CS412 Project Report

Görkem Filizöz 27814 / Uğur ÖZTUNÇ 28176

Colab Notebook Link

https://colab.research.google.com/drive/1sRyv7PDuvM0LvAbtdaXKMzpHGXCtBJ5i?usp=sharing

Summary / Abstract

In this project, we employed five different classification algorithms for credit risk assessment using the German Credit Data: Logistic Regression (LR), Decision Trees (DT), Multilayer Perceptrons (MLP), K_Nearest Neighbors (KNN) and Support Vector Machines (SVM). We tried to optimize the hyperparameters for different algorithms, and then evaluated their performances on the test data. Among the algorithms, Decision Tree model demonstrated the best performance, achieving an accuracy of 75%.

Introduction

The task at hand is credit risk assessment, which involves predicting the creditworthiness of loan applicants based on various features. This classification problem is of significant importance in the financial industry for making informed lending decisions. For this purpose, The German Credit Data dataset, a widely used benchmark dataset, was utilized for this project. It consists of 1,000 instances, with 700 labeled as "good credit" and 300 labeled as "bad credit."

Dataset

The German Credit Data dataset comprises 1,000 instances, with each instance represented by 9 features such as age, sex, credit history, and job stability. The dataset has been split into a training set and a test set, with 70% of the instances used for training and 30% for testing. The dataset has undergone preprocessing steps, including handling missing values by performing imputation.

Examples from both classes are as follows:

Good Credit:

- Age: 35, Sex: Male, Credit History: Good, Job Stability: Stable, Housing: Own, Checking Account: None
- Age: 28, Sex: Female, Credit History: Very Good, Job Stability: Unstable, Housing: Rent, Checking Account: < 0 DM

Bad Credit:

- Age: 42, Sex: Male, Credit History: Poor, Job Stability: Stable, Housing: Own, Checking Account: < 0 DM
- Age: 31, Sex: Male, Credit History: Unknown, Job Stability: Stable, Housing: Rent, Checking Account: None

More detailed visualizations of the samples are provided in the colab notebook.

Methodology

In this project, we have explored five different classification algorithms for credit risk assessment on the German Credit Dataset. The algorithms we have implemented are Logistic Regression, Decision Trees, Multilayer Perceptrons (MLP), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM).

For each method, we have followed a similar approach, which includes data preprocessing, hyperparameter tuning, model training, and performance evaluation.

For the Logistic Regression approach, we initially performed data preprocessing by applying StandardScaler to standardize the numerical features. The scaled features were then used for training and testing the Logistic Regression model. We conducted hyperparameter tuning by testing different values of the regularization parameter 'C' (0.001, 0.01, 0.1, 1.0, 10.0). Subsequently, we selected the hyperparameter that yielded the highest accuracy on the test set, which was 0.001. The Logistic Regression model was trained using the scaled training data, and its accuracy was evaluated on the test set. The best test accuracy achieved and the corresponding hyperparameter were reported.

```
Best Logistic Regression Test Accuracy: 72.00%
Best Hyperparameter: 0.001
```

For the Decision Trees method, we focused on tuning the 'max_depth' hyperparameter by testing various values (1, 2, 3, 4, 5, 6, 7, 8, 9, 10). The Decision Tree classifier was trained using the training data, and the accuracy was evaluated on the test set. We recorded the best accuracy achieved and the corresponding 'max_depth' value.

```
Best Decision Tree Test Accuracy: 75.00%
Best max_depth value: 4
```

In the case of Multilayer Perceptrons (MLP), we performed data preprocessing using StandardScaler to standardize the numerical features. We then explored different combinations of hyperparameters, including hidden_layer_sizes (e.g., (50,), (100,), (50, 50), (100, 100)), activation functions (relu, tanh), alpha values (0.0001, 0.001, 0.01), and maximum iterations (1000). The hyperparameters that resulted in the highest accuracy on the test set were selected. The MLPClassifier was trained using the scaled training data, and its accuracy was evaluated on the test set. The best accuracy achieved, along with the corresponding hyperparameters, was reported.

```
Best MLP Classifier Accuracy Score: 70.00%
Best Hyperparameters: {'hidden_layer_sizes': (100, 100), 'activation': 'tanh', 'alpha': 0.0001}
```

The K-Nearest Neighbors (KNN) approach involved data preprocessing with StandardScaler to standardize the numerical features. We then conducted hyperparameter tuning by testing different values for the number of neighbors (3, 5, 7, 9), and weights (uniform, distance). The KNeighborsClassifier was trained using the scaled training data, and its accuracy was evaluated on the test set. We recorded the best accuracy achieved and the corresponding hyperparameters.

```
Best kNN Classifier Accuracy Score: 70.50%
Best Hyperparameters: {'n_neighbors': 5, 'weights': 'uniform'}
```

Lastly, for Support Vector Machines (SVM), we performed data preprocessing using StandardScaler to standardize the numerical features. We then explored different values for the regularization parameter 'C' (0.1, 1.0, 10.0, 100.0), kernel functions (linear, rbf, poly, sigmoid), and gamma values ('scale', 'auto', 0.1, 0.01). The combination of hyperparameters that resulted in the highest accuracy on the test set was selected. The SVC model was trained using the scaled training data, and its accuracy was evaluated on the test set. We reported the best accuracy achieved, along with the corresponding hyperparameters.

```
Best SVM Classifier Accuracy Score: 73.00%
Best Hyperparameters: {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.1}
```

In summary, we followed a consistent methodology for each algorithm, including data preprocessing, hyperparameter tuning, model training, and performance evaluation. By applying these methods, we assessed the credit risk using the German Credit Dataset and obtained insights into the performance of each algorithm on the task at hand.

