Assessing Probability of Consumer Default based on Historical Banking Transactions

Lakshmikethan Bethamcharla
lbethamcharla@ucsd.edu

Darren Jiang
dzjiang@ucsd.edu

Sheng Zhou
shz040@ucsd.edu

Victor Thai
v3thai@ucsd.edu

Brian Duke, Berk Ustun
brian.duke@prismdata.com, berk@ucsd.edu

Abstract

Throughout this project, our team is focusing on analyzing the process behind classifying credit card transactions and estimating consumer income to later determine a final credit score. This credit score would determine if a specific consumer is likely to default or not. Given datasets of inflows and outflows with categorical transactions, we focused on establishing a broad set of features to assess how risky a consumer is. For example, given categories amongst a user's outflow transactions, we extrapolated their average spending in non-essential categories to understand their spending habits and build towards a more holistic credit score. With the help of our industry partner Prism Data, our team's goal is to maximize our ROC-AUC on the amount of people likely to default and not. To achieve a high AUC-ROC, we utilized a robust feature building process that looked into the patterns of how consumers spend, save, and make money. For example, our income feature stems from summed recurring payments that came in within respective quarterly, monthly, or weekly segments. Overall, stacking features like these into our models helps us determine how good of an understanding we have on determining the difference between someone who doesn't default compared to those who do in order to come up with a final credit scoring for consumers.

Code: https://github.com/JiaDarren/DSC_180A

1 Introduction 2
2 Methods 8
3 Results 10
4 Conclusion 11

1 Introduction

Throughout this two quarter long task, our project's ultimate goal is to build a representative credit score for any given individual. We've split this task into two major objectives: categorizing the inflow and outflow transactions, and building the actual model and features for the risk predictor.

Quarter 1 was focused on classifying the category of credit card transactions for the inflow and outflow datasets provided to us. Each transaction contains information on the amount, date, and a brief text field providing a short description/note on what that transaction's purpose was intended for. This text field was the main focus on data engineering where we were aiming to find a way to categorize the transactions in the most correct manner. One example for this is shown when a customer purchases an item from Costco Wholesale, a leading retail company. This respective transaction provides a text field that describes where the purchase was made. Using this text description, we want to classify the potential outflow category that the transaction belongs to. However, taking a look into the initial dataset, this text field was very messy and had to be cleaned in order for us to separate our transaction into its true respective category. With such little text amongst each transaction it was also important to highlight key words to use in an overall categorization while also keeping in mind of any filler/distracting words that could alter the true categorization of a transaction.

To address these problems, we have applied the use of multiple methods such as the removal of unnecessary white-spaces, special characters, and phrases in the text field. Combining these cleaning methods we created a stratified and down-sampled model using Bidirectional Encoder Representations from Transformers, also known as BERT, that provided a model which resulted in a 88% accuracy. It is important to note that this high accuracy could be due to the stratification of our data from when we initially sampled it. Our team's next steps were to create some of our own models with our personalized hyperparameters to determine if we can come up with a similar accuracy as with stratifying our initial sampled data in compairson to using an established BERT model. We proceeded to create logistic regression and decision tree classifier models with TF-IDF as an additional feature vector. By using much simpler models that used tree based splits to come up with a categorization or even simple feature weights, these two models performed extremely well giving us an accuracy of about 97% and 98%, highlighting the impact TF-IDF has on our classification accuracy.

During Quarter 2, the major focus was on building features from the inflow and outflow datasets and creating models in order to predict a given consumer's likelihood for defaulting on their first loan payment. Many of our features looked at different facets of a consumer's transaction history, like the average monthly spending of each category, the balance of each account a consumer had at the time of evaluation, and an estimate of a consumer's income based on inflow transactions. Beyond feature creation, we've also implemented a basic linear regression and a more complex sequential neural network which allows the model itself to recognize patterns for the final risk assessment model. This serves to showcase a simple and complex approach to conclude with a middle ground in outcomes.

