Team 48 ML Project

Introduction/Background

Heart disease is a prevalent and fatal illness, regularly diagnosed in a diverse set of people. Heart disease prediction is a beneficial tool in healthcare that studies previous diagnoses, patient statistics, and additional patient conditions to recognize heart disease in early stages. Since heart disease continues to be a global health concern, several studies have been conducted on the usage of machine learning to predict heart disease risk, in order to prevent the development of the disease and ensure patients continue to live a healthy lifestyle. In *Predicting Disease with Deep Learning [3]*, researchers used a deep neural network and the LinearSVC method for feature selection to effectively predict heart disease. Additionally, in *Using Machine Learning to Predict Heart Disease [1]*, different algorithms and their effectiveness in predicting heart disease are compared. The dataset we are using is a prelabeled Heart Disease Dataset from the University of California Irvine which is suitable for predictive modeling.

Problem definition

The primary motivation for this project is to improve recoverability and survivability rates of heart disease through early prediction and detection. Approximately 5.7 million Americans experience heart failure with 825,000 new cases and 280,000 mortalities per year [4]. Early detection of heart disease can help patients make positive lifestyle changes and receive medical intervention before the disease progresses too far [2]. By training a model using data from previously diagnosed patients, we aim to predict if a patient is prone to heart disease in order to begin prevention and treatment as soon as possible.

Methods

Data Preprocessing

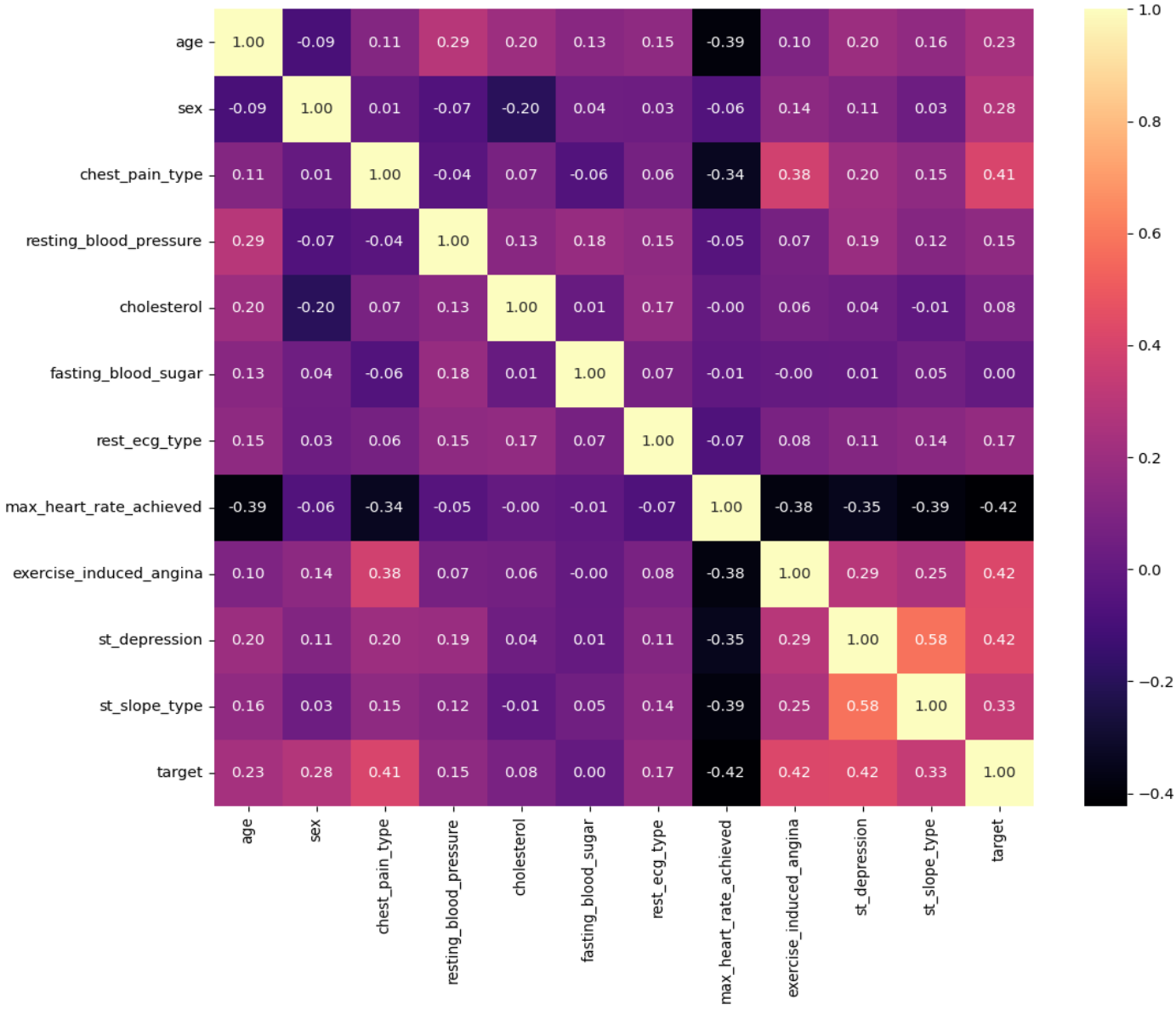
Our initial step involved managing missing and null values by dropping them and transforming our target variable into a binary representation (0 or 1) to denote the absence or presence of the risk of heart disease. Next, we conducted exploratory data visualizations to gain a deeper understanding of the dataset. Finally, multiple dimensionality reduction methods were implemented on our dataset.

