Multi-Class Credit Score Health Classification

Sachit Patel

College of Computing and Digital Media, DePaul University

DSC 345: Machine Learning

Dr. Casey Bennett

June 6th, 2024

1. Abstract

Credit scores involve a wide array of factors and time within calculating such a three digit number. Advancements have been made over the years in predicting and categorizing credit scores. Lenders who have analyzed related data have learned how to categorize common groups of people and their similar credit scores given similar financial statistics. And because of these similar financial statics, they have in-range gained the capability to profile groups of people and the level of lending risk they take on with them.

However, something is lost in each advancement of accuracy- the explainability and interpretability of such models. Using interpretability techniques on top of several different modeling attempts and combinations, patterns can be found within the data to create new findings for lenders to net higher safety in decision-making.

2. Introduction

As a system, credit scores indicate an individual's trustworthiness with payments, loans, and other forms of borrowing. A higher credit score is able to net purchases or transactions otherwise not possible, and at lower interest rates, making these transactions further justifiable. Evaluating credit scores efficiently, effectively, and accurately gives lenders more consistency and coverage in how they approach lending with various common types of financial statistic combinations (if they do not have access to credit score information).

When they do not have access to such information, they may aim to predict the level of the individual's credit score. This is reflected in the dataset given and the prediction problem given. The problem given is to predict a level of credit score (Poor, Standard, or Good) given input features. This turns the problem into a multi-class classification problem, some machine learning techniques will be altered in their impact because of this.

There are a high amount of features at the start within the data. These features range from information on the individual, the amount of credit cards and bank accounts they have, annual income and outstanding debt, to the number of delayed payments and days of delay for these payments. Other statistics such as interest rate and credit history are also recorded within the dataset, spanning 100,000 observations.

Features were heavily cleaned and narrowed down after data processing, sizing down the dataset to about 31,000 observations. Multiple machine learning methods were used for feature selection and model-building. Random Forest was used in backwards feature selection (alongside Mutual Information scores as an alternative feature selection method), and various kinds of methods were used for model building (Random Forest Classification, Gradient Boost Classification, XGBoost Classification, Multilayer Perceptron Classification). Confusion matrices and Classification Reports were created specifically with Random Forest Classification in two sets: a train-test split, and a train-test split involved using SMOTE to rebalance the amount of observations in each class.

Afterwards, SHAP was used to help explain the findings of the results, inherent to the features chosen by feature selection. In particular, the beeswarm plot outlined a recurring relation that could be useful to lenders in identifying long-term customers.

Other studies have been done regarding credit score classification and the use of various interpretation algorithms (LIME or SHAP), which have found tangible results and effective use of these algorithms.

3. Literature Review

Literature referenced for the writing of the paper involved an overview of why classification analysis of credit score (and thus credit risk) is becoming increasingly important, previous research done on this topic, and research into results and explanation of interpretation algorithms. In the early stages of research, LIME was a point of interest in terms of model interpretation. However, focus soon shifted to SHAP given the results within the literature and the ease-of-use plot creation built into the SHAP package.

According to Moscato et al., revenue for global payments is increasing through "peer-to-peer" social lending platforms, in which banks and clients can complete transactions without the medium of financial institutions. There are various statistics regarding these platforms. In short, the revenue and usage of these platforms has increased within the past decade. Their rise has posed several issues, such as "the high dimension, sparse and imbalanced data" and the "large amount of unlabeled data that require online analysis for supporting lenders' real-time decisions [1]. The high amount of complexity and prerequisite preprocessing for such data is costly. It also makes statistical methods improper, since they struggle to cover non-linear relationships.

Different machine learning methods were attempted to compare results (alongside being combined with various resampling techniques). In the case of Moscato et al. Logistic Regression, Multi-Layer Perceptron, and Random Forest were used (see table 7 within their paper for results) [1]. Due to the dataset used for this paper involving multi-class classification, logistic regression can not be used reliably.

Various resampling techniques have been used across research of credit score and credit risk assessment. A table can be found in Chen et al. of several related studies done with their interpretation methods, sampling methods, and more. Resampling to avoid class imbalances avoids several kinds of errors within model making. As stated by Chen et al., such errors can happen in "Logistic Regression. King & Zeng (2001a,b) theoretically and empirically showed that the estimation bias of coefficients in Logistic Regression could be greatly magnified by class imbalance. Owen (2007) also suggested that, in the case of extreme class imbalance, the minority class only contributes to the Logistic Regression estimation via its sample mean vector, and this issue cannot be solved by using penalisation or likelihood weighting" [2].

