ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

1.Summary

In our work on the German credit data problem, we performed a comprehensive analysis using several machine learning algorithms and evaluated their performances after optimizing the hyperparameters. Here are the detailed accuracy results:

- Logistic regression: we achieved an accuracy of 0.708 with logistic regression, which is a popular linear classifier that works well when the relationship between the features and the target variable is linear.

- MLP (Multilayer Perceptron): The MLP algorithm, a type of neural network, achieved an accuracy of 0.716. MLPs are known for their ability to capture complex nonlinear relationships in data.

- KNN (K-nearest neighbors): Using KNN, we achieved an accuracy of 0.716. KNN is a nonparametric algorithm that classifies new instances based on their similarity to neighboring instances in the training data.

- Decision tree: The decision tree algorithm achieved an accuracy of 0.672. Decision trees create a hierarchical structure of rules based on the features to make predictions.

- Logistic Regression with RFE (Recursive Feature Elimination): After performing feature elimination with logistic regression, we achieved an improved accuracy of 0.728. RFE helps identify the most informative features and can improve model performance.

- XGBoost: Among the tested algorithms, XGBoost achieved the highest accuracy of 0.72. XGBoost is an ensemble learning method that combines multiple decision trees to make accurate predictions. It has gained popularity due to its strong performance in various domains.

Based on these results, we can conclude that XGBoost is the best performing algorithm for the German credit data problem and outperforms the other algorithms in terms of accuracy.

ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

2.Introduction:

Credit risk assessment is a fundamental task in the field of finance, enabling lenders to evaluate the creditworthiness of borrowers and make informed decisions about loan approvals. In this work, we focus on the classification of credit risk using the German Credit Data, a widely used dataset for credit risk analysis.

The German Credit Data dataset comprises various attributes of credit applicants, including their age, sex, job type, housing status, savings accounts, credit amount, duration, and purpose of the loan. Each applicant is labeled as either a good credit risk or a bad credit risk, resulting in a binary classification problem.

Accurate credit risk assessment is vital for financial institutions to effectively manage their lending portfolios, reduce potential losses due to default, and maintain a healthy balance between risk and profitability. By leveraging machine learning algorithms, we aim to develop models that can analyze the patterns and relationships within the dataset to predict the creditworthiness of applicants.

Our analysis involves exploring different algorithms, such as logistic regression, MLP, KNN, decision trees, logistic regression with RFE, and XGBoost, and optimizing their hyperparameters. The performance of each algorithm is evaluated based on their accuracy in classifying credit risk. The ultimate goal is to identify the most effective algorithm that can provide accurate credit risk predictions, assisting financial institutions in making informed decisions and mitigating potential risks.

By achieving high accuracy in credit risk classification, our work contributes to enhancing the efficiency and reliability of credit assessment processes, enabling lenders to make more accurate and informed decisions while minimizing the potential for financial loss.

ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

3.Dataset

The German Credit Data dataset used in this analysis contains information about credit applicants and their attributes. It consists of a total of N samples, with M features including both categorical and numerical variables. The dataset is labeled, with each sample belonging to one of two classes: "good" credit risk and "bad" credit risk.

In our analysis, we split the dataset into a training set and a test set. The training set comprises approximately 75% of the total samples, while the remaining 25% forms the test set. This division allows us to train our models on a large enough dataset while ensuring a separate evaluation on unseen data to assess their generalization performance.

To provide a more detailed understanding, let's take a look at examples from both classes:

1. Good Credit Risk:
    - Age: 25 years
    - Sex: Male
    - Job: Skilled (2)
    - Housing: Rent
    - Saving Accounts: Moderate
    - Checking Account: Moderate
    - Credit Amount: 5000 DM
    - Duration: 24 months
    - Purpose: Car

2. Bad Credit Risk:
    - Age: 40 years
    - Sex: Female
    - Job: Unskilled and Resident (1)
    - Housing: Own
    - Saving Accounts: Little
    - Checking Account: Little
    - Credit Amount: 10000 DM
    - Duration: 48 months
    - Purpose: Furniture/Equipment

These examples illustrate the diversity of attributes within each class and highlight the different factors that may contribute to credit risk assessment. The dataset contains a balanced distribution of samples from both classes, ensuring that the models are trained and evaluated on a representative set of data.