Handling Missing Values

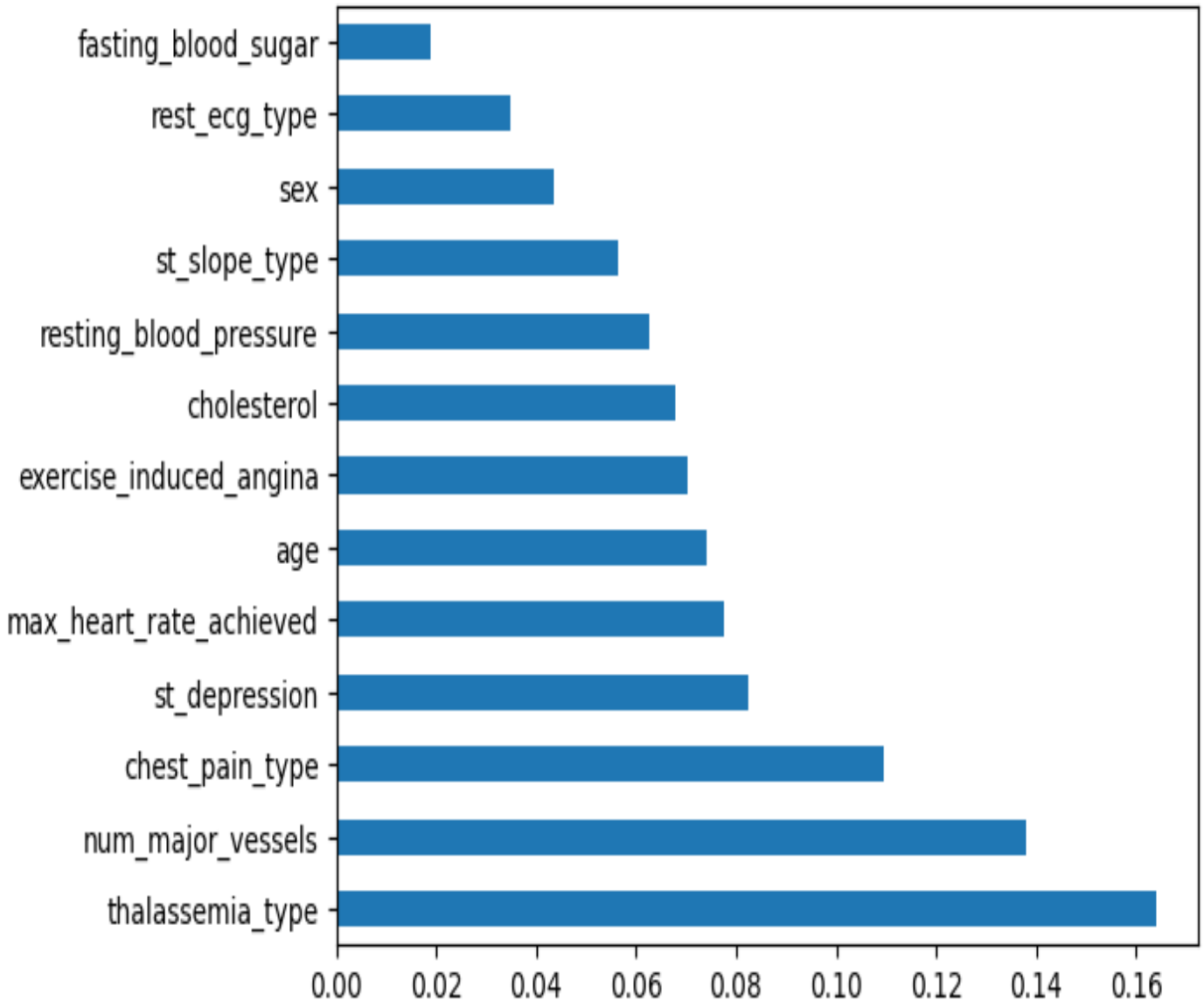
Our data had only 6 missing values, so we were able to handle the missing values by simplifying and removing the rows with the missing values. This is because those rows made up less than 0.01% of our data set and from later testing, we saw that it did not have much of an impact on the total dataset.

Data Visualization

We conducted a correlation analysis to see the dependencies of features, and use these results for feature selection. We started with a heatmap to understand the feature correlation to the outcome and to the remaining features. This allowed us to determine which features have a stronger correlation to the outcome and to each other. We can remove strongly correlated features to prevent double counting their weight in the model

