Support Vector Machines for Credit Default Prediction

Matthew Collins
Department of Computer Science
Clemson University
mbc3@clemson.edu

Abstract

This project presents an analysis of support vector machines (SVMs) applied to the UCI Default of Credit Card Clients dataset. The study investigates how different kernal functions and optimization techniques influence classification performance. Support Vector Machine models using linear, radial basis function (RBF), polynomial, and sigmoid kernels are evaluted. A grid search is conducted for hyperparameter optimization, and PCA is utilized to visualize decision boundaries. Performance metrics, including accuracy, precision, recall, F1 score, and runtime are used to evaluate the results, highlighting the effectiveness of SVM's in binary classification tasks involving tabular data.

1 Introduction

Support vector machines (SVMs) are a widely used supervised machine learning approach, recognized for their strong performance in diverse classification tasks. This project explores the application of Support Vector Machines to the UCI Default of Credit Card Clients dataset, focusing on the role of kernal functions and optimization techniques in shaping model performance. Additionally, hyperparameter tuning is investigated as a means of improving both effiency and accuracy. To improve interpretability, Principal Component Analysis (PCA) is used to visualize the decision boundries. The main goal is to evaluate the effectiveness of SVM's for predicting credit card default and to analyze the computational trade-offs associated with various kernel functions and optimization strategies. Results are evaluated using accuracy, recall, precision, and the F1-score.

2 Dataset

The UCI Default of Credit Card Clients dataset is a multivariate dataset containing data from 30,000 credit card holders in Taiwan. It is primarily designed for binary classification tasks, specifically predicting whether a client would default on their next months payment. The dataset includes 23 features, which consist of demographic variables such as age, sex, education, and marital status, along with financial metrics like credit limits, historical bill statements, and repayment amounts. The target variable is binary, indicating whether a client defaulted (1) or did not default (0) on their credit card payment.

The dataset includes a mix of categorical and numerical variables. Important variables include repayment statuses (`PAY_0` to `PAY_6`), historical bill amounts (`BILL_AMT1` through `BILL_AMT6`), and historical repayment amounts (`PAY_AMT1` to `PAY_AMT6`). Additionally, demographic variables such as `SEX`, `EDUCATION`, and `MARRIAGE` provide insights into customer profiles. Table 1 summarizes the key features in the dataset.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Table 1: Summary of Key Features in the Dataset

Variable Name	Type	Description	Units
SEX	Categorical	Gender	1 = Male, 2 = Female
EDUCATION	Categorical	Education level	1 = Graduate, etc.
MARRIAGE	Categorical	Marital status	1 = Married, etc.
BILL_AMT1-6	Numerical	Monthly bill amounts (6 months)	NT Dollars
PAY_AMT1-6	Numerical	Historical repayment amounts (6 months)	NT Dollars
PAY_0-6	Categorical	Repayment status (-2: advance payment)	Discrete values
DEFAULT (Target)	Binary	Default status for the next month	0 = No, 1 = Yes

To prepare the dataset for modeling, several preprocessing steps were applied. Variables like `SEX`, `EDUCATION`, `MARRIAGE`, and repayment statuses (`PAY_0` to `PAY_6`) were categorical. Since SVMs require numerical input, these categorical variables were transformed into one-hot encoded binary variables. This ensured that each category was treated as an independent feature, avoiding any implicit ordinal assumptions. Additionally, the dataset exhibited significant class imbalance, as most clients did not default on their payments. To address this, the majority class (non-default clients) was downsampled to create a balanced dataset. This ensured the model learned patterns equally well from defaulting and non-defaulting clients.

Another essential step was standardizing numerical features like bill amounts and repayment amounts. These features had varying scales, with some measured in monetary units and others being categorical. Without standardization, features with larger magnitudes could disproportionately impact the model. Thus, all continuous features were scaled to have a mean of zero and a standard deviation of one, ensuring fair contribution from all variables.

The preprocessing steps of one-hot encoding, downsampling, and standardization were essential to making the dataset compatible with SVMs and producing reliable predictions. These transformations addressed challenges posed by mixed feature types, class imbalance, and variable scaling, ultimately improving the efficiency and accuracy of the machine learning models used in this study.

