Default Risk Prediction using Banking Transaction Data

Jessica Guzman jeguzman@ucsd.edu
Haicheng Xu hax004@ucsd.edu
Kuangyu Zou kzou@ucsd.edu
Xuewen Yang xuy001@ucsd.edu
Brian Duke brian.duke@prismdata.com
Berk Ustun berk@ucsd.edu

Abstract

Within the realm of financial lending, the FICO score has long served as the primary tool for assessing potential borrowers. In this paper, we suggest leveraging the capabilities of machine learning alongside borrowers' transactional data to forecast the likelihood of customer default. Specifically, income, balance, and transaction category emerge as pivotal factors correlated with credit risk for each individual. This paper aims to construct a model that integrates all these features and combinations therein to anticipate whether a customer is likely to default on their banking transactions.

Keywords: inflow/outflow/transaction data, income estimation, cumulative balance trend, feature engineering, default score

Code: https://github.com/daphneyyy/dsc-capstone-project

1 Introduction

1.1 Introductory

In the current dynamic economic landscape, financial institutions are increasingly seeking methods to broaden their customer base while maintaining a cautious approach towards risk. When it comes to credit risk analysis in particular, financial institutions tend to have access to multitudes of important user data, but using it to accurately and efficiently predict risk can become difficult based on the data structure. In this project, we will focus on combining customers' income, balance, and transaction category information to create features that help the model predict the default probabilities for each customer. To get the most accurate income estimation, we take into account all possibilities for paychecks by basing inflow on a given definition of regularity. For the balance feature, we calculate the cumulative balance sum, standardize these values and use the balance regression coefficients as well as moving averages. For the transaction features, we calculate the customer's percentage spent by both the inflow and outflow categories. Additionally, we also define a feature that correlates to the account type since different account type has different inflow and outflow amounts per customer. By further refining the important features, we use roughly the most important 40 features to put into the XGBoost model to predict the probability of a customer's default, along with the top three reasons why each user would default.

1.2 Literature Review and Discussion of Prior Work

The first piece of literature we reviewed is "Looking at credit scores only tells part of the story – cash flow data may tell another part" by Alexandrov, Brown and Jain (2023). In this article, the authors claimed while credit scores are widely used to assess creditworthiness, they provide only a partial view of a person's financial health. Credit scores primarily focus on credit history and repayment behavior but do not consider factors such as income and expenses. This led to the idea of creating a cash score - a score determined by a consumer's cash flow. The article suggests three ways of measuring cash flow: regularly saving and no overdrafts, paying bills on time, and high accumulated savings. We used these ideas as starting points for constructing our cash score estimation.

1.3 Data Description

1.3.1 Inflow Data

We utilized a pre-clean dataset containing 507,943 data entries of inflow data and a total of six columns: customer ID, account ID, inflow transaction information, inflow amount, the date for the inflow transaction, and the inflow category. This data is from Prism's internal database provided by our mentors. Examples are shown in Table 1.

In the "memo" column of the inflow data, there are digit-character combinations represent-

Table 1: Unclean Inflow Data

	prism_consumer_id	prism_account_id	memo	amount	posted_date	category_description
23	0	acc_0	ATM CHECK DEPOSIT 08/06 XXXX BAILEY RD CUYAHOG...	537.50	2022-08-08	DEPOSIT
52	0	acc_0	ATM CHECK DEPOSIT 09/10 XXXX BAILEY RD CUYAHOG...	524.48	2022-09-12	DEPOSIT
79	0	acc_0	Zelle payment from CODY CRANO CTZ0H8QOXXXX	50.00	2022-12-12	EXTERNAL_TRANSFER
151	2	acc_3	ATM CHECK DEPOSIT ON 04/08 213 STEPHANIE STE...	19.00	2021-04-08	DEPOSIT
161	2	acc_3	ATM CHECK DEPOSIT ON 10/25 213 S STEPHANIE ST...	320.00	2021-10-25	DEPOSIT

ing confirmation numbers of transactions, dates and times, etc. We blurred transaction data by replacing all digit-related phrases with "X" to ensure confidentiality and consistency. After cleaning the data, we gained a cleaner and more consistent view of the memo where examples are in Table 2. We will need the clean transaction information for our future data analysis.

