mlgroup45

Credit Risk Assessment (Team 45) – Final Checkpoint:

Credit Risk Assessment Machine Learning model for CS4641 (Machine Learning) at Georgia Tech. Team is composed of 5 undergraduate students, namely Marc-Alain Adjahi, Christopher Farrington, Ansh Gadodia, Kashyap Nathan, and Gowtam Kommi.

Introduction

For financial institutions, risk management is imperative, and credit risk assessment is a fundamental component of this. Institutions need to accurately evaluate the likelihood that a borrower will default on a loan in order to make informed decisions when lending money. In the past, analysts manually combed through data and formed predictions based on their analysis. However, with the rise of advanced computing techniques like ML, more data can be assessed to improve predictive accuracy.

Literature

ML-based credit risk models can quantitatively assess loan default likelihood of different clients with greater accuracy by formulating the problem as binary classification (Xueyan 2012). By leveraging various methods such as logistic regressions, decision trees, and support vector machines as methods of detecting credit risks, models can be enhanced and thus trusted to provide reliable predictions. At the same time, a cautious approach is still required due to challenges arising with model training. Presence of data bias can directly impact fairness in determining whether or not to provide a loan (Strivasta 2021). Additionally, a large amount of data with high quality is required for accurate model predictions, and this data is not always readily available. Regulatory compliance, like data privacy and usage laws, often restrict usable data, and institutions must deploy models that continue to adhere to regulations.

Dataset

We will be using a German credit risk dataset, which has 1000 data points with 20 features for each data entry. The dataset supports binary classification, as each data point has a “good/bad

credit risk" label that our predictions can compare to. We hope to deploy various models trained on this dataset, and determine the model with the most accurate predictive performance.

Problem Definition and Motivation

Credit Risk: the likelihood of a counterparty defaulting on their debt. We want to improve financial institutions' predictive ability when it comes to default rates, and most of that is going to come from improving their ability to assess credit risk. Beyond that, we would like to provide open-source models so that anyone (not just financial institutions) can provide credit assessments. This will pave the way for future innovations such as peer-to-peer lending platforms, or even providing investors the ability to utilize credit risk-dependent financial securities by creating an open-source, accessible way to assess credit risk for a customer.

Methods

Note for File Directories going forwards:

- SVM (with PCA and standardization built in) was implemented in /SVM/svm.py, and used as an imported class in a notebook
- Test train split, data encoding, and results from the SVM model being trained and run on the data are in /SVM/svm_notebook.ipynb
- GMM and KMeans from our inclass implementations are in /UnsupervisedModel/gmm.py and /UnsupervisedModel/kmenas.py, respectively
- The results from using these models (with PCA, test train split, data encoding, etc.) are in /UnsupervisedModel/unsupervised_notebook.ipynb
- Simple log reg is implemented in /log_reg/logistic_regression.py, and the model is carried out in /log_reg/model.ipynb
- SGD-LR is done in /log_reg/std_logreg.ipynb

Preprocessing – Standardization and PCA

- We started by encoding categorical features to make the dataset compatible with numeric-based machine learning algorithms. Specifically, we used one hot encoding for categorical features without any logical ordering to them, and used integer encoding for features that were ordinal (i.e. low class, middle class, upper class).
- Then, we used standardization to scale all features, ensuring that attributes like Credit Amount and Duration don't dominate others. This makes each feature contribute equally to distance-based methods, enhancing model stability and interpretability.

- Finally, we applied PCA to our encoded and standardized data. PCA allows us to reduce dimensionality and improve processing efficiency (although, for 1000 data points, the latter is not a primary concern). With PCA, we retained components that capture highest variance, allowing us to work with a compact representation of the data while preserving key information. This variance maximization, while also preserving key trends, can be helpful for our algorithms to more easily and effectively find a decision boundary.

Machine Learning Algorithm – SVM (Supervised Learning)

- For a supervised classification algorithm, we implemented SVM to predict credit risk. SVM was chosen due to its ability to create clear boundaries between classes by maximizing the margin, allowing it to generalize well to points that may be close to the decision boundary. This is especially beneficial for credit risk predictors, as this margin maximization inherently aims to reduce misclassification risk (i.e. preventing misclassifying a bad risk as a good one). In addition, by focusing on support vectors (boundary data points), SVM minimizes the impact of outliers (which can be prominent in credit risk data, i.e. crazy spenders). Finally, SVM can be implemented with class weight adjustments to aid in the class imbalance within a credit risk dataset.
    - In addition, SVM allows us to use kernels to alter its effectiveness. We can experiment with linear and RBF kernels to see what works best. Specifically, RBF kernels are notably good at capturing non linear relationships, which could be beneficial for our credit risk dataset.

Machine Learning Algorithm – KMeans and GMM (Unsupervised Learning)

- For an unsupervised approach, we implemented KMeans and GMM (using our homework implementations) to predict credit risk. Clustering helps find inherent groups in data, allowing us to group credit risk profiles without the use of predefined labels. Specifically, our thought process was to use two clusters, assuming that low credit risk profiles would share similar features, and that high credit risk features would also share similar characteristics, allowing our clustering algorithms to seperate them into two groups. KMeans provides a simple way to group credit risk profiles into spherical groups of data, which can provide us with a good baseline of results for clustering algorithms. GMM, assuming that credit risk data exhibits complex patterns, should provide us with better results, as it allows for overlapping clusters and can capture more complex patterns in the data (ie soft boundaries between low and high risk groups, and more complex, non-spherical shapes for clusters).

Machine Learning Algorithm – Logistic Regression and SGD + L2 Regularized Logistic Regression (Supervised Learning)

- For a supervised classification algorithm, we implemented logistic regression to predict credit risk. Logistic regression is a fundamental algorithm for binary classification, particularly well-suited for scenarios where we want interpretable results and probabilistic outputs. It models the relationship between the input features and the binary outcome using a logistic function, ensuring outputs between 0 and 1, which can be directly interpreted as probabilities of credit risk. Logistic regression is particularly valuable for credit risk prediction as it assumes a linear decision boundary, which can simplify interpretability. However, real-world credit risk datasets often contain noise, multicollinearity, and class imbalances. To address these challenges, we extended our implementation with SGD and L2 regularization. L2 regularization adds a penalty to large coefficients, which reduces overfitting and helps the model generalize better by focusing on the most significant features. Additionally, stochastic gradient descent optimizes the model iteratively, improving convergence for large datasets.
    - This approach ensures that the model is less sensitive to outliers and better suited to handle imbalanced data, as credit risk datasets typically have many “good” risks and fewer “bad” risks. The combination of logistic regression, L2 regularization, and SGD allows us to strike a balance between simplicity, interpretability, and improved performance by refining the decision boundary for both balanced and imbalanced datasets.

Overfitting Considerations

In credit risk assessment, overfitting poses a particular challenge as it could lead to:

- Poor generalization to new credit applications
- Over-reliance on noise in historical data
- False confidence in risk predictions

This is especially critical in our context, where misclassifications can have significant financial implications.