Oversampling and undersampling were the most widely considered within the table provided by Chen et al., with some hybrid sampling. While these approaches are strong, SMOTE (hybrid sampling) was chosen due to the context of the dataset used for research. SMOTE is more useful for the heavy class imbalance found within the dataset after preprocessing, with a significant number of observations making up the "Standard" credit score category.

With machine learning models becoming increasingly complex, especially with the high-dimensionality of the dataset. Two different explainer algorithms were assessed during research of literature: LIME and SHAP. As explained by Aljadani et al., LIME (short for Local Interpretable Model-Agnostic Explanations) approximates local results through random sampling. Then, local linear models or decision trees that similarly reflect the results are created, mitigating the mystique of the “black box” nature of complex models in regard to credit score classification [3].

SHAP works differently than LIME, but has the same end goal in model explainability and interpretation through Shapley values for each observation and each class. As described by Gramenga et al., Shapley values are “expressing model predictions as linear combinations of binary variables that describe whether each covariate is present in the model or not” [4]. They find within their results that “SHAP values seem to constitute an input space more suitable to be divided into clusters, with a clear advantage in discriminative power in this unsupervised setting” [4]. This is more suitable to the multi-class classification problem, since SHAP can work with the high-dimensionality data effectively, identify data as part of “clusters” (classes), and give weights for each prediction in the set, reflective of each observation’s predictive probability.

4. Methodology

Preprocessing the data posed several challenges. As mentioned previously in the literature review, data generated by peer-to-peer transactional platforms tend to be complex, unstructured, and have high amounts of missing values. One challenge that came up is the data types of each attribute. Due to random characters inserted in columns and underscores being used to represent missing values, a lot of the attributes end up being an object type. Float and Int types are most usable. Additionally, a lot of values are not just outliers, but unrealistic and unusable entries (negative values, age of a person being over 1000). These were removed for clarity and also to reduce the standard deviation in each feature.

After preprocessing, there are no missing values in the dataset, and the dataset has sized down to 31,000 observations. One issue is that there is a heavy class imbalance in the data after preprocessing, as the majority of the credit scores are classified under “Standard” credit. Multiple categorical features were encoded using ordinal and label encoders for proper use.

New features were also engineered through combinations of pre-existing features. Refer to table 1 for the finalized features list. The engineered features were chosen to combine existing features to increase parsimony (total monthly payments, total number of accounts), generate debt specifics (debt per account, number of delayed payments per account) and debt to income ratio (outstanding debt divided by annual income). Debt to income ratio in particular is a useful statistics in viewing if someone is in debt that is difficult to pay off. Debt to income ratio over 1 means there’s over a year’s worth of the individual’s income in debt, less than 1 is the opposite case. This gives context to the amount of outstanding debt by tying it to the individual’s ability (or lack thereof) to pay it off.

Multiple machine learning methods and feature selection methods were tested. Between the two feature selection types used and the toggle of Cross Validation, four model combinations were produced. A seed was generated early on throughout the code to generate similar results between runs. More feature selection types and model variants would make the analysis more thorough and validated, and could be pivotal in future work.

First for feature selection is the use of SequentialFeatureSelector with RandomForestClassifier used as a wrapper in backward feature selection method, done in

Pandas. The number of estimators was set low (20) due to the high amount of computational runtime Random Foresting takes). A hard cutoff of 5 features was selected to parallel the choice with the second feature selection method, as well as compensating for runtime, as letting the algorithm choose a variable amount of features significantly increases runtime.

Second for feature selection is the use of Mutual Information, or MI scores. MI scores indicate how useful each feature is to a model overall. A hard cutoff of MI scores lower than 0.2 was used after looking at the results of the MI scoring (see figure 1 for the plot output), as there is a significant fall-off in MI scores after the 5th feature.

Multiple machine learning methods were tested with the dataset. A train-test split was done to separate the data randomly before model-building, at about a 70:30 split for training and testing sets respectively. Random Forest Classification, Gradient Boost Classification, XGBoost Classification, and MLPClassifier were tested. These were chosen for an overview of many different methods, and in specific Neural Networks (MLPClassifier) for its extremely efficient computational runtime.

SMOTE was then used to resample the dataset. This served not only to handle class imbalancing (as the Standard class had the majority of observations, refer to the Classification Report within the appendix), but also generate new data points. These new data points can help test how efficient a model is against new data points.

SHAP is a model explainer and interpreter based on Game Theory created by Lundberg and Lee [5]. Here, it is used with XGBoost with the multi-softmax parameter to handle multi-class classification. By creating SHAP values for each observation and each class within, the SHAP values for each observation reflect how much influence each predictor has on the observation. There are plots used that give an overview of which features are the most important across all observations. These plots are discussed later on within the discussion section.