Table 2: Clean Inflow Data

	prism_consumer_id	prism_account_id	memo	amount	posted_date	category_description
23	0	acc_0	ATM CHECK DEPOSIT X/X XXXX BAILEY RD CUYAHOG...	537.50	2022-08-08	DEPOSIT
52	0	acc_0	ATM CHECK DEPOSIT X/X XXXX BAILEY RD CUYAHOG...	524.48	2022-09-12	DEPOSIT
79	0	acc_0	Zelle payment from CODY CRANO X	50.00	2022-12-12	EXTERNAL_TRANSFER
151	2	acc_3	ATM CHECK DEPOSIT ON X/X X STEPHANIE STE...	19.00	2021-04-08	DEPOSIT
161	2	acc_3	ATM CHECK DEPOSIT ON X/X X S STEPHANIE ST...	320.00	2021-10-25	DEPOSIT

1.3.2 Outflow Data

The outflow data has data of size 2,576,829 and a total of six columns which are customer ID, account ID, outflow transaction information, outflow amount, the date for the outflow transaction, and the outflow category. This data is provided from Prism's internal database provided by our mentors. Examples are shown in Table 3.

Table 3: Outflow Data

	prism_consumer_id	prism_account_id	memo	amount	posted_date	category_description
62	0	acc_0	Online Transfer to CHK...XXXX transaction#: X...	280.00	2023-01-06	SELF_TRANSFER
818	0	acc_0	Walmart	121.04	2022-04-18	GROCERIES
819	0	acc_0	Kroger	25.67	2022-05-16	GROCERIES
820	0	acc_0	AFTERPAY 185-XXXXXXXX CA 02/09	8.19	2023-02-10	GENERAL_MERCHANDISE
821	0	acc_0	Walmart	45.97	2022-11-21	GROCERIES

1.3.3 Account Data

The account data has data of size 4,696 and a total of five columns which are customer ID, account ID, account type, final balance amount, and final balance date when they are evaluated by the bank. This data is provided from Prism's internal database provided by our mentors. Examples are shown in Table 4.

1.3.4 Previous Evaluation Data

The evaluation data has data of size 2,978 and a total of four columns which are customer ID, the Approved column indicating whether the customer was approved by the banking

Table 4: Account Data

	prism_consumer_id	prism_account_id	account_type	balance	posted_date
0	0	acc_0	SAVINGS	6182.60	2023-04-13
1	0	acc_1	CHECKING	9907.23	2023-04-13
2	2	acc_12	SAVINGS	17426.83	2022-02-15
3	2	acc_11	CHECKING	8079.43	2022-02-15
4	4	acc_16	SAVINGS	0.00	2021-08-13

company for low risk, and the FPF_TARGET column indicating whether the customer will default the money. In the FPF_TARGET column, 0.0 means the customer does not default on money, and 1.0 means the customer defaults on money. Examples are shown in Table 5.

Table 5: Evaluation Data

	prism_consumer_id	evaluation_date	APPROVED	FPF_TARGET
8	0	2023-04-13	1	0.0
43	2	2022-02-15	1	0.0
49	4	2021-08-13	1	0.0
69	7	2021-08-08	1	0.0
77	9	2023-04-19	1	0.0

1.3.5 Data Summary

These five datasets contain crucial information for us to define the income, balance, and transaction categories features. We can get inflow and outflow amounts from certain categories and the date for these transactions from both inflow and outflow datasets. We can get the balance amount for each account per customer to calculate the cumulative balance sum over time for each customer and also the account type features from the account dataset. We can also get our predicted value which is the binary column FPF_TARGET from the evaluation dataset. However, it's important to acknowledge certain limitations within the dataset. The dataset lacks information about the geographic locations where transactions occurred, limiting our ability to capture comprehensive spending patterns. Spending and income levels may vary across different locations. Despite these limitations, we aim to derive meaningful insights from the available data, understanding that certain factors, such as geographical context and income disparities, may not be fully represented.

2 Methods

2.1 Feature Creation

Our prediction model heavily depended on the features derived from the datasets. This section outlines the methodologies employed to generate these features, crucial for the efficacy of our predictive modeling framework.

2.1.1 Income Estimation

We first approached this task by determining which categories of inflow transactions should be considered income. Two categories, "paycheck" and "placeholder_paycheck", were determined to always be considered income, by the nature of the category description. For some particular categories, 'deposit', 'external_transfer', 'investment_income', 'unemployment_benefits', and 'miscellaneous', we determined certain transactions could possibly be income. We split our data into two sub-datasets: paycheck_inflow and relevant inflow (rel_inflow) to make this distinction. Transactions in relevant inflow could be considered income if they were proven to be consistently coming into the user's accounts. We define consistency for this task as time recurrent and/or amount recurrent transactions.

