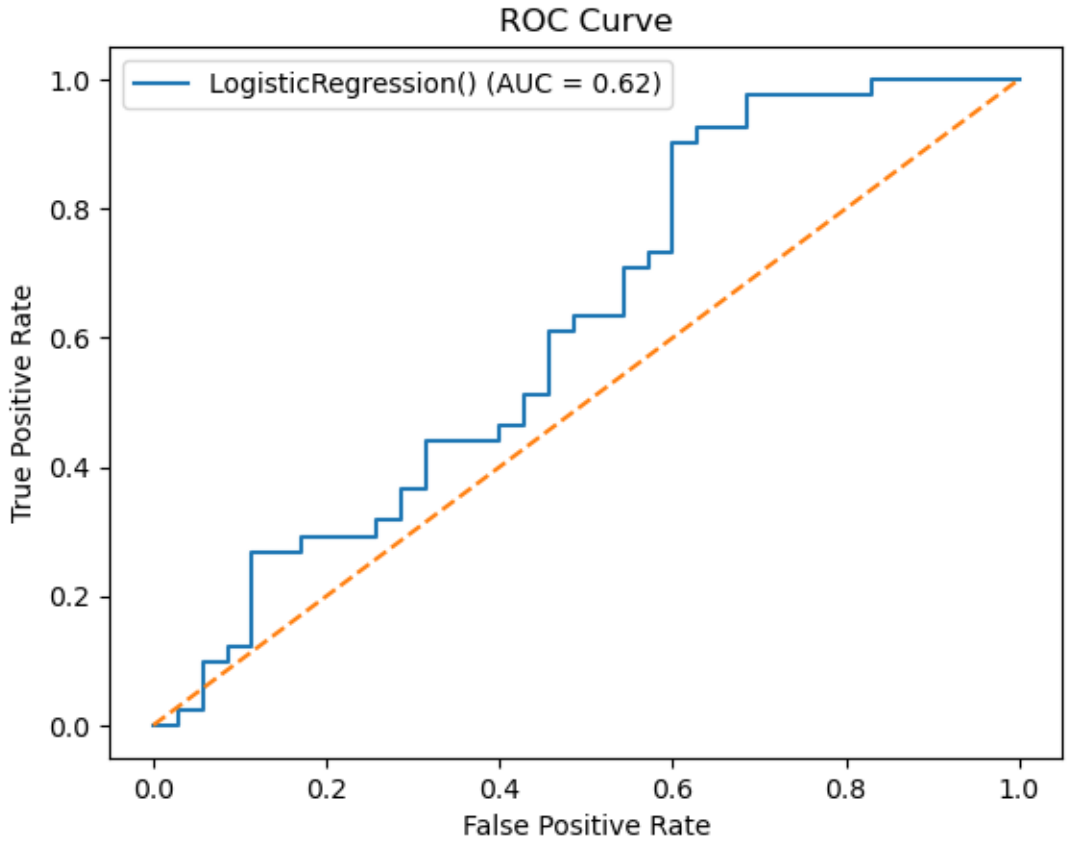
```
y_train.head()
```

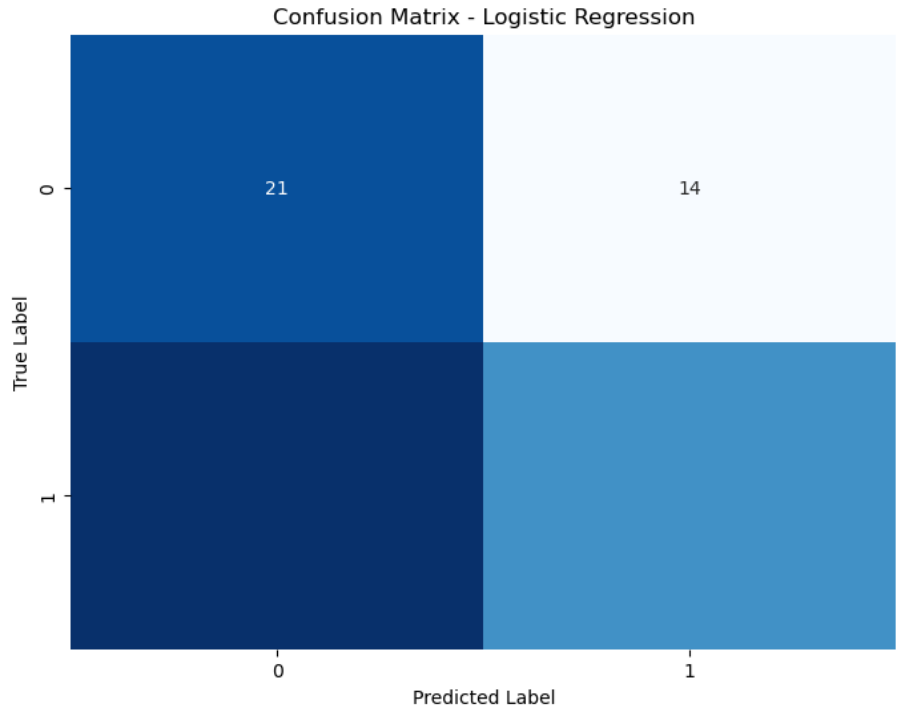
```
253    0
37     1
169    0
197    1
203    0
Name: Outcome Variable, dtype: int32
```