5. Results

Between the two types of feature selection and the Cross Validation toggle, there are 4 sets of results. Between all runs, the results will be listed within parentheses for all available run parameters with specifications.

The two forms of feature selections result in a different set of features selected. With backwards feature selection, the following features are chosen. For MI Score, the following 5 features are selected: 'Annual_Income', 'Debt_to_Income_Ratio', 'Outstanding_Debt', 'Debt_Per_Account', and 'Total_EMI_per_month'.

These feature choices are consistent, but the choices for backwards feature selection vary from run to run. In the results listed, the following features chosen with backwards feature selection and without Cross Validation are 'Num_Credit_Card', 'Delay_from_due_date', 'Num_of_Delayed_Payment', 'Amount_invested_monthly', and 'Credit_Mix_Encoded'.

A baseline basic model using Pandas is fitted with no extra parameters and is built alongside the Random Forest model without Cross Validation. The baseline MSE comparison scores are as follows: (0.366 Baseline MSE, 0.299 Backwards Feature Selection Random Forest

MSE, 0.171 MI Score Feature Selection). Both models decrease the baseline MSE, with MI Score Feature Selection offering the lowest MSE.

With Cross Validation, RMSE and Explained Variance (EV) are calculated. The scores are as follows: (0.51 RMSE, 0.29 EV, Backwards Feature Selection), and (0.42 RMSE, 0.52 EV, MI Score Feature Selection). Explained Variance is significantly higher with MI Scores, which gives the model more flexibility compared to backwards feature selection. RMSE is still middling with either model, but MI Score still offers a decrease in RMSE.

Scores between GradientBoostClassifier and MLPClassifier varied slightly throughout available toggles. However, one issue arises with these machine learning methods in particular: their AUC values will always return NaN. This is due to the classification problem being multi-class. These methods can be coerced, set up in a way to generate AUC scores for multi-class classification problems, but these scores are faulty.

Scores for MLPClassifier accuracy are as follows: (0.69 without Cross Validation, 0.72 with Cross Validation, Backwards Feature Selection), and (0.62 without Cross Validation, 0.62 with Cross Validation, MI Score Feature Selection). MLP performed best with Backwards Feature Selection and Cross Validation. It’s the best available combination with Backwards Feature Selection.

Scores for GradientBoostClassifier accuracy with Cross Validation are as follows: (0.73 with Backwards Feature Selection, 0.67 with MI Score Feature Selection). Overall, boosting and neural networks seem to perform better with backwards feature selection. However, due to the AUC statistic being faulty in multi-class classification, these machine learning methods are lackluster choices.

Random Forest as a model performs the best of the results with the MI score features, and has the most statistics available. Without Cross Validation, there is higher concern of overfitting, as the accuracy on the original set is extremely high (>99%), but 75% when involving SMOTE data. As such, Cross Validation Random Forest with MI score feature selection, with about 93% accuracy on the training set, and 85% on the SMOTE test set.

A Confusion Matrix and a Classification Report was generated for these model settings (MI Score feature selection, Cross Validation on). See figure 2 and figure 3 within the appendix for results. These matrices overall cover a majority of the cases, but there’s still a fair amount of misclassification. This is particularly notable in the standard credit class, despite the rebalancing. In the classification report, we see lower precision scores and higher recall scores for the base test set. Therefore, it struggles to predict “negative” cases (false negatives and true negatives in a multiclass scenario). This translates to the standard class having the lowest precision score in the SMOTE set.

6. Discussion

With all the results now listed, some elaboration can be given. With real world context, the numbers don't seem justifiable as good results. It's not to the scale of severity as a false cancer diagnosis, as even one life lost/falsely treated is one too many, and letting such a model slide as a primary basis for diagnosis is unethical. It's not quite the same, but a misclassification on assessment could lead to a huge risk financially, possibly jeopardizing the lender if they overestimate someone's credit health (a false positive in this multi-class case, where someone's credit health is listed as a higher class than it actually is).

The choice in models didn't have much foresight. The boosting and neural network methods all posed significant weaknesses. Random Forest is quite effective, although it did cause overfitting in some cases. XGBoost did find a use through SHAP however. As discussed in the literature review, SHAP helps explain and interpret models. Included in the appendix are two plots, a beeswarm plot and a mean bar plot. Each feature delivers some impact on the model, but primarily debt per account and outstanding debt have the most influence.