3 Methodology

The Support Vector Machine (SVM) algorithm aims to find an optimal hyperplane that maximizes the margin between two classes while minimizing classification errors. For a training dataset with n points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, the primal optimization problem is formulated as:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

subject to the constraints:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$$

where $\mathbf{w}$ is the normal vector to the hyperplane, b is the bias term, ξ_i are slack variables allowing for soft margin violations, and C is the regularization parameter controlling the trade-off between margin maximization and error minimization.

The dual formulation of this optimization problem, derived using Lagrange multipliers, is:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

subject to:

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

To handle nonlinear classification problems, kernel functions $K(\mathbf{x}_i, \mathbf{x}_j)$ are introduced to implicitly map the input features into a higher-dimensional space. The kernel trick transforms the dual optimization problem to:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

In this project, four kernel functions were evaluated:

1. Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$
2. Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^\top \mathbf{x}_j + r)^d$
3. RBF (Gaussian): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
4. Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^\top \mathbf{x}_j + r)$

The decision function for classifying new points becomes:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right)$$

Grid search cross-validation was employed to optimize the hyperparameters C and γ for each kernel. The search space included:

- $C \in \{0.1, 1, 10, 100\}$
- $\gamma \in \{0.001, 0.01, 0.1, 1\}$

To facilitate visualization and interpretation, Principal Component Analysis (PCA) was applied to reduce the feature space to two dimensions. Given the centered data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features, PCA first computes the covariance matrix:

$$\boldsymbol{\Sigma} = \frac{1}{n} \mathbf{X}^\top \mathbf{X}$$

PCA then performs eigendecomposition of the covariance matrix:

$$\boldsymbol{\Sigma} = \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^\top$$

where $\mathbf{V} \in \mathbb{R}^{d \times d}$ contains the eigenvectors as columns (sorted by decreasing eigenvalues) and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$ contains the corresponding eigenvalues. The transformed features are obtained by projecting the data onto the first two principal components:

$$\mathbf{X}_{\text{transformed}} = \mathbf{X} \mathbf{V}_{1:2}$$

where $\mathbf{V}_{1:2} \in \mathbb{R}^{d \times 2}$ contains the first two eigenvectors. This projection preserves the maximum possible variance in the data while reducing dimensionality to facilitate visualization of the decision boundaries.

4 Results

The comparative analysis of kernel functions yielded results, summarized in Table 2. The RBF kernel outperformed other kernels, achieving the highest accuracy (68.75%) and F1-score (0.65). The scree plot in Figure 3 highlights the explained variance by the principal components, providing insights into the dimensionality reduction process.

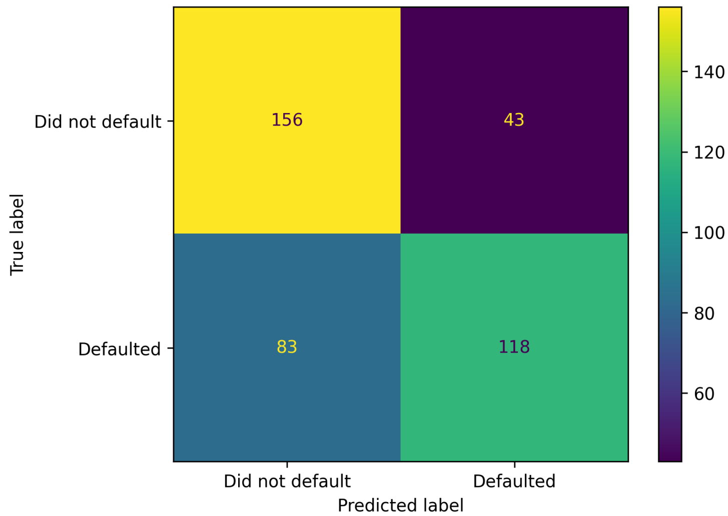

Figure 1: Confusion Matrix Before Grid Search.

