# CS-4641-Project

# Heart Failure Prediction Project

## Introduction/Background

Heart failure affects millions of people worldwide annually. Known risk factors include hypertension, age, gender, etc [3]. Existing literature supports that if patients take preventative measures towards their risk factors, their heart failure risk decreases significantly [2].

## Literature Review

- **Lin and Haug**: This study discusses an approach to automating the data cleaning process using metadata from health records, streamlining data preparation for ML models. This approach can improve heart failure models by quickly processing clinical data [5].
- **Sitar-Tăut et al.**: This paper covers the utilization of J48 decision trees to predict cardiovascular disease risks. The models were less successful for predicting stroke and peripheral artery disease, highlighting the need for more nuanced machine learning approaches [1].
- **Mahmud et al.**: The authors proposed a metamodel that was shown to be more accurate than individual machine learning models like Decision Tree, Random Forest, etc., achieving an accuracy of 87% on the dataset, an increase from 67% [4].

## Dataset

- **Name**: Heart Failure Prediction Dataset
- **Link**: Kaggle

- **Description**: The dataset has various biometric features (age, blood sugar, cholesterol) and a label for whether a patient has heart disease or not (0 or 1).

## Problem Definition

The problem for this report is to address the issue of heart failure risk awareness. Many people do not know when they have a serious cardiovascular condition, leading to delayed medical interventions, medication prescription, and likely worse health outcomes. Our motivation is to improve early detection capabilities and inform individuals of their risk through ML Methods, allowing them take preventive measures.

## Methods

### Data Preprocessing Methods

1. **LabelEncoder**: LabelEncoder from sklearn.preprocessing is a tool used to preprocess data by transforming categorical data into numerical data. It scans through a column that is filled with text data and replaces each entry with a unique integer representing the original data. The supervised learning model used for this report is Random Forest, which expects numerical values. Thus, LabelEncoder is an important preprocessing tool necessary for encoding categorical data in this dataset. The columns “Sex”, “ChestPainType”, “RestingECG”, “ExerciseAngina”, “ST_Slope” were encoded to fulfill the numerical input requirement of the random forest model. The algorithm can then interpret these parameters as discrete values for decision making.

2. **StandardScaler**: Additionally, we used the StandardScaler technique to further preprocess data. In the dataset the features such as blood pressure and cholesterol levels have different ranges of values. Without applying a standardization technique, some features with larger ranges would overpower features with smaller ranges. However, because we used StandardScaler, the feature values were standardized to ensure that our Random Forest Model would account for all features and thus accurately predict the risk of an individual’s heart disease.

### ML Algorithm/Model

1. **Random Forest**: Random Forest is a supervised learning model that is perfect for handling both small and large datasets effectively. It is also known to be robust for overfitting. For our data, we decided to use RandomForestClassifier from sklearn.ensemble to help examine complex interactions between features such as categorical data and numerical data. This is needed because the data we are using is diagnostic data including diverse features like discrete categories like sex and chest pain type but at the same time we have continuous measurements such as age and cholesterol levels. Something we also noticed was that medical conditions are going to involve complex interactions between different health indicators. An example of this is how factors like age, cholesterol, and blood pressure might all have a combined impact to influence heart disease risk. Random Forest's tree structure is particularly good at getting this sort of detail and intricate relationships between features.
2. **SVM**: The second supervised learning model used for the heart failure dataset is the Support Vector Machine (SVM) algorithm. SVM classifies data by finding the optimal hyperplane that maximizes the separation of data points between classes, associated with the largest margin. In the project proposal, we initially planned to use logistic regression because it is a reliable model that can be easily interpreted. However, after some further consideration, we found that SVM offered more advantages for problems with non-linear decision boundaries such as the heart failure dataset. This model is very effective in handling binary classification tasks as well as high dimensional data. Features such as cholesterol levels and blood pressure could interact nonlinearly, and SVM can handle these features by finding the best hyperplane and finding the best performance. Further, even though our dataset contains a combination of categorical and continuous features, the SVM model can continue to accurately detect heart failure risk because it is resistant to overfitting when properly tuned. For this reason, SVM is a great choice of supervised model for this task.
3. **Gradient Boosting**: The last supervised learned model we used for the heart failure dataset is eXtreme Gradient Boosting (XGBoost). Our data is a mix of categorical and numerical features. It contains critical data points like non linear relationships between features like chest pain type, resting ECG results, and various numerical measurements such as blood pressure and cholesterol. This is perfect for XGBoost, because it effectively handles this type of data through its boosted decision tree structure. XGBoost is going to perform very well here because it can actually figure out which medical measurements actually matter for predicting heart disease. For example, it might learn that certain ECG patterns or chest pain types are more important warning signs than age or resting blood pressure. Additionally something super important to note and one of the requirements we were trying to focus on is the speed, which is where XGBoost shines. XGBoost can do all of this analysis in the matter of seconds because of the way it is arcticured with features like parallel processing, block structure, and cache awareness. These are the reasons why we have decided to use XGBoost for our project, specifically for the heart failure dataset.