Implemented Prevention Measures

1. Data Splitting Strategy:
    - Utilized a 70-30 train-test split, ensuring no data leakage
    - Maintained class distribution across splits
    - Used random_state=42 for reproducibility
2. Regularization Parameters:
    - Implemented C=1.0 in our SVM model
        - This regularization parameter controls the trade-off between margin maximization and classification error
        - Lower C values create larger margins but allow more misclassifications

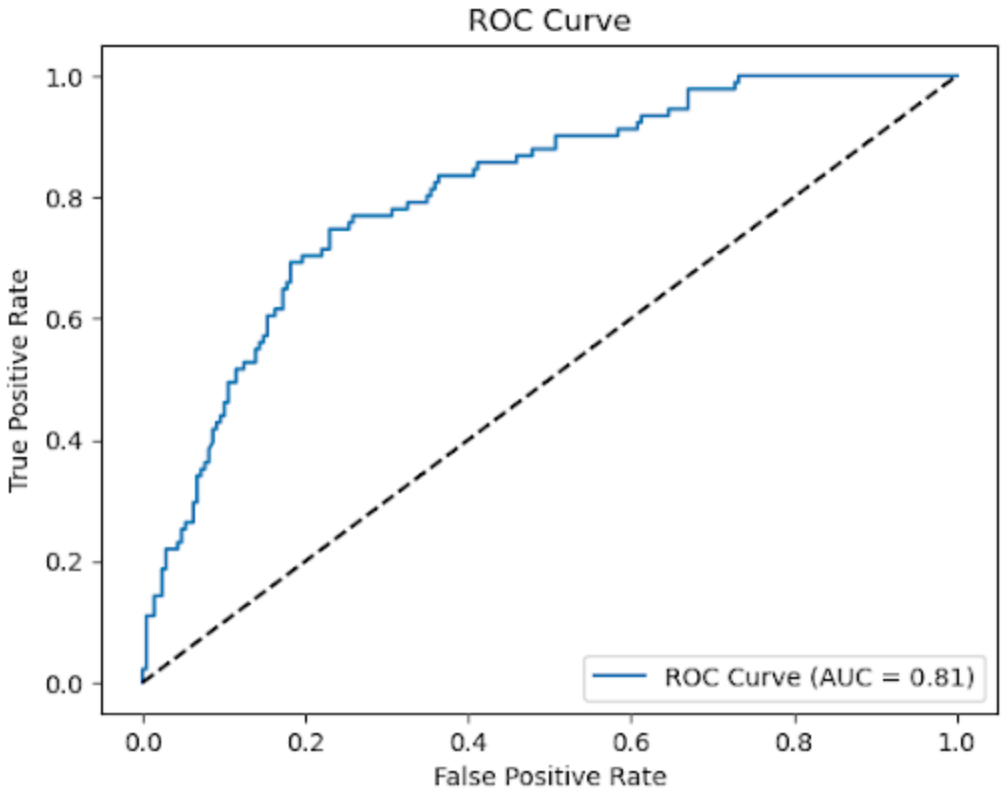
        - Higher C values aim for fewer misclassifications but risk overfitting
    - Our SGD-LR uses L2 regularization, which directly prevents overfitting.
3. Feature Engineering and Selection
    - Applied PCA for dimensionality reduction
    - Reduced feature space complexity
    - Removed potential noise in less significant components
    - Standardized features to prevent scale-based overfitting

Results

Quantitative Metrics and Visualizations - SVM

Our SVM model achieved balanced performance across key metrics:

- Accuracy: 0.76
- Precision: 0.75
- Recall: 0.76
- AUC-ROC Score: 0.81



- Figure 1: ROC Curve - Shows model's ability to distinguish between classes

Interpreting Our Metrics:

- These balanced results (0.76 / 0.75 across Accuracy, Precision, and Recall) indicate that the SVM model provides a consistent and reliable prediction performance without a heavy bias toward one metric.
- This is promising, as it suggests that the model is effectively capturing high-risk cases while controlling false positives to an acceptable level.
- The AUC-ROC score of 0.81 suggests good discriminative ability, with a relatively high true positive rate while keeping false positives to a minimum.

