Can Çiftçioğlu – 26532

Summary/Abstract: We tried several algorithms for the German Credit Data classification problem and observed their performances after hyperparameter optimization. The best-performing algorithm was the Decision Tree Classifier with a maximum depth of 7, minimum samples split of 10, and the Gini criterion, achieving an accuracy of 73.67%.

Introduction: The task at hand is to classify German Credit Data into good or bad credit risks. This classification problem is important in assessing creditworthiness and making informed lending decisions. The dataset contains various features such as savings and checking accounts, sex, housing, and purpose, among others. The goal is to build a model that accurately predicts the credit risk based on these features.

Dataset: The dataset used is the German Credit Data, which consists of a total of N samples. The dataset was split into a training set and a test set, with a test size of 30%. The training set was used for model training and hyperparameter optimization, while the test set was used to evaluate the final performance of the models.

The dataset contains examples from two classes: good and bad credit risks. Here are some examples:

* Example 1: A customer with a moderate saving account, little checking account, male, owning a house, and the purpose of the credit is education.
* Example 2: Another customer with no saving account, moderate checking account, female, renting a house, and the purpose of the credit is a car.

Methodology: We applied several algorithms to the German Credit Data classification problem. Before training the models, we performed some preprocessing steps such as replacing categorical values with numerical equivalents and one-hot encoding certain features. This allowed us to work with a complete numerical dataset.

For each algorithm, we tuned the hyperparameters using grid search and cross-validation to find the best combination of hyperparameters that maximizes the performance of the model. The hyperparameters tuned for each algorithm are as follows:

1. K-Nearest Neighbors (KNN):

* Number of neighbors (n\_neighbors)
* Weight function used in prediction (weights)
* Power parameter for the Minkowski metric (p)

1. Logistic Regression:

* Regularization penalty (penalty)
* Inverse regularization strength (C)
* Solver algorithm (solver)

1. Decision Tree Classifier (CART):

* Maximum depth of the tree (max\_depth)
* Minimum number of samples required to split an internal node (min\_samples\_split)
* Criterion to measure the quality of a split (criterion)

Experiments: We conducted experiments for each algorithm and recorded their performances after hyperparameter optimization.

1. K-Nearest Neighbors (KNN): The best hyperparameters found were: n\_neighbors = 7, weights = 'distance', and p = 1. The accuracy achieved on the test set was 65.33%.
2. Logistic Regression: The best hyperparameters found were: penalty = 'l2', C = 0.1, and solver = 'liblinear'. The accuracy achieved on the test set was 70.33%.
3. Decision Tree Classifier (CART): The best hyperparameters found were: criterion = 'gini', max\_depth = 7, and min\_samples\_split = 10. The accuracy achieved on the test set was 73.67%.

Discussion: Among the algorithms we tried, the Decision Tree Classifier (CART) yielded the highest accuracy of 73.67%. This model performed better than both K-Nearest Neighbors and Logistic Regression. The decision tree model's ability to capture complex relationships between features and the target variable might have contributed to its superior performance in this case.

A brief error analysis reveals that the most common errors made by the model were misclassifying bad credit risks as good risks. This could be an area of improvement for future iterations of the model.

Bonus: As a bonus, we also applied feature selection using the Random Forest Classifier. The selected features were used to train a Logistic Regression model. The accuracy achieved with the selected features was 71%.

Conclusion: In conclusion, we explored various algorithms for the German Credit Data classification problem. The Decision Tree Classifier outperformed the other models, achieving an accuracy of 73.67% on the test set. The findings aligned with our intuition that decision tree-based models would perform well in this scenario due to their ability to capture complex relationships. Further improvements can be made by addressing the misclassification of bad credit risks as good risks. Overall, the experiments yielded insightful results and provided a better understanding of the dataset and its classification problem.