Credit Score Classification using Model Selection and Machine Learning

Dominique Kellam
Department of *Computer Science & Information Technology*
Eastern Kentucky University
Richmond, Kentucky, US
dominique_kellam@mymail.eku.edu

***Abstract*— This project focuses on building and evaluating machine learning models to classify credit scores based on financial and credit-related information. The dataset used comprises various features such as annual income, number of bank accounts, outstanding debt, and credit utilization ratio, with the goal of predicting whether a person's credit score falls into one of three categories: Good, Standard, or Poor. I implemented various feature selection techniques and machine learning algorithms, including Lasso, chi-squared (chi2), Ridge, Recursive Feature Elimination, Principal Component Analysis, Max Voting, and Stacking. After a detailed evaluation using metrics like accuracy, precision, recall, F1-score, and ROC-AUC, I analyzed the pros and cons of each method and selected the best-performing models. The results showed that the Stacking model outperformed others, with an accuracy of 92.17%. This report outlines the dataset, feature selection techniques, machine learning models, evaluation metrics, results, and potential areas for future work.**

I. Introduction

Credit score classification is a critical problem in the financial industry, as it helps assess the risk associated with lending to individuals. Financial institutions rely on accurate classification to make informed decisions regarding loans and credit lines, aiming to minimize risk while maintaining profitability. A person's credit score is a strong indicator of their financial reliability, with higher scores signifying lower lending risk. Incorrect classification could result in financial loss for lenders or unjustified credit denial for qualified individuals, making this an essential area for machine learning applications.

In this project, I tackle the problem of credit score classification by employing various feature selection techniques and machine learning models. I aim to identify the most relevant features in the dataset and determine the most accurate models for predicting credit scores. Through this, we hope to provide insights into how different selection techniques and algorithms perform, ultimately contributing to the financial sector's ability to accurately assess lending risks.

II. Literature Review

Credit scoring has seen substantial advancement due to machine learning, which outperforms traditional methods such as logistic regression by capturing complex relationships between financial variables. A systematic literature review by Noriega and Rivera highlighted the evolution of credit scoring models, particularly ensemble learning, which has shown superior performance in handling large datasets and imbalanced classes [1]. Similarly, Dastile et al. emphasized the effectiveness of machine learning models like Random Forest and Gradient Boosting in improving predictive accuracy and handling diverse financial data [2].

Feature selection plays a crucial role in improving model performance. Studies by González et al. discussed how techniques like Lasso, Ridge, and RFE help in reducing overfitting while improving model interpretability [5]. These methods were vital in my project, where they helped identify the most influential financial variables for predicting credit scores, such as annual income and credit utilization ratio.

Addressing the challenge of class imbalance is critical in credit scoring. Techniques like SMOTE, as discussed by Gicić and Subasi, have been applied to create balanced datasets, ensuring models don't disproportionately favor the majority class [6]. I used RandomOverSampler in my project to resolve this issue and ensure the model performed well across all credit score categories (Good, Standard, Poor).

Finally, ensemble models such as Stacking and Max Voting have proven to significantly improve classification accuracy. Suhadolnik et al. demonstrated the effectiveness of combining multiple models for credit scoring, particularly in reducing errors and improving prediction accuracy in imbalanced datasets [3].

III. Data Set

The dataset used in this project consists of over 50,000 records, each containing various financial attributes related to individuals' credit history. Key attributes include annual income, the number of bank accounts, delayed payments, credit utilization ratios, and the number of credit inquiries for a total of 28 features. The target variable, Credit_Score, is categorized into three classes: Class 0 = Standard, Class 1 = Good and Class 2 = Poor.

The dataset was preprocessed to handle outliers, missing values, and scaling. Outliers were removed using the Interquartile Range method, and both Min-Max Scaling and Standard Scaling were applied to the features depending on the specific requirements of the feature selection methods. After preprocessing, the dataset was balanced using oversampling to ensure equal representation of all classes.

Here, the class distribution before and after oversampling is summarized in Table 1. The distribution shows that the dataset initially had an imbalance, which was corrected to ensure that each class was equally represented during model training.

TABLE I.

Credit Score	Data Distribution		
	Class 0	***Class 1***	***Class 2***
Before Oversampling	53.17%	17.83%	29.00%
After Oversampling	33.33%	33.33%	33.33%

IV. Main Approach

The task involved applying various feature selection methods, training machine learning models, and then evaluating their performance using a range of metrics.