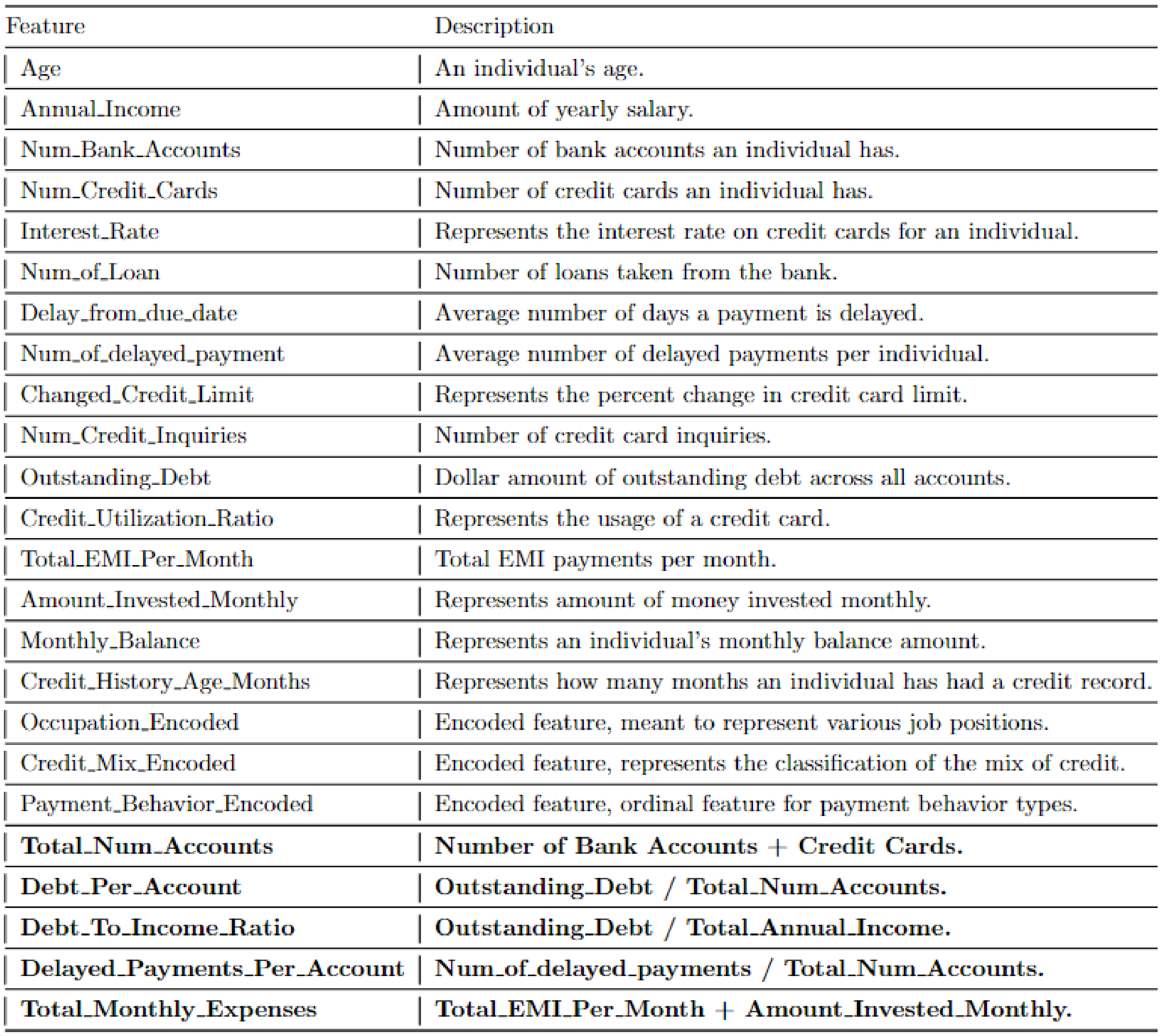
When looking at these two features on the Beeswarm plot, there's a unique relation that can be observed. There is a range between 0 and 1 on the x-axis (SHAP value, impact on model output) where the outstanding debt is high, but the debt per account is low. This means as outstanding debt has a higher impact on the model, the amount of it is higher. This checks out; but the unique detail is that within this range, there are a low amount of accounts per individual. This suggests a group of customers who have a high amount of debt centralized in one account that they are struggling to pay off, rather than multiple instances of debt intentionally taken on. This is further supported by the feature value being low in this same range for annual income, which reveals a probable middle class which can be exploited to be in a perpetual loop of lending, unable to make major changes to their financial situation in the long-term.