Before we determined our exact logic for establishing recurrence, we ensured there was a way to measure the change of time for transactions. We engineered an "age" feature for each transaction by determining the number of days elapsed since the earliest date of a transaction within the same category for the respective user. By considering the category when calculating age, we can further ensure we are calculating the recurrence of transactions from the same income source.

We first examined patterns in all transaction ages. Considering a lot of paychecks are disbursed biweekly and monthly, we wanted to see how well this idea represented the distribution of ages. Below we plotted the distribution of ages mod 30(Figure 1) and mod 7(Figure 2).

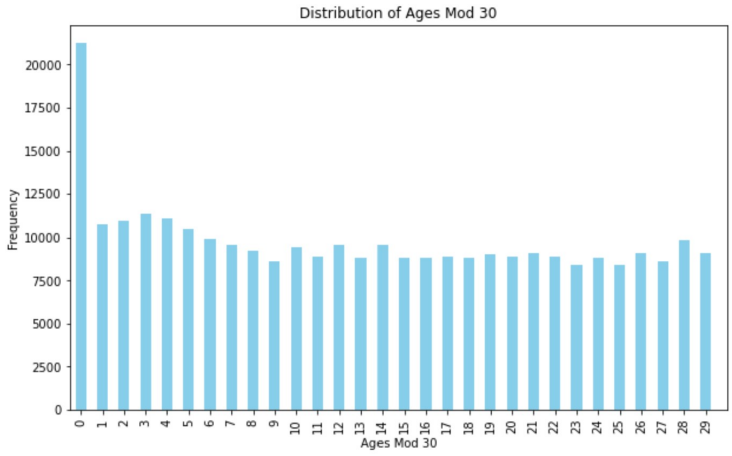

Figure 1: Age Mod 30 Distribution

The fact there is a significant number of transactions that fall under the 0 value means the transactions are occurring every 30 days (or on multiples of 30) or every 7 days(or multiples of 7). However, since the task is to estimate income at the user level and many transactions don't fall exactly on this pattern, we decided to take another approach.

We then began to examine recurrence at a user level. All steps hereafter are applied by user. We first filtered our dataset for memos with more than one occurrence. Seeing a repeated memo within the same category indicated to us that these transactions were likely from the same source. Upon further examination, this turned out to not always be the case and we

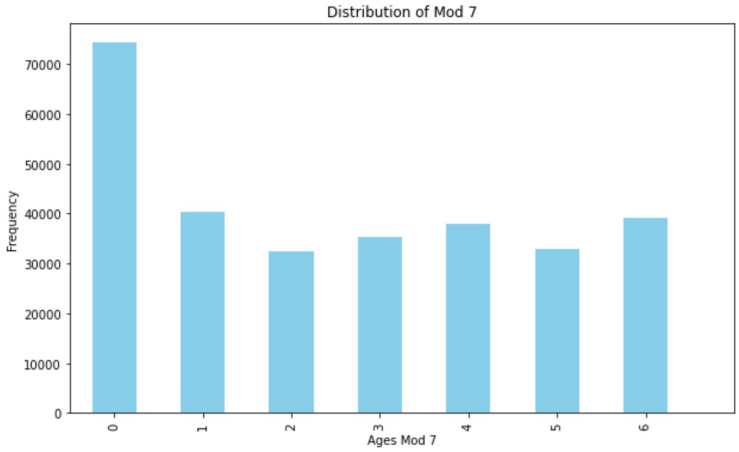

Figure 2: Age Mod 7 Distribution

further split the memos into two groups: one where all the transaction amounts are close to the median amount of the group, and another where the amounts are not close to the median. We defined "close" to be amounts within 10% of the given amount values. The logic behind this is that perhaps not all transactions with the same memo can be considered from the same source, but transactions with the same memo and similar amounts can be. For the first group, this is already checked and we move on to checking recurrence by applying the `regular_amount()` function. For the second group, we further split transactions into subgroups where the transaction amounts are close to the median amount of the sub-group, then apply the `regular_amount()` function for each sub-group. The transactions that are classified as regular income would be aggregated with the paycheck_inflow and become the income estimate for the customer.