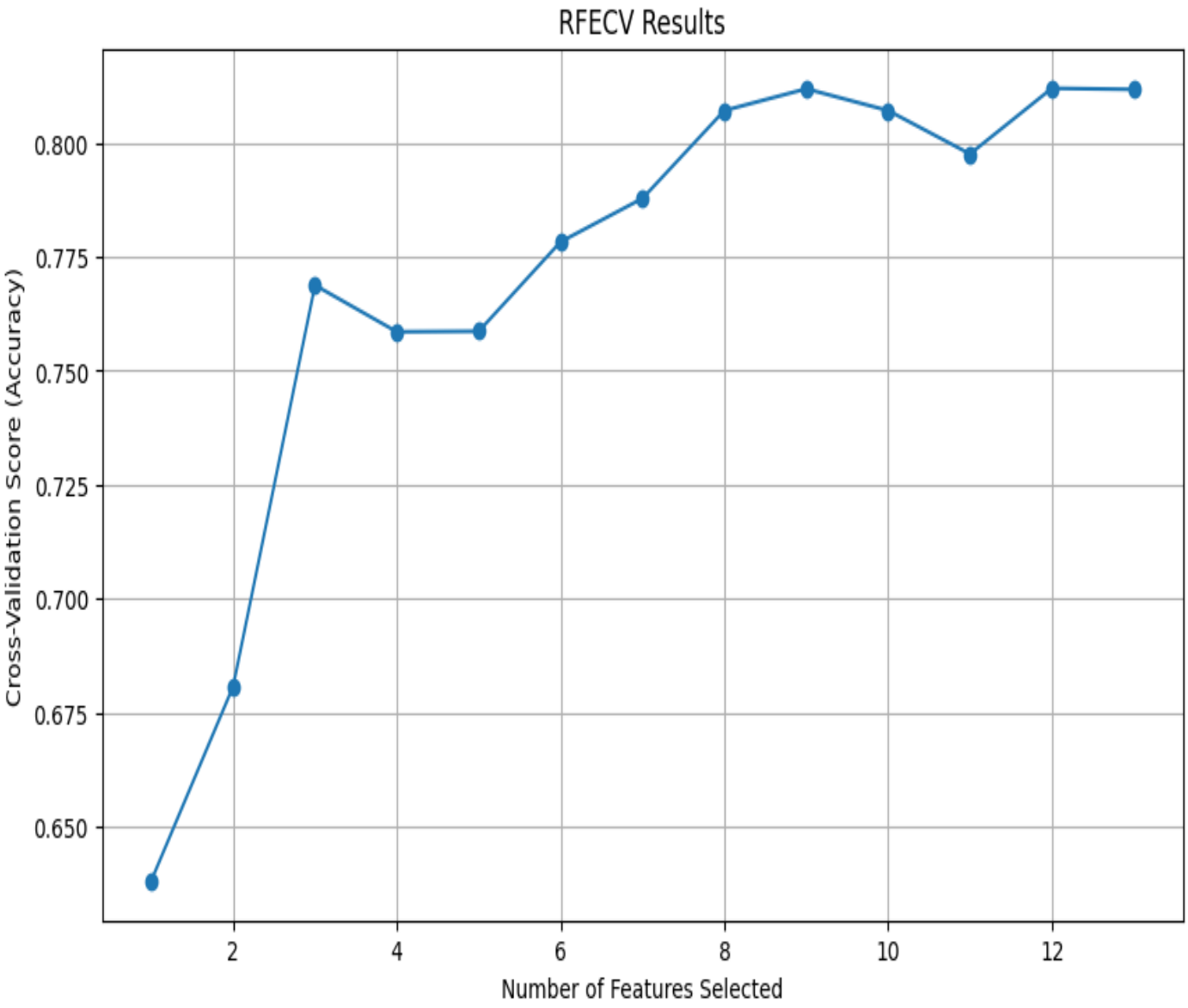


We additionally created a tree based classifier to visualize feature importance, aiding us when we perform feature elimination on our dataset.



Recursive Feature Elimination with Cross Validation

We will then use the recursive feature elimination with cross-validation to determine which clinical and demographic values are more accurate indicators. By using cross-validation we don't have to input a select number of features and can allow the algorithm to recursively eliminate and add features assessing the performance of the model

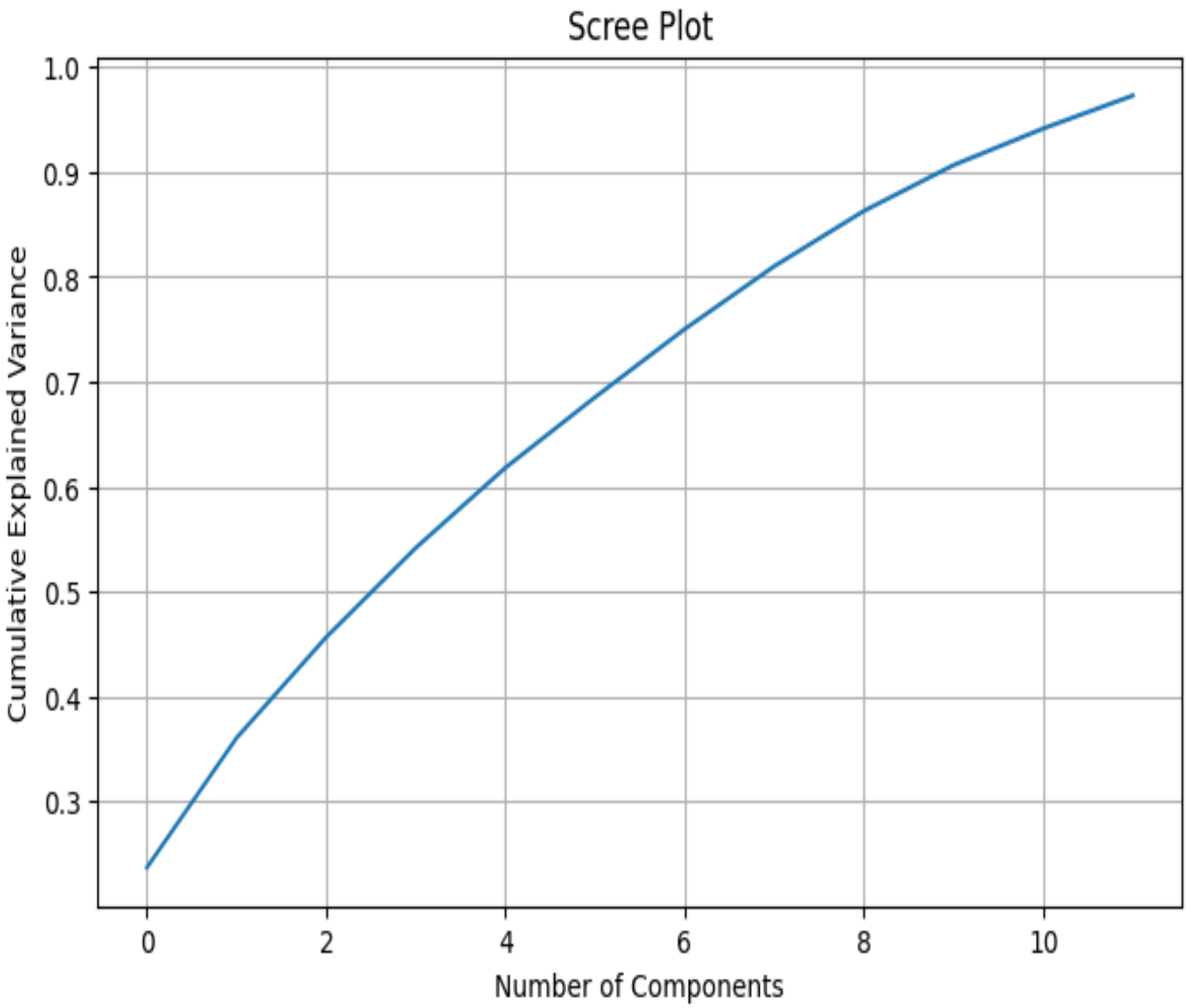


As shown in the graph the cross-validation accuracy is highest with 12 features. So when we run our models using this preprocessed data we will use all features except cholesterol levels.