7. Conclusion and Future Work

The research objective on this dataset is to help predict credit score classification for taking on credit risk. Individuals are assessed into one of 3 categories of credit health: Poor, Standard, and Good. Multiple models are built on this dataset after heavy preprocessing and variations of feature selection and Cross Validation combinations. The most effective result found within this analysis is Random Forest with MI score feature selection and Cross Validation, with about a 93% accuracy on the test set, 85% accuracy on the SMOTE set, 0.42 RMSE and 0.52 Explained Variance. The SMOTE classification report is found to be weakest with the standard class, which also had a class imbalance skewed towards it by a heavy amount of observations.

Plenty of optimizations and alternate routes to model building could be used in future work. Grid Search could be used to optimize the parameters of each model, potentially boosting Random Foresting results. Other machine learning methods could be considered as well, such as Decision Trees and Naive Bayes. A more expansive literature review may help give necessary context. Additionally, SMOTE splits and Cross Validations can also be tweaked, to see if results still hold up with further validation and more data given to the SMOTE split.

Appendix



Feature	Description
Age	An individual's age.
Annual_Income	Amount of yearly salary.
Num_Bank_Accounts	Number of bank accounts an individual has.
Num_Credit_Cards	Number of credit cards an individual has.
Interest_Rate	Represents the interest rate on credit cards for an individual.
Num_of_Loan	Number of loans taken from the bank.
Delay_from_due_date	Average number of days a payment is delayed.
Num_of_delayed_payment	Average number of delayed payments per individual.
Changed_Credit_Limit	Represents the percent change in credit card limit.
Num_Credit_Inquiries	Number of credit card inquiries.
Outstanding_Debt	Dollar amount of outstanding debt across all accounts.
Credit_Utilization_Ratio	Represents the usage of a credit card.
Total_EMI_Per_Month	Total EMI payments per month.
Amount_Invested_Monthly	Represents amount of money invested monthly.
Monthly_Balance	Represents an individual's monthly balance amount.
Credit_History_Age_Months	Represents how many months an individual has had a credit record.
Occupation_Encoded	Encoded feature, meant to represent various job positions.
Credit_Mix_Encoded	Encoded feature, represents the classification of the mix of credit.
Payment_Behavior_Encoded	Encoded feature, ordinal feature for payment behavior types.
Total_Num_Accounts	**Number of Bank Accounts + Credit Cards.**
Debt_Per_Account	**Outstanding_Debt / Total_Num_Accounts.**
Debt_To_Income_Ratio	**Outstanding_Debt / Total_Annual_Income.**
Delayed_Payments_Per_Account	**Num_of_delayed_payments / Total_Num_Accounts.**
Total_Monthly_Expenses	**Total_EMI_Per_Month + Amount_Invested_Monthly.**

Table 1: List of finalized and engineered features. Note all dollar amounts are represented in USD. Engineered Features are marked in bold.