Feature Analysis and Dimensionality Reduction

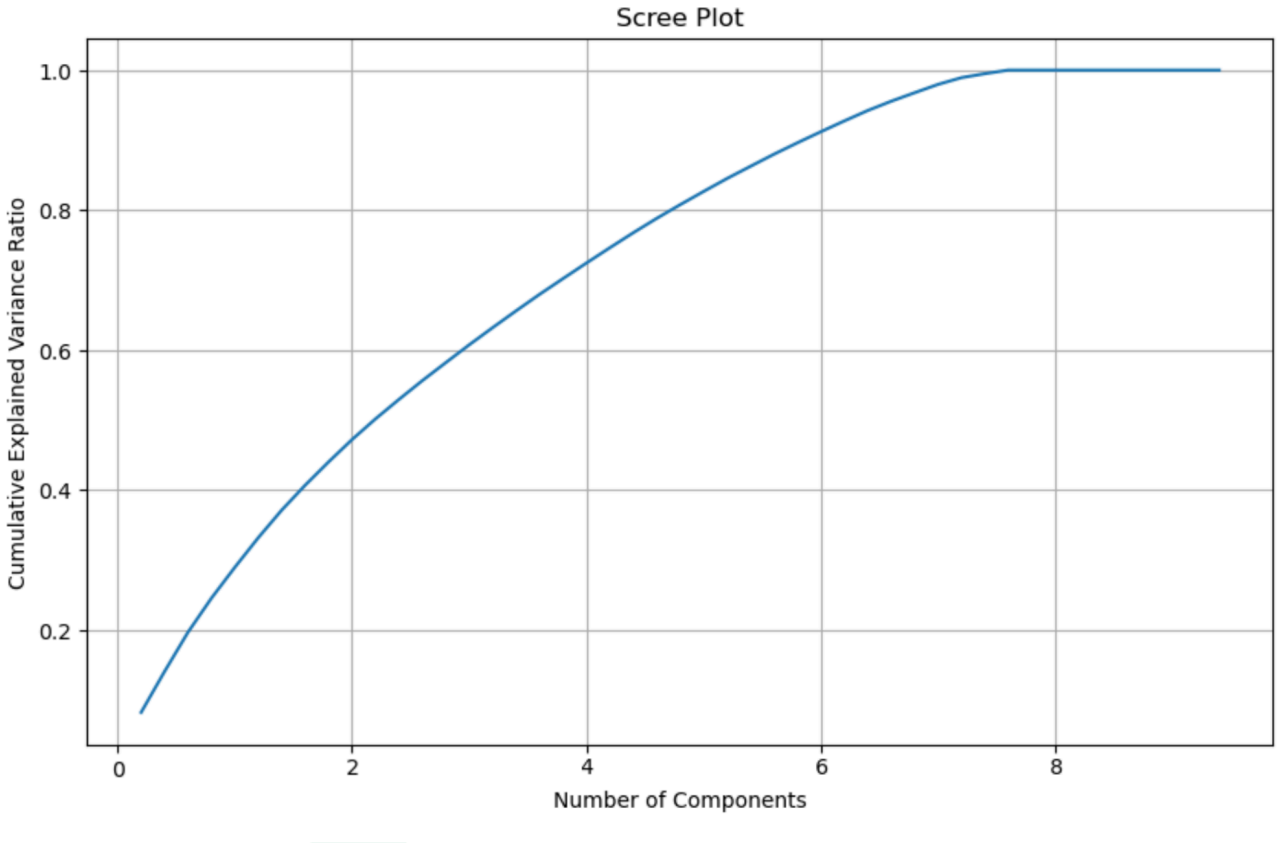


Figure 2: PCA Scree Plot - Shows cumulative explained variance

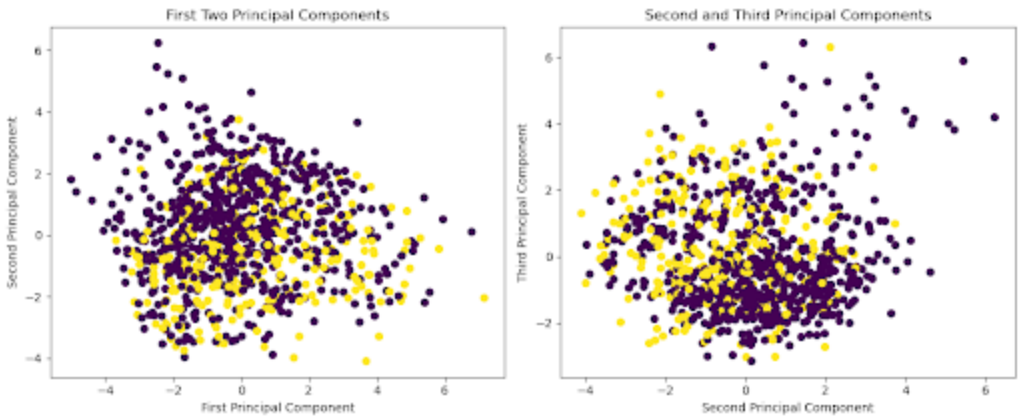


Figure 3: Principal Components Scatter Plot

As seen in Figures 2 and 3, our PCA analysis reveals that:

- The first 6 components explain approximately 95% of the variance
- Some overlap exists between classes, as can be seen in the visualization of the first three principle components, indicating an inherent challenge in a perfect classification

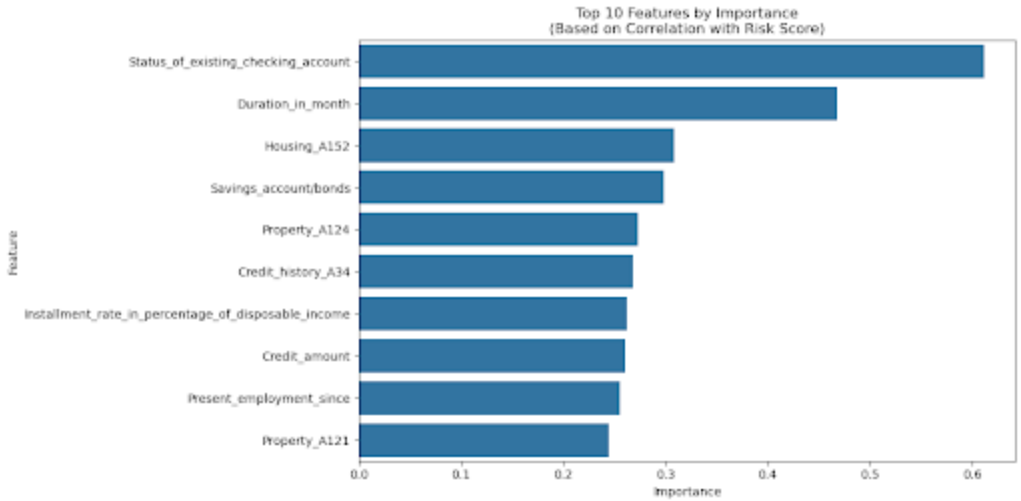


Figure 4: Top 10 Feature Importance Plot

The feature importance analysis shows that:

- Duration and Credit Amount are the strongest predictors
- Account status and employment history also play significant roles
- Demographic features have relatively lower importance

Confusion Matrix

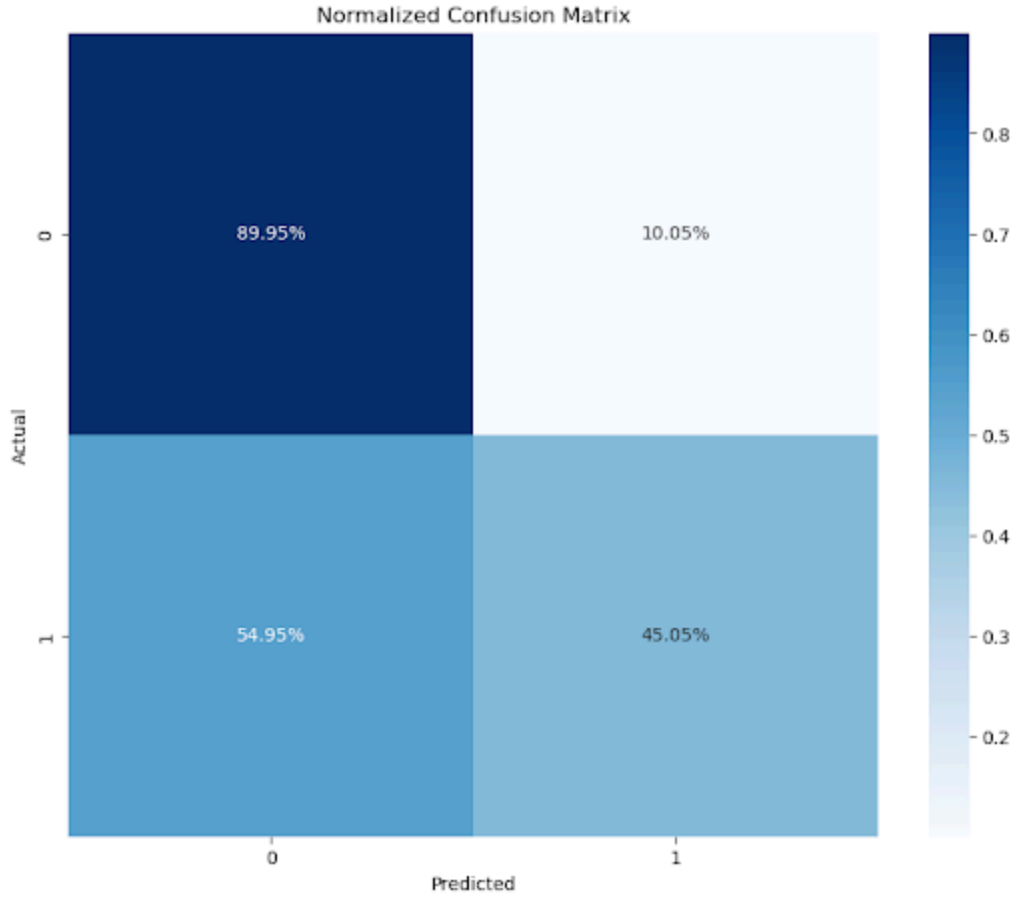


Figure 5: Normalized Confusion Matrix with Percentages

Particularly important in credit risk assessment is the analysis of false positives, as seen in Figure 6 below:

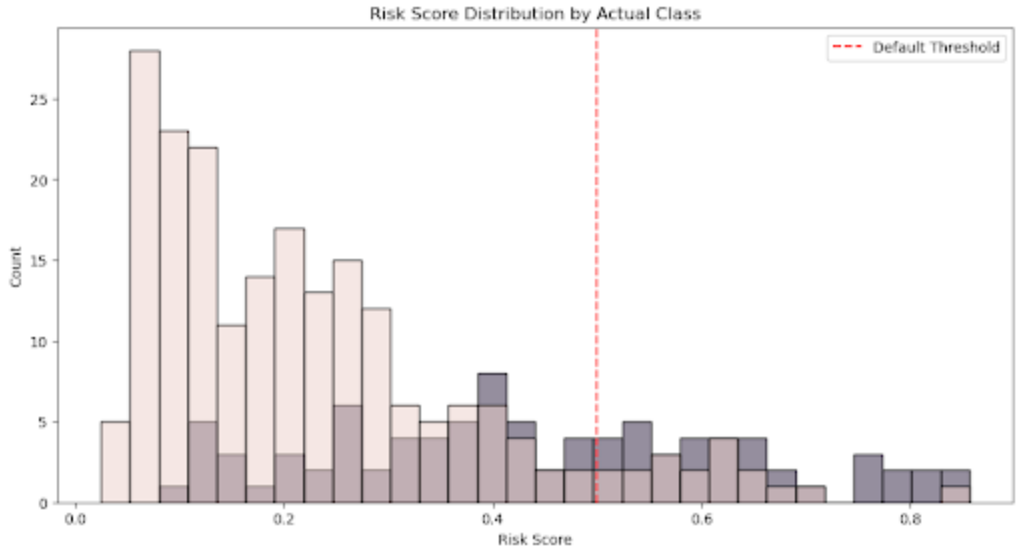


Figure 6: Risk Score Distribution by Actual Class

The error analysis reveals:

- Most false positives occur near the decision boundary
- The overlap (shown by the medium color) visualizes the difficulty the model has discriminating between classes, especially around the decision boundary

Model Performance Discussion - SVM

Our SVM model achieved balanced performance across the board based on our metric selection. This is evidenced by an accuracy of 0.76, precision of 0.75, recall of 0.76, and an AUC-ROC score of 0.81. These metrics indicate that the model is well-calibrated and strikes a reasonable balance between identifying true positive cases and minimizing false positives, which is essential in credit risk assessment. However, with an accuracy of 75%, the models performance (though consistent) could be considered modest and thus has room for improvement. Because we are doing credit risk, we could specifically focus on minimizing false positives even more than we already do.

Building off of that, the AUC-ROC score shows that the model has reasonably good discriminative ability (as AUC-ROC above .8 can be considered sufficient). Nonetheless, our visual PCA analysis shows that class overlap is inevitable, which may make increasing our models discriminative ability in the future quite difficult.

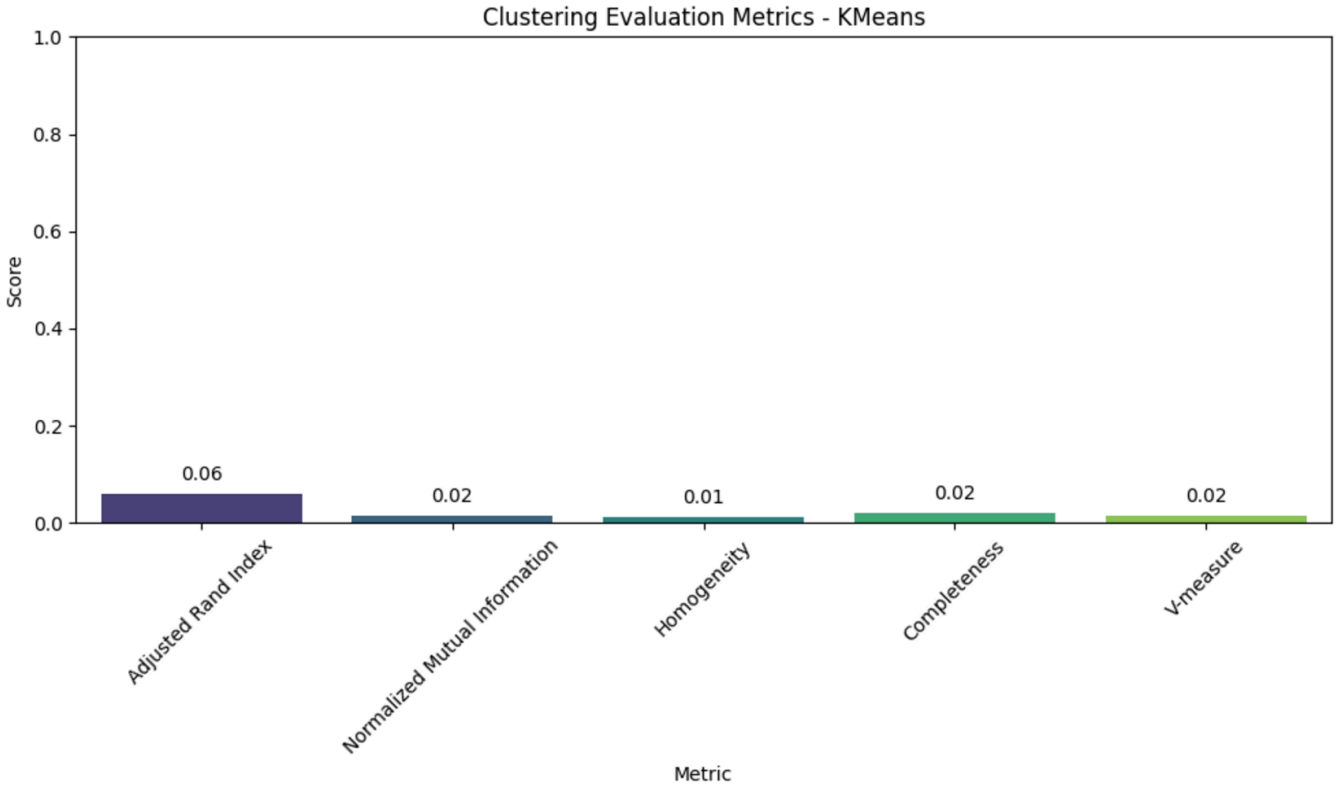
Discussing our other visualizations, our feature importance graph shows that factors like duration and status of checking account are by far the most influential predictors, with demographic features having relatively lower significance. Error analysis revealed that most false positives occurred near the decision boundary, particularly among cases with higher credit amounts. This suggests that while the model captures most patterns effectively, there are edge cases—particularly higher credit amounts—where misclassification could result in higher financial costs. This limitation points to the model's difficulty in fully differentiating between classes in areas where boundaries are less distinct.