Principal Component Analysis

We also implemented PCA do cross-check our results from RFECV.

From this Scree Plot I can see that the optimal features is also 12.



Logistic Regression, Random Forest, and Naive Bayes Models

For the actual model implementations, we used Logistic Regression, Random Forest, and Naive Bayes to implement a binary classification.

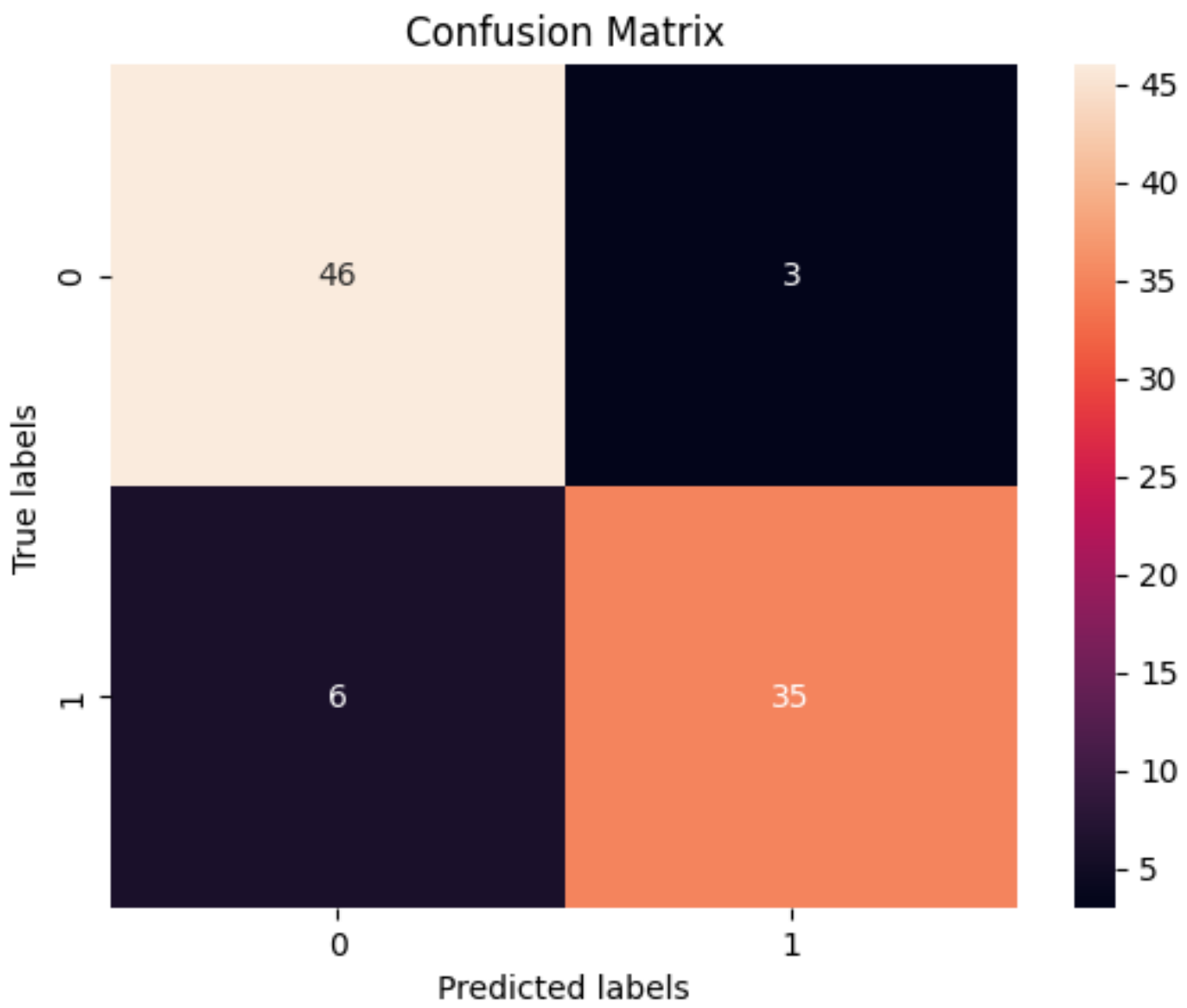
For Logistic Regression and Random Forest, we used our two different preprocessed data sets: One using RFECV for dimensionality reduction, and one using PCA for dimensionality reduction. With this, we can directly compare which preprocessed dataset yields more accurate results.

Results

Logistic Regression with RFECV Dataset

After training and fitting our model with the dataset processed by RFECV, we got an overall accuracy of 90%

We represented our results in the a confusion matrix and a classication matrx below.