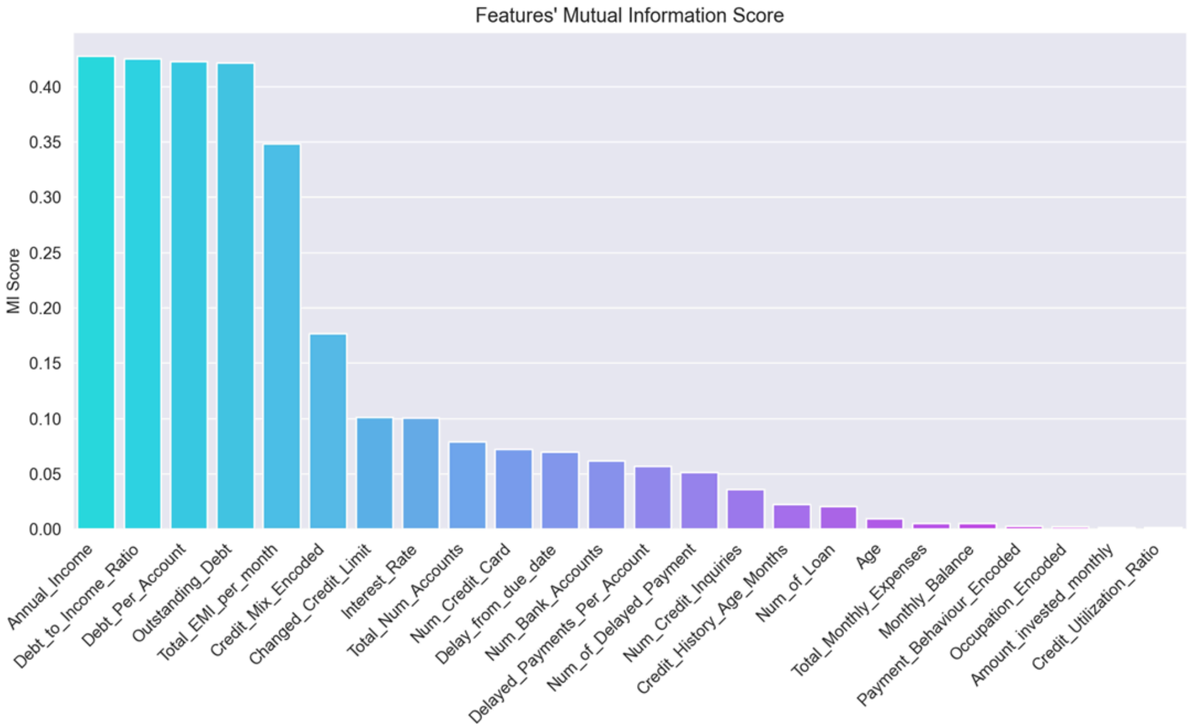


Figure 1. Mutual Information (MI Score) plot output made using Seaborn in Python.

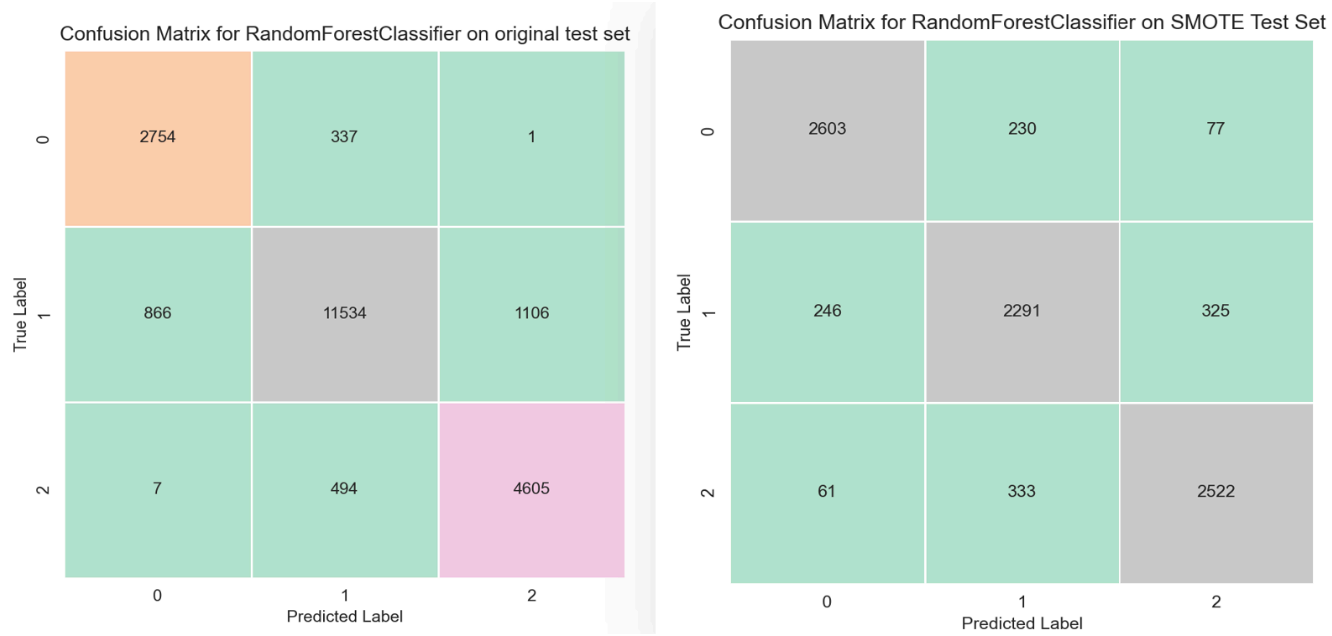
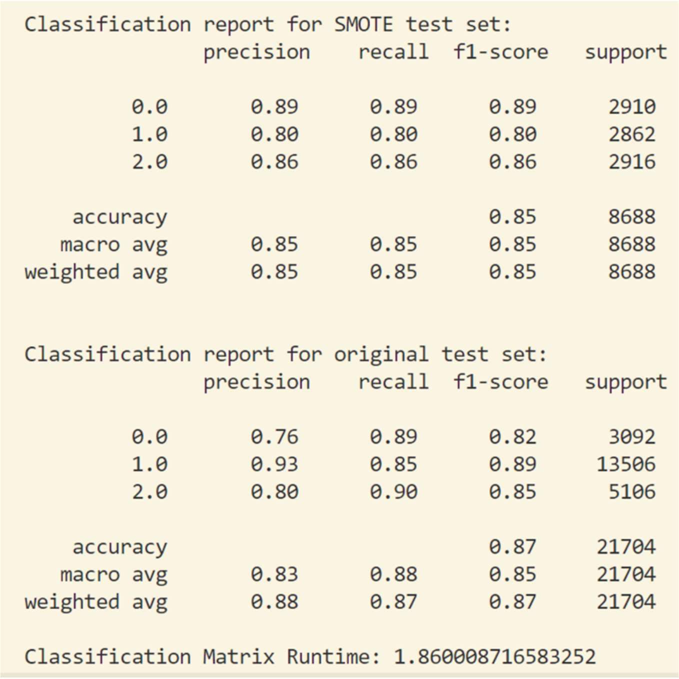