“What kind of results did you get across the 3 different classifiers?”

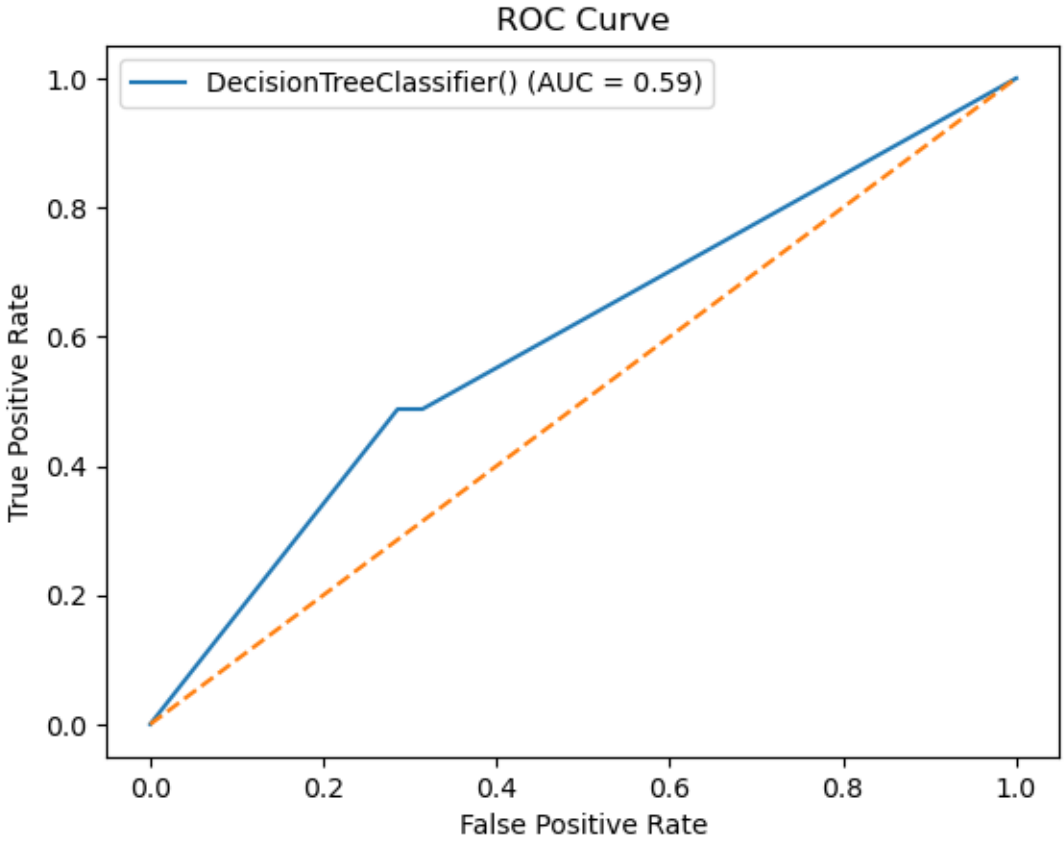
The results for the various classifiers we are using are as follows