A. Feature Selection Methods

I used several techniques such as Lasso, chi-squared (chi2), Ridge, Recursive Feature Elimination (RFE), and PCA to reduce the dimensionality of the dataset. Each method selected a subset of features based on different criteria.

- Lasso regression helps by applying L1 regularization, which forces some coefficients to be zero, effectively selecting a subset of the most important features.
- The chi-squared test selects features based on their correlation with the target variable.
- Ridge applies L2 regularization and selects features based on the magnitude of the coefficients.
- Recursive Feature Elimination ranks features by recursively eliminating the least important ones.
- Principal Component Analysis transforms the data into a lower-dimensional space by projecting it onto principal components.

The general workflow was as follows:

1. Preprocess the data (remove outliers, scale features).
2. Apply feature selection techniques to reduce the dimensionality of the dataset.
3. Train and evaluate models using the selected features.
4. Compare performance metrics such as accuracy, F1 score, precision, recall, and ROC-AUC.

B. Machine Learning Algorithms

In my project, I implemented several machine learning algorithms to classify credit scores into one of three categories: Good, Standard, or Poor. Each algorithm brought a different approach to the credit score classification problem.

- Random Forest: Random Forest is an ensemble learning method that I used to handle the complexity and variability of the dataset. It constructs multiple decision trees, each trained on random subsets of the data and features, and outputs the class prediction that is most common among all the trees (majority vote).

  Random Forest was particularly useful because it handled the complex, non-linear relationships between financial variables like annual income, number of loans, and credit utilization ratio effectively. By averaging predictions from multiple decision trees, Random Forest reduced the risk of overfitting and gave reliable predictions.

  $$\hat{y} = \text{mode}(\{h_1(x), h_2(x), \ldots, h_T(x)\})$$

  $\hat{y}$ is the predicted credit score class (Good, Standard, or Poor), $h_i(x)$ is the prediction from the

i-th decision tree based on the individual's financial data, *T* is the total number of decision trees in the forest.

- Decision Tree: Decision Tree classifiers split the dataset into smaller and smaller subsets based on feature values, eventually arriving at a class prediction. Each node in the tree represents a decision based on one financial feature (e.g., number of bank accounts), and the leaves represent the predicted credit score class.

  In this project, Decision Trees provided an easy-to-interpret model that helped reveal how certain financial factors lead to different credit score classifications. For instance, the tree might split on interest rates or delayed payments to decide whether a credit score falls into the Poor or Standard class.

  $$G(X_m) = 1 - \sum_{k=1}^{K} p^2 mk$$

  $G(X_m)$ is the Gini impurity for a particular node in the tree, p_{mk} is the proportion of class *k* and node *m*, *K* is the number of credit score classes (Standard, Good, Poor).

  The tree makes splits to minimize the impurity, ensuring that nodes contain individuals with more similar credit scores.

- Gradient Boosting: Gradient Boosting builds models sequentially, with each new model correcting the mistakes made by the previous ones. It uses decision trees as weak learners, which are gradually improved to minimize a loss function. For my credit score classification project, Gradient Boosting allowed the model to handle complex interactions between financial variables, such as how number of delayed payments and monthly balance affect credit scores.

  $$F_m(x) = F_{m-1}(x) + \eta h_m(x)$$

$F_m(x)$ is the model's prediction at stage m, $F_{m-1}(x)$ is the prediction from the previous stage, η is the learning rate, controlling how much each tree contributes to the overall model, $h_m(x)$ is the new tree added at stage m.

In practice, Gradient Boosting sequentially corrects for the previous model's misclassifications, resulting in better predictions for credit scores, especially for borderline cases like Standard and Poor.

- Max Voting: Max Voting is an ensemble method that combines the predictions of multiple models and chooses the majority vote as the final prediction. For the credit score classification task, I combined predictions from Random Forest, Decision Tree, and Gradient Boosting models. Max Voting takes the predicted credit scores from each model and assigns the final class based on the majority vote.

$$\hat{y} = \text{mode}(\{h_1(x), h_2(x), \ldots, h_N(x)\})$$

$h_1(x), h_2(x), \ldots, h_N(x)$ are the predictions from different models, $\hat{y}$ is the predicted credit score class (Good, Standard, or Poor) based on the majority vote.