# Results and Discussion

## Random Forest Analysis

We found the accuracy, generated a confusion matrix, and plotted the ROC Curve after generating predictions using the Random Forest Classifiers performance. We achieved an accuracy of 84.4 percent, measuring the proportion of correct predictions for both heart disease and no heart disease. This achieves our project’s goal of having an accuracy of over 80%, but we also analyzed our models in terms of true positives (TP) , true negatives (TN) , false positives (FP) , and false negatives (FN) using the following confusion matrix since our data is imbalanced.

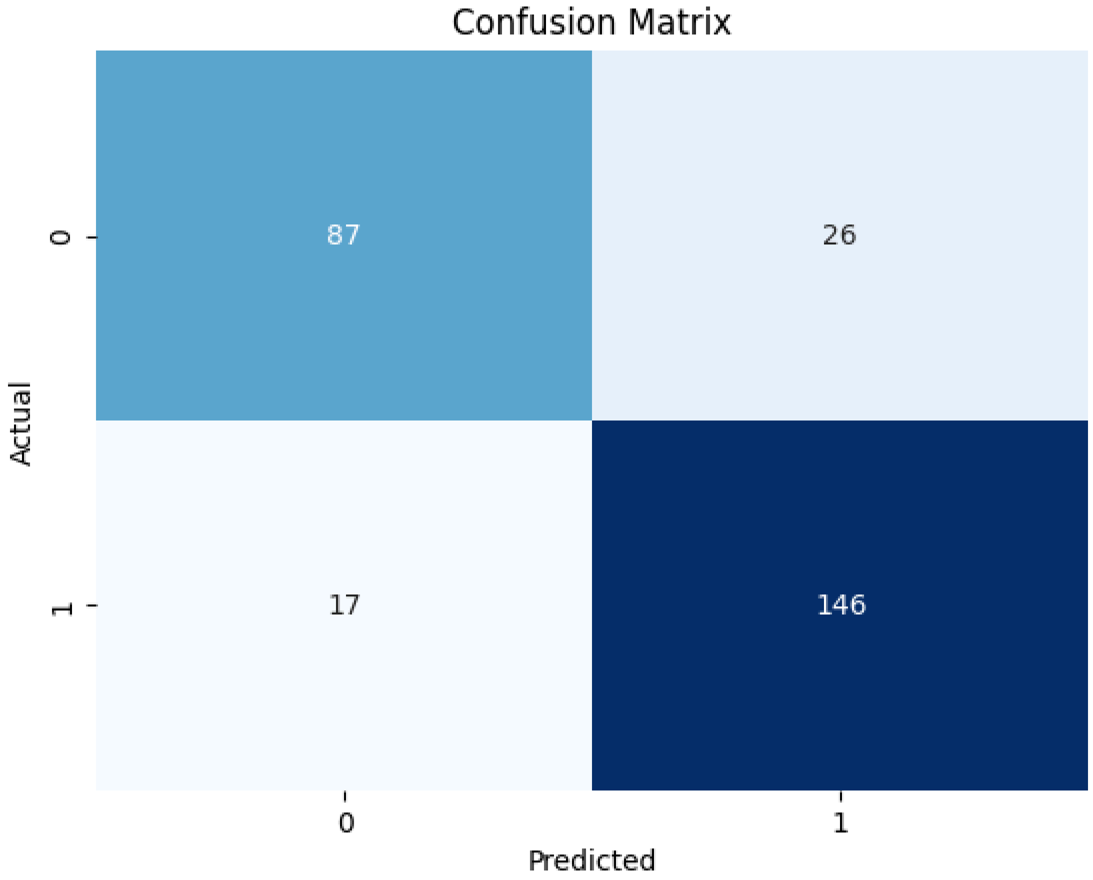


*Figure 1: Confusion matrix for Random Forest Classifier Model detailing the model's true positive, true negative, false positive, and false negative rates.*

The confusion matrix shows that we have a TP rate of 146, a TN rate of 87, FP rate of 26 , and a FN rate of 17. Using these values, we calculated the precision of 0.849 which is the ratio of true positive predictions (patient correctly identified as having heart disease) to all positive predictions (both true positives and false positives). This means that out of all the patients the model predicted as having heart disease, 85% actually have it. This is crucial in medical predictions because a lower precision would mean more false positives—patients incorrectly flagged as having heart disease, which could lead to unnecessary testing or anxiety. We also calculated recall which is the ratio of the true positives to all actual positives (true positives and false negatives). With a recall of 0.896, the model correctly identifies ~90% of patients who actually have heart disease. High recall is important in medical contexts because it minimizes false negatives, ensuring that fewer patients with heart disease are missed. Missing a positive case could have serious health consequences if a patient with heart disease goes untreated.