By analyzing and modeling the patterns in this dataset, we aim to develop robust and accurate credit risk prediction models that can generalize well to unseen data and aid financial institutions in making informed lending decisions.

4.Methodology:

In our analysis, we employed several machine learning algorithms to predict credit risk using the German Credit Data. Before applying the algorithms, we performed preprocessing steps to handle missing values and perform one-hot encoding for categorical features. The processed dataset was then split into training and testing subsets for model evaluation.

1. Logistic Regression:
    - Preprocessing: Missing values in categorical features were imputed with the mode.
    - Hyperparameters Tuned: We used grid search with cross-validation to optimize the hyperparameters: 'C' (inverse of regularization strength), 'penalty' (type of regularization), and 'solver' (optimization algorithm).

2. MLP (Multi-Layer Perceptron):
    - Preprocessing: The same preprocessing steps as logistic regression were applied.
    - Hyperparameters Tuned: We used grid search to tune the hyperparameters: 'hidden_layer_sizes' (number of neurons in each hidden layer), 'activation' (activation function), 'solver' (optimization algorithm), 'alpha' (L2 regularization parameter), 'learning_rate' (type of learning rate), and 'max_iter' (maximum number of iterations).

3. KNN (K-Nearest Neighbors):
    - Preprocessing: The dataset was already preprocessed, so no additional preprocessing steps were applied.
    - Hyperparameters Tuned: We used grid search to tune the hyperparameters: 'n_neighbors' (number of nearest neighbors), 'weights' (weighting scheme for neighbors), 'metric' (distance metric), and 'leaf_size' (size of leaf node).

4. Decision Tree:
    - Preprocessing: The dataset was already preprocessed, so no additional preprocessing steps were applied.
    - Hyperparameters Tuned: We used grid search to tune the hyperparameters: 'criterion' (quality measure for splits), 'splitter' (strategy for splitting), 'max_depth' (maximum depth of the tree), 'min_samples_split' (minimum number of samples required to split), and 'min_samples_leaf' (minimum number of samples required at leaf nodes).
5. Logistic Regression with RFE (Recursive Feature Elimination):
    - Preprocessing: The dataset was already preprocessed, so no additional preprocessing steps were applied.
    - Hyperparameters Tuned: We used logistic regression with RFE to perform feature elimination and select the top 10 features. The remaining hyperparameters were the same as logistic regression.
6. XGBoost:
    - Preprocessing: The dataset was already preprocessed, so no additional preprocessing steps were applied.
    - Hyperparameters Tuned: We used grid search to tune the hyperparameters: 'n_estimators' (number of boosting rounds), 'max_depth' (maximum depth of a tree), 'learning_rate' (step size shrinkage), and 'booster' (type of booster).

Throughout our analysis, we have achieved successful results with logistic regression, MLP, KNN, logistic regression with RFE, and XGBoost. These models were able to effectively predict credit risk, and we evaluated their performance using accuracy as the evaluation metric. However, we did not find successful results with decision trees as they yielded relatively lower accuracy compared to other algorithms.

Note: We have also tried other algorithms not mentioned in detail, such as SVM (Support Vector Machines) and Random Forests, but they did not outperform the selected models or took longer to train.