1.1 Discussion of previous work

For our project our industry partner Prism Data has been guiding us through our process of understanding what a credit score is, how it has been typically generated in the past, and steps in order to solve our current problem: credit card transaction classification. Prism Data has provided us with the datasets to work on this specific task. Since we aren't looking into any specific literature works, Prism Data has been our guide for the flow of this project in which they have been sharing their approaches for data cleaning, splitting, and testing. For example, when it comes to cleaning our text description data, Prism has recommended techniques they personally use themselves, such as removing whitespaces, unnecessary characters, and understanding the significance of certain phrases that could be related to a specific outflow category. Apart from cleaning they also said the feature that helps them best classify their transactions is TF-IDF. Building off of this, during Quarter 2, they also provided us with business standard techniques when suggesting new features that can be built off of transactions. Coming to model classification, Prism Data uses accuracy to measure their model's performance along with simple evaluation models such as Logistic Regression to keep prediction tasks light weight especially since that is the standard if such a model is deployed in the business setting. For income estimation Prism Data uses frequency of payments as their top indicator as well. Amongst consumer default, they also recommended the use of simpler models and determining an effective metric to use for model evaluation. Their suggestions have greatly aided us in approaching our ultimate task at hand to predict consumer default in a new credit scoring system.

1.2 Dataset

For this past quarter, we looked at four separate datasets: inflows, outflows, accounts, and consumer.

Inflows/Outflows Datasets:
Both the inflows and outflows datasets have the same key columns shown in Table 1. The only difference is that the transactions in the outflow dataset are negative transactions (since they are transactions that decrease the amount of money in the consumer's bank account, e.g. purchases) while the transactions in the outflow dataset are positive transactions (since they are transactions that increase the amount of money in the consumer's bank account, e.g. income). Overall, the outflow dataset holds around 1,306,452 transactions, whereas the inflow dataset has a total of 513,115 transactions.

Table 1: Column Description of Outflow/Inflow Transaction Dataset

Column Name	**Description**
prism_consumer_id	An identifier for individual consumers.
prism_account_id	An identifier for the associated bank account.
amount	The amount of money associated with the transaction.
memo_clean	A cleaned version of the transaction description
amount	The amount of money associated with the transaction
posted_date	The date that the transactions was made/recorded.
category	The category to which each transaction was categorized to, based on models done by Prism Data and Quarter 1 project

Accounts Dataset:
The accounts dataset contains information about a consumer's different banking accounts as well as the balance of each account at the time of evaluation. The columns of this dataset can be seen in Table 2. Overall, the accounts dataset holds 4695 unique accounts.

Table 2: Column Description of Accounts Dataset

Column Name	**Description**
prism_consumer_id	An identifier for individual consumers.
prism_account_id	An identifier for the associated bank account.
account_type	The type of account the account is, e.g. checking account, savings account, etc.
balance	The balance of the associated account on the balance_date
balance_date	The date that the account balance was recorded

Consumer Dataset:
The consumer dataset contains information about when the consumer applied/was evaluated for a loan, and whether that consumer defaulted on their first loan payment or not. The columns of this dataset can be seen in Table 2. Overall, the consumer dataset holds 1978 unique consumers.

Table 3: Column Description of Consumer Dataset

Column Name	**Description**
prism_consumer_id	An identifier for individual consumers.
evaluation_date	The date that the loan evaluation is considered
FPF_TARGET	If the person failed to pay/defaulted on their first loan payment.

With the use of these datasets, our team's goal is to develop a more comprehensive and equitable credit assessment system, utilizing past banking records, transaction data, and existing account values. From our analysis so far, we have discovered that the 'memo' descriptions and provided transaction categories contain a majority of an individuals' financial behaviors and patterns which we plan to use for further analysis. In addition to this, consumers that have defaulted in the past will most likely default once again.

1.3 Potential Biases/Problems

In our previous work, we mentioned the main problems that we decided would be factors in which we would have to consider are memo categorizations, determining income estimates from inflow transactions, along with creating the ultimate credit scoring. To follow up with our considerations on these potential problems we may run into, this quarter we were able to find more clarity on the approach to these problems.

For our first problem on memo categorizations, we were able to confirm the categories that were presented to us in the dataset obtain from Prism Data. This allowed for us to use their categorizations for our analysis since they align with the categorizations that we can confirm on our work.

For this quarter, the main problem we're working on is determining the amount of risk a consumer has for not paying their bills, based on their transaction history. The goal is to create a new score similar to one's current credit score that predicts and communicates that risk to a potential client, but tailored more towards their credit card transaction history instead. This problem can be broken down into several sub-problems, each with their own potential biases and problems:

Problem 1: Memo Categorization for Outflow Transactions
The first step in determining risk is categorizing each transaction a consumer makes. We know from our current dataset that a majority of the outflow transactions are categorized as "FOOD_AND_BEVERAGE" or "GENERAL_MERCHANDISE" as seen in Figure 1.