Figure 2. Confusion Matrices created with RandomForestClassifier for the original test set and the SMOTE test set, respectively.



```
Classification report for SMOTE test set:
              precision    recall  f1-score   support

         0.0       0.89      0.89      0.89      2910
         1.0       0.80      0.80      0.80      2862
         2.0       0.86      0.86      0.86      2916

    accuracy                           0.85      8688
   macro avg       0.85      0.85      0.85      8688
weighted avg       0.85      0.85      0.85      8688


Classification report for original test set:
              precision    recall  f1-score   support

         0.0       0.76      0.89      0.82      3092
         1.0       0.93      0.85      0.89     13506
         2.0       0.80      0.90      0.85      5106

    accuracy                           0.87     21704
   macro avg       0.83      0.88      0.85     21704
weighted avg       0.88      0.87      0.87     21704

Classification Matrix Runtime: 1.860008716583252
```

Figure 3. Code output for Classification reports on the SMOTE and original test sets, respectively.

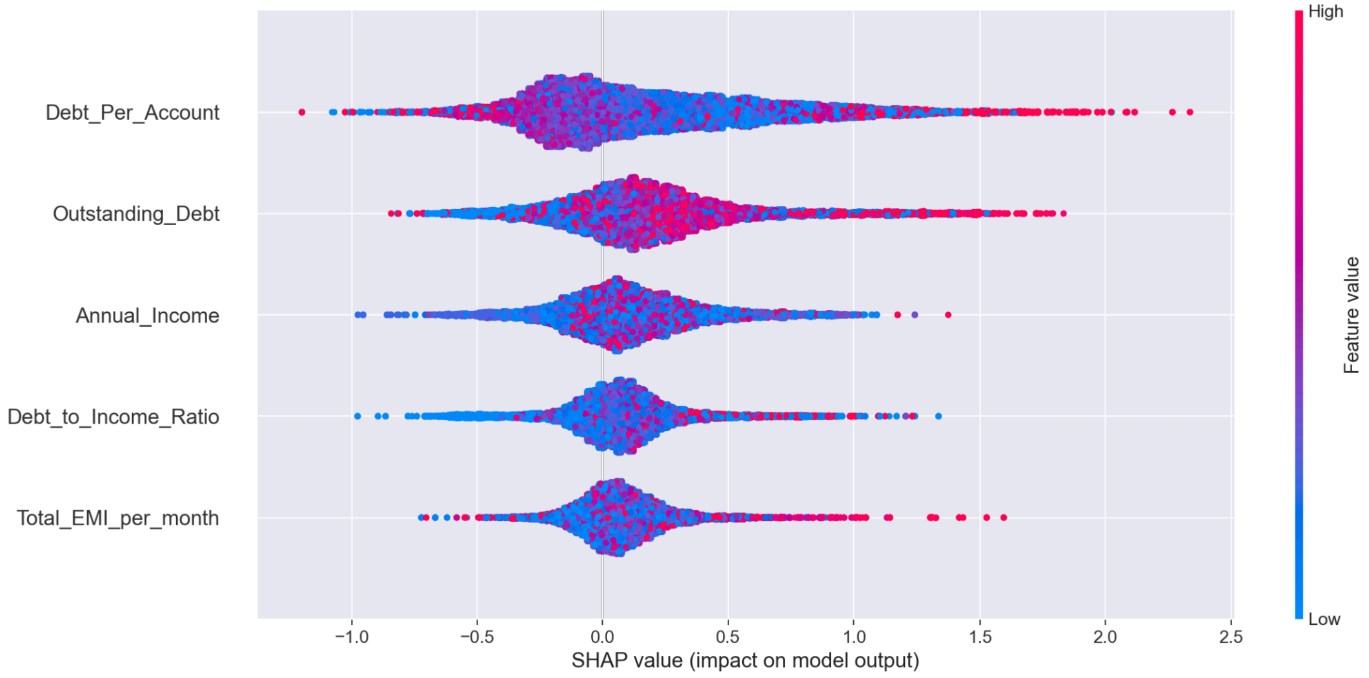


Figure 4. SHAP Beeswarm plot output for the features chosen by Mutual Information scores with Cross Validation.