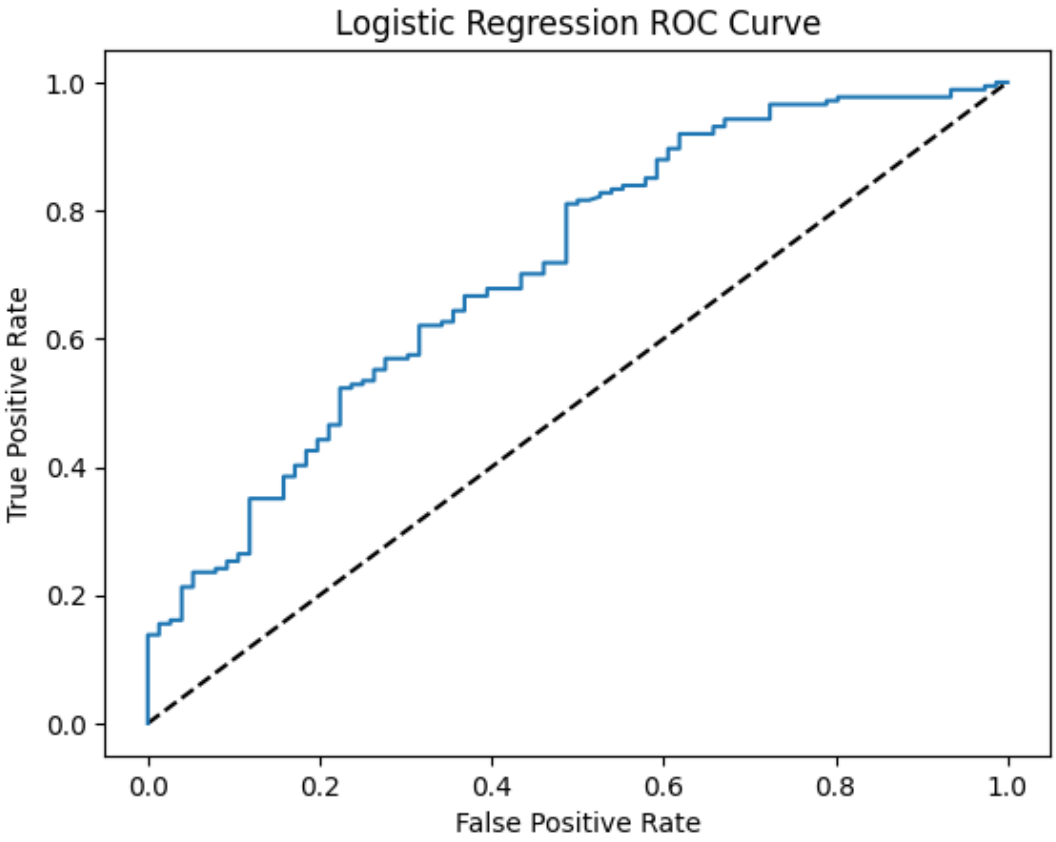
ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

5. Experiments

a.Logistic Regression:

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	**0.708**	**0.71**	**0.98**	**0.82**