If not addressed, this could create a categorization model that's unduly biased towards categorizing transactions as those categories, reducing the overall accuracy of the model. Additionally, it's likely that there are many more transactions during holiday months, such as December, which could create additional bias that may affect the model's accuracy. In

other words, we would also need to consider seasonality amongst our predictions.

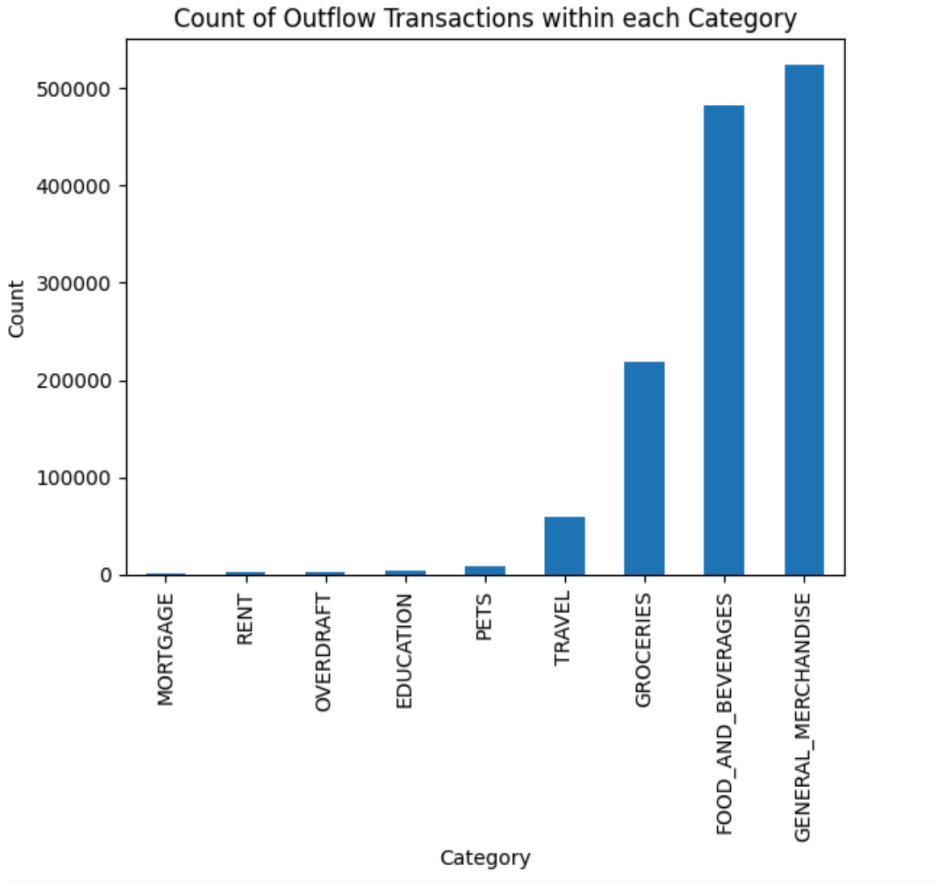

Figure 1: Histogram of Income Distribution for Consumers

Problem 2: Default Rate Imbalance
Similar to the previous problem about category imbalance, there's a significant imbalance when looking at the proportion of people that default on their first loan vs. not defaulting on their first loan, as seen in Figure 2.