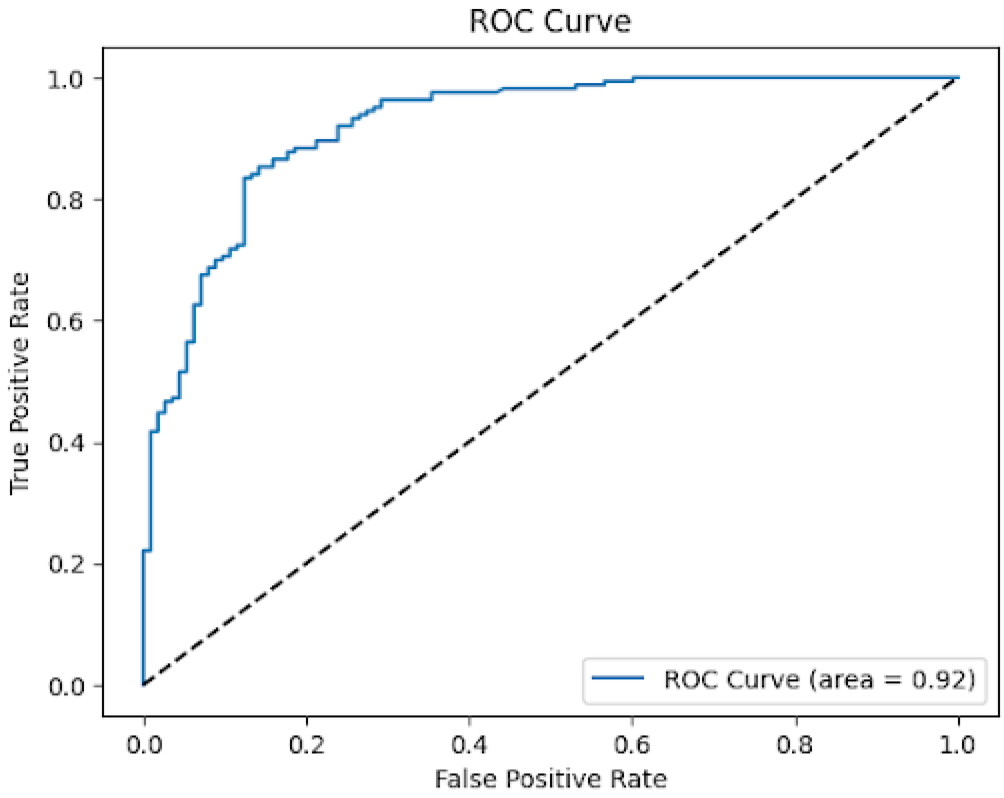


*Figure 2: The Receiver Operating Characteristic (ROC) Curve plots the true positive rate against the false positive rate at various threshold settings.*

The area under the ROC curve tells us if the model performs well in terms of distinguishing between binary classes, and an AUC of 0.92 indicates that there is a high probability that a randomly chosen heart disease patent will be ranked higher than a randomly chosen non-heart disease patient, which is ideal for a medical prediction task. In addition, the curve is visually far from diagonal (random guessing) which indicates a strong ability of the model to achieve high recall while keeping the false positive rate low.

Overall, the evaluation suggests that the model is effective at distinguishing between patients with and without heart disease, which is critical for medical prediction tasks. However, some areas of improvement include reducing false negatives, so we plan on tuning our random forest classifier such as tuning the decision threshold to achieve a better balance between precision and recall to minimize false negatives. In addition, we plan on performing more feature preprocessing and engineering, such as using SelectKBest from sklearn.feature_selection to select features that are the most important instead of using all the features we have. Finally, plan on training Gradient Boosted models since they provide more control over hyper parameters and may handle imbalanced data sets better since it prioritizes errors and adjusts to the minority class better than random forest.

## SVM Analysis

Using the SVM classifier, we have determined the accuracy and efficiency by different metrics, plotted the confusion matrix, and plotted the ROC curve. The model had an accuracy of 86.95 percent, which is above our threshold of 80 percent. The Confusion Matrix for the model is as follows: TN: 102, TP: 138, FP: 22, FN: 14.