The `regular_amount()` function relies on one particular metric, the mean time between successive transactions. We used this metric to check whether transactions grouped together were recurring for at least approximately one month's duration. Depending on the mean time found, then the requirement for the number of transactions was changed accordingly to meet this. Transactions that passed this test were flagged to be later considered part of income.

We also considered cases where the memo was not the same, but the amounts were repeated. This is meant to capture income from jobs where the source might not be consistent, but the amount itself is. This could include income from services provided for any type of work, freelance gigs, or sporadic projects where the payment remains constant despite variations in the job or client. For this case, transactions of each user were grouped by amount and were passed through the `regular_amount()` function as well to ensure recurrence.

2.1.2 Probability of Default Based on Percentage Spent of Inflows and Income by Outflow Category

Our approach began with aggregating the total inflow amount for each consumer, followed by calculating their total spending by categories. Subsequently, we computed the percentage of the inflow amount that each consumer spent on each outflow category.

Next, we employed a Logistic Regression model, fitting the percentage spent for each outflow category of each consumer alongside their corresponding `FPF_Target` value. Utilizing this model, we generated an initial estimation of probabilities for the positive class (default).

In addition to calculating percentage spent by outflow category over total inflow, we also calculated percentage spent by outflow category over estimated income. Using the income estimation method from 2.1.1, we assigned income estimates to every user and used those values as our denominator rather than the sum of total inflow. After that, we again used a Logistic Regression Model to predict the probabilities for the consumer to default on a loan.

2.1.3 Count of Account Type

To understand consumers' financial habits, we tracked the number of different account types they have. We started by grouping inflow transactions by consumer, account type, and account ID. Then, we tally the occurrences of each account type for every consumer. We organized this information into a table, which shows the count of each account type for each consumer.

This table, see Table 6, serves as a summary of the account distribution among consumers. Each row corresponds to a unique consumer ID, while the columns represent different types of accounts. The values in the table indicate the count of each respective account type associated with a particular consumer.

Table 6: Count of Account Type

prism_consumer_id	CASH MANAGEMENT	CD	CHECKING	CREDIT CARD	MONEY MARKET	PREPAID	SAVINGS
217	0	0	2	0	0	0	0
801	0	0	8	12	4	0	2
1356	0	0	2	0	0	1	2
1469	0	0	3	2	0	0	0
1890	0	1	0	0	0	0	1
1964	1	0	1	0	0	0	0

2.1.4 Balance and Difference in Balance Regression Coefficients

The next subject we analyzed for feature creation was account balances. While the data maps out the transaction details for all the users' accounts, the only direct balance information provided was the final account balance captured on the date of the users' loan decision. Because of this issue, we decided to calculate monthly balance summaries by user and account using the cumulative sum of positive inflow amounts and negative outflow amounts.

Once we had the users' final month calculated balance, we then subtracted this from the final balance snapshot mentioned previously. This operation provided us with the starting account balances for all users, which we could retroactively add to the beginning of our user transactions. Applying a monthly cumulative sum again yielded the accurate balance values.

In order to handle the variety in scale of user transaction amounts, we standardized the monthly balance by converting the balance values to z-scores. This allowed us to compare the user-specific balances relative to the standard deviation of their mean values. This way, we can focus on the trends present in balance variation by user and account.

Once we had done this, we applied linear regression to find coefficients representative of the trajectory of user balances. We note we applied this by account type rather than account id with the consideration that individual accounts trends are less significant than account type trends. This led to the creation of 7 features, each representing the regression coefficient for a different account type.

In addition to these features, we also wanted to create features based off of the difference in balance. In order to calculate this, we grouped transactions amounts by month and converted them to z-score values. After that, we applied linear regression to find the trend in balance differences across users and account types. Like before, this created 7 new features.

2.1.5 Moving Average of Standardized Balance

In order to estimate a more stabilized pattern of the users' balances across account types, we decided to also calculate the moving average of the standardized balance. Working with the standardized balances we had already calculated, we applied both the simple moving average (SMA) and exponential moving averages (EMA) with a window of 2 months. These moving averages are computed for each group of consumers and account types within the dataset. Subsequently, the data is transformed and aggregated to select the most recent seven months of transactions for each user. Thus, features are labeled in reverse chronological order with `month1` being the most recent month to `month7` being the seventh most recent month. In addition, based on the findings for most common account types, we focus particularly on specific account types: checking, savings, and credit card. This generated 42 new features for our model.