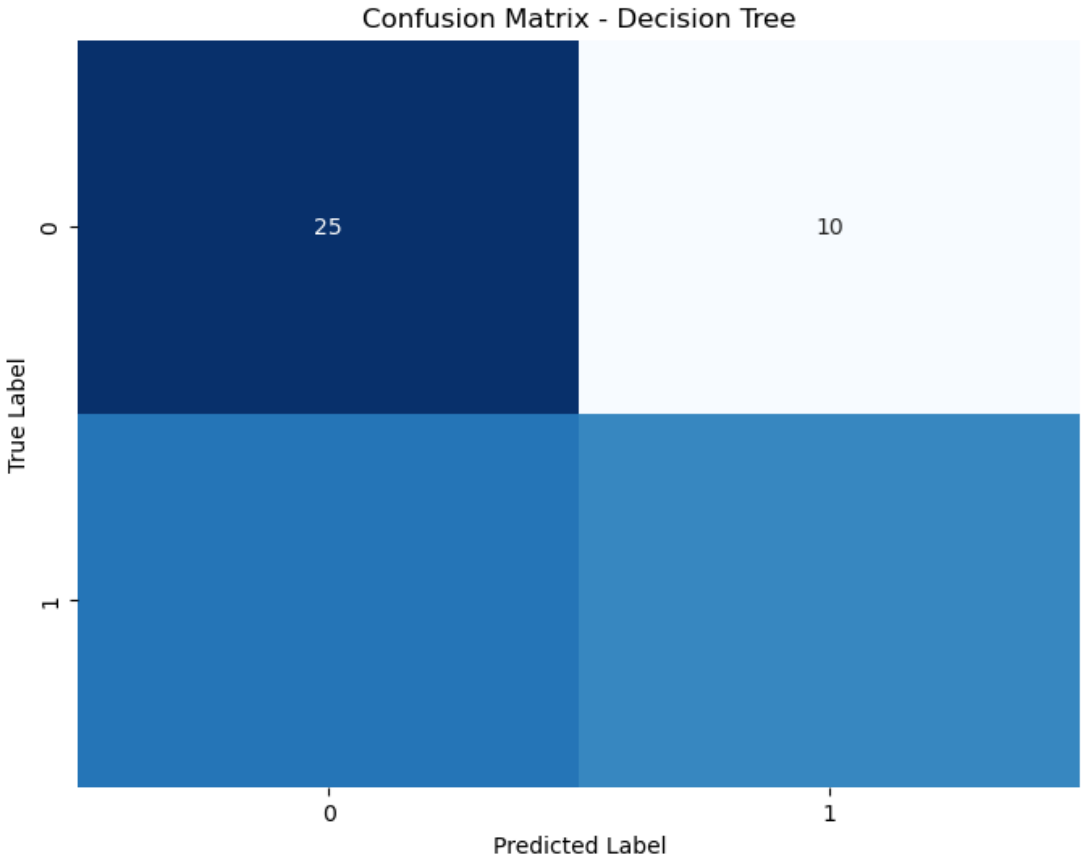
Classifier 1 - Logistic Regression:





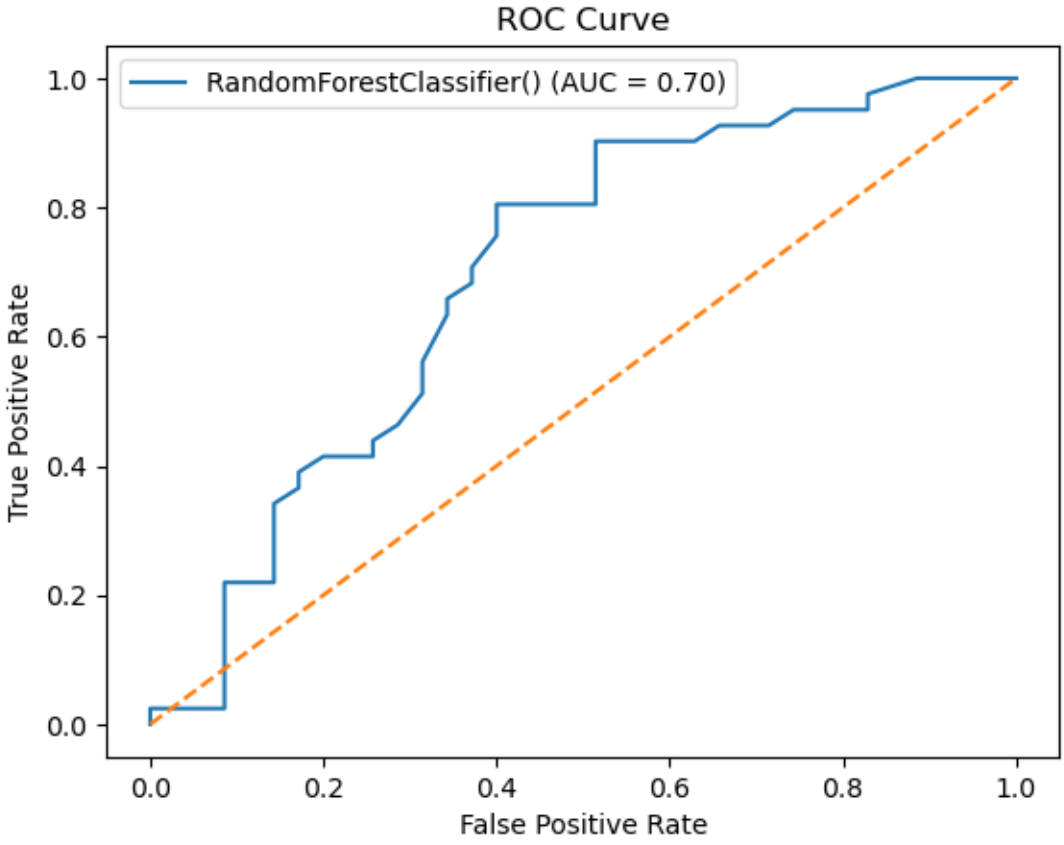
Classifier 2 - Decision Tree:

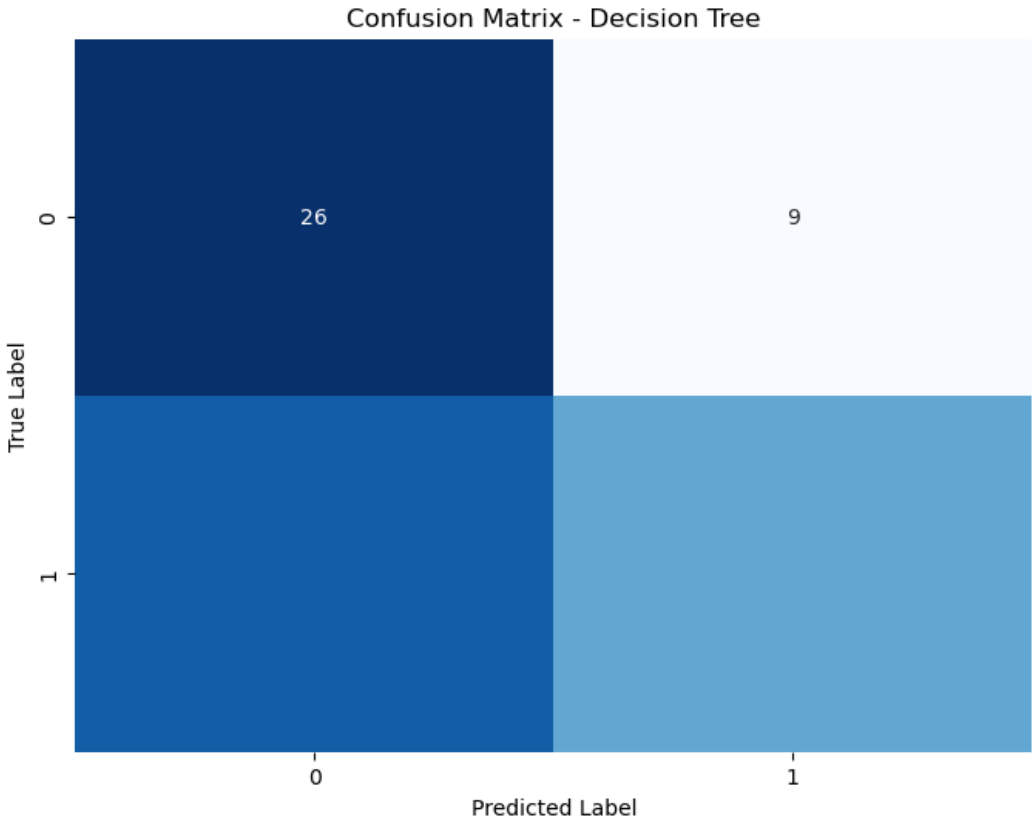




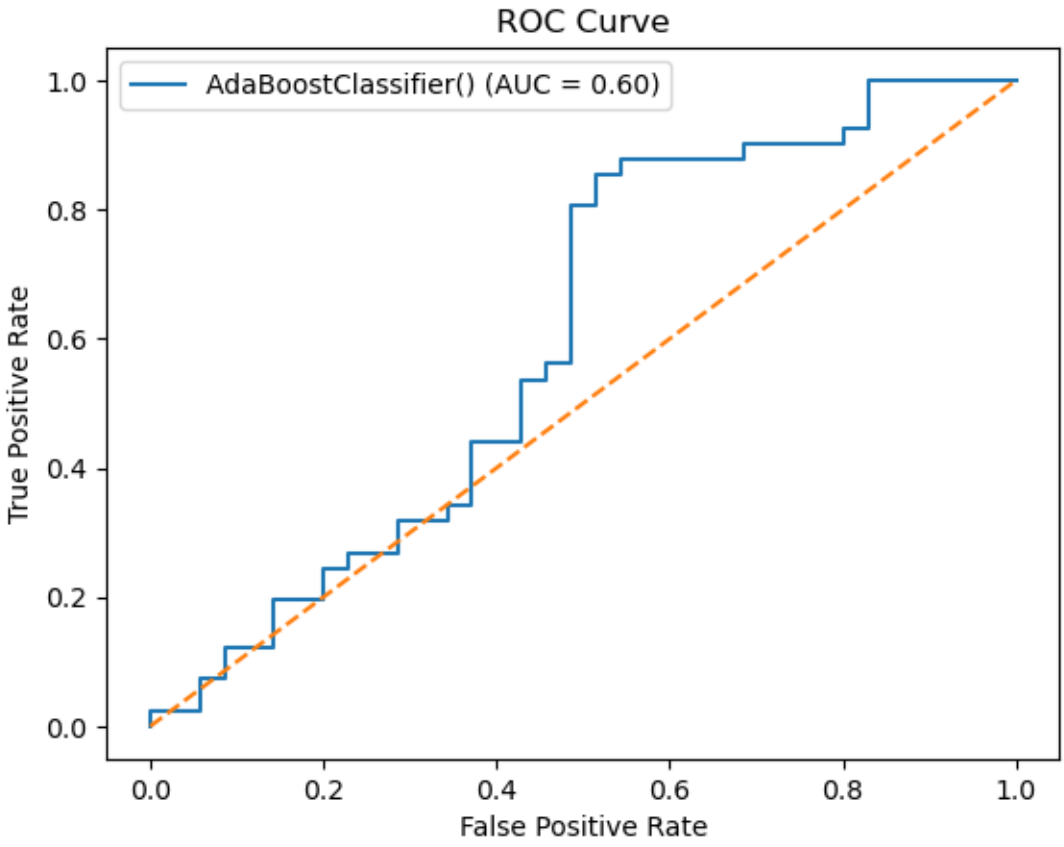