In summary, the SVM model demonstrates robust and consistent performance for the dataset, capturing key patterns necessary for credit risk prediction while maintaining a balance across metrics. However, there is room for improvement in handling boundary cases and further refining the model's performance, especially for cost-sensitive applications.

Quantitative Metrics and Visualizations - KMeans and GMM

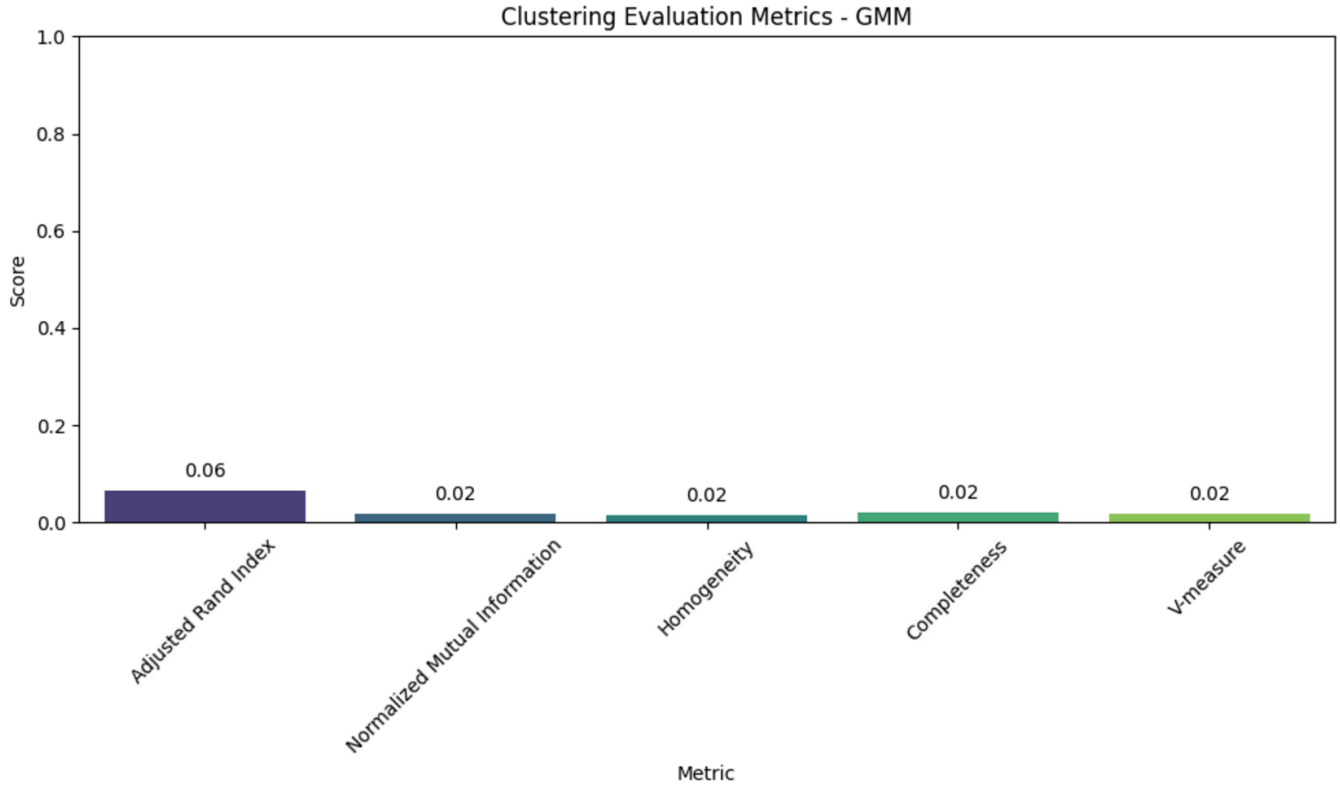
KMeans: </br>

Accuracy Metric: 0.57 </br> Precision Metric: 0.60 </br> Recall Metric: 0.78 </br> Fowlkes Mallow Meausre: 0.68 </br>



GMM: </br>

Accuracy Metric: 0.56 </br> Precision Metric: 0.60 </br> Recall Metric: 0.71 </br> Fowlkes Mallow Meausre: 0.65 </br>



Model Performance Discussion - KMeans and GMM

In terms of these evaluation metrics, both models performed poorly. For both models, accuracy hovers a little above 50 percent, meaning that both models are essentially randomly guessing for each class label. In addition, both models had a precision of about 60 percent, highlighting the presence of many false positive pairs where points from different true classes were incorrectly grouped into the same cluster.

Recall, however, was higher, with KMeans achieving 78% and GMM 71%, suggesting that both models were relatively effective at grouping points from the same class into the same cluster. Despite this, the high recall values are offset by the moderate precision, as the clusters are not "pure," and many true class pairs are likely split across multiple clusters.

Finally, the fowlkes mallow score provides a metric that balances both precision and recall. An FM of .68 and .65 for KMeans and GMM indicates a very moderate performance, with a lot of room for improvement. These scores indicate that while the models capture some relationships between points, they fail to achieve the separability required for reliable predictions.

In regards to the metrics presented in our visualizations, both GMM and KMeans showed extremely low scores across label-invariant metrics such as Adjusted Rand Index (0.06), Normalized Mutual Information (0.02), Homogeneity (0.02), Completeness (0.02), and V-measure (0.02). Specifically:

- **ARI**: 0.06 means poor agreeement between clustering assignments and true class labels beyond what would be expected by random chance.
- **NMI**: 0.02 means minimal shared mututal information.
- **Homogenity, Completeness, V-Measure**: 0.02 across the board means that clusters contain mixes of classes / true classes are fragmented.