2.1.6 Inflow Percentage Features

First, we calculated the total inflow of each consumer by inflow categories. There were 14 inflow categories in total, but we omitted the "Unemployment Benefits" category for the rest of the calculations because we believed it was ethical to do so.

With the remaining 13 categories, we created 3 sets of features by dividing those categories by 3 different aggregates to transform these inflow categories features into percentage terms.

The first aggregate was total outflows per consumer, where we grouped by consumer id of the outflow dataset and summed up their outflows. We then merged the aggregated outflow data with the inflow category data and divided each category by their matching total outflow. This resulted in 13 features that showed the percent inflow per total spending for each customer.

The second aggregate was total inflow per consumer, where we grouped by consumer id of the inflow dataset and summed up their inflows. Using the same merging process, we created another 13 features that show the percent inflow per total inflow.

The last aggregate was the estimated income per consumer. We used the estimated income generated from the process described in section 2.1.1. Since some customers have an estimated income of 0 and dividing by 0 would result in infinity, we filled the zeros with the total inflow of the consumer. Using the same merging process, we created 13 features that show the percent inflow per total income.

In total, we created 39 features relating to the per category inflow for each consumer.

2.1.7 Outflow-to-Income Ratio Features

Additionally, we introduced a new set of features focusing on outflow over income. Following a similar process outlined in Section 2.1.6, we calculated the percentage of outflows for each expenditure category relative to the total income of each consumer.

This involved aggregating inflow transactions by consumer ID and summing their amounts. Subsequently, we merged this data with the income dataset, addressing cases where income is zero. Next, we aggregated outflow transactions by consumer ID and outflow categories. Finally, we generated the percentage of outflows over income for each category and consumer. This resulted in 28 features.

2.2 Feature Selection

In our feature selection process, we aimed to refine the feature set for optimal model performance. With a total of 132 features generated from previous methodologies, we employed various techniques to streamline and prioritize features. This involved using exclusion criteria to remove features with potential bias, such as those containing sensitive information related to healthcare, benefits, and dependents. The remaining features were then evaluated based on the proxy for feature importance selected for each model. In logistic regression and SVM models, we determined the significance of each feature by analyzing the coefficients assigned to them. Conversely, in the case of XGBoost, we directly utilized the feature importance function to assess the relevance of each feature. Afterwards, we employed cross-validation across diverse thresholds of feature importance. This iterative analysis facilitated the identification and retention of features that significantly enhance the predictive capacity of the model.

2.3 Model Fitting

2.3.1 Baseline model - Logistic Regression

The Logistic Regression is our baseline model for predicting the default result of customers. We use this model as a baseline model because the default result is a binary column, which is a classification task. Besides, our features contain non-numerical features including categorical and text features. Logistic regression is good at doing classification tasks, and it has high versatility which means it can be used with different types of independent variables, including continuous, categorical, and binary data.

The way to use this model is by doing the feature selection depending on the feature's importance that we describe in the Feature Selection Part above. We then use a stratified train-test split to get a new set of training data and testing data. We trained a new Logistic Regression model on the new training set with features selected above, and then used it in the new testing dataset to see model performance, including accuracy, AUC score, and classification report metrics.

2.3.2 SVM

After trying the Logistic Regression model, we try to use the Support Vector Machine model(SVM) to do our prediction task. The SVM model is also known for handling high-dimension spaces which makes them particularly suitable for applications where the number of features exceeds the number of samples. This capability is beneficial in fields such as text classification, where the data can be very high-dimensional. The way for using this model is the same as we used in Logistic Regression, which is selecting features based on their importance and running the model using our selected features. We also check the model performance using accuracy, AUC score, and classification report metrics.

2.3.3 XGBoost Classifier

Following the evaluation of logistic regression and SVM models, we turn our attention to XGBoost, an ensemble learning method renowned for its effectiveness in handling complex datasets and delivering high predictive performance. XGBoost stands out for its ability to handle a variety of data types, including categorical and numerical features, making it a suitable choice for our classification task involving default prediction.

The utilization of XGBoost involves a similar methodology to the previous models. We begin by conducting feature selection based on the importance of features derived from the XGBoost model. Following model training on the selected features, the XGBoost model is evaluated on the testing dataset to assess its predictive performance. Metrics such as accuracy, AUC score, and classification report metrics are utilized.