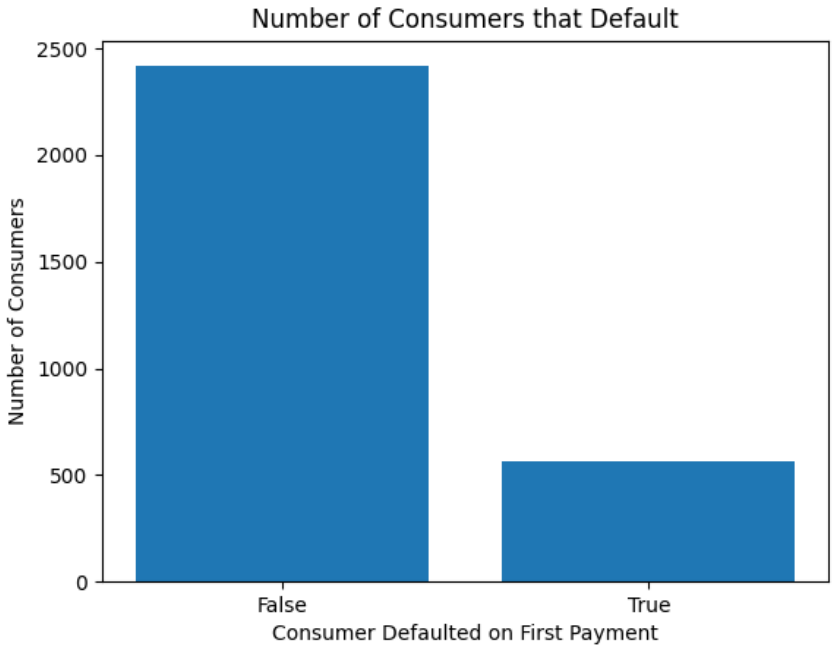

Figure 2: Ratio of Number of Consumers that Default

If not addressed, it could result in risk assessment models that less capable of recognizing when a person has a high risk of default, which defeats the point of the entire project.

Problem 3: Model Explainability
Because the model's assessment of a consumer's likelihood to default would theoretically impact real people, we must be able to justify the model's decisions, what factors contributed to the model making the decision that it did, and what a consumer could do to improve it. This means that the model cannot be too complex, use too many features, or use features that are protected. It doesn't necessarily rule out complex models like neural nets, but it does mean that the model can only be as complex as we can explain, putting a hard cap on the complexity and amount of input features that we can use.

1.4 Model Assessment Metrics

Memo Categorization Model:
Our group is currently using accuracy to assess how well our categorization model is doing at categorizing memo statements. This accuracy metric is simply the number of correct categorizations compared to the actual category of that transaction. However, during model training for BERT we used F1-Score to optimize the model. Ideally, this would allow the model to be less affected by outliers or biases within the data during training, and after some testing, we determined that this metric led to better results compared to just using accuracy for the training process. Besides BERT, the rest of the models were built and optimized upon their accuracies.

Risk Assessment Model:
To evaluate the effectiveness of the risk assessment model, we use the ROC-AUC score. This is used instead of accuracy because there are significantly more consumers that will not default on their loan compared to the number of consumers that will default on their loan. As a result, models could achieve a high accuracy by predicting that everyone it encounters will not default, which defeats the point of the model in the first place.

2 Methods

2.1 Categorizing outflow memos

One of the first steps towards modeling was to present a new schema for the way that the outflow memos were initially categorized. Our dataset was first given to us with predefined categories which had a total of 9 categories for the outflow data. In order for our model to raise significantly more accurate results rather than just using the 9 general category types, we decided to come up with a significant amount of additional categorizations that included more tailored categories to our memo data. For this step, we decided to use TF-IDF Vectorization, also recognized as Term Frequency - Inverse Document Frequency, which measures the importance of terms within a document in relation to the collection of words per document. This was a pivotal step in our data preprocessing especially since we were able to generate 96,000 additional features for our model to use in its predictions. With the addition of 96,000 features in our vectorization process allows the model to use more clues in its analysis. Looking at the effects of this on our models after vectorization, it generated a 97% accuracy for Logistic Regression and a 98% accuracy for the decision tree classifier. This is an increase of 10% compared to our initial accuracy of 88% from BERT.

2.2 Feature Creation

All features that we created can be found below:

- **balance_summary_stats**: min, max, and standard dev of balance a consumer has in any of their accounts
- **category_summary_stats**: mean, median, min, and max of spending in each category throughout dataset
- **disposable_income**: amount remaining after combining all inflow and outflow transactions
- **avg_monthly_spending**: The average amount of money spent in a month
- **num_avg_monthly_purchases**: The average number of purchases in a month
- **avg_monthly_cat_spending**: The average amount of money spent in a month for each category
- **avg_monthly_cat_count**: The average amount of purchases in a month for each category

- **num_savings_transfer**: The number of times someone has pulled from savings account
- **insufficient_bal**: A boolean output for whether a consumer has an account that is negative or near 0 at the time of evaluation.
- **num_accounts**: total number of accounts a consumer has
- **monthly_cat_slope**: slope of spending for each category
- **non_essential_ratio**: proportion of spending done in non-essential vs essential categories
- **stand_balance_slope**: slope of monthly balance
- **positive remaining ratio**: the amount of months where cashflow is positive out of all months in the dataset for each consumer
- **credit ratio**: the maximum number of consecutive months in which a consumer pays of a loan
- **prop spending**: percentage of total consumer spending for each category
- **overdraft freq**: boolean value of users with more than 1 monthly overdraft transaction
- **cumulative weighted default monthly avg**: get sum of average weighted value of overdraft per month

While creating each of these features, it was essential to use a variety of group by operations, filtering on respective categories, along with calculating features amongst a monthly split. This all required the use of a number of splits which was important to gain the final features we came up with in our model. For example, for any monthly related features, there required a split on each year, each month, and a median calculated from this cumulative total in order to demonstrate an accurate monthly average. To elaborate on this further, using the median is a pivotal change that we took compared to the prior quarter since the median a less volatile to changes affected by outliers, in comparison to the mean.

