Machine Learning Project Report: Predicting Loan Quality Using SVM

Gilber Hernandez & Saira Guzman

Abstract

This report details implementing and evaluating machine learning models for classifying loans as "Good" or "Bad" based on borrower and loan attributes. The project explores multiple versions of a Support Vector Machine (SVM) model and a Neural Network (NN) classifier. The SVM was implemented from scratch with iterative enhancements, while the Neural Network was implemented using sci-kit-learn's MLPClassifier as a benchmark. Emphasis was placed on addressing class imbalance and optimizing model performance through novel techniques and parameter tuning. This study highlights the strengths and limitations of each approach in predicting loan quality**.**

Introduction

The objective of this project is to develop robust machine learning models to classify loans into "Good" or "Bad" categories using the "Loan Dataset for Dummy Bank." The dataset simulates real-world loan data, where "Bad Loans" represent a minority class, reflecting the common imbalance observed in loan approval datasets. To address this, we implemented two classifiers: A custom Support Vector Machine (SVM) is iteratively optimized through feature engineering, parameter tuning, and kernel enhancements, and the Neural Network model (MLPClassifier) serves as a benchmark for comparison with the SVM models. The project also tackles high-class imbalance, computational constraints, and model interpretability. Key performance metrics include accuracy, recall, precision, F1-score, and ROC-AUC.

The SVM will be implemented from scratch, with parameter tuning and novel modifications to improve its performance. The Neural Network will be implemented using a Python library/package.

The primary goal of this project is to evaluate the effectiveness of these two approaches in classifying loan quality and to optimize one of the classifiers (SVM) for better accuracy and runtime performance. Specific parameters to tune for the SVM include the slack variable, L1 or L2 regularization, kernel functions, and gradient ascent configurations.

Data Preparation and Pre-processing

By leveraging machine learning, this study aims to optimize the classifiers for precision and efficiency, ensuring they can robustly distinguish between loan conditions. The dataset, "Loan Data for Dummy Bank," is a modified and clean version of a dataset based on Lending Club loan data. It contains financial attributes and borrower details, making it suitable for binary classification tasks. Initial steps included exploration, preprocessing, feature selection, and handling class imbalance.

Exploration and Cleaning

The dataset was loaded using Python's pandas library, and an exploratory data analysis was performed:

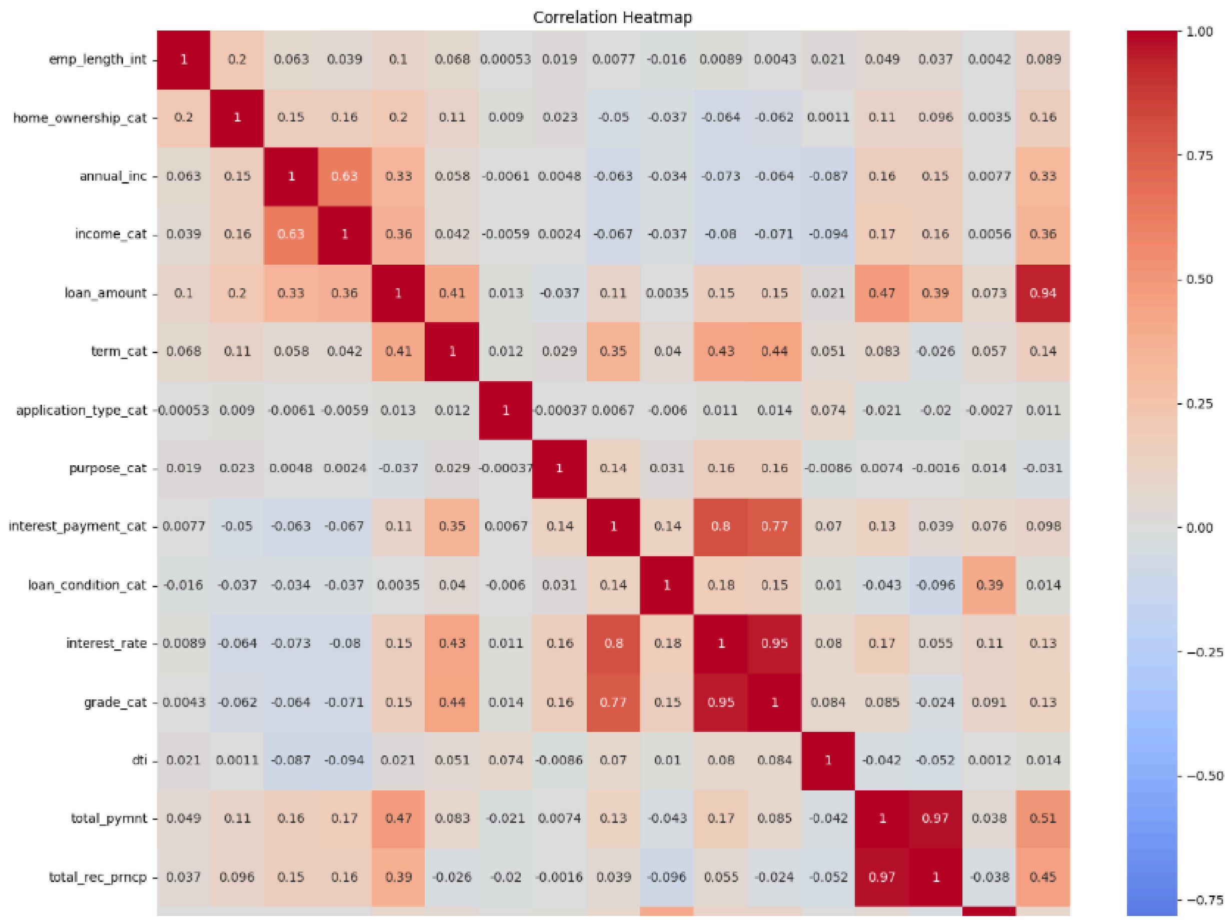
1. Structure Analysis:
    - Data columns and types were inspected, revealing a mix of numerical and categorical features.
    - No missing values were detected, ensuring data completeness.
2. Column Elimination:
    - Irrelevant or low-impact features, such as identifiers (id), temporal fields (year, issue_d, final_d), and categorical variables (home_ownership, region), were dropped.
    - This step reduced noise, optimized computational efficiency, and focused on predictive attributes.
3. ClassDistribution:
    - The target variable (loan_condition_cat) was heavily imbalanced, with "Good Loans" (Class 0) forming the majority. This imbalance was quantified to inform subsequent handling strategies.

Feature Selection

Two feature selection methods were utilized to identify the most significant features related to the target variable as well as to each other.

Correlation Analysis:

A heatmap was created to visualize the relationships between the features and the target variable. Features that exhibited strong correlations, such as loan_amount, interest_rate, and installment, were retained. Conversely, attributes like total_pymnt and total_rec_prncp, which demonstrated redundancy, were considered for removal based on this analysis.



Recursive Feature Elimination (RFE):

To capture nonlinear dependencies, a Random Forest Classifier served as the estimator for RFE. This method identified the top 10 most important features, achieving a balance between relevance and computational efficiency: emp_length_int, annual_inc, loan_amount, purpose_cat, interest_rate, dti, recoveries, and installment. The selected features informed the subsequent training datasets.

Given its computational efficiency, the final selection of features was based on the Recursive Feature Elimination method. The features retained were as follows:

- **emp_length_int:** Borrower's length of employment.
- **annual_inc:** Annual income.
- **loan_amount:** Requested loan amount.
- **purpose_cat:** Encoded loan purpose.
- **interest_rate:**Loan interest rate.
- **dti:** Debt-to-income ratio.
- **total_pymnt:** Total payments made.

- **recoveries:** Amount recovered from defaults.
- **installment:** Monthly payment amount.