Furthermore, we utilized SHAP (SHapley Additive exPlanations) values to interpret the predictions of our models. SHAP values provide insights into the contribution of each fea-

ture to the model's predictions, enabling us to understand the rationale behind individual predictions and identify the most influential features driving the model's decision-making process.

3 Results

3.1 Classification Report - Logistic Regression

The logistic regression model's feature selection process identified a subset of **13** features out of the total available features. This selection was based on maximizing the area under the ROC curve (AUC), which resulted in an **AUC score of 0.824** for a training dataset. The **threshold** corresponding to the best AUC score is **0.42293391**.

Using this threshold and fitting a new logistic regression model on the selected features of a new training set, the **testing accuracy** of the model was found to be **81.89%**, and the **AUC score** on a new testing dataset was **0.806**.

Table 7 shows the classification report of the testing dataset.

Table 7: Logistic Regression Classification Report

	precision	recall	f1-score	support
0	0.82	0.99	0.90	798
1	0.63	0.09	0.16	185
accuracy			0.82	983
macro avg	0.73	0.54	0.53	983
weighted avg	0.79	0.82	0.76	983

3.2 ROC Curve - Logistic Regression

3.3 Classification Report - SVM

The SVM model's feature selection process identified a subset of **49** features out of the total available features. This selection was based on maximizing the area under the ROC curve (AUC), which resulted in an **AUC score of 0.799** for a training dataset. The **threshold** corresponding to the best AUC score is **0.03344907**.

Using this threshold and fitting a new SVM model on the selected features of a new training set, the **testing accuracy** of the model was found to be **81.18%**, and the **AUC score** on a new testing dataset was **0.794**.

Table 8 shows the classification report of the testing dataset.

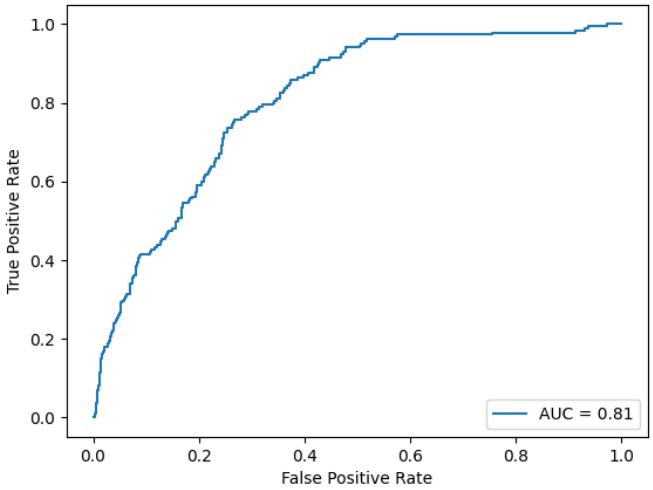

Figure 3: ROC Curve of Logistic Regression Model

Table 8: XGB Classification Report

	precision	recall	f1-score	support
0	0.81	1.00	0.90	798
1	0.00	0.00	0.00	185
accuracy			0.81	983
macro avg	0.41	0.50	0.45	983
weighted avg	0.66	0.81	0.73	983

3.4 ROC Curve - SVM

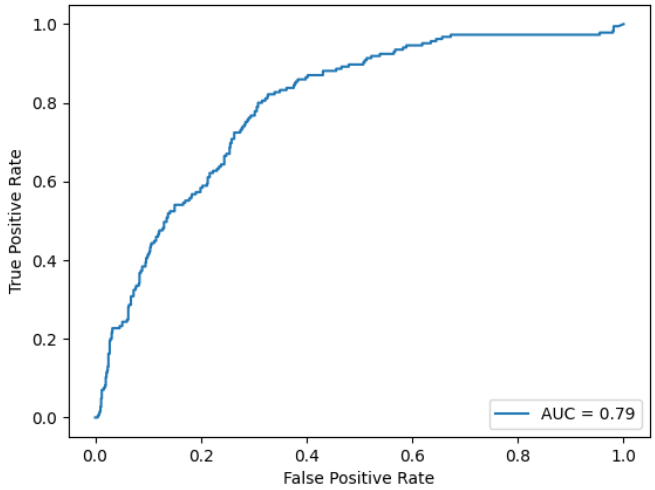

Figure 4: ROC Curve of SVM Model

3.5 Classification Report - XGB

The XGBoost model's feature selection process identified a subset of **35** features out of the total available features. This selection was based on maximizing the area under the ROC curve (AUC), which resulted in an **AUC score of 0.859** for a training dataset. The **threshold** corresponding to the best AUC score is **0.00978852**.