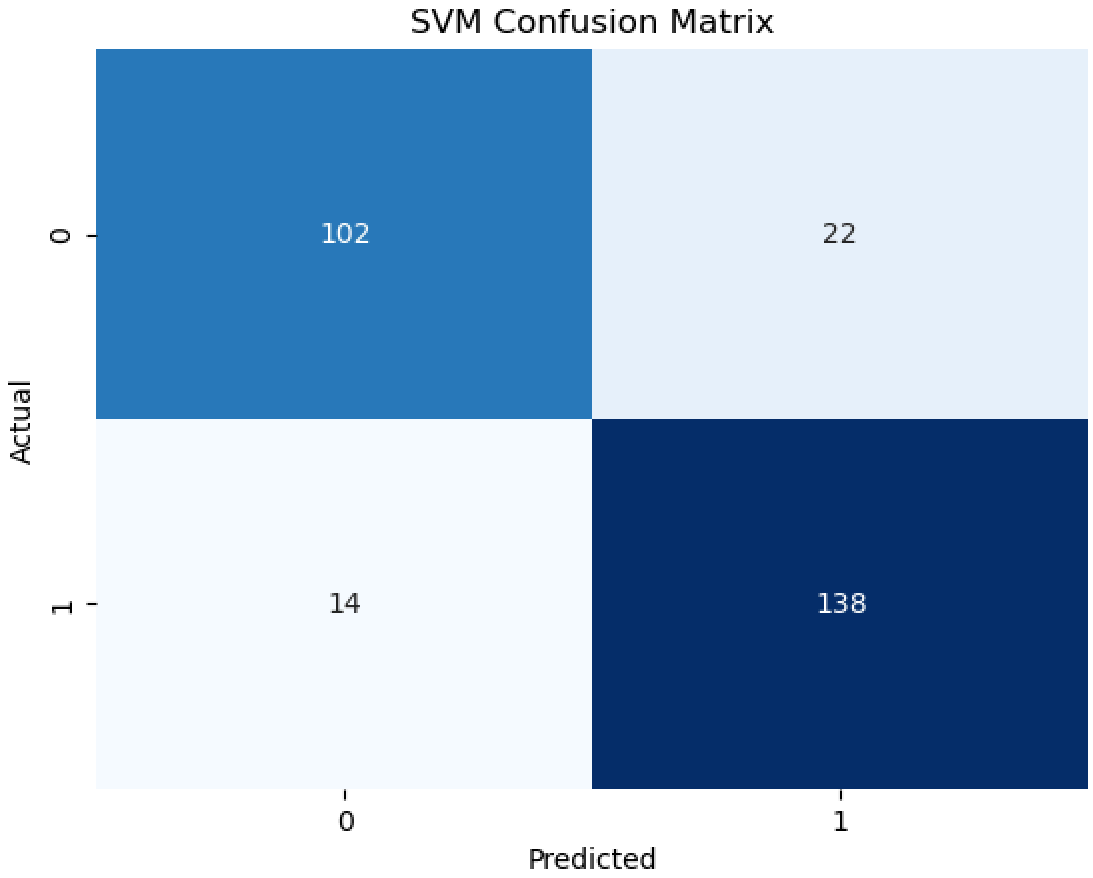


*Figure 3: Confusion matrix for SVM Classifier Model detailing the model's true positive, true negative, false positive, and false negative rates.*

The precision value of 0.8625 details that approximately 86 percent of patients were correctly identified to have heart disease, thus avoiding the false positives. Recall of 0.9078 defines that the model can predict approximately 90 percent of actual heart disease cases correctly, which means that we can effectively reduce false negatives. Finally, we plotted the ROC Curve for the SVM Classifier to check the true positive rate against the false positive rate at different threshold settings.