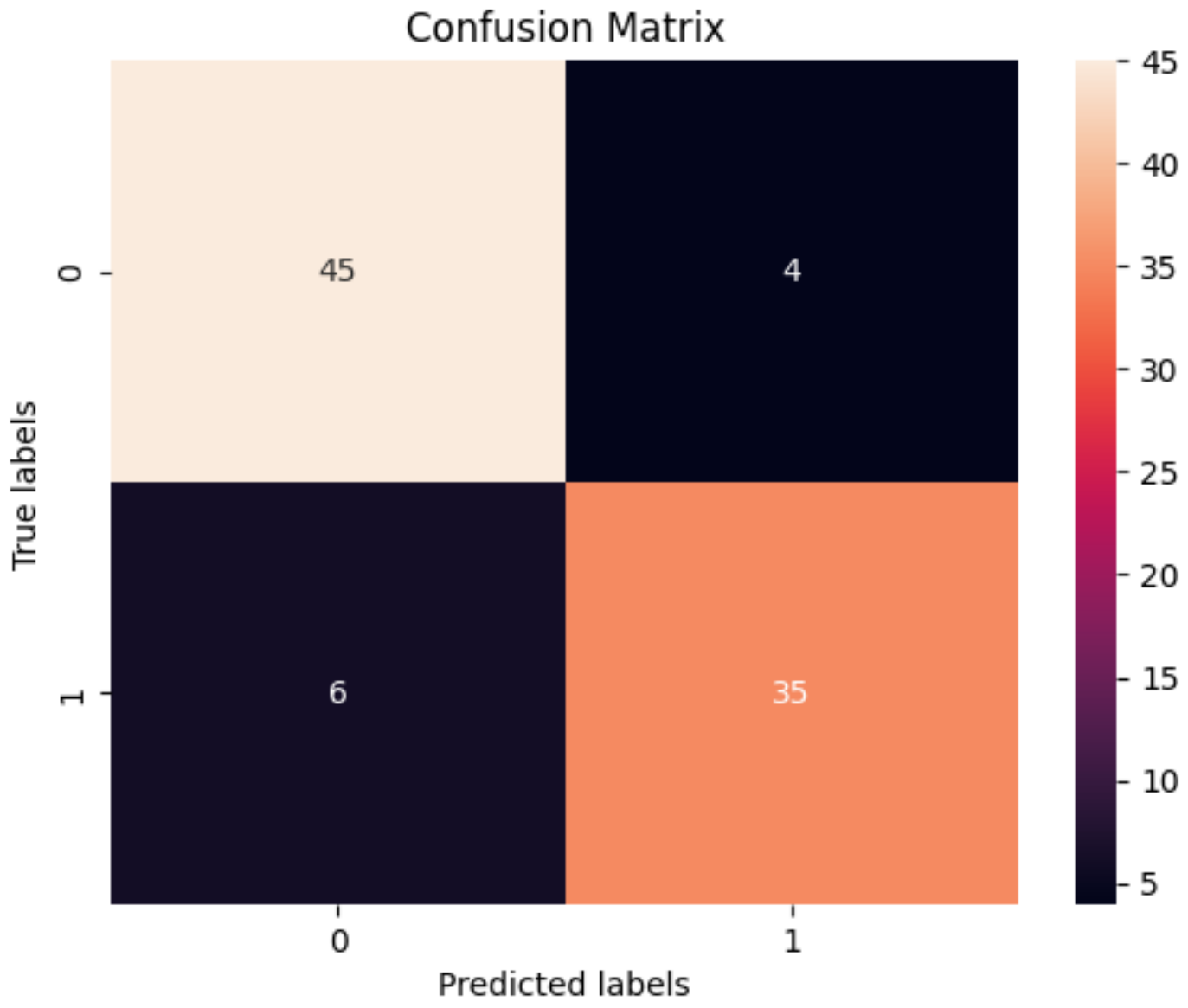


	precision	f1-score	support	
0	0.88	0.94	0.91	49
1	0.92	0.85	0.89	41
accuracy			0.9	90
macro avg	0.9	0.9	0.9	90
weighted avg	0.9	0.9	0.9	90

Logistic Regression with PCA Dataset

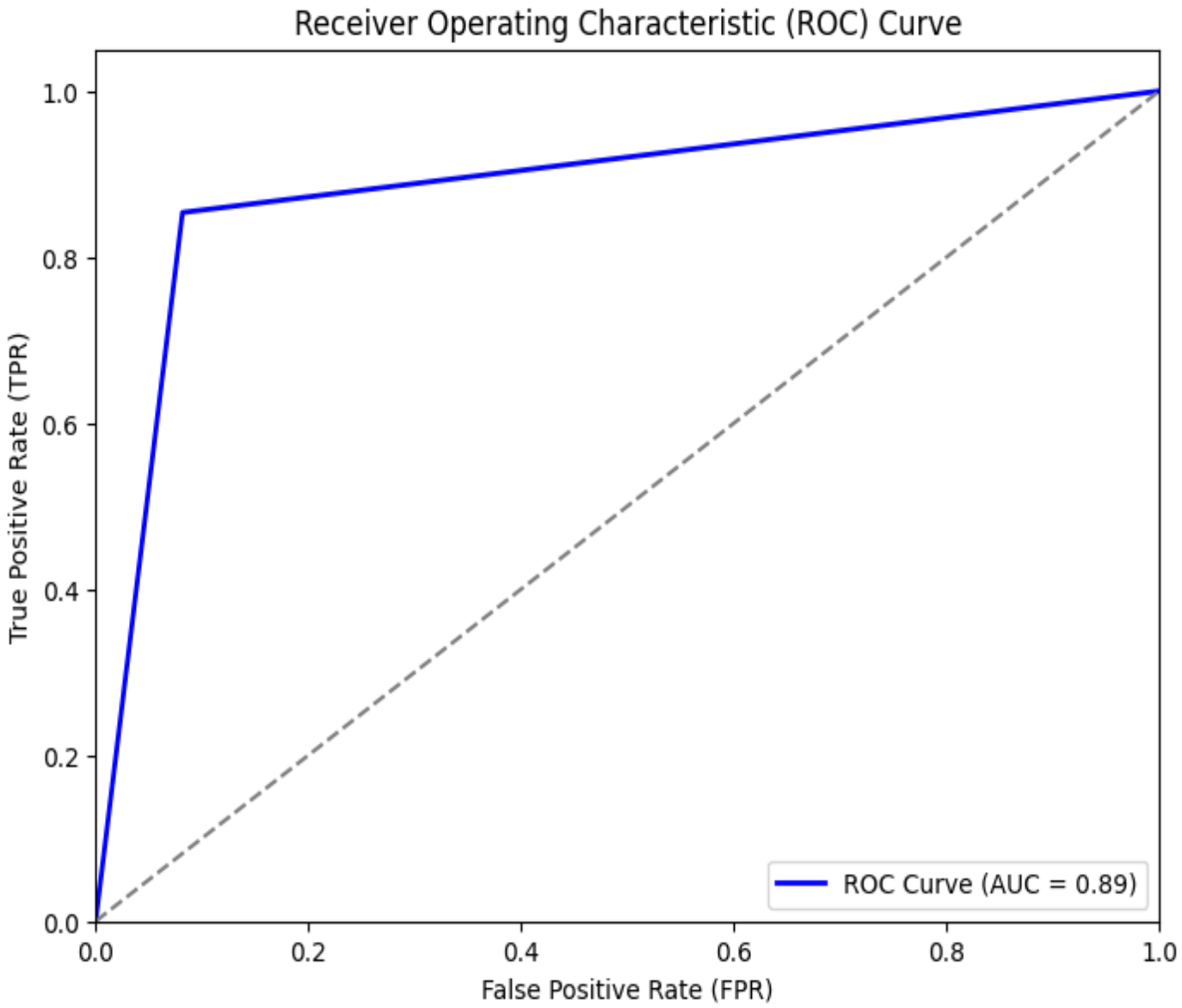
After training and fitting our model with the dataset processed by PCA, we got an overall accuracy of 89%