Using this threshold and fitting a new XGBoost model on the selected features of a new training set, the **testing accuracy** of the model was found to be **83.72%**, and the **AUC score** on a new testing dataset was **0.867**.

Table 9 shows the classification report of the testing dataset.

Table 9: XGB Classification Report

	precision	recall	f1-score	support
0	0.88	0.93	0.90	798
1	0.59	0.44	0.50	185
accuracy			0.84	983
macro avg	0.73	0.68	0.70	983
weighted avg	0.82	0.84	0.83	983

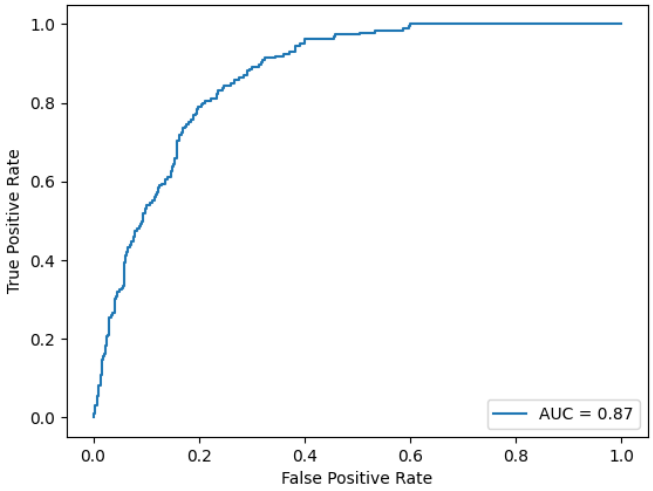

Figure 5: ROC Curve of XGB Model

3.6 ROC Curve - XGB

3.7 SHAP Value Analysis for Feature Impact on Predictions

We used SHAP value to determine the features that make customers risky. Then count the top three features for each customer, getting the result in the sorted bar graph, see graph 6.

Table 10: Top 10 of Most Common Reasons in Percentage

Feature	Percentage (%)
CREDIT_CARD_PAYMENT_outflow_over_income	33.93
Predictions_cat_proba	26.43
checking_month7_EMA	18.29
checking_month4_EMA	18.26
CHECKING_balance_std_diff_regress_coeff	16.68
EXTERNAL_TRANSFER_inflow_over_income	16.27
EXTERNAL_TRANSFER_inflow_over_inflow	15.70
checking_month5_SMA	14.66
MISCELLANEOUS_inflow_over_income	14.02
SMALL_DOLLAR_ADVANCE_inflow_over_outflow	13.82

4 Conclusion

4.1 Interpretation

Based on the results of our three models, all the accuracy scores of our models reached 80 percent or higher, indicating the models are performing well across both classes. This