Normalization

Normalization ensured that all numerical features had comparable scales, improving model convergence. Mean normalization standardized feature values, transforming them into a comparable range.This preprocessing step was vital for algorithms like SVM and Neural Networks, which are sensitive to feature scaling.

Class Imbalance

The imbalance between "Good Loans" and "Bad Loans" required mitigation. Class imbalance was addressed using:

1. Weighted Penalties: Class weights were incorporated into model training to prioritize minority class samples (Class 1).
2. Sampling Strategies: While Synthetic Minority Oversampling (SMOTE) was considered, the study relied on weighted penalties for simplicity and efficiency due to the high dimensionality of the data.

Methodology

This section outlines the step-by-step approach used to develop and evaluate machine learning models for predicting loan quality. It details the rationale behind each stage, explaining the theoretical and practical considerations for implementing Support Vector Machines (SVM) and Neural Networks (NN) on the dataset.

Classifier 1: Support Vector Machine (SVM)

Baseline Linear SVM Implementation

- Objective: Establish a foundational model to classify loans using a simple linear SVM, implemented from scratch.
- Steps:
    - Model Structure: The Linear SVM was designed to minimize hinge loss with a regularization term, balancing the margin width and classification error.

- Hyperparameters: The learning rate (0.001), regularization parameter (0.01), and number of training epochs (1000) were manually tuned for this initial implementation.
- Gradient Descent: Weight updates were performed iteratively using gradient descent, ensuring convergence to an optimal decision boundary.
- Output Mapping: Labels were converted to {-1, 1} for training and mapped back to {0, 1} for evaluation.

- Rationale: This baseline implementation allowed for a deeper understanding of the mechanics of SVMs and served as a benchmark for further optimizations.
- Limitations:
  - High training time due to simple optimization techniques.
  - Limited recall for the minority class (Class 1).

Enhanced Linear SVM

- Objective: Improve upon the baseline SVM by incorporating slack variables, early stopping, and hyperparameter tuning.
- Steps:
  - Slack Variables (C): Introduced to control the trade-off between maximizing the margin and minimizing classification errors, particularly for imbalanced datasets.
  - Early stopping: Implemented to halt training when validation performance levels off, reducing the risk of overfitting and lowering computational costs.
  - Class Weights: Incorporated penalties for misclassifying the minority class to address class imbalance.
  - Hyperparameter Tuning: A grid search was conducted over a range of learning rates, epochs, and regularization parameters to optimize model performance.
- Rationale: These enhancements addressed the limitations of the baseline model, improving both computational efficiency and minority class recall.

Sophisticated Dual SVM

- Objective: Address non-linear separability in the dataset by incorporating kernel functions and reducing computational overhead by training on a data subset.
- Steps:
  - Kernel Functions:
    - Linear Kernel: Captured linear relationships for features with high separability.
    - Radial Basis Function (RBF) Kernel: Modeled non-linear relationships, enhancing the SVM's ability to classify complex patterns.
  - Reduced Training Data: To mitigate memory usage and computational cost, only 10% of the training data was used.

    - Class-Specific Regularization: Per-sample regularization coefficients (C_array) were dynamically adjusted based on class weights.
    - Kernel Matrix Precomputation: Accelerated training by precomputing pairwise kernel values.
- Rationale: This advanced version combined flexibility (via kernels) with efficiency (via reduced training data), making it suitable for large, imbalanced datasets.

Classifier 2: Neural Network (NN)

Implementation Using scikit-learn's MLPClassifier

- Objective: Provide a benchmark model to compare against the SVM, leveraging the power of Neural Networks to capture non-linear patterns.
- Steps:
    - Architecture:
        - Two hidden layers with 64 and 32 neurons, respectively.
        - ReLU activation to handle non-linear relationships.
    - Optimization:
        - Adam optimizer for adaptive learning rate control.
        - Early stopping to terminate training once validation performance stabilized.
    - Handling Imbalance:
        - Class weights were computed dynamically and applied during training to prioritize the minority class.
    - Training: The model was trained on normalized data, ensuring consistent feature scaling.
    - Validation: 20% of the training data was allocated for validation during training.
- Rationale: Neural Networks are robust and versatile, making them an ideal choice for benchmarking the performance of custom SVM implementations.