We represented our results in the a confusion matrix and a classication matrx below.



	precision	f1-score	support	
0	0.88	0.92	0.90	49
1	0.90	0.85	0.88	41
accuracy			0.89	90
macro avg	0.89	0.89	0.89	90
weighted avg	0.89	0.89	0.89	90

To further assess the robustness of the model, we plotted a ROC curve.

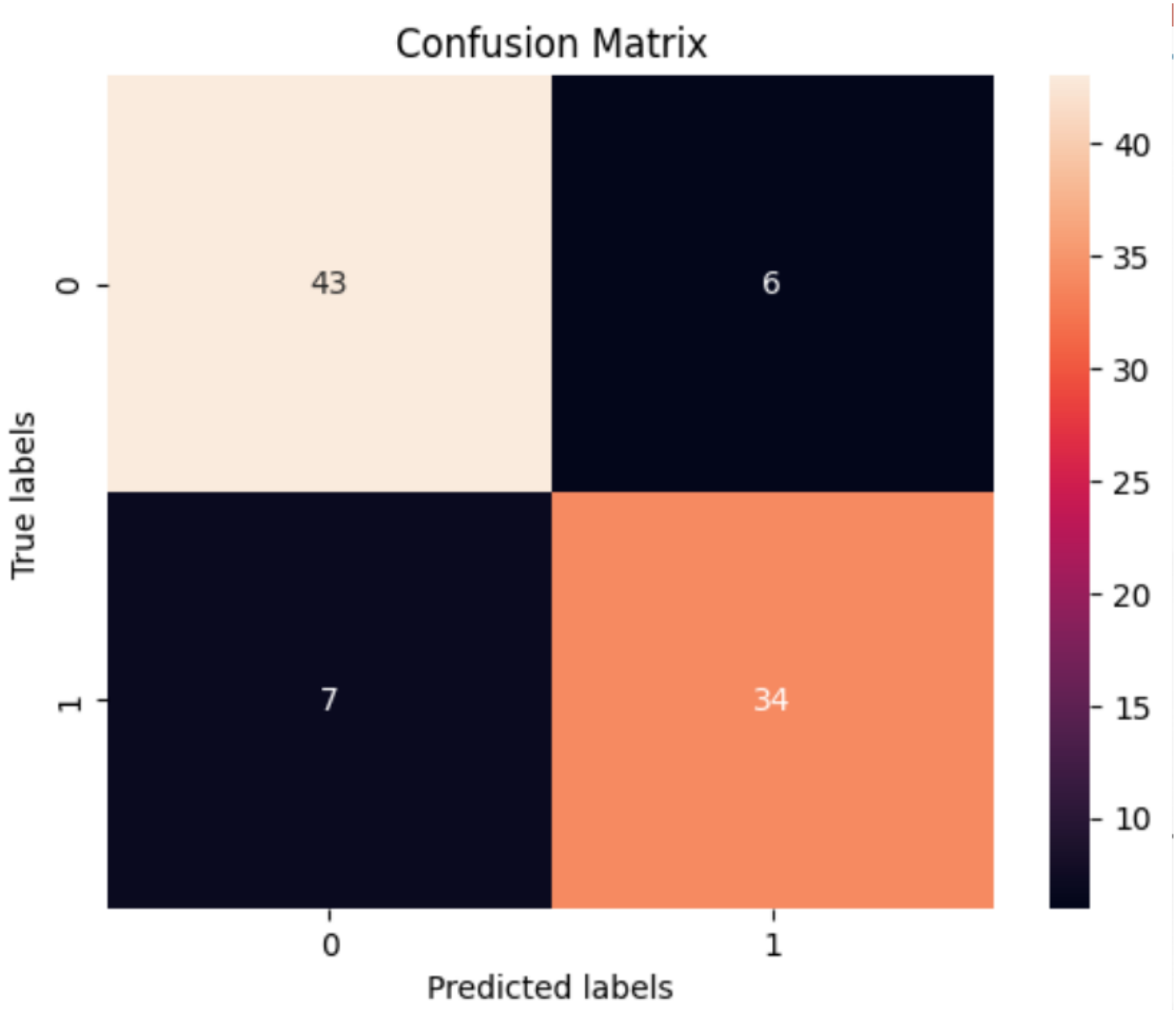


Based on the shape of the curve, the model is able to distinguish true positives while maintaining a low false positive probbility.

Random Forest with RFECV Dataset

After training and fitting our model with the dataset processed by RFECV, we got an overall accuracy of 86%

We represented our results in the a confusion matrix and a classication matrx below.