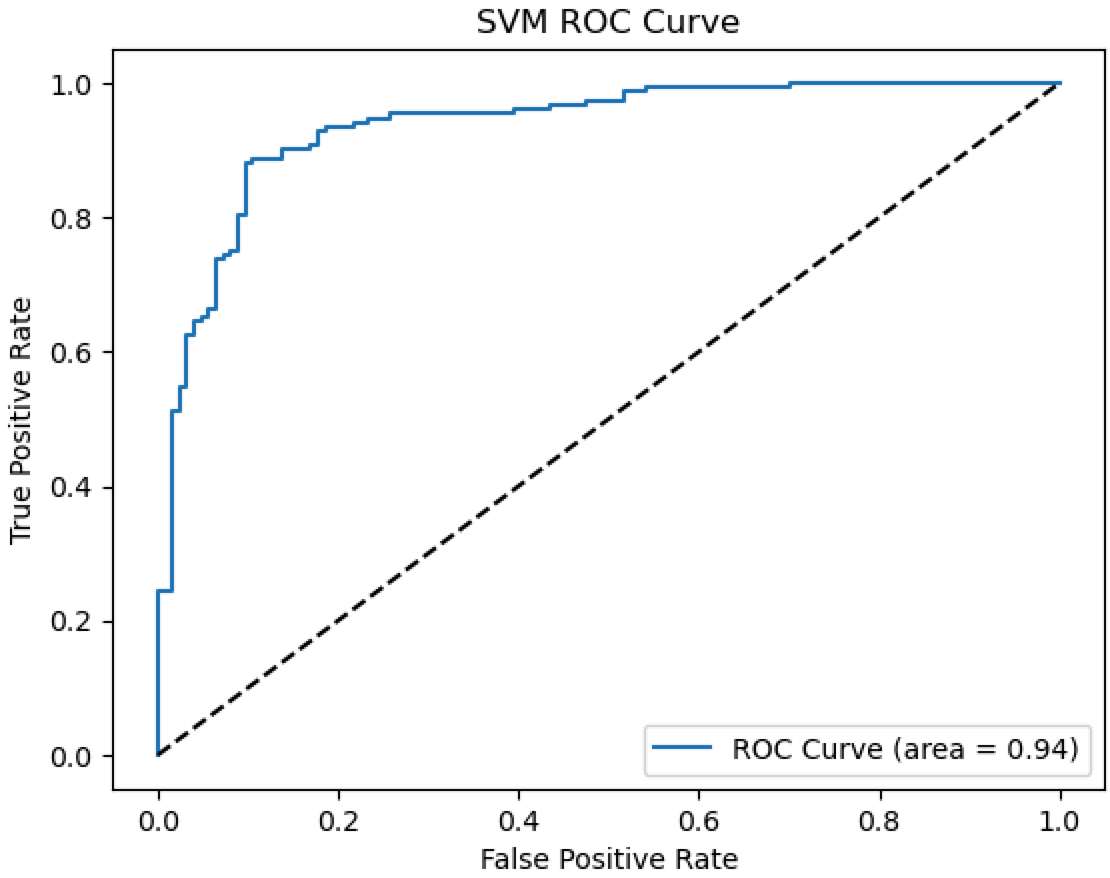


*Figure 4: The Receiver Operating Characteristic (ROC) Curve for SVM Classifier.*

The AUC of 0.936 and the curve's separation from the diagonal indicate that the model has a good ability in differentiating between heart disease and non-heart disease cases. Overall, SVM is good at minimizing false negatives, but we can try to further improve the accuracy by tuning kernel selection or using feature selection methods. This would help to handle the interaction of features and imbalance in data more effectively.