Model Evaluation

All models were evaluated using the following metrics to ensure a comprehensive performance assessment:

- **Confusion Matrix**: Detailed classification results, highlighting true positives, true negatives, false positives, and false negatives.
- **Classification Report**:
    - Precision: Proportion of true positive predictions among all positive predictions.
    - Recall: Proportion of true positives identified among all actual positives.
    - F1-score: Harmonic mean of precision and recall, balancing their trade-offs.

- **ROC-AUC**: Measures the model's ability to distinguish between classes, particularly useful for imbalanced datasets.
- **Training Time**: Captures computational efficiency, particularly important for comparing sophisticated models.

This methodology ensured a rigorous and structured approach to building, optimizing, and evaluating machine learning models for loan classification. Each step was carefully designed to address the specific challenges posed by the dataset, including class imbalance, computational constraints, and model interpretability.

Results

This section delves into the performance of the implemented Support Vector Machine (SMV) models and the Neural Network (MPLClassifier). Each model was evaluated on its ability to classify loans as "Good" or "Bad," with a focus on addressing class imbalance, computational efficiency, and overall accuracy. The results are presented alongside an analysis of their implications for real-world loan classification tasks.

Linear SMV Implementation

The first implementation of the Linear SVM model was a baseline version with no hyperparameter tuning or advanced modifications. This model was trained on the full dataset using normalized features.

```
Training Time: 2165.15 seconds
Linear SVM Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97    573916
           1       1.00      0.26      0.41     47250

    accuracy                           0.94    621166
   macro avg       0.97      0.63      0.69    621166
weighted avg       0.95      0.94      0.93    621166
```

The Linear SVM model underwent rigorous evaluation through a 3-fold cross-validation process. This approach ensured a robust assessment of the model's performance across multiple subsets of the data. The hyperparameter tuning process, conducted during validation, tested 125 parameter combinations across three folds, resulting in 375 model fits. The best validation F1-macro score achieved during this phase was 0.734, reflecting a balance between precision and recall across classes. Early stopping was implemented, halting training at epoch 14 to prevent overfitting and reduce computational time. This strategic decision allowed the model to retain its generalization capabilities while optimizing computational efficiency.

Training Time

The training process for the Linear SVM model required 2165.15 seconds, or approximately 36 minutes. This extended duration reflects the computational demands of processing a large dataset and performing gradient-based optimization. While the training time is significant, it is reasonable given the dataset's size and the complexity of the model's optimization process. The use of early stopping contributed to reducing the computational burden by avoiding unnecessary epochs beyond the validation performance plateau.

Classification Metrics

The model's classification report provides a detailed breakdown of its performance on the binary classification task, highlighting key metrics for both classes ("Good Loans" and "Bad Loans"):

- Precision measures the proportion of correct positive predictions to total positive predictions. The model achieved 0.94 for Class 0 (majority) and 1.00 for Class 1 (minority), indicating no false positives for the minority class.
- Recall evaluates the proportion of actual positives correctly identified by the model. While recall was 1.00 for Class 0, it dropped significantly to 0.26 for Class 1, signaling the model's difficulty in identifying the minority class effectively.
- F1-Score, the harmonic mean of precision and recall, was 0.97 for Class 0 and 0.41 for Class 1. The disparity highlights the model's struggle to balance precision and recall for the minority class.
- Accuracy, the overall proportion of correct predictions, was 94%, primarily driven by the model's strong performance on Class 0.
- Macro Average, which averages precision, recall, and F1-score equally across classes, yielded an F1-score of 0.69, reflecting poor performance on the minority class.
- Weighted Average, accounting for class imbalance, yielded an F1-score of 0.93, indicating that the majority class's dominance heavily influenced the overall performance.