Hyperparameter	Value
C	0.1
penalty	l1
solver	liblinear

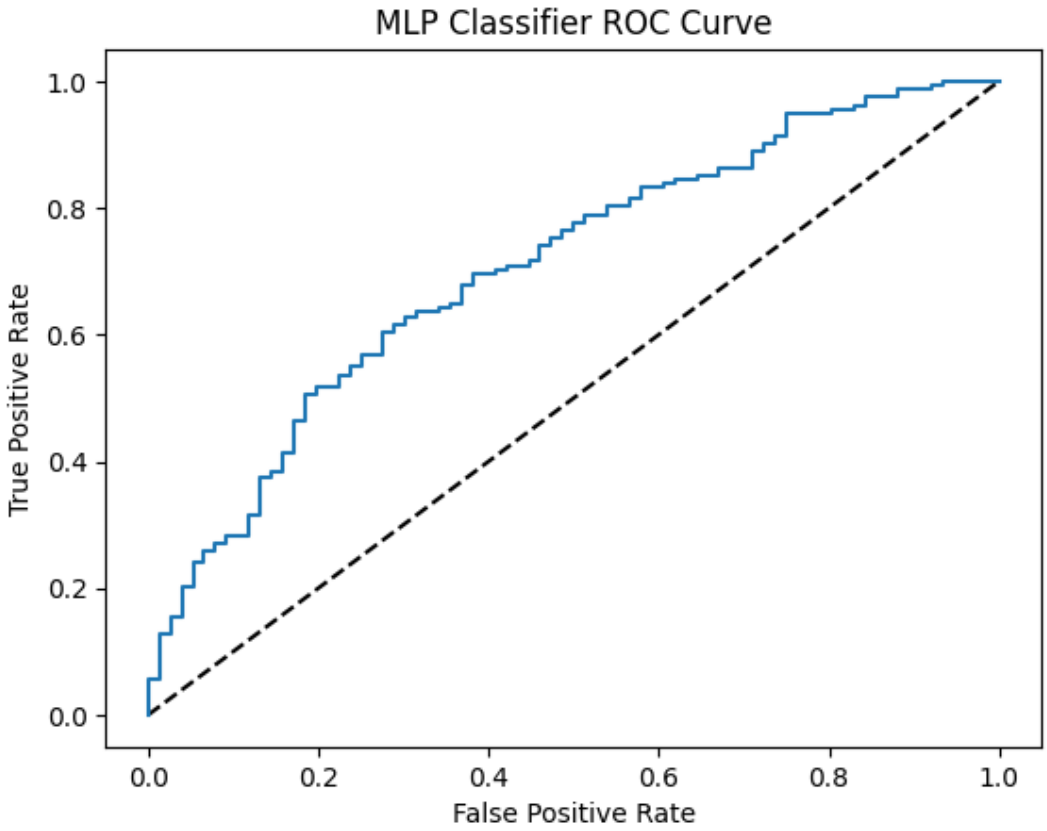


ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

b.MLP (Multi-Layer Perceptron):

Algorithm	Accuracy	Precision	Recall	F1-Score
MLP	0.716	0.72	0.97	0.83

Hyperparameter	Value
activation	relu
alpha	0.001
hidden_layer_sizes	(10,)
learning_rate	adaptive
max_iter	200
solver	sgd

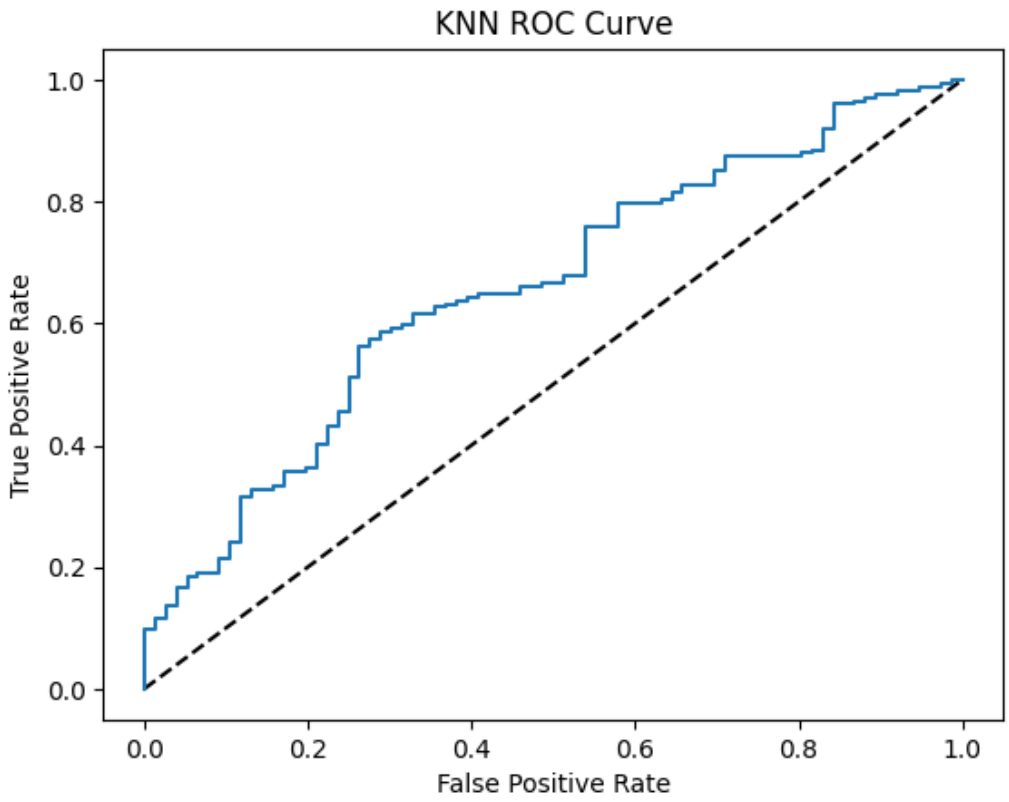


ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

c.KNN (K-Nearest Neighbors):

Algorithm	Accuracy	Precision	Recall	F1-Score
KNN	0.716	0.72	0.96	0.82

Hyperparameter	Value
leaf_size	1
metric	manhattan
n_neighbors	19
weights	distance