## Gradient Boosting Analysis

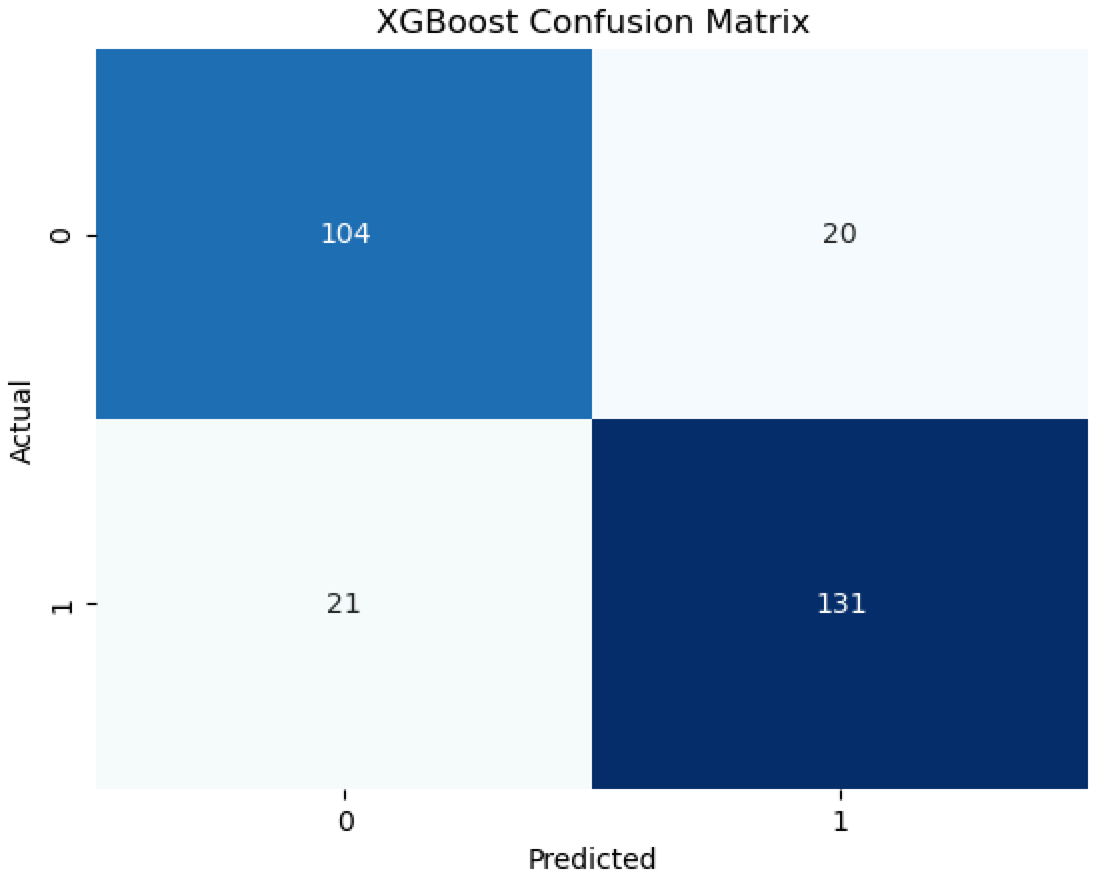


*Figure 5: Confusion matrix for XGBoost Classifier Model detailing the model’s true positive, true negative, false positive, and false negative rates.*

The XGBoost model achieved an accuracy of 85.1%, surpassing our target of 80%. The confusion matrix reveals 104 true negatives (TN), 131 true positives (TP), 20 false positives (FP), and 21 false negatives (FN). With a precision of 86.8%, the model correctly identifies 87% of patients predicted to have heart disease, reducing false positives. The recall of 86.2% ensures most heart disease cases are detected, minimizing false negatives—critical in medical diagnostics.