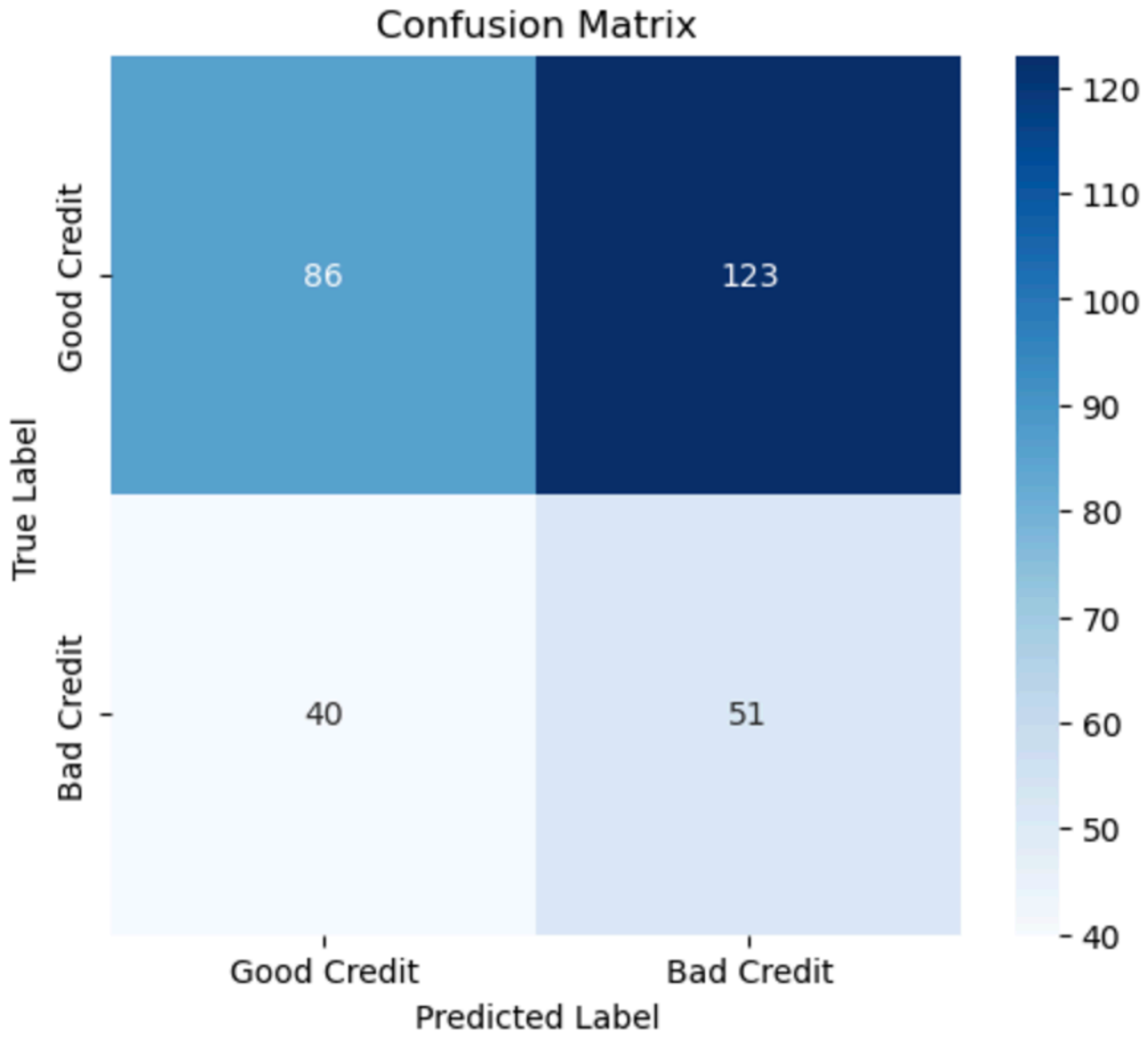
These results highlight that the clusters poorly align with the true credit risk classes, with clusters being impure and fragmented. The low values reflect the models' inability to capture meaningful class structures, further supporting that these clustering methods are unsuitable for distinguishing between good and bad credit risks in this dataset.

The poor performance is likely due to the assumptions of the models — KMeans assumes spherical clusters with equal variance, and GMM assumes Gaussian distributions — neither of which may align with the true data distribution. In addition, the complexity of credit risk data, which involves non linear relationships and overlapping feature spaces, makes it unsuitable for mundane clustering methods like KMeans and GMM. Other clustering methods could be tried, but supervised methods are usually going to provide much better results.

Quantitative Metrics and Visualizations - Simple Logistic Regression & Stochastic Gradient Descent, L2 Regularized Logistic Regression

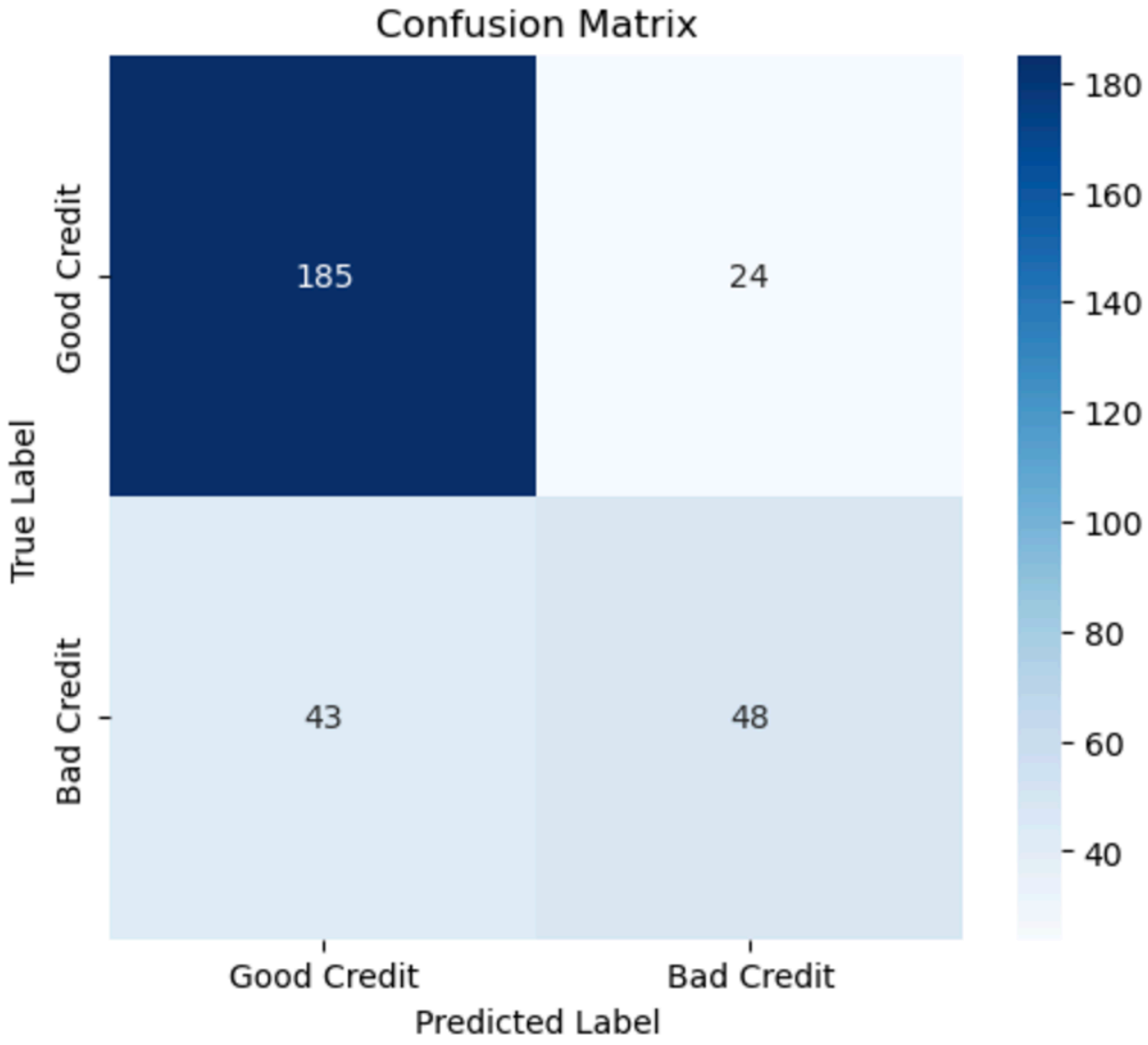
Simple Logistic Regression: </br>

Accuracy Metric: 0.46 </br> Precision Metric: 0.56 </br> Recall Metric: 0.46 </br> F1 Score: 0.47 </br>