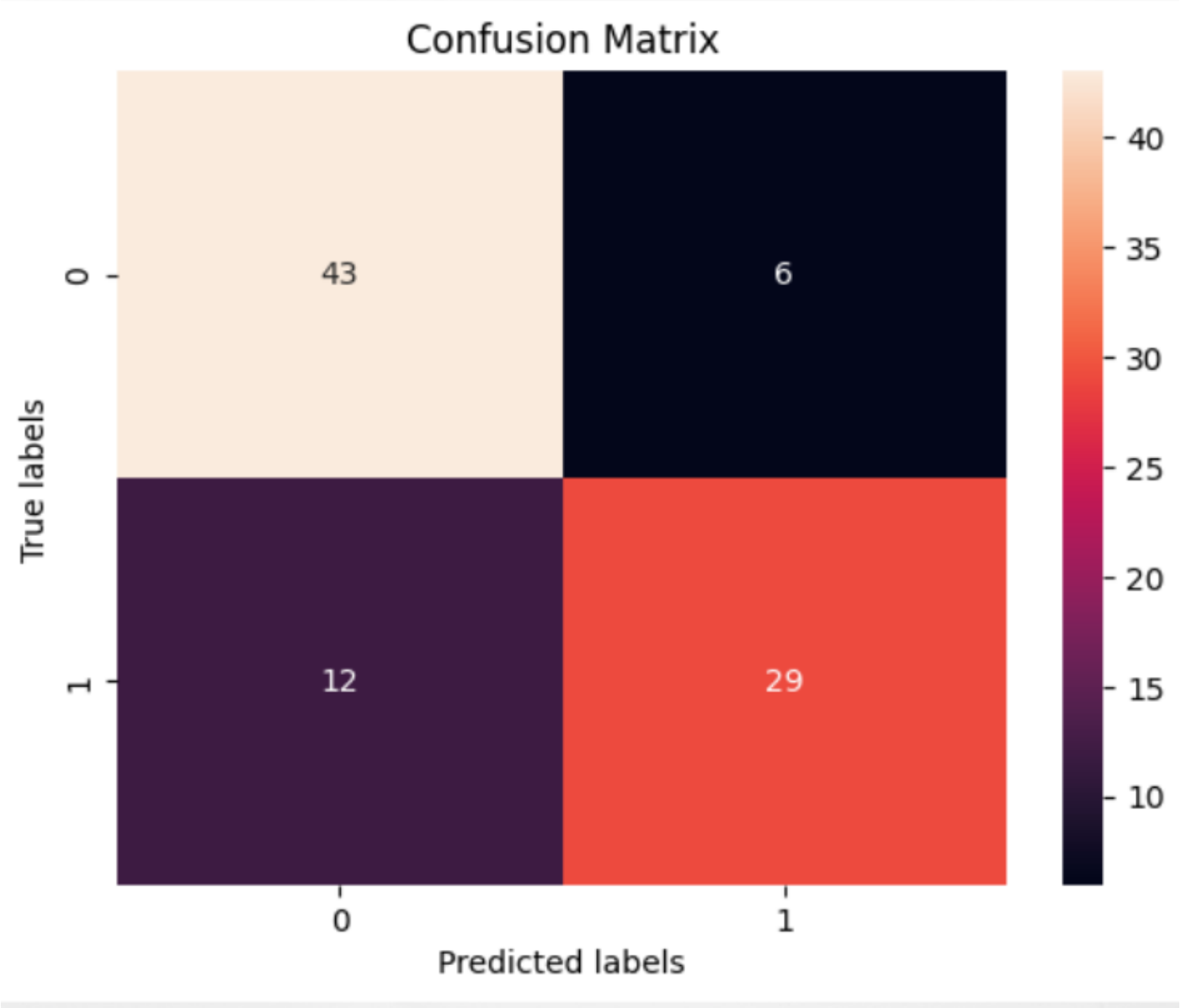


	precision	f1-score	support	
0	0.86	0.88	0.87	49
1	0.85	0.83	0.84	41
accuracy			0.86	90
macro avg	0.85	0.85	0.85	90
weighted avg	0.86	0.86	0.86	90

Random Forest with PCA Dataset

After training and fitting our model with the dataset processed by PCA, we got an overall accuracy of 80%

We represented our results in the a confusion matrix and a classication matrx below.

