Machine Learning Project: Classification with SVM

Naim BADAOUI, Ilyas MESKINE, Ozan GUNES

March 16, 2025

Contents

1 Introduction

In this project, the goal is to apply a **Support Vector Machine (SVM)** algorithm to solve a classification problem. The original dataset had a target variable called **"approved"** which was initially balanced, with a total of 690 data points. However, we deliberately introduced an imbalance in this variable to simulate a realistic scenario where some classes are under-represented, as often happens in real-world classification problems. After this imbalance, the dataset contains 490 data points.

2 Data Preprocessing

2.1 Loading and Initial Exploration

The dataset was loaded from a CSV file called *clean_dataset.csv*. An exploratory analysis showed that the target variable **"approved"** was initially balanced, with 690 samples.

2.2 Data Imbalance

For the experiment, we deliberately introduced an imbalance in the **"approved"** class by reducing the number of samples from one of the classes. Before the imbalance, the dataset had 690 data points, but after introducing the imbalance, it has 490 data points. This manipulation allows us to test the effectiveness of the SVM algorithm under more realistic conditions where classes are often imbalanced.

2.3 Visualization of the Imbalanced Classes

The following figure shows the distribution of the different variables after introducing an imbalance in the **"approved"** class. We can observe that some variables have more significant imbalances, which could affect the model's performance.

2.4 Data Cleaning

The following steps were carried out:

- Removal of outliers using statistical methods.
- Splitting the dataset into explanatory variables (`X`) and target variable (`y`).

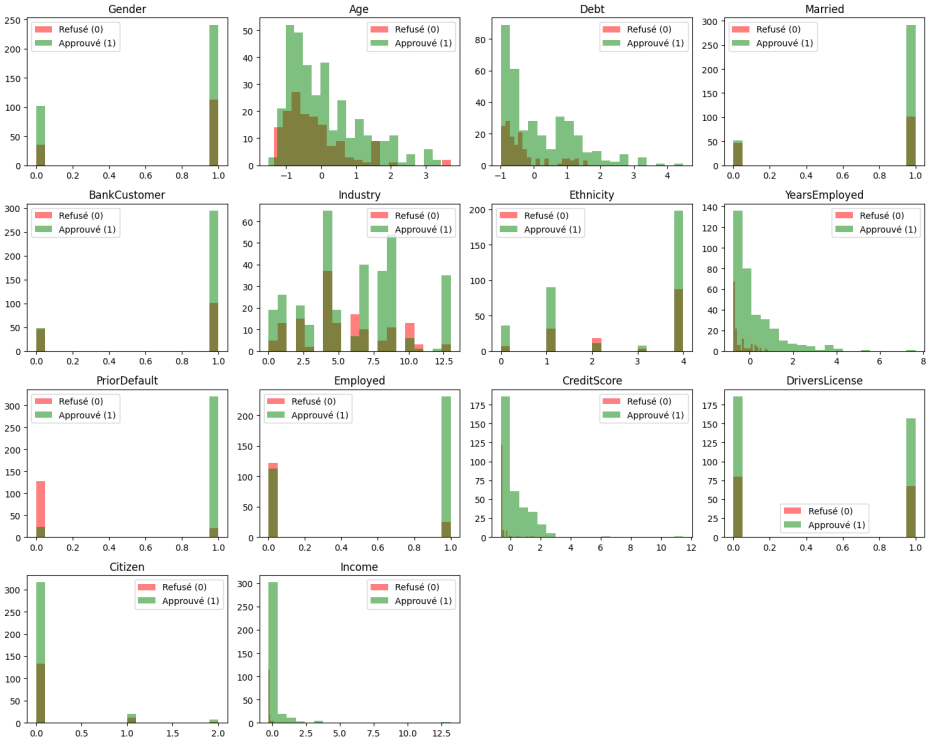

Figure 1: Distribution of classes after imbalancing the 'approved' variable.

2.5 Splitting the Dataset

The dataset was divided into two sets:

- **Training Set** : 70%
- **Test Set** : 30%

The function `train_test_split` was used with the `stratify` argument to ensure a proportional distribution of classes in each set.