In my case, if two out of three models predicted a credit score as *Standard*, Max Voting would classify the individual as *Standard*.

- Stacking: Stacking is a more advanced ensemble technique where multiple base classifiers are trained, and their predictions are used as inputs for a meta-model. In my project, I used Stacking to combine the strengths of several models (Random Forest, Decision Tree, Gradient Boosting) and then trained a meta-model (Random Forest) to make the final credit score classification.

$$\hat{y} = g(\{f_1(x), f_2(x), \ldots, f_N(x)\})$$

$f_1(x), f_2(x), \ldots, f_N(x)$ are the predictions from base classifiers, g is the meta-model, in this case, a Random Forest trained on the base model predictions, $\hat{y}$ is the final predicted credit score class.

In practice, Stacking allowed me to combine multiple models, with the meta-model learning how to best combine their predictions, resulting in the highest accuracy of 92.17%.

V. Evaluation Metric

I evaluated the performance of each model using a combination of the following metrics:

1. Accuracy: This measures the overall correctness of the model by calculating the proportion of correct predictions out of the total predictions. For multi-class classification tasks like mine, accuracy is essential in assessing the model's ability to distinguish between the three classes.

2. Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positives. This metric is particularly useful when the cost of false positives is high, as it tells me how reliable the model is in predicting a particular credit score category.

3. Recall: Recall is the ratio of correctly predicted positive observations to the actual positives. For my task, this was critical to understanding how well the model identifies individuals in each credit score class without missing any critical cases.

4. F1-Score: The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both. This was crucial when dealing with imbalanced classes in the dataset.

5. ROC-AUC: The area under the ROC curve measures how well the model separates the classes. For a multi-class classification task, this gives insight into how well the model distinguishes between 'Good,' 'Standard,' and 'Poor' credit scores.

These metrics provided comprehensive insights into the performance of each model, considering both the precision and recall trade-offs, along with the overall effectiveness of the classification task.

VI. Results and Analysis

A. Baseline Performance and

Initially, we trained the models without applying any feature selection techniques. The baseline results indicated that the Random Forest classifier performed best, achieving an accuracy of 87.5%. This was expected, given the robustness of Random Forest in handling high-dimensional data. However, it was clear that some features were not contributing significantly to the model's performance.

B. Feature Selection and Model Performance

After applying various feature selection techniques, we observed significant improvements in model accuracy and other performance metrics. The chi-square method (chi2) emerged as the best feature selection technique when paired with Random Forest, achieving an accuracy of 91.54%. This was followed closely by RFE and Ridge, both of which improved model accuracy to over 91%. Lasso, while effective, did not perform as well as the other methods, achieving an accuracy of 87.53%.

One of the most surprising findings was the performance of ensemble methods. Both MaxVoting and Stacking significantly outperformed individual models, with Stacking achieving the highest overall accuracy of 92.18%. The ROC curves for each model showed that Stacking consistently had the highest AUC values across all credit score categories, indicating near-perfect classification ability.

The ROC-AUC for Stacking reached 0.986, indicating a near-perfect classification capability. Chi-square and RFE also showed strong performance, with AUC values close to 0.98. Gradient Boosting underperformed compared to Random Forest, likely due to its sensitivity to the number of features and the lack of hyperparameter tuning.

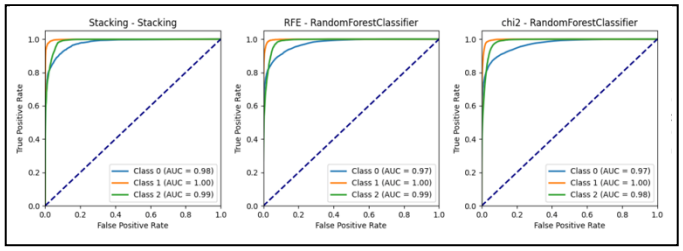
TABLE II.