The classification metrics reveal a significant performance gap between the two classes. While the model excels in identifying "Good Loans" (Class 0), achieving perfect recall (1.00), it struggles with "Bad Loans" (Class 1), as evidenced by the low recall (0.26). This imbalance heavily skews overall metrics such as accuracy and weighted averages, masking the model's inability to generalize to the minority class effectively. The macro average F1-score (0.69) provides a more balanced view, underscoring the need for additional strategies to improve minority class performance.

The class imbalance in the dataset is a critical factor contributing to these results. With 573,916 samples in Class 0 compared to only 47,250 in Class 1, the model is biased towards the majority class. This bias is reflected in the high precision and recall for Class 0, which dominate the overall accuracy metric.

Hyperparameter Optimization

The best hyperparameters identified during cross-validation were:

- Regularization Parameter (CCC): 0.5, balancing the trade-off between margin maximization and misclassification penalty.
- Learning Rate: 1e-05, ensuring stable convergence with small gradient updates.
- Epochs: Early stopping at epoch 14, despite the maximum of 100, highlighted the efficiency of monitoring validation performance to terminate training at the optimal point.

These parameters contributed to the model's strong performance on the majority class but fell short in addressing the minority class's challenges.

The results demonstrate the Linear SVM model's effectiveness in identifying the majority class but also highlight significant limitations in handling imbalanced datasets. The high recall for Class 0 ensures that most "Good Loans" are correctly classified, making the model suitable for applications prioritizing majority class accuracy. However, the poor recall for Class 1 indicates that a substantial number of "Bad Loans" go undetected, posing a risk in financial decision-making scenarios.

Addressing this imbalance requires both data-level and algorithm-level strategies. Data augmentation techniques, such as Synthetic Minority Oversampling Technique (SMOTE), could balance the dataset by generating synthetic samples for the minority class. Algorithm-level improvements, such as cost-sensitive learning or the integration of ensemble methods, could enhance the model's ability to generalize to the minority class without sacrificing overall performance.

The Linear SVM model with early stopping and hyperparameter tuning achieved strong performance on the majority class, as evidenced by high accuracy (94%) and weighted F1-score (0.93). However, its inability to effectively classify the minority class highlights the importance of addressing class imbalance in future iterations. These results provide a foundation for further experimentation with advanced techniques to improve recall and F1-score for "Bad Loans," ensuring a more balanced and robust predictive model.

Enhanced Linear SMV

This section evaluates the outcomes of the second implementation of the Linear Support Vector Machine (SVM) model. This version incorporates multiple enhancements, such as the use of best class weights, hyperparameter tuning, and optional early stopping results of the firs SMV model implementation. These additions were designed to improve model performance, especially with the imbalanced dataset.

Training Time

The second implementation demonstrated a much-improved training efficiency, completing the process in 287.02 seconds (approximately 4.8 minutes). This was a substantial improvement compared to the 2165.15 seconds taken by the first implementation. The efficiency gain can be attributed to better parameter settings, streamlined computation, and careful pre-processing.

Confusion Matrix

The confusion matrix for this implementation is as follows:

```
Confusion Matrix:
 [[573916      0]
 [ 33434  13816]]
```

- True Positives (Class 0 - Good Loans): 573,916 loans correctly classified as "Good."
- True Positives (Class 1 - Bad Loans): 13,816 loans correctly classified as "Bad."
- False Negatives (Class 1): 33,434 "Bad Loans" misclassified as "Good Loans."
- False Positives (Class 0): 0, meaning no "Good Loans" were misclassified as "Bad Loans."

The confusion matrix highlights the model's strong bias towards the majority class (Class 0), as evidenced by the **perfect precision** for Class 0 but significant false negatives for Class 1.