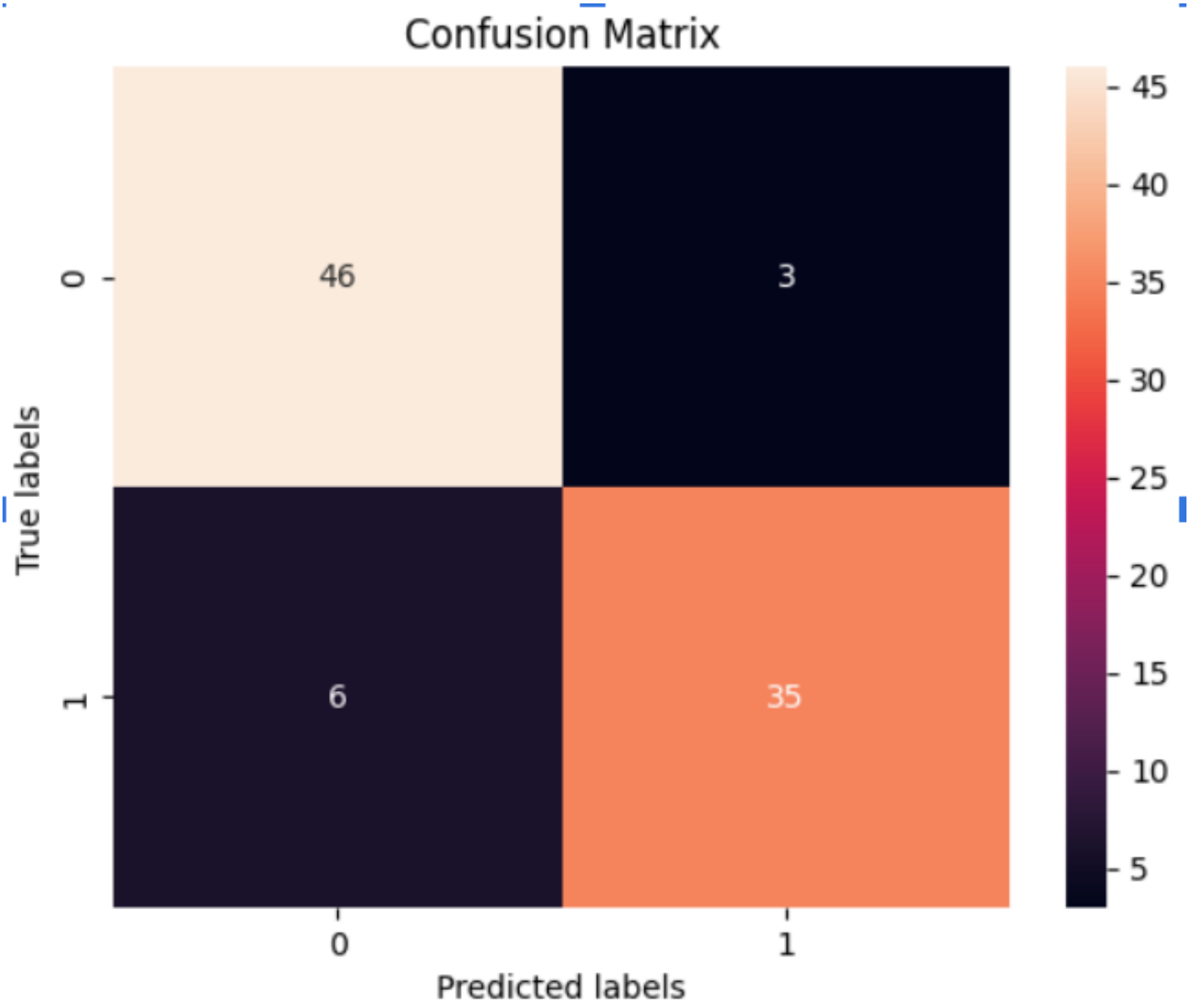


	precision	f1-score	support	
0	0.78	0.88	0.83	49
1	0.83	0.71	0.76	41
accuracy			0.80	90
macro avg	0.81	0.79	0.80	90
weighted avg	0.80	0.80	0.80	90

Naive Bayes

After training and fitting our model, we got an overall accuracy of 90%

We represented our results in the a confusion matrix and a classication matrx below.



	precision	f1-score	support	
0	0.88	0.94	0.91	49
1	0.92	0.85	0.89	41
accuracy			0.90	90
macro avg	0.9	0.9	0.9	0.9
weighted avg	0.9	0.9	0.9	90

Discussion

Preprocessing Analysis

For preprocessing, the recursive feature elimination method works better than PCA, shown through increased accuracy and precision. This is because RFE uses model specific feature selection whereas PCA does not, and RFE retains feature shape and information whereas PCA transforms it. Furthermore, RFECV elimnates feature redundancy while retaining information.

Model Analysis

We assesed all the models with a confustion matrix and a classification matrix. Since we are using binary classification a random guess has a 50% accuracy. As such we wanted to create a model that performed well in the 85-90% percentile.

Another important consideration with regards to the data is that on average there are more people without heart disease than with heart disease. This imbalance is reflected in the data as well. However since it is a smaller dataset the models were quite robust to this imbalance, but we still kept this mind and utilized the precision score(amount of true positives) to determine the effectiveness of each of the models.

Logistic Regression Analysis

Logistic regression is a simple linear model with an impressive capability to avoid overfitting on smaller datasets giving us stable and understandable predcitions whereas complex modles may not. Additionally the usuage of RFECV allowed for a 3% increase in overall accuracy and a precision value of 0.92 for positive values(1).

Random Forest Classfier Analysis

Random Forest Trees typically performs well on models with large dimensionality or data with missing values. However, the model did not perform as well on our dataset and this is likely due to the size of the dataset. Additionally, random forest has its own technique for feature importance and selection and that is why when we performed PCA on the dataset before fitting it to the model the accuracy went down. By performing two feature recutions we diluted the importance of some of the less strongly correlated features, but still useful features.

Naive Bayes Analysis

Next we performed Naive Bayes which is a differennt type of model compared to the first two. One pelimanary drawback of this model is that it assumes that all the features are independent which is why we did not perform any dimensionality reduction on the data before training naive bayes. We were able to confirm the linear independence by creating the heatmap and seeing a low correlation between each of the variable (highest correlation was 0.58 so we could safely assume linearly independece). The model performed really well getting a 90% overall accuracy and a 0.92 precision score for prediciting positives. However, there is decreased f1-score for predicting positive values which is an important indicator for our dataset.

Through all this we have determined that Logistic Regression is the best model for our dataset. It handles the simplicty and linearity our our dataset the best without overfitting and allows for a sensitive feature scaling. Random Forest is a robust model not necessary for our dataset, and finally naive bayes doesn't have any utilize feature selection which is important to our dataset which has multiple features of varying correlation to the target variable.

Dataset

Timeline

Contribution Table

Shreya Puvvula	Gantt chart/timeline creation, results and discussion section Data preprocessing (implementing RFECV)
Shreya Jha	Methods research and section content, introduction + background section Data preprocessing (testing models with continuous and discrete data) Creating quantitative models, conclusion slide
Manasvi Gaddam	Gantt chart/timeline creation, dataset research, heart disease topic proposal Logistic Regression with both data sets, Random Forest implementation Final presentation GitHub updates
Amogha Thodpunuri	Introduction + background section, results and discussion section Naive Bayes implementation Conclusion presentation slide
Rhea Chitanand	Problem definition section, data collection methods, presentation slides Data preprocessing (implementing PCA) Methods presentation slides

References

Bora, N. (2021, December 3). Using Machine Learning to Predict Heart Disease. California State University San Marcos. https://scholarworks.calstate.edu/downloads/nc580s739

Muhammad, Y., Tahir, M., Hayat, M., & Chong, K. T. (2020, November 12). Early and accurate detection and diagnosis of heart disease using intelligent computational model. Scientific reports. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7665174/

Wang, Y., Ng, K., Byrd, R. J., Hu, J., Ebadollahi, S., Daar, Z., deFilippi, C., Steinhubl, S. R., Stewart, W. F. (2015, January 12). Early detection of heart failure with varying prediction windows by structured and unstructured data in Electronic Health Records. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5233460/

Zhang, D., Chen, Y., Chen, Y., Ye, S., Cai, W., Jiang, J., Xu, Y., Zheng, G., & Chen, M. (2021, September 29). Heart disease prediction based on the embedded feature selection method and Deep Neural Network. Journal of Healthcare Engineering. https://www.hindawi.com/journals/jhe/2021/6260022/