SGD + L2 Regularized Logistic Regression: </br>

Accuracy Metric: 0.78 </br> Precision Metric: 0.77 </br> Recall Metric: 0.78 </br> F1 Score: 0.77 </br>



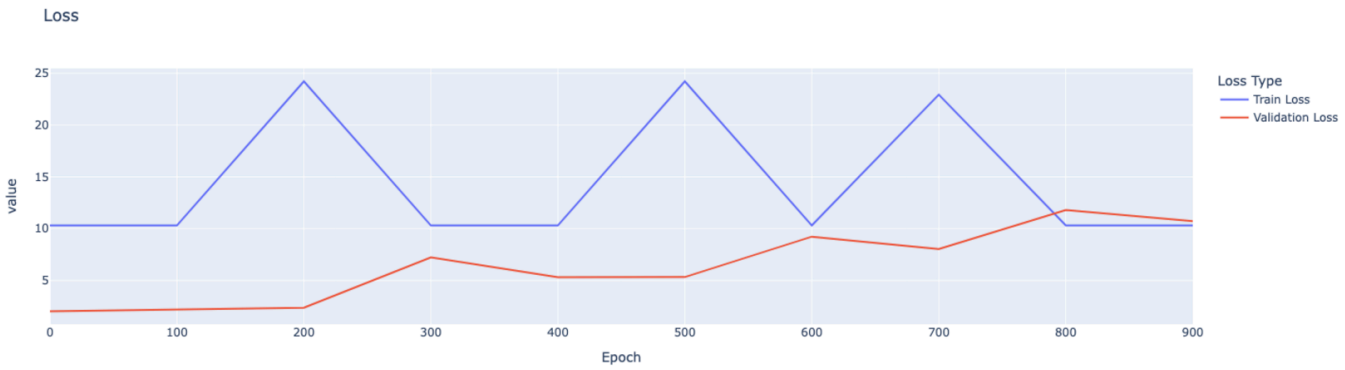
Model Performance Discussion - Simple Logistic Regression and SGD-LR with L2 Regularizer

Based on the evaluation metrics, it is clear that using a simple logistic regression is worse than guessing, and that optimizing the logistic regression model by using stochastic gradient descent and L2 regularization provides a significant improvement on the model performance.

For the first model, overall accuracy is below 50 percent, meaning that it is worse than guessing at identifying labels. Further, we identified that the model has a better F1 score for positive (good credit) labels at 0.51, but performance drastically falls when the label is negative with a 0.38 F1 score. The model identified a large amount of false negatives, which in the context of credit risk assessment, is especially concerning because it can lead to bad credit risks being approved for loans.

The low precision and recall metrics, especially for the minority class, indicate that the model is biased towards the majority class. This bias is likely because the model assigns higher probability to the majority class to minimize overall error, a common issue in imbalanced datasets. Without mechanisms to counteract this bias, the model inadequately learns the characteristics of the minority class.

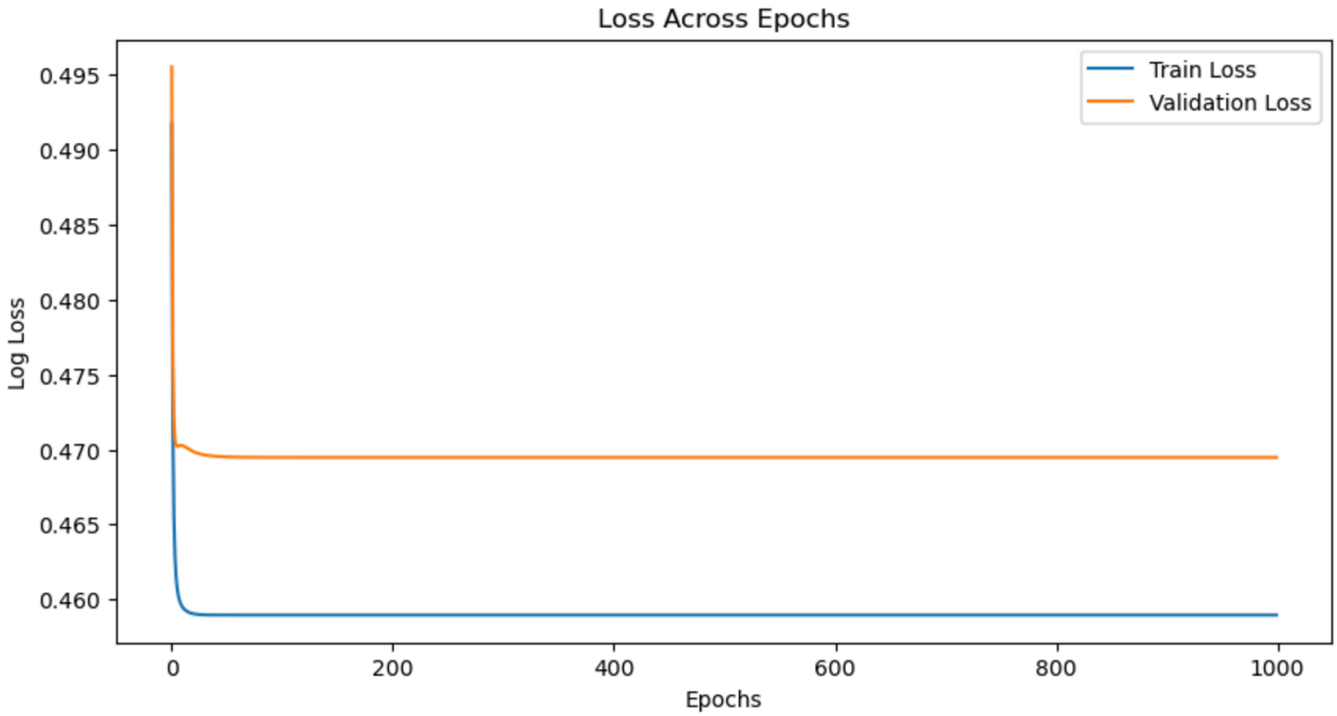
Finally, we observed that the model does not converge to an optimal solution during training, hence its poor performance across the board.



The optimized model with stochastic gradient descent and an L2 regularizer displays stark performance differences across the board. First, all 4 metrics show drastic improvements, and the more important ones (accuracy and F1 score) are close to 0.8. This suggests that the implemented penalty by the L2 regularization combined with using SGD for parameter updates is a worthwhile addition.

Specifically, the L2 term penalizes large coefficients in the model, which discourage the model form adding too much weight to one feature. This directly prevents overfitting, where the model captures noise instead of actual patterns. Overfitting was likely an issue in the original simple model, which lead it to perform poorly on its validation set.