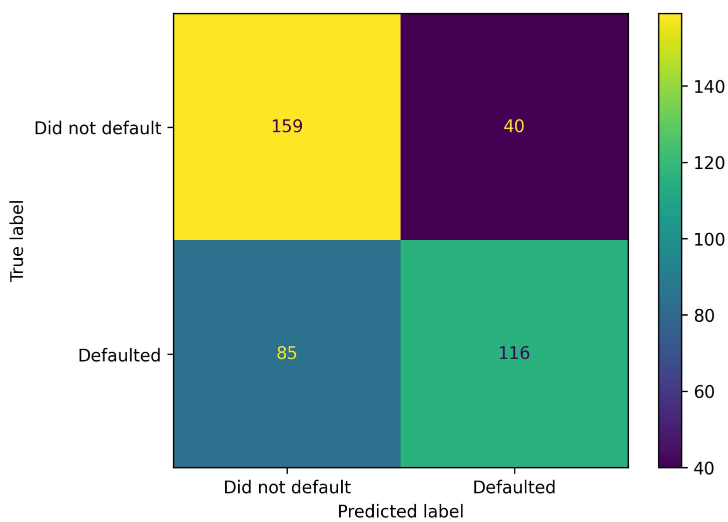

Figure 2: Confusion Matrix After Grid Search.

Table 2: Kernel Comparison Results

Kernel	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Runtime (s)**	**Best Hyperparameters (C, γ)**
Linear	0.6525	0.7095	0.5224	0.6017	0.2457	(1, Scale)
Polynomial	0.6700	0.7226	0.5572	0.6292	0.0629	(10, 0.01)
RBF	0.6875	0.7436	0.5771	0.6499	0.0634	(1, 0.01)
Sigmoid	0.6450	0.7007	0.5124	0.5920	0.0891	(100, 0.001)

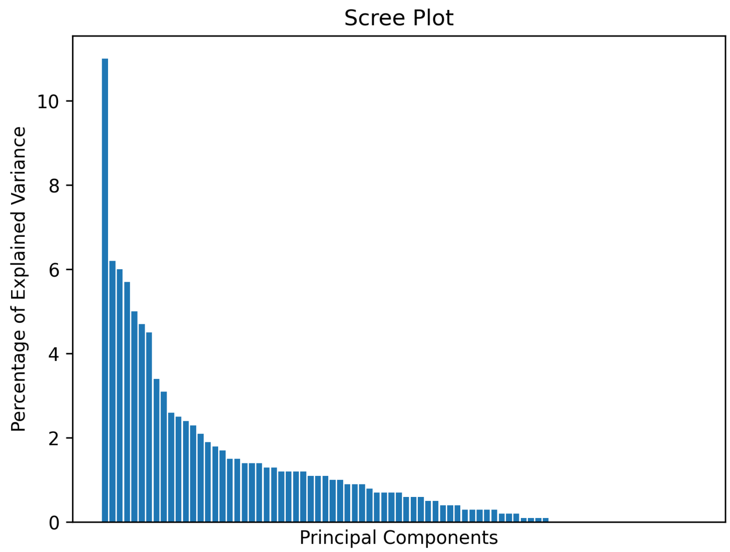

Figure 3: Scree Plot Showing Percentage of Explained Variance.