Baseline Accuracy for Models							
Method	***Model***	***Accuracy***	***F1***	***Recall***	***Precision***	***ROC***	***# Feat***
Lasso	Random Forrest	0.875	0.872	0.875	0.875	0.954	6
Lasso	Decision Tree	0.853	0.850	0.852	0.851	0.912	6
Lasso	Gradient Boosting	0.721	0.719	0.721	0.720	0.857	6
Chi2	Random Forrest	0.907	0.905	0.907	0.909	0.976	15
Chi2	Decision Tree	0.879	0.876	0.878	0.878	0.909	15
…	…	…	…	…	…	…	…
PCA	Random Forrest	0.837	0.833	0.837	0.836	0.878	21
PCA	Decision Tree	0.725	0.722	0.725	0.724	0.871	21
PCA	Gradient Boosting	0.876	0.873	0.876	0.878	0.976	21

C. *Takeaways and Surprises*

The key takeaway from these results is that feature selection plays a crucial role in improving model performance. Simple methods like chi2 can be highly effective, even outperforming more complex regularization techniques. Ensemble models, particularly Stacking, significantly boost classification accuracy by leveraging the strengths of multiple classifiers.

One surprising result was the relatively poor performance of Lasso, which did not improve model accuracy as much as anticipated. This might be due to the linear assumptions inherent in Lasso, which may not capture the complex relationships in the dataset.



[a.] Credit Score Classification Juptyer Notebook

Fig. 1. ROC Curves for Top Models

D. *Potential Errors*

There were some misclassifications, particularly between the "Standard" and "Good" credit score categories. This may be due to the overlap in feature values between these two classes, causing the model to struggle with distinguishing between them. Another potential issue is the limited hyperparameter tuning for Gradient Boosting, which could explain its underperformance compared to Random Forest.

VII. Future Work

There are several directions for future work to improve the results of this project. First, I plan to explore deep learning models such as neural networks, which could capture more complex patterns in the dataset. Additionally, more sophisticated hyperparameter tuning methods, such as GridSearchCV, could be applied to optimize model performance further.

I also plan to incorporate domain-specific feature engineering, creating new features based on financial rules or expert knowledge. This could help the model capture more meaningful patterns and improve classification accuracy. Finally, I would explore the use of techniques like SMOTE to handle class imbalance more effectively.

VIII. Conclusion

This project demonstrated the importance of feature selection techniques and ensemble methods in improving credit score classification accuracy. Stacking was the most effective method, achieving an accuracy of 92.18%, followed by RFE and chi2 feature selection methods. The results highlight that combining feature selection with ensemble techniques provides the best performance in credit score classification.

With further improvements, such as advanced hyperparameter tuning and deep learning models, this system could be deployed in real-world financial applications to assist lenders in making more accurate credit decisions.

References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] Noriega, J.P., Rivera, L.A. (2023). Machine Learning for Credit Risk Prediction: A Systematic Literature Review. Data, 8(11), 169. https://doi.org/10.3390/data8110169

[2] Dastile, X., Celik, T. (2020). Statistical and Machine Learning Models in Credit Scoring: A Systematic Literature Survey. Applied Soft Computing, 91, 106263. https://doi.org/10.1016/j.asoc.2020.106263

[3] Suhadolnik, N., Ueyama, J., Da Silva, S. (2023). Machine Learning for Enhanced Credit Risk Assessment: An Empirical Approach. Journal of Risk and Financial Management, 16(12), 496. https://doi.org/10.3390/jrfm16120496

[4] Edla, D.R., Tripathi, D., Cheruku, R., Kuppili, V. (2018). An Efficient Multi-layer Ensemble Framework with BPSOGSA-based Feature Selection for Credit Scoring Data Analysis. Arabian Journal for Science and Engineering, 43(12), 6909–6928. https://doi.org/10.1007/s13369-018-3523-5

[5] González, S., García, S., Del Ser, J., Rokach, L., Herrera, F. (2020). A Practical Tutorial on Bagging and Boosting-Based Ensembles for Machine Learning: Algorithms, Software Tools, Performance Study. Information Fusion, 64, 205–237. https://doi.org/10.1016/j.inffus.2020.07.008

[6] Gicić, A., Subasi, A. (2019). Credit Scoring for a Microcredit Dataset Using SMOTE and Ensemble Classifiers. Expert Systems with Applications, 36(2), e12363. https://doi.org/10.1111/exsy.12363

[7] Kaggle (2023). Credit Score Classification Dataset. https://www.kaggle.com/datasets/parisrohan/credit-score-classification