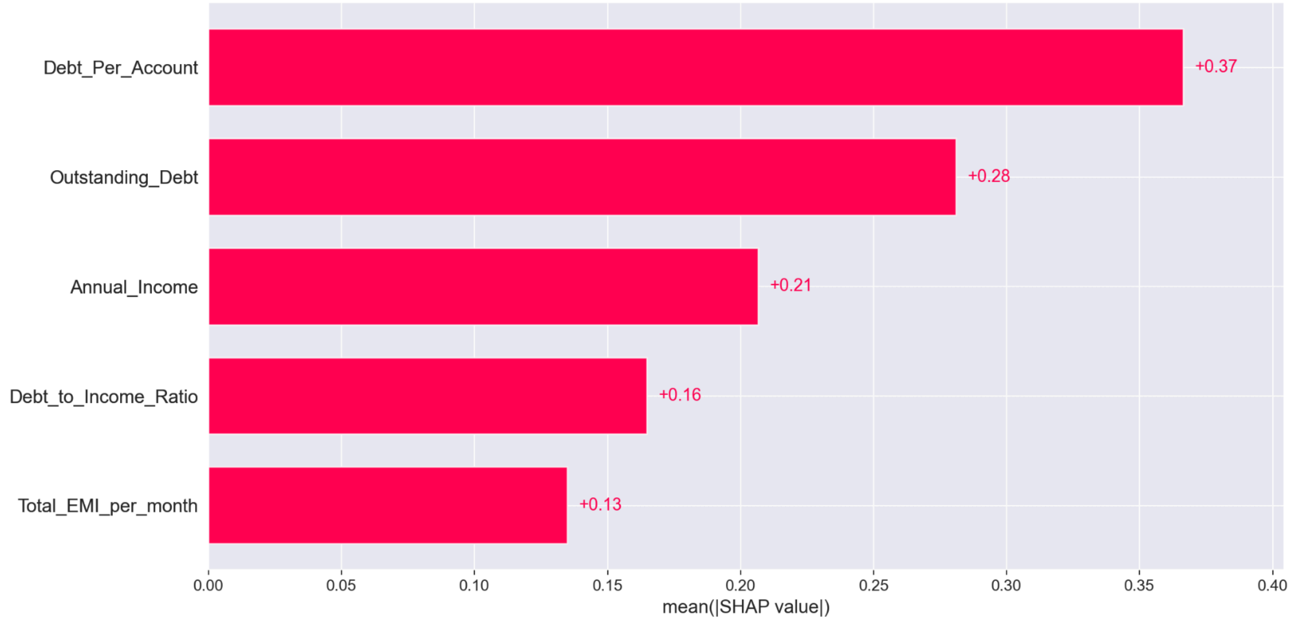


Figure 5. SHAP mean contribution (across all observations) bar plot output for the features chosen by Mutual Information scores with Cross Validation.

References

1. Moscato, Vincenzo, et al. “A Benchmark of Machine Learning Approaches for Credit Score Prediction.” *ScienceDirect*, Department of Electrical Engineering and Information Technology (DIETI), University of Naples, 9 Sept. 2020, www.sciencedirect.com/science/article/pii/S0957417420307636.
2. Chen, Yujia, et al. “Interpretable Machine Learning for Imbalanced Credit Scoring Datasets.” *European Journal of Operational Research*, North-Holland, 23 June 2023, www.sciencedirect.com/science/article/pii/S0377221723005088
3. Aljadani, Abdussalam, et al. “Mathematical Modeling and Analysis of Credit Scoring Using the Lime Explainer: A Comprehensive Approach.” *MDPI*, Multidisciplinary Digital Publishing Institute, 25 Sept. 2023, www.mdpi.com/2227-7390/11/19/4055
4. Gramegna, A., & Giudici, P. (2021, August 30). *Shap and lime: An evaluation of discriminative power in credit risk*. Frontiers. https://www.frontiersin.org/articles/10.3389/frai.2021.752558/full
5. Lundberg, S., & Lee, S.-I. (2017, November 25). *A unified approach to interpreting model predictions*. arXiv.org. https://arxiv.org/abs/1705.07874