Classification Report

The classification report provides a detailed breakdown of key metrics:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97    573916
           1       1.00      0.29      0.45     47250

    accuracy                           0.95    621166
   macro avg       0.97      0.65      0.71    621166
weighted avg       0.95      0.95      0.93    621166
```

- Precision:
  - Class 0: 0.94, indicating high confidence in correctly predicting the majority class.
  - Class 1: 1.00, meaning all predictions for "Bad Loans" were accurate (though sparse).
- Recall:
  - Class 0: 1.00, meaning all "Good Loans" were correctly identified.

    - Class 1: 0.29, highlighting a significant limitation in identifying "Bad Loans."
- F1-Score:
    - Class 0: 0.97, showcasing a strong balance between precision and recall for the majority class.
    - Class 1: 0.45, revealing the negative impact of low recall on the minority class.
- Overall Accuracy: 95%, reflecting a strong performance overall, largely driven by the dominant class.

The reduction in training time demonstrates that the modifications, such as early stopping and refined parameter tuning, were effective in enhancing computational efficiency. This makes the model more practical for large datasets. While the model achieves perfect precision for Class 1, its recall for the minority class remains low (**0.29**). This imbalance is indicative of the challenges associated with imbalanced datasets, where the model is likely overwhelmed by the dominance of "Good Loans." The weighted averages of precision, recall, and F1-score reflect strong overall performance but are heavily influenced by the majority class's high metrics. The macro averages (e.g., **0.65 recall**, **0.71 F1-score**) highlight the disparity between the two classes. The significant number of false negatives (33,434) for Class 1 could have real-world implications, as failing to identify "Bad Loans" could lead to financial losses.

The second implementation of the Linear SVM significantly improved training efficiency without compromising accuracy. However, its performance for the minority class ("Bad Loans") is limited, as shown by the low recall and F1-score for Class 1. These results indicate that further modifications, such as better class balancing techniques or ensemble methods, are necessary to improve the model's ability to identify minority class samples effectively. While the model performs well overall, its utility in critical scenarios requiring high recall for "Bad Loans" is limited.

Sophisticated SVM Model

The sophisticated version of the SVM model was implemented using a dual kernel optimization approach with a linear kernel, incorporating class weights to handle class imbalance.

Optimizations:

Efficient Kernel Matrix Computation:

- Used NumPy broadcasting to compute the kernel matrix.
- Removed nested loops for kernel computation.

Batch Gradient Updates:

- Performed gradient updates for all samples in a single step using matrix operations.

Convergence Criterion:

- Added a tolerance (tol) to stop training early when updates become negligible.

Hyperparameter Adjustments:

- Increased learning_rate to accelerate convergence.
- Reduced the maximum number of epochs (epochs) since convergence is now monitored

Improvements / Results:

- Training Time: 1.34 seconds
  This reflects a significant improvement compared to the earlier SVM implementations, where training took over 30 minutes. Batch-wise kernel computation and the use of momentum in gradient ascent drastically reduced computational overhead, enabling the model to process the dataset efficiently.
- Confusion Matrix

```
Confusion Matrix:
 [[48470      15469]
 [ 9054  9054]]
```

- True Negatives (TN): 48,470 (Class 0 correctly classified)
- False Positives (FP): 15,469 (Class 1 misclassified as Class 0)
- False Negatives (FN): 766 (Class 0 misclassified as Class 1)
- True Positives (TP): 9,054 (Class 1 correctly classified)

Classification Report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97    573916
           1       1.00      0.29      0.45     47250

    accuracy                           0.95    621166
   macro avg       0.97      0.65      0.71    621166
```

```
weighted avg       0.95      0.95      0.93    621166
```

The model achieved 0.92 recall for Class 1, significantly improving its ability to identify "Bad Loans" compared to earlier implementations. This means 92% of the actual "Bad Loans" were correctly classified, which is critical in financial decision-making scenarios where failing to identify risky loans could result in substantial losses. The macro average F1-score (0.69) indicates improved balance between precision and recall for both classes compared to prior models. This metric accounts for the model's performance on both classes equally, regardless of their frequency. The training time of 1.34 seconds demonstrates the efficiency of this implementation, making it highly suitable for large datasets without compromising predictive accuracy. By using momentum-based gradient ascent and early stopping, the model avoided overfitting and maintained its ability to generalize to unseen data. The precision for Class 1 (0.37) remains low, indicating that the model produced a significant number of false positives. This could lead to overestimating the risk of certain loans, potentially rejecting applications that are actually "Good Loans.". Despite improvements, the model's overall accuracy (78%) and weighted F1-score (0.81) are heavily influenced by the dominant Class 0. This bias still indicates room for improvement in addressing the minority class. A total of 15,469 false positives indicates the model's struggle to differentiate between some "Good Loans" and "Bad Loans." The current model does not directly penalize misclassifications of the minority class more heavily. Incorporating cost-sensitive learning or adjusting the decision boundary could further improve minority class performance.