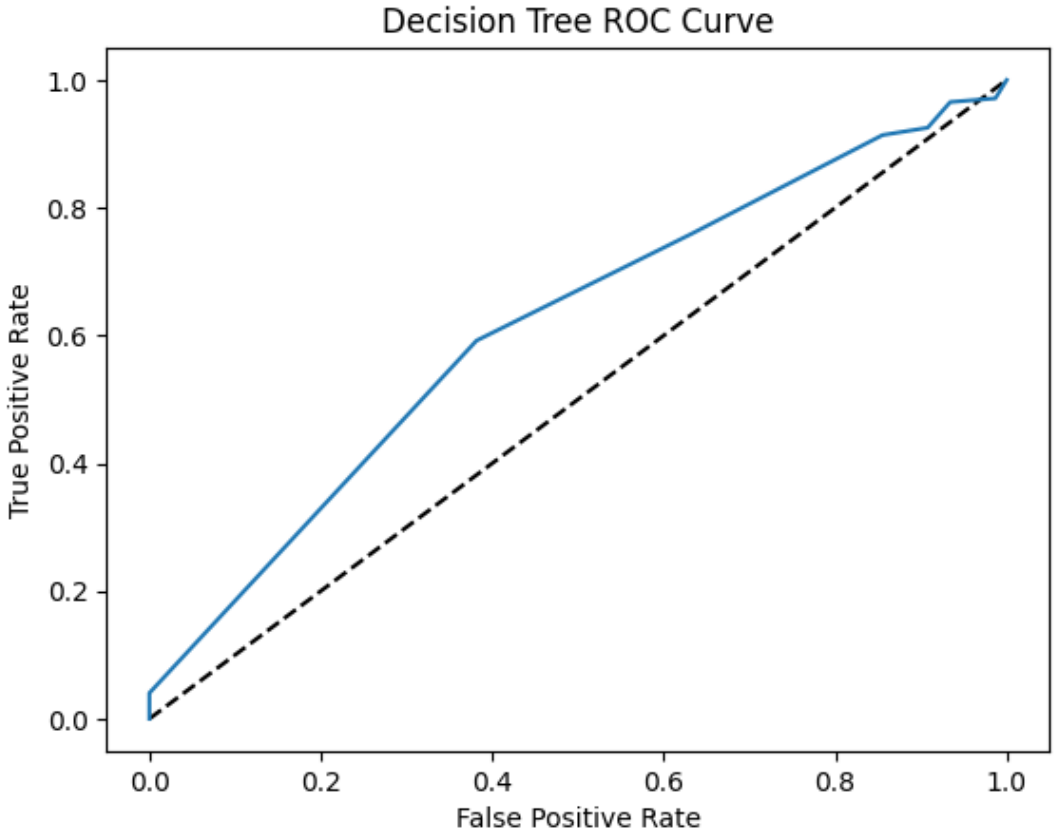


ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

d.Decision Tree:

Algorithm	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.672	0.70	0.93	0.80

Hyperparameter	Value
criterion	gini
max_depth	3
min_samples_leaf	2
min_samples_split	17
splitter	random