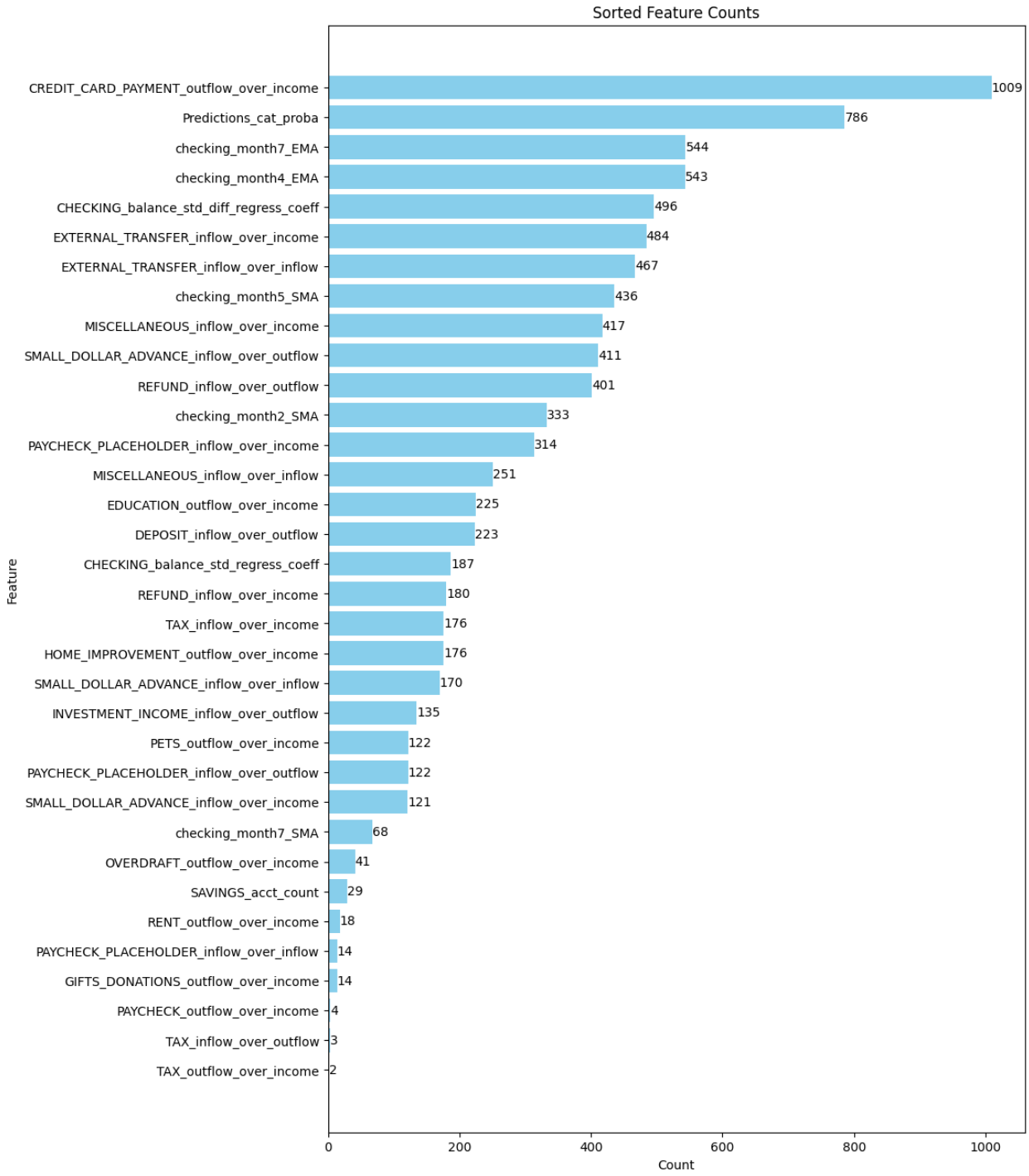

Figure 6: Count of Most Common Reasons

means that for a given set of data, our models can correctly identify whether a customer will default or not 80 percent of time. Additionally, based on the AUC score, the xgboost model has the best result. An AUC score, or Area Under the Receiver Operating Characteristic (ROC) Curve, of 0.87 indicates a high level of model performance in distinguishing between the positive class (e.g., customers who will default) and the negative class (e.g., customers who will not default). So there is an 87 percent chance that the model will be able to distinguish between a randomly chosen positive instance and a randomly chosen negative instance. This is considered to be a very good performance, indicating that the model has a high likelihood of correctly classifying customers who will default on loans versus those who will not.

4.2 Limitations

The model's performance in predicting class 1 is suboptimal, with precision, recall, and F1-score values of 0.59, 0.44, and 0.5, respectively. This limitation stems primarily from two factors: the composition of the data and the inherent characteristics of the model. In our training dataset, more than 80 percent of customers belong to class 0 (non-defaulters), while less than 20 percent are in class 1 (defaulters). Consequently, the model tends to prioritize optimizing precision and recall for class 0 at the expense of class 1.

4.3 Contributions Beyond

By leveraging transaction data from user accounts, we broaden the scope of potential borrowers we can cater to, all while striving for the most accurate distinction between sound and risky loans. Significantly, we also try to adhere to strict ethical standards, refraining from utilizing features that could lead to discrimination against protected classes. We also attempted to avoid indirect discrimination by normalizing and analyzing users' transaction histories against themselves. In summary, our model expands access to the market ethically, accommodating a broader range of users, while concurrently optimizing loan quality by emphasizing good loans and mitigating the risk of bad ones. Overall, our model promotes financial justice and ethical decision-making while guaranteeing inclusion without sacrificing the integrity of our lending operations.

References

Alexandrov, Alexei, Alyssa Brown, and Samyak Jain. 2023. "Looking at credit scores only tells part of the story – cashflow data may tell another part." July. [Link]