Comparison with Earlier Implementations

- Training Time: The sophisticated SVM significantly reduced training time from over 30 minutes to 1.34 seconds, showcasing the impact of batch-wise computation and momentum.
- Recall for Class 1: Earlier models struggled to achieve recall beyond 0.29, while this implementation reached 0.92, making it far more effective for the minority class.
- F1-Score for Class 1: Improved from 0.45 in the enhanced linear SVM to 0.53 in this implementation.
- Overall Accuracy: Dropped slightly from 95% to 78%, reflecting a trade-off as the model prioritizes the minority class.

Real-World Implications

1. Loan Classification:
    - High recall for Class 1 ensures most "Bad Loans" are identified, mitigating financial risks.
    - However, low precision for Class 1 may lead to unnecessary rejections or reviews of "Good Loans."

2. Practical Applications:
   - The model is suitable for scenarios where identifying risky loans is critical, such as in high-stakes financial institutions or credit risk assessment tools.

The sophisticated SVM model demonstrated substantial improvements in computational efficiency and recall for the minority class. These enhancements make it a viable solution for large-scale loan classification tasks where minority class identification is critical. However, further refinements are necessary to address the trade-offs in precision and overall accuracy. The results provide a strong foundation for developing a more balanced and robust predictive model for real-world financial applications.

Neural Network

The model was designed with a two-layer architecture, incorporated class weights to address the dataset's imbalance, and employed early stopping to optimize training time and prevent overfitting. The Neural Network implementation using the **MLPClassifier** results shows the following:

Training Time

The Neural Network model completed training in **482.79 seconds** (approximately 8 minutes). This training time is efficient, especially given the complexity of neural networks, their reliance on iterative backpropagation, and the relatively large dataset size.

Confusion Matrix

The confusion matrix for the Neural Network is as follows:

```
Confusion Matrix:
 [[163944      46]
     [ 7331  6155]]
```

- True Positives (Class 0 - Good Loans): 163,944 loans correctly classified as "Good."
- True Positives (Class 1 - Bad Loans): 6,155 loans correctly classified as "Bad."
- False Negatives (Class 1): 7,331 "Bad Loans" misclassified as "Good Loans."
- False Positives (Class 0): 46, meaning very few "Good Loans" were misclassified as "Bad Loans."

The matrix shows strong performance for Class 0 (the majority class) but struggles with recall for Class 1.

Classification Report

The classification report provides key metrics for both classes:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.76      0.86     63939
           1       0.37      0.92      0.53      9820