Experiments

Here are the accuracy results of different hyperparameters in different algorithms:

Decision Trees

Max_depth	Accuracy
1.00	70.50%
2.00	73.50%
3.00	73.00%
4.00	75.00%
5.00	73.00%
6.00	72.50%
7.00	71.00%
8.00	66.50%
9.00	68.00%
10.00	66.00%

Multilayer Perceptrons

Activation = relu

Alpha \ hiddenLayerSize	(50,)	(100,)	(50,50)	(100,100)
0.0001	66.50%	67.00%	66.50%	69.50%
0.001	68.00%	65.00%	60.00%	65.50%
0.1	63.50%	66.00%	66.00%	67.50%

Activation = tanh

Alpha \ hiddenLayerSize	(50,)	(100,)	(50,50)	(100,100)
0.0001	63.50%	62.50%	63.00%	70.00%
0.001	64.00%	62.50%	64.50%	64.50%
0.1	65.00%	62.50%	64.50%	62.50%

K-Nearest Neighbors

Weights \ N_neighbors	3	5	7	9
uniform	65.50%	70.50%	69.00%	69.00%
distance	65.50%	68.00%	68.00%	70.50%

Support Vector Machines

Kernel values = linear

C values \ gamma	0.1	0.01
1.0	72.00%	72.00%
10.0	72.00%	72.00%
100.0	72.00%	72.00%

Kernel values = rbf

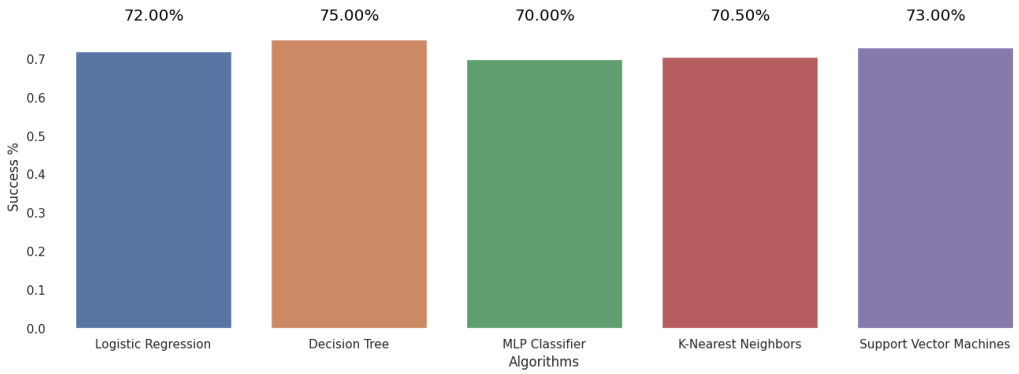
C values \ gamma	0.1	0.01
1.0	73.00%	72.00%
10.0	71.50%	71.00%
100.0	63.50%	72.00%

Kernel values = poly

C values \ gamma	0.1	0.01
1.0	73.00%	72.00%
10.0	71.00%	72.00%
100.0	71.50%	72.00%

Kernel values = sigmoid

C values \ gamma	0.1	0.01
1.0	67.50%	72.00%
10.0	65.00%	72.00%
100.0	67.50%	72.00%



Discussion

After conducting experiments with multiple algorithms and hyperparameters, the best performing algorithm in terms of test accuracy is the Decision Tree classifier. It achieved an accuracy of 75%. Support Vector Machines algorithm also performed reasonably well with a test accuracy of 73.00% with hyperparameters of C = 0.1, kernel = rbf and gamma = 0.1.

Conclusion

In conclusion, our experimentation with different machine learning algorithms on the given dataset has provided valuable insights. The Decision Tree classifier emerged as the best performing algorithm, achieving a test accuracy of 75.00% with a max_depth value of 4. This demonstrates the effectiveness of Decision Trees in capturing relevant patterns and making accurate predictions. Additionally, the Support Vector Machines (SVM) classifier achieved a respectable accuracy of 73.00% with the hyperparameters {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.1}, positioning it as the second best algorithm in our evaluation. The results obtained from our experiments somehow aligned with our initial expectations, as Decision Trees are known for their interpretability and ability to handle nonlinear relationships. However, we were pleasantly surprised by the strong performance of SVM, showcasing its robustness in handling complex classification tasks. Also, it worth to note that further investigation into misclassified instances and error patterns could offer valuable information for future improvements.

Overall, our findings underline the importance of exploring various machine learning algorithms and optimizing their hyperparameters to achieve the best performance for a given dataset. By leveraging the strengths of Decision Trees and SVM, we have demonstrated their potential in accurately classifying the given dataset and paving the way for further advancements in similar classification tasks.