Classifier 3 - Ensemble Trees:

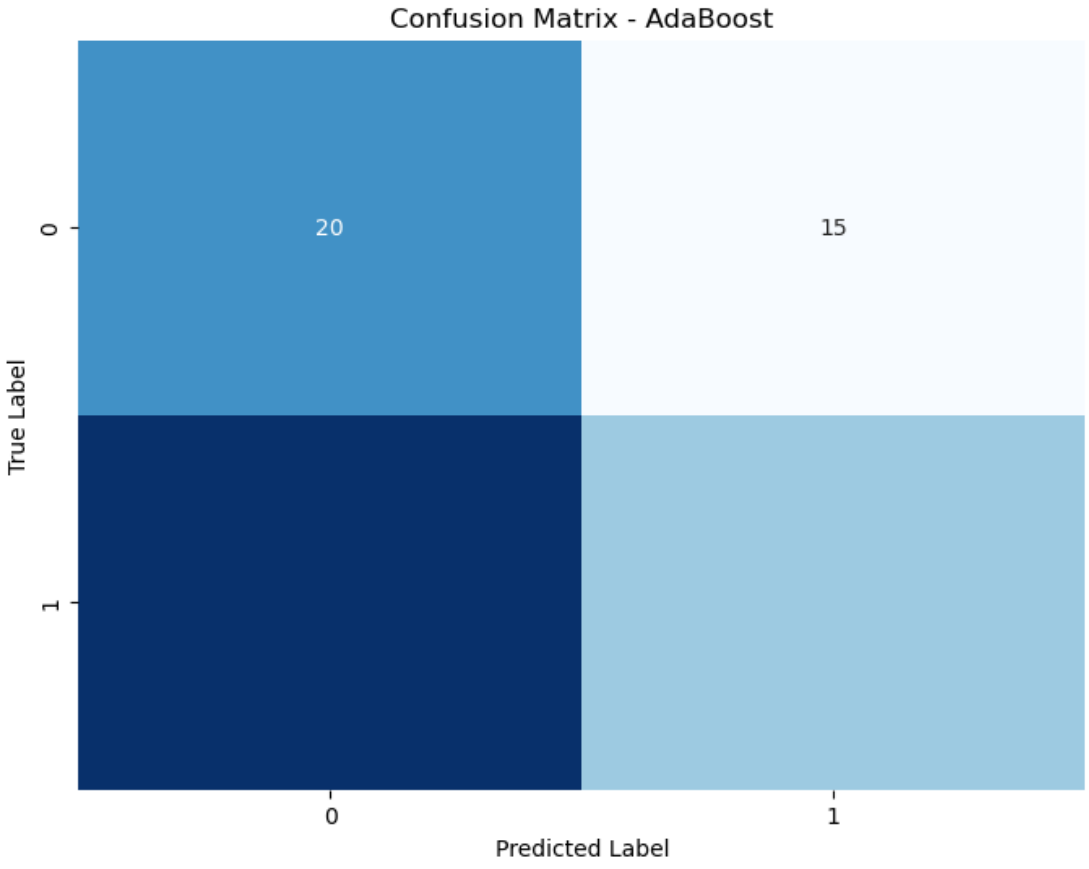
3.1 Random Forest Classifier:





3.2 AdaBoost Classifier:

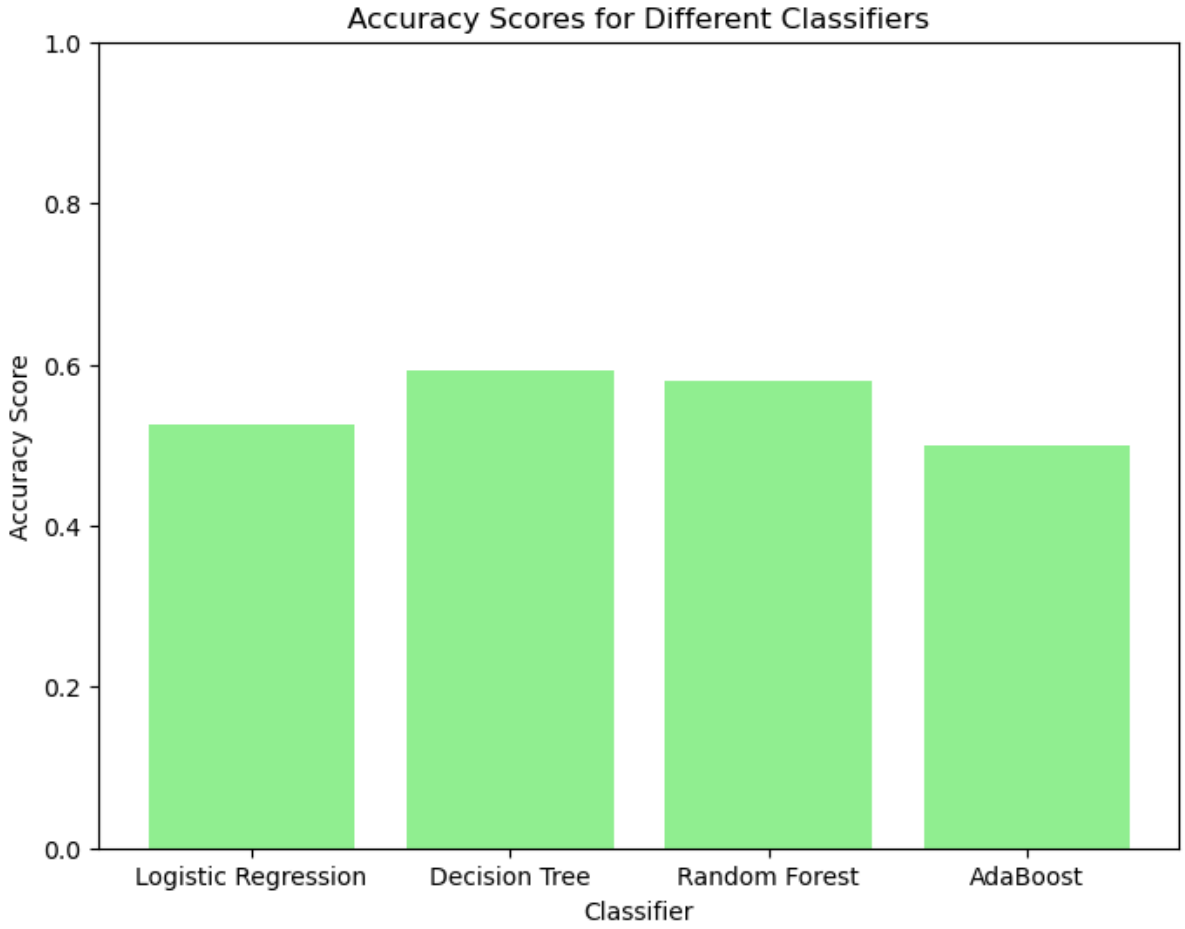




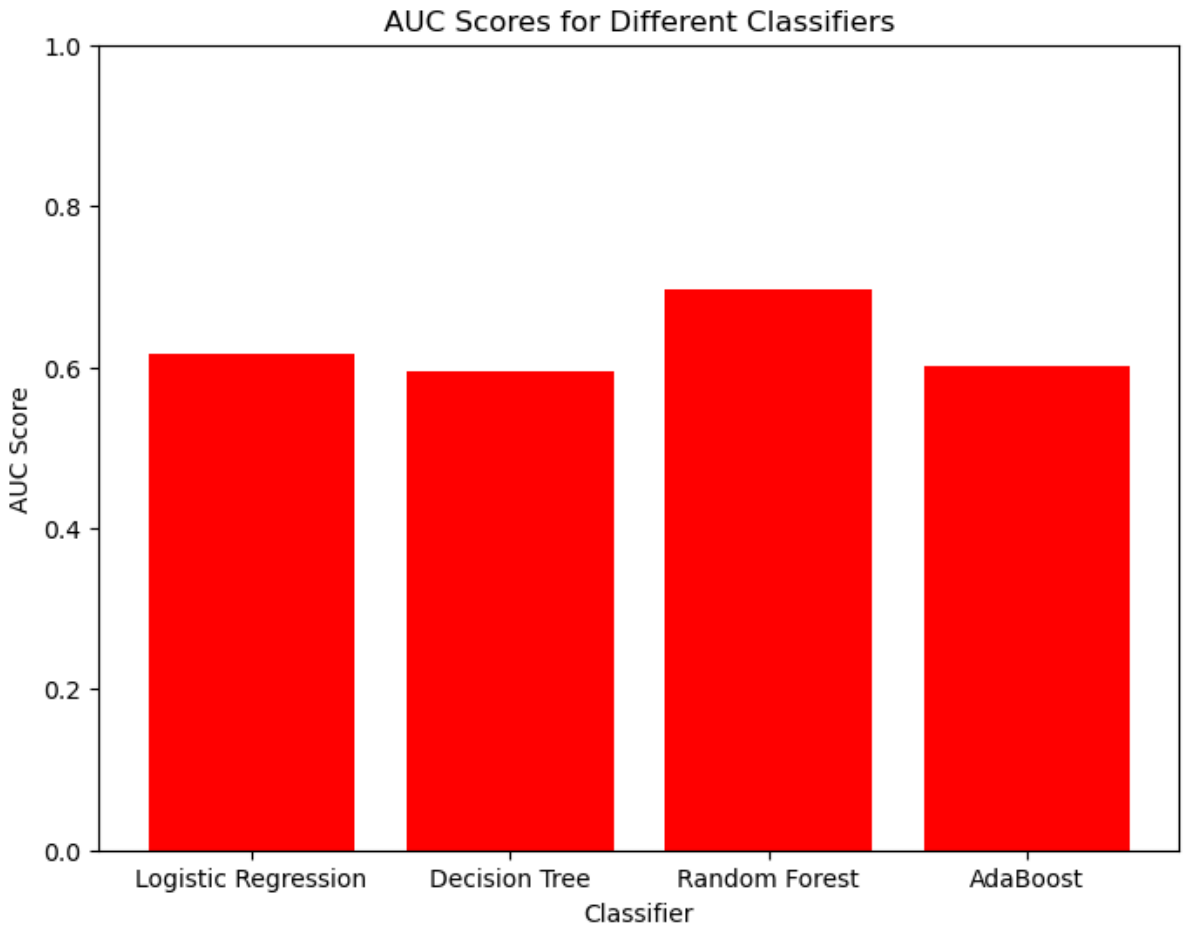
“What metrics did you use to compare the three classifiers?”

To compare the three classifiers, we used two primary evaluation metrics:

1. **Accuracy Score**:
    - This metric measures the proportion of correctly predicted outcomes (both positive and negative) by each classifier. A higher accuracy score indicates better overall predictive performance.



2. **ROC Curve AUC Score**:
    - The Receiver Operating Characteristic (ROC) curve is a graphical representation of the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different threshold values.
    - The Area Under the ROC Curve (AUC) summarizes the ROC curve, providing a single scalar value that measures the classifier's ability to discriminate between positive and negative outcomes. A higher AUC score indicates better discriminative ability.