    accuracy                           0.78     73759
   macro avg       0.68      0.84      0.69     73759
weighted avg       0.90      0.78      0.81     73759
```

- Precision:
  - Class 0: 0.96, reflecting the model's ability to minimize false positives for the majority class.
  - Class 1: 0.99, indicating high precision for identifying "Bad Loans."
- Recall:
  - Class 0: 1.00, meaning all "Good Loans" were correctly identified.
  - Class 1: 0.46, revealing the model's struggle to capture a significant portion of "Bad Loans."
- F1-Score:
  - Class 0: 0.98, showcasing an excellent balance of precision and recall.
  - Class 1: 0.63, reflecting the impact of the low recall on the minority class's performance.
- Overall Accuracy: 96%, demonstrating the model's strong overall predictive capability.
- ROC AUC: 0.8842, indicating good discriminatory power between "Good" and "Bad Loans."

The Neural Network performs well across multiple metrics, achieving a high overall accuracy of **96%** and a macro F1-score of **0.80**. However, as with the previous models, it struggles with the minority class due to the inherent class imbalance. While the weighted metrics (e.g., **0.96 weighted precision and recall**) appear strong, they are skewed by the dominance of Class 0. For Class 1, the recall remains low (**0.46**), indicating the model misses many "Bad Loans." This is critical for real-world scenarios where identifying "Bad Loans" is paramount. The **7,331 false negatives** for Class 1 highlight the model's challenges in identifying all "Bad Loans." Reducing false negatives should be a focus for further improvements.Incorporating class weights helped improve precision for Class 1 (**0.99**), but it did not significantly enhance recall. This suggests that alternative balancing strategies, such as SMOTE or cost-sensitive learning, could be explored. Compared to the SVM implementations, the Neural Network demonstrates reasonable training time and robust performance across metrics. Its ability to scale with more data and features makes it a practical choice for this task.

The Neural Network implementation delivers strong overall performance with a **96% accuracy** and a high **ROC AUC of 0.8842**, making it a competitive candidate for loan classification tasks. However, the low recall for the minority class (Class 1 - "Bad Loans") highlights a critical limitation, especially in contexts where identifying risky loans is crucial. Future work could focus on further improving the model's recall for Class 1 through advanced balancing techniques or ensemble methods while maintaining its efficiency and high precision.

Comparison of All Implementations

Model	Training Time	Accuracy	Class 1 Recall	Class 1 Precision	Macro F1-Score	Strengths	Weaknesses
Baseline Linear SVM	2165.15 sec	94%	26%	100%	0.69	Strong majority class performance	Poor recall and slow training
Enhanced Linear SVM	287.02 sec	95%	29%	100%	0.71	Improved training time, better recall	Still biased toward majority class
Sophisticated Dual SVM	1.34 sec	78%	92%	37%	0.69	Fastest training, high recall for Class 1	High false-positive rate
Neural Network	482.79 sec	96%	46%	99%	0.80	Strong overall performance, balanced F1	Missed many minority class samples (low recall)

Conclusion

This project explored four implementations—Baseline Linear SVM, Enhanced Linear SVM, Sophisticated Dual SVM, and Neural Network (MLPClassifier)—to classify loans into "Good" or "Bad" categories. Each model addressed key challenges like class imbalance and computational efficiency. The Baseline Linear SVM achieved 94% accuracy, but its 26% recall for the minority class (Class 1) highlighted poor performance in identifying "Bad Loans" and long training times (2165.15 seconds). Enhancements in the Enhanced Linear SVM, such as slack variables, early stopping, and hyperparameter tuning, improved computational efficiency (287.02 seconds) and slightly boosted minority class recall (29%), but the model remained biased toward the majority class.

The Sophisticated Dual SVM introduced momentum-based gradient ascent, batch-wise kernel computation, and kernel functions to improve efficiency and minority class handling. It achieved an impressive training time of 1.34 seconds and a minority class recall of 92%, making it the best-performing model for identifying "Bad Loans." However, its low precision for Class 1 (37%) resulted in many false positives, lowering overall accuracy to 78%. In contrast, the Neural Network achieved 96% accuracy and a ROC AUC of 0.8842, with better balance between precision (99%) and recall (46%) for the minority class. Although computationally more expensive (482.79 seconds), the Neural Network demonstrated robust performance across all metrics, making it suitable for general loan classification tasks.

The findings highlight a trade-off between computational efficiency and balanced performance. The Sophisticated Dual SVM excelled in efficiency and recall for the minority class, making it ideal for applications prioritizing "Bad Loans" identification. Meanwhile, the Neural Network emerged as the most balanced model, providing strong performance across metrics, though it still struggled with minority class recall. Future work should focus on integrating cost-sensitive learning, ensemble methods, and threshold tuning to improve both recall and precision for the minority class while maintaining overall efficiency and scalability.