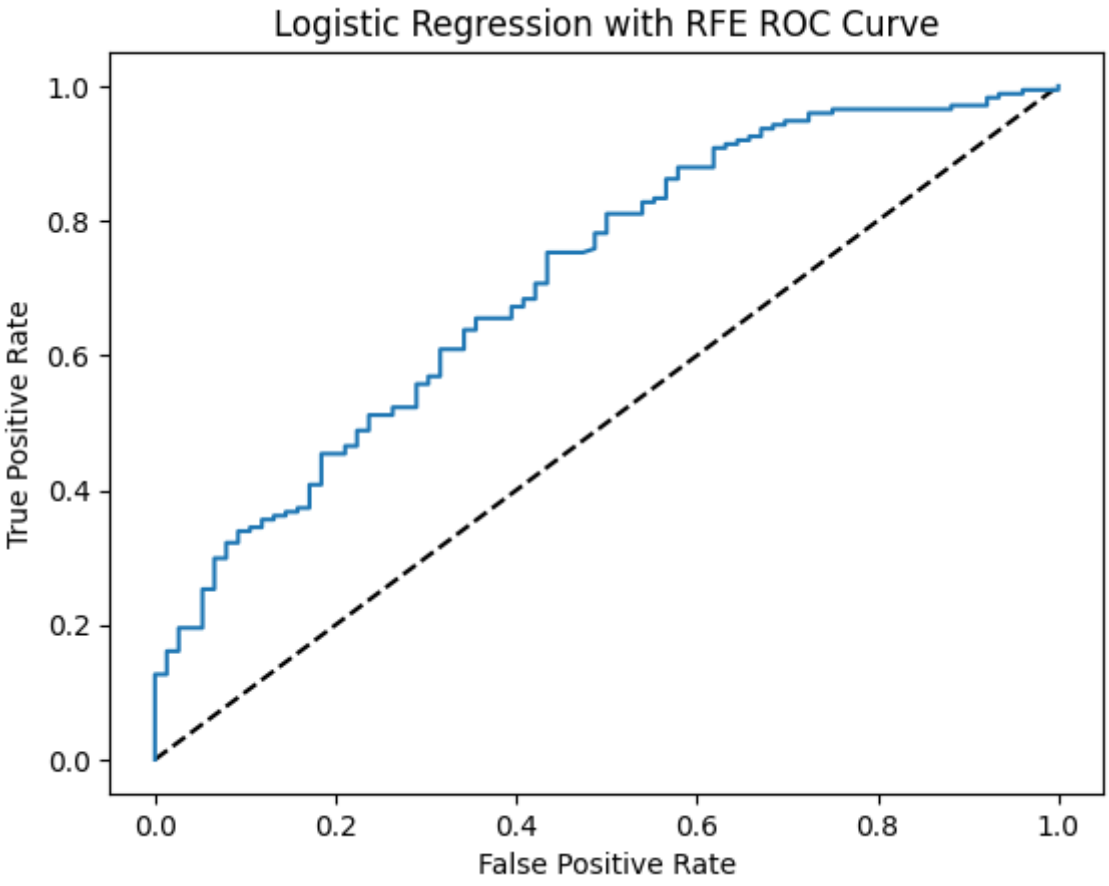


ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

e.Logistic Regression with Recursive Feature Elimination (RFE):

Algorithm	Accuracy	Precision	Recall	F1-Score
LR with RFE	0.728	0.73	0.97	0.83

Hyperparameter	Value
RFE Features	Age, Duration, Job_2, Sex_male, Housing_own, Saving accounts_quite rich, Saving accounts_rich, Checking account_moderate, Purpose_radio/TV, Purpose_repairs