By comparing accuracy scores and ROC AUC scores across the three classifiers, we were able to assess their overall performance in predicting the outcome variable.

Let's break down the results obtained across the three different classifiers:

1. **Classifier 1 - Logistic Regression:**
    - **Accuracy Score (0.526)**: This indicates that the Logistic Regression model correctly predicted approximately 52.6% of the outcomes in the test set.
    - **ROC Curve AUC Score (0.62)**: The Receiver Operating Characteristic (ROC) curve is a graphical representation of the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different threshold values. The Area Under the ROC Curve (AUC) measures the model's ability to distinguish between positive and negative outcomes. An AUC score of 0.62 suggests that the Logistic Regression model performs moderately well in discriminating between positive and negative outcomes.
2. **Classifier 2 - Decision Tree:**
    - **Accuracy Score (0.592)**: The Decision Tree model achieved an accuracy of approximately 59.2%, indicating that it correctly predicted a higher proportion of outcomes compared to the Logistic Regression model.

- **ROC Curve AUC Score (0.59)**: The AUC score of 0.59 suggests that the Decision Tree model's ability to discriminate between positive and negative outcomes is slightly lower than that of the Logistic Regression model.

3. **Classifier 3 - Ensemble Trees:**
    - **Random Forest Classifier**:
        - **Accuracy Score (0.579)**: The Random Forest Classifier correctly predicted approximately 57.9% of the outcomes, which is slightly lower than the Decision Tree model but higher than the Logistic Regression model.
        - **ROC Curve AUC Score (0.70)**: The AUC score of 0.70 indicates that the Random Forest Classifier outperforms both Logistic Regression and Decision Tree models in discriminating between positive and negative outcomes, with a higher degree of accuracy.
    - **AdaBoost Classifier**:
        - **Accuracy Score (0.500)**: The AdaBoost Classifier achieved an accuracy of 50%, suggesting that its predictive performance is similar to random chance or the baseline accuracy.
        - **ROC Curve AUC Score (0.60)**: While the AUC score of 0.60 indicates better discriminative ability compared to random chance, it is lower than that of the Random Forest Classifier, indicating inferior performance in distinguishing between positive and negative outcomes.

“Conclusion section”

In this study, we explored the predictive performance of three different classifiers—Logistic Regression, Decision Tree, and Ensemble Trees (Random Forest and AdaBoost)—for identifying outcomes related to disease based on symptom profiles and patient demographics. We evaluated the classifiers using accuracy scores and ROC Curve AUC scores as performance metrics.

The results revealed varying levels of performance among the classifiers. The Logistic Regression model exhibited moderate predictive accuracy (52.6%) and discriminative ability (ROC AUC: 0.62). The Decision Tree model showed slightly higher accuracy (59.2%) but lower discriminative ability (ROC AUC: 0.59). Among the ensemble methods, the Random Forest Classifier demonstrated the highest accuracy (57.9%) and superior discriminative ability (ROC AUC: 0.70), outperforming the other classifiers. However, the AdaBoost Classifier yielded suboptimal results with an accuracy of 50% and an ROC AUC score of 0.60.

Recommendations for Future Research:

1. **Feature Engineering**: Investigate additional features or feature transformations that may enhance predictive performance. This could involve exploring interactions between variables or incorporating domain-specific knowledge to create more informative features.
2. **Model Tuning**: Optimize hyperparameters for each classifier to improve performance further. Grid search or Bayesian optimization techniques can be employed to find the optimal parameter settings.
3. **Ensemble Methods**: Explore advanced ensemble techniques such as Gradient Boosting Machines (GBM) or XGBoost to potentially achieve even better predictive accuracy and robustness.
4. **Data Collection and Quality**: Ensure the collection of high-quality data with sufficient sample size and representative diversity to improve model generalization. Address any issues related to data imbalance or missing values.
5. **Evaluation Metrics**: Consider additional evaluation metrics such as precision, recall, and F1-score to gain a more comprehensive understanding of classifier performance, especially in scenarios with imbalanced class distributions.
6. **Interpretability**: Investigate techniques for enhancing the interpretability of predictive models, particularly in healthcare applications, to facilitate trust and understanding among healthcare practitioners.
7. **External Validation**: Validate the developed models on external datasets to assess their generalizability and robustness across different populations or healthcare settings.

By addressing these recommendations, future research can advance the state-of-the-art in disease prediction using data mining techniques, ultimately leading to improved healthcare outcomes and decision-making processes.