In addition to this, the fact that SGD+L2 improves the performance this much could mean that there is high multicollinearity between features in our dataset. Multicollinearity leads to more unstable coefficients, making the model itself unstable and sensitive to minor changes in data. Regularization mitigates this. We also theorize that the regularization fixes or at least lessens the bias towards the majority class that we initially observed in the simple implementation. Finally, we also conclude that the SGD assists with the convergence of the model, as observed in the figure below:



Both confusion matrices support our observations. As previously stated, the simple implementation is culprit to a high amount of false negatives. This is adressed by our optimization.

Model Performance Comparison

Model	Accuracy	Precision (Bad Credit)	Recall (Bad Credit)	Fowlkes-Mallows Score
SGD-LR	0.78	0.77	0.78	N/A
SVM	0.76	0.75	0.76	0.72
K-Means (Raw Features)	0.57	0.60	0.78	0.68
GMM (PCA Features)	0.56	0.60	0.71	0.65

What was compared

Four models were selected for comparison: Stochastic Gradient Descent Logistic Regression (SGD-LR), Support Vector Machine (SVM), and unsupervised methods using K-Means and Gaussian Mixture Models (GMM).

Comparing Supervised Models

SVM vs SGD-LR with L2 Regularizer: Both models had robust results and nearly the same performance across key metrics. SVM had an accuracy of 76%, precision of 75%, and recall of 76%, with SVM being two points higher on all of these metrics. So, SGD-LR technically outperformed the SVM implementation, but by a small margin.

- **SVM Strengths**: SVM has the ability to handle non linear relationhsips with its kernels, which can provide an edge when there are complex non linear relationships present. In addition, it's focus on boundary points reduces the effects of outliers, making it more robust against noise and skewed data points. However, SVM does require more computational resources for training and fine tuning.
- **SGD-LR Strengths**: Despite its linear assumptions, SGD-LR was able to slightly outperform SVM with the help of SGD for convergence and L2 regularization for overfitting prevention. In addition, SGD-LR is less computationally expensive, making it more favorable for scalable applications.
- **SVM Tradeoffs**: Better for data with complex, nonlinear relationships, at the expense of needing more computational resources.
- **SGD-LR Tradeoffs**: Comparable performance with a simpler, less expensive implementation. However, linear assumptions could make future improvement and accurate performance on broader data harder to achieve.

Comparing Unsupervised Models

KMeans vs GMM: KMeans achieved an accuracy of 57% with a recall of 78% and precision of 60%, while GMM had similar performance with an accuracy of 56%, recall of 71%, and precision of 60%. Both models struggled to align their clusters with true labels due to the complex nature of the credit risk dataset.

- **KMeans Strengths**: KMeans is distance based, so it works well when there are well defined feature distinctions. So, in the context of our credit risk data set, this strength lead to poor performance on the non linear and overlapping patterns in the dataset.
- **GMM Strengths**: GMM's probablistic approach allows it to have overlapping clusters and more complex cluster shapes when compared to KMeans. However, despite this advantage that we theorized earlier, this did not give it an edge over KMeans.
- **KMeans Tradeoffs**: Computationally inexpensive and very simple to implement, but fails to capture complex relationships due to its simplicity.
- **GMM Tradeoffs**: Greater flexibility when it comes to cluster shapes and assignments, but did not actually outperform KMeans. Also, GMM has a more complex implementation compared to KMeans.

Supervised vs Unsupervised Models

Performance Comparison: Supervised models (SVM and SGD-LR) outperformed unsupervised models (KMeans and GMM) across all metrics. While supervised methods leverage labeled data to optimize decision boundaries, unsupervised models inherently lack this guidance, making them less sufficient for tasks requiring good alignment with true labels.</br> **Tradeoffs**:

- Supervised models provide better metrics and interpretability, but require labeled data and usually more complex implementations.
- Unsupervised models, while much less effective, can be good for exploratory analysis (hence why we opted for all other models to be supervised after seeing its results).

Future Improvements/Next Steps

1. Model Enhancements:
    - **SVM**: A grid search method for hyperparameter tuning was utilized after the midterm report, slightly increasing accuracy to 79%. However, these new hyperparameters had negative effects on precision, recall, and AUC-ROC scores. Ultimately, we decided that our original model showed more well rounded metrics, and thus kept it's results in this report. Going forwards, we could employ more advanced forms of hyperparameter tuning such as Bayesian optimization, which can be done with open source libraries like Optuna. In addition, it might be beneficial to explore different types of kernels outside of those provided by the sklearn library.
    - **SGD-LR**: Addressing this models limitations with class imbalance and linearity is key to enhancing its performance. Adjusting class weights, or introducing synthetic data for the minority class, would improve recall for high risk credit predictions. Exploring non linear transformations could also improve the model's ability to capture complex relationships. Finally, regularization fine tuning could be done with grid search, and the SGD algorithm could be improved with an adaptive learning rate.
    - **KMeans and GMM**: These models are inherently limited due to their unsupervised nature. However, there is potential for improvement: refined cluster initilization methods, kernelized KMeans, and GMM with flexible covariance types could all lead to better unsupervised results. Experimenting with different numbers of clusters could also increase metrics across the board, but would not logically fit the two inherent groups in the data. Finally, it might make sense to experiment with other clustering methods, like DBScan or hierarchical clustering.
2. Cost Matrix Refinement:
    - Develop more sophisticated cost matrices based on actual business impact (penalize false positives more heavily)
    - Implement dynamic thresholds based on risk tolerance
    - Consider variable costs based on credit amount

3. Feature Engineering:
    - Create interaction features between key predictors
    - Develop more sophisticated encoding for categorical variables
    - Incorporate domain-specific feature transformations
4. Real-world Considerations:
    - Address potential bias in the model
    - Implement interpretability techniques for regulatory compliance
    - Develop monitoring systems for model performance in production

References

1. S. Lessmann et al., “Benchmarking state-of-the-art classification algorithms for credit scoring,” Journal of the Operational Research Society, vol. 63, no. 10, pp. 1508–1519, 2015.
2. L. Xueyan et al., “Sparse representation for credit risk evaluation,” IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 8, pp. 1206–1214, 2012.
3. T. Fawcett, “An introduction to ROC analysis,” Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.
4. R. K. Srivastava et al., “Mitigating bias in credit risk scoring using Bayesian hierarchical modeling,” Journal of the Operational Research Society, vol. 72, no. 6, pp. 1219–1235, 2021.

Gantt Chart

Our Gantt Chart for this project.

Contribution Table

Team Member	Contribution
Marc-Alain Adjahi	Problem Definition/Motivation, Results, Model Performance Discussion
Christopher Farrington	Methods, Results, Unsupervised Model, SVM Model
Ansh Gadodia	Introduction/Literature, Model Comparisons
Kashyap Nathan	Methods, Results

Team Member	Contribution
Gowtam Kommi	Discussion, References/Citations
All	Video Recording, Presentation, GitHub Page