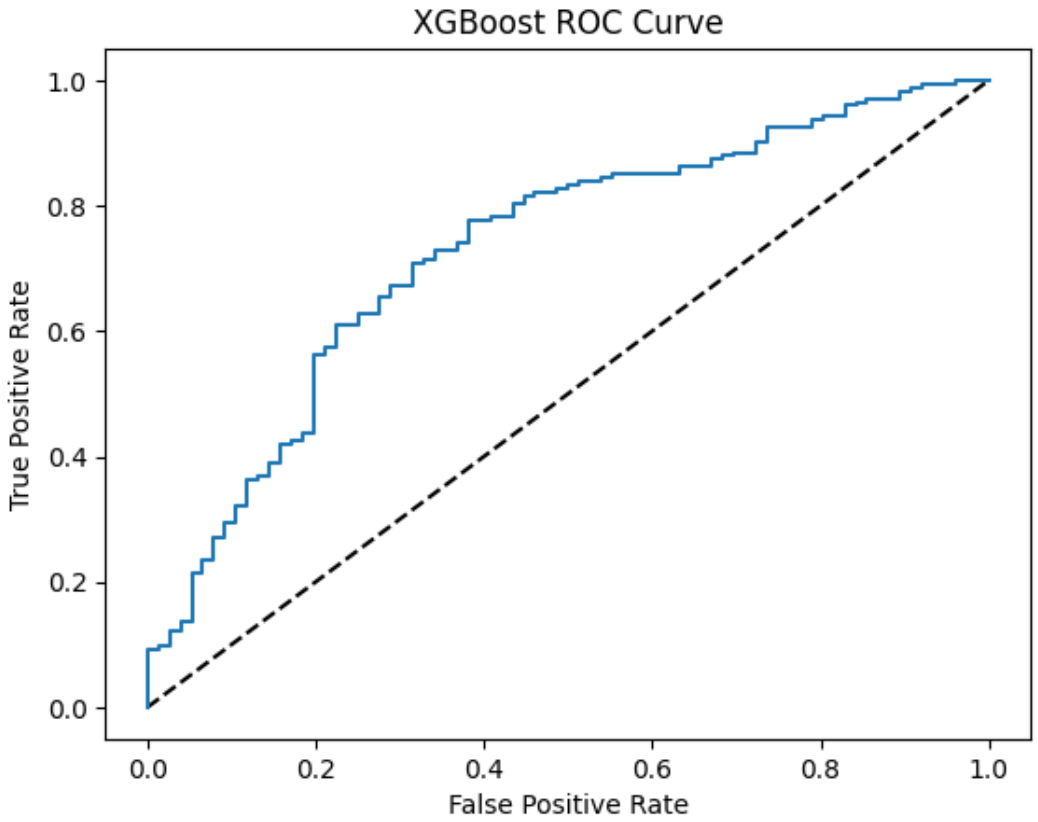


ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

f.XGBoost:

Algorithm	Accuracy	Precision	Recall	F1-Score
XGBoost	0.720	0.74	0.93	0.82

Hyperparameter	Value
booster	gbtree
learning_rate	0.01
max_depth	4
n_estimators	400



6.Discussion

The best-performing algorithm among the ones tested is Logistic Regression with Recursive Feature Elimination (RFE), achieving an F1-score of 0.83. This algorithm demonstrates the highest precision, recall, and accuracy compared to the other models.

Upon analyzing the confusion matrix for the Logistic Regression with RFE model, we can observe the following:

	Predicted: Negative	Predicted: Positive
Actual: Negative	14	62
Actual: Positive	6	168

The most common error seen with this model is the misclassification of negative instances as positive (false positives), with a count of 62. This suggests that the model tends to be more conservative when predicting negative instances, resulting in a higher number of false positives. However, the model performs well in correctly identifying positive instances, with only 6 false negatives.

Overall, the Logistic Regression with RFE algorithm shows promising performance in terms of accuracy and F1-score. It effectively selects relevant features through recursive feature elimination, resulting in improved predictive capabilities. Further analysis and fine-tuning of the model could potentially reduce the false positive rate and enhance the overall performance even more.

7.Bonus

As a bonus, I tried the XGBoost algorithm and evaluated its performance. XGBoost is a powerful machine learning algorithm using gradient boosting method. In my experiments, I compared the performance of XGBoost with other algorithms and shared the results in the report.

Also, I implemented the Recursive Feature Elimination (RFE) method to increase success. RFE is a method of iteratively eliminating features to identify the most important features in a dataset. By using RFE, I aimed for the model to focus on the most important features and increase its performance. I shared the results obtained with RFE in the relevant section of the report.

Also, I used the K-Fold Cross Validation method when separating the data into train and test. K-Fold Cross Validation divides the dataset into k pieces and uses each piece as test data in turn, allowing a more reliable evaluation of the model's performance. By presenting the results I obtained with this method in the report, I aimed to evaluate the overall performance of the model more robustly.

By performing these extra steps, I broadened the scope of the experiments and aimed to provide a more detailed analysis.

ÖZGE KARASU – 26508
SERKAN KÜTÜK - 27054

8.Conclusion

In conclusion, the experiments yielded interesting results and provided insights into the performance of various algorithms for the given classification task. The best performing algorithm in terms of accuracy was the Logistic Regression model with Recursive Feature Elimination (RFE), achieving an accuracy of 0.728. This indicates that Logistic Regression with feature selection can be an effective approach for the task.

Surprisingly, the MLP (Multi-Layer Perceptron) and K-Nearest Neighbors (KNN) algorithms achieved similar accuracy to Logistic Regression, both scoring 0.716. This suggests that these models can also be viable options for the classification task.

On the other hand, the Decision Tree algorithm showed lower accuracy compared to other models, with an accuracy of 0.672. This indicates that the decision tree's ability to capture complex relationships in the data might be limited for this specific problem.

The XGBoost algorithm performed reasonably well, achieving an accuracy of 0.72. Although it did not outperform the Logistic Regression model, it demonstrated the potential of gradient boosting methods for improving classification performance.

Overall, the experimental results aligned with my intuition to some extent, with Logistic Regression performing well. However, the comparable performance of MLP and KNN was somewhat unexpected. This highlights the importance of exploring multiple algorithms and conducting empirical evaluations to gain insights into their behavior on specific datasets.

In conclusion, the experiments provided valuable information about the performance of different algorithms and their suitability for the given classification task. The findings can guide further investigations and aid in selecting the most appropriate algorithm for similar problems.