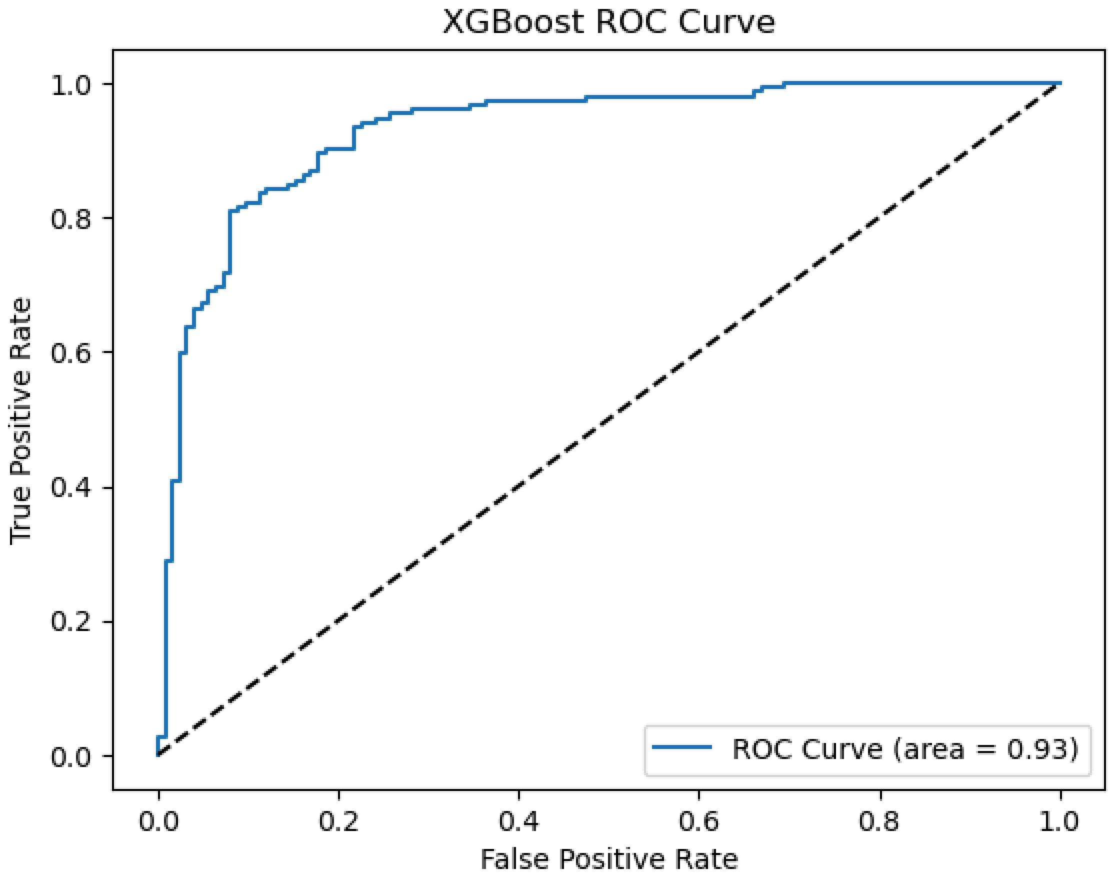
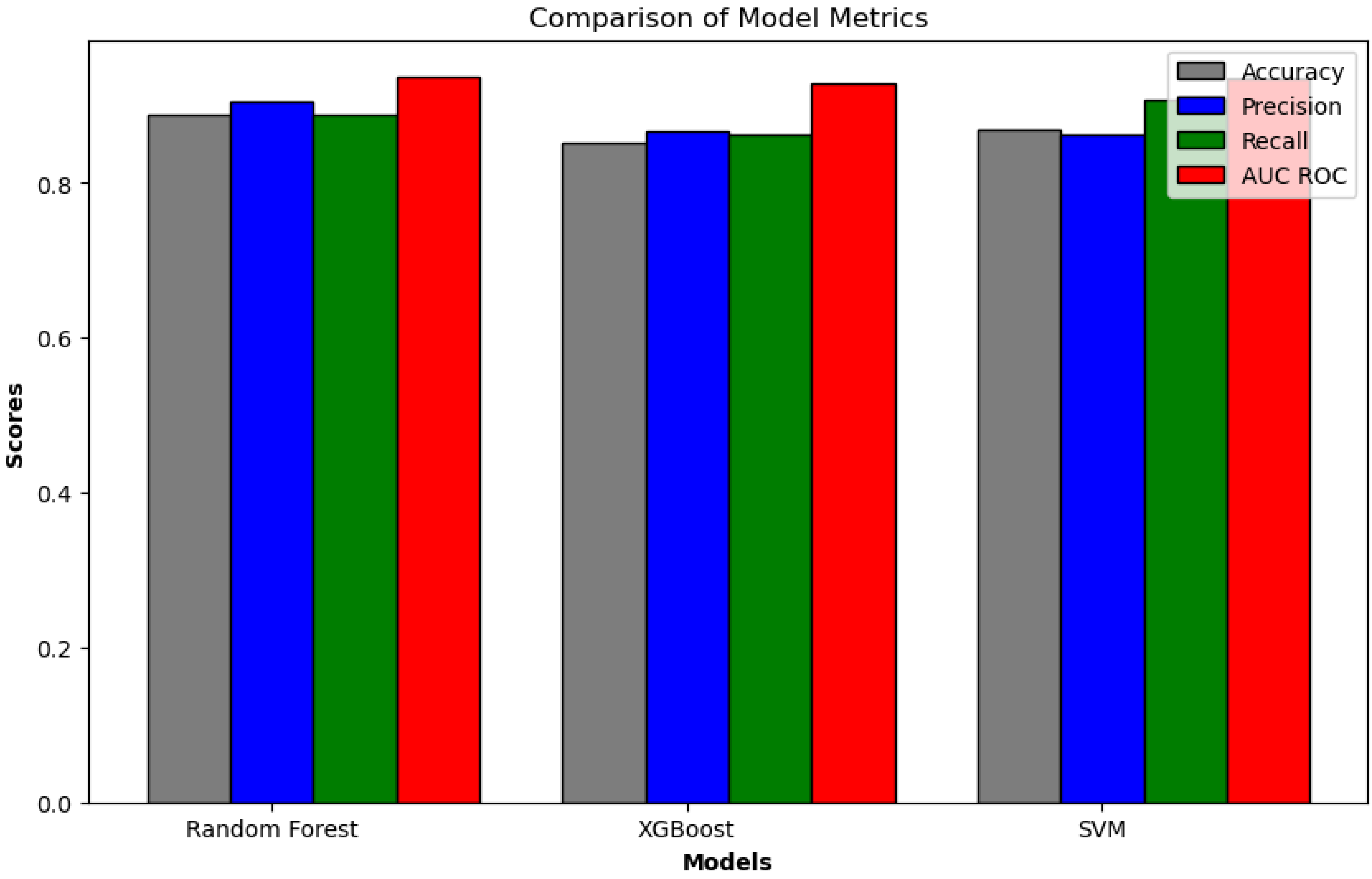


*Figure 6: The Receiver Operating Characteristic (ROC) Curve for XGBoost Classifier.*

The ROC AUC of 0.93 highlights the model's strong ability to distinguish between heart disease and non-heart disease cases, reflected in a curve far from the random baseline. While the model performs well, we aim to reduce false negatives by adjusting decision thresholds and improving feature selection with tools like SelectKBest. Additionally, exploring Gradient Boosting could enhance performance, as it better handles imbalanced datasets and prioritizes errors effectively.