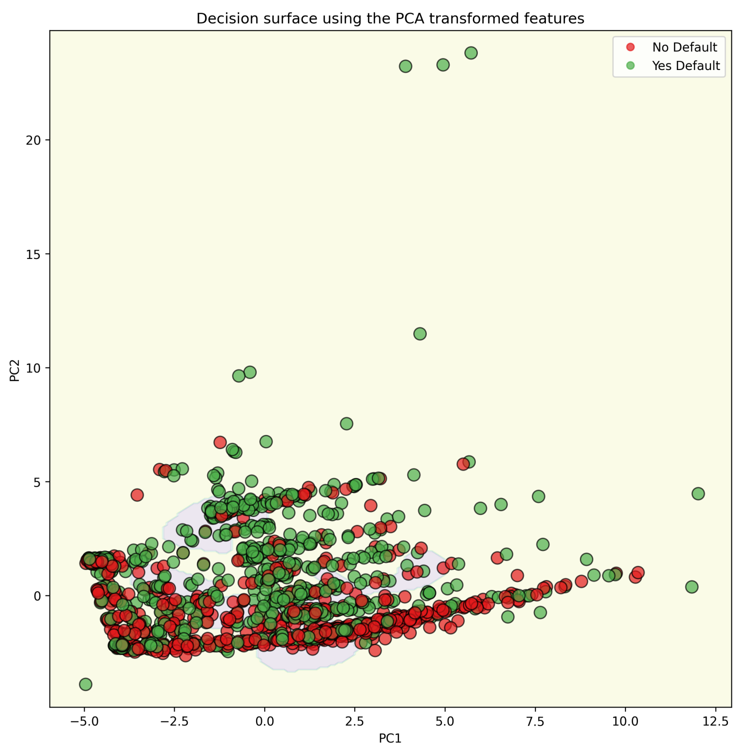

Figure 4: Decision Surface Using PCA Transformed Features.

5 Discussion

The results highlight the importance of kernel selection in SVMs for binary classification tasks. Among the evaluated kernels, the radial basis function (RBF) kernel stood out as the best performing option, achieving an accuracy of 68.75%. This performance underscores the RBF kernel's strength in mapping data into higher-dimensional spaces, allowing it to capture nonlinear decision boundaries.

However, the achieved accuracy remains relatively modest, raising questions about whether SVMs are the most suitable choice for this dataset. While the polynomial kernel showed comparable performance with an accuracy of 67.00%, its sensitivity to parameter tuning limited its robustness. In contrast, the linear kernel underperformed, achieving an accuracy of only 65.25%, which likely results from its inability to handle the nonlinear structure in the data. The sigmoid kernel, with its more complex parameterization, struggled to converge effectively and resulted in the lowest accuracy of 64.50%.

The relatively low accuracy across all kernels suggests that SVMs, while effective in many applications, may not be the optimal model for this task. The imbalanced and high-dimensional nature of the dataset likely contributed to these results. Although hyperparameter tuning through grid search improved the models as seen in the enhanced confusion matrix after optimization (Figure 2)—the improvements were incremental and insufficient to achieve significant gains in accuracy. This points to challenges inherent in the dataset, such as overlapping class distributions and complex feature relationships, that SVMs alone might not effectively address.

The PCA visualization of the decision boundary (Figure 4) offered insights into the separability of the classes. However, the limited explained variance from the first two principal components (Figure 3) suggests that most of the dataset's complexity lies in higher dimensions. This indicates a limitation of PCA for interpretability and hints that other dimensionality reduction techniques could provide more nuanced insights into the data's structure.

6 Conclusion

This project demonstrates the limitations of SVMs for predicting credit card default, with even the best performing RBF kernel achieving only 68.75% accuracy. While the findings emphasize the importance of kernel selection, none of the tested kernels achieved performance levels that would be satisfactory for real-world application. Although hyperparameter tuning via grid search led to minor improvements, the gains were not substantial enough to overcome the inherent limitations of SVMs for this specific problem.

The suboptimal performance across all kernel types, ranging from 64.50% to 68.75% accuracy, indicates that SVMs may not be well-suited for credit card default prediction on this dataset. The complex, high-dimensional nature of the data, combined with likely class overlap, presents significant challenges that SVMs struggle to address. Future work should focus on exploring more advanced machine learning techniques, such as ensemble methods or deep learning models, which could be better equipped to handle the complexities of credit default prediction. Additionally, investigating alternative feature engineering techniques and addressing the dataset's underlying issues may yield greater improvements than further optimizing SVM models.

References

- Agarwal, A. (2023). Everything About SVM Classification: Above and Beyond. *Towards Data Science*. Available at `https://towardsdatascience.com/everything-about-svm-classification-above-and-beyond-cc665bfd993e`.
- Bhatia, N. (2023). Support Vector Machines (SVM): An Intuitive Explanation. *Medium*. Available at `https://medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106`.
- Liu, K. (2024). Support Vector Machines [SVM.pdf]. CPSC 8420 Advanced Machine Learning, Clemson University.
- Liu, K. (2024). Principal Component Analysis [PCA.pdf]. CPSC 8420 Advanced Machine Learning, Clemson University.