3 Modeling with SVM

3.1 Hyperparameter Tuning

To optimize the SVM model, a grid search (`GridSearchCV`) was performed, testing several combinations of hyperparameters:

- **C** : [0.1, 1, 10]

- **Kernel** : ['linear', 'rbf', 'poly', 'sigmoid']
- **Gamma** : ['scale', 'auto', 0.01, 0.1, 1]
- **Degree** : [2, 3, 4] (for the polynomial kernel)
- **Decision Function Shape** : ['ovr', 'ovo']

The model was optimized with the parameter `class_weight='balanced'` to account for the class imbalance.

3.2 Best Parameters Found

After training, the best hyperparameters identified were:

- **C** : 10
- **Kernel** : `rbf`
- **Gamma** : `scale`
- **Degree** : 2
- **Decision Function Shape** : `ovr`

3.3 Model Training

The SVM model was trained using the best hyperparameters obtained.

4 Model Evaluation

4.1 Classification Results

The model's performance was evaluated on the test set using the following metrics:

- **Accuracy** : 94.29%
- **Classification Report** : See the table below.

	precision	recall	f1-score	support
0	0.93	0.89	0.91	44
1	0.95	0.97	0.96	96

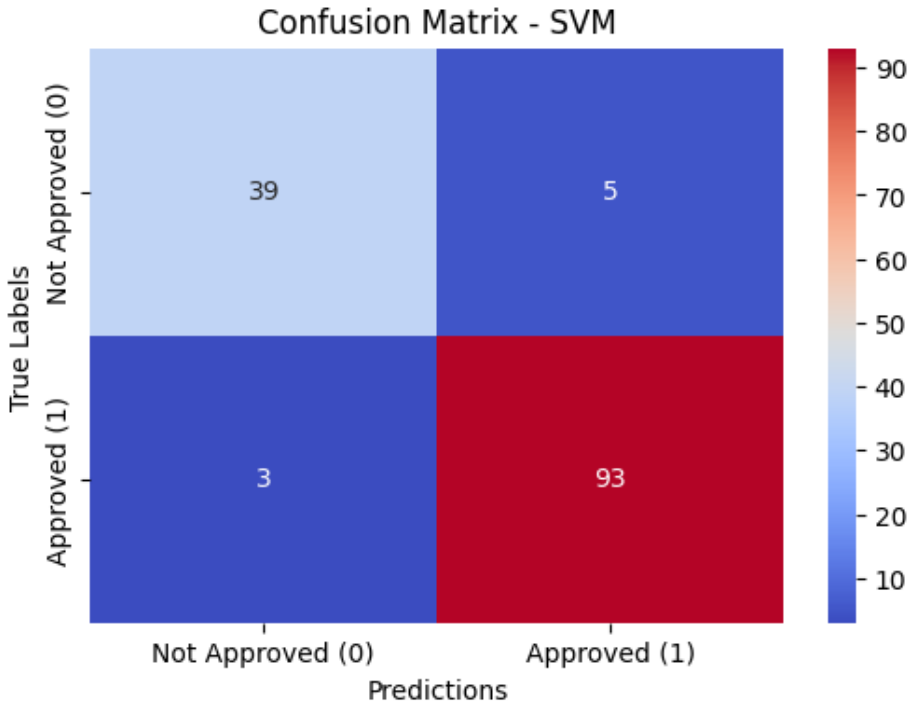

Figure 2: The following confusion matrix summarizes the SVM model's performance.

4.2 Confusion Matrix Interpretation

- **39** : True Negatives (class 0 correctly predicted - Rejected).
- **93** : True Positives (class 1 correctly predicted - Approved).
- **5** : False Positives (predicted as Approved but actually Rejected).
- **3** : False Negatives (predicted as Rejected but actually Approved).

This matrix shows that the model is performing well, with few classification errors. It correctly identifies most examples in both classes, even in the context of imbalanced data.

5 Conclusion

The results obtained, with an overall accuracy of 94.29%, show that the model is effective and able to distinguish the two classes well, even with the imbalance. However, there are opportunities for improvement, such as using

oversampling techniques like SMOTE or experimenting with other classification algorithms.