## Comparison of 3+ Algorithms/Models



When comparing the three algorithms and looking at their performance, we noticed that the random forest model had the best accuracy at 88.77% as well as the highest precision at 90.60%. Due to its high area under the ROC curve as well, we concluded that random forest was our best performing model for heart failure detection. Although we concluded this, we found a couple of key trade-offs that are worth noting. The first one is how Random Forest is actually more computationally intensive than the other models we used here. This is because of it having to aggregate all of the decision trees and then giving us our final output. Even though SVM had the precision at 86.25%, it had the highest recall at 90.79%. This indicates that SVM might be better for identifying true positives, but a limitation/tradeoff is that this could result in more false positives. However, in the context of heart failure prediction, a false negative can cause significantly more damage than a false positive, which shows that SVM can be an effective predictor. Lastly, XGBoost is more of a balanced precision which basically means more measured predictions in both directions. But with this there is a tradeoff which is slightly lower overall accuracy compared to the other models. But when we look at our use case of heart failure prediction, this balanced approach could actually be pretty valuable in clinical settings where there is a need to weigh multiple factors and prefer a model that doesn't lean too heavily toward either false positives or false negatives.

### Next Steps

Firstly, some future steps we can take to optimize random forest include altering the number of trees and the maximum depth. Random Forest had good accuracy and precision, but these changes can also help boost up the recall. For SVM, one area of experimentation to fine-tune the model is to experiment with the type of kernel and try a polynomial kernel in addition to the RBF kernel which we are currently using. Another change we can experiment with to further optimize the model is adjusting the regularization parameter which can bump up the lower precision numbers we were getting. Thirdly, next steps for our XGBoost algorithm is to make sure we focus on hyperparameter tuning to simply improve its balanced prediction capabilities while at the same time targeting its weak spot in overall performance. Additionally, we want to explore incorporating additional medical features to make sure that the model predictive power can be enhanced in identifying heart disease cases. Lastly, we can also try searching for more data to create better predictors and also use different models in order to find the most accurate one. With medical advancements happening frequently, stronger and better indicators are emerging as ways to detect potential heart failure. New features and data could help make our models perform even more effectively.

## References

[1] A. Sitar-Tăut, D. Zdrenghea, D. Pop, and D. Sitar-Tăut, "Using machine learning algorithms in cardiovascular disease risk evaluation," Age, vol. 1, no. 4, pp. 4, 2009.

[2] Djoussé, L., Driver, J. A., & Gaziano, J. M. (2009). Relation between modifiable lifestyle factors and lifetime risk of heart failure. JAMA, 302(4), 394-400.

[3] Groenewegen, A., Rutten, F. H., Mosterd, A., & Hoes, A. W. (2023). Epidemiology of heart failure. European Journal of Heart Failure, 25(2), 200-213.

[4] I. Mahmud, M. M. Kabir, M. F. Mridha, S. Alfarhood, M. Safran, and D. Che, "Cardiac Failure Forecasting Based on Clinical Data Using a Lightweight Machine Learning Metamodel," *Diagnostics (Basel)*, vol. 13, no. 15, p. 2540, Jul. 2023, doi: 10.3390/diagnostics13152540.

[5] Lin JH, Haug PJ. Data preparation framework for preprocessing clinical data in data mining. AMIA Annu Symp Proc. 2006;2006:489-93. PMID: 17238389; PMCID: PMC1839316.

## Gantt Chart

Link to Gantt Chart

## Contribution Table

| Name | Contribution |
|---|---|
| Anay | Model Selection and Model Coding and Presentation |
| Vivek | Data Sourcing and Cleaning and Presentation |
| Akshay | Method Explanation and Results Analysis and Presentation |
| Rishi | Method Explanation and Results Analysis and Presentation |
| Daiwik | Model Coding and Data Cleaning and Presentation |

## Directory Structure

`/data/` : Contains the data files that are going to be used for the training and testing on model

`/data/heart.csv` : Main dataset that has heart disease patient records with some features like age, sex, chest pain type and many more `heartDisease.ipynb` : This is a Jupyter Notebook containing the main model implementation for all 3 models and has things like:

- Data preprocessing steps
- Random Forest, SVM, and XGBoost model training
- Model evaluation including accuracy, precision, recall
- Visualization of results using confusion matrix and ROC curves

`/requirements.txt` : Lists all Python dependencies required to run the project such as scikit-learn, pandas, numpy, and matplotlib

`/visualizations/` : Directory containing visualizations for our machine learning models

`/visualizations/comparison_of_models.png` : PNG image of the model comparison

`/visualizations/random_forest_confusion_matrix.png` : PNG image of the Random Forest Model's Confusion Matrix

`/visualizations/random_forest_roc_curve.png` : PNG image of the Random Forest Model's Receiver Operating Characteristic (ROC) Curve

`/visualizations/svm_confusion_matrix.png` : PNG image of the SVM Model's Confusion Matrix

`/visualizations/svm_roc_curve.png` : PNG image of the SVM Model's Receiver Operating Characteristic (ROC) Curve

`/visualizations/xgboost_confusion_matrix.png` : PNG image of the XGBoost Model's Confusion Matrix

`/visualizations/xgboost_roc_curve.png` : PNG image of the XGBoost Model's Receiver Operating Characteristic (ROC) Curve