The final feature matrix created by all of these features results in over 200 distinct input features that for the model. It is important to note that the reduction of using more distinct and unique features allows our model to visualize the trends that are essential to creating an accurate score. In comparison to a model that uses all the features we generated which would be redundant and have features that may overfit to the training data, we decided to filter it to features that would give our model the best performance. This feature matrix is split into a training, validation, and testing dataset.

Even with the reduction of our features to be at a manageable volume, this vast amount of features still makes the explainability of the model rather difficult, the next step would be to evaluate the individual impacts of each respective feature. After determining each features relevance, we can select the top 20-50 most effective features to use in a final modeling assessment and remove the remaining features.

2.3 Model Creation

To evaluate model effectiveness, we created a Logistic Regression model to serve as a baseline. Once we did that, we experimented with other models like Random Forests, XGBoost,

and Neural Nets. Our goal for model creation is to create a model that's more effective than the baseline model.

Baseline Logistic Regression Classifier

The Logistic Regression Classifier is created using standard sklearn functions. The initial maixmum epochs to converge were set to 100 epochs. The baseline logistic regression failed to converge at 100 epochs, and while increasing the maximum number of epochs to 1000 did have a small increase in ROC-AUC score, it ultimately didn't have much of an effect on the overall model performance. The ROC-AUC score listed in the results use the higher maximum epoch limit.

Random Forest Classifier

The Random Forest Classifier was created using standard sklearn funtions. While there was some experimentation on the max depth hyperparameter, we ultimately set it to 2. Experimentation on the hyperparameter didn't result in significant change.

XGBoost Classifier

The XGB Classifier was created with default hyperparameters. Experimentation with different hyperparameter values didn't significantly affect the resulting ROC-AUC score.

Sequential Neural Net

The Sequential NN is initially trained with across the training dataset for 500 epochs in batches of size 8, optimizing for BCELoss. This was eventually reduced to 200 epochs with batches of size 64 to speed up training and feature evaluation time while having little impact on the resulting ROC-AUC score. During training, the validation loss is recorded for each epoch and the epoch with the lowest loss is used as the final model.

The Sequential NN has five hidden layers of size 12, 24, 24, 12, and 6 respectively. The final probability is calculated by running the output value through a sigmoid function, giving the overall probability of default. The input layer is the number of features used, which, after feature evaluation, is size 40.

Because the model generates probabilities, we create an optimal threshold by running potential thresholds through the validation set and choosing the threshold that generates the best resulting ROC-AUC score.

3 Results

For classifying memos by category of credit card transactions, we utilized our cleaned data with three different models to see which could perform the best. Our accuracy results on the classification task are shown in Table 4.

Table 4: Classification Accuracy of Memo Categorization Models

Model	**Accuracy**
BERT	88%
Logistic Regression with TF-IDF	97%
Decision Tree Classifier with TF-IDF	98%

When looking at the results for the risk-assessment models, the ROC-AUC scores can be found in Table 5.

Table 5: Test ROC-AUC score of Risk Assessment Models

Model	**ROC-AUC Score**
Logistic Regression	0.58630
Decision Tree Classifier	0.56834
XGB Classifier	0.59218
Sequential NN	0.78500

4 Conclusion

The models that we decided to use in our determination of how effective our features are included using a simple Logistic Regression model, along with a Sequential Neural Network. We decided to a model that is very straightforward just simply using previous classification outputs and another model that uses tensors / layers that would provide more granularity on our final output. This is reflected in our outputs 4, where the Sequential Neural Network provides more accurate results compared to our Logistic Regression model. It is important to note that the time required for training is increased by using a more complex model, along with unknown filters that may be applied to our NN.

In regards to the relevancy of both scores we retrieved from our risk assessment models, each score is used to measure whether or not a consumer is approved or declined for a request in credit. The factors on how often they have defaulted or received overdraft payments are factors into this ultimate decision. In the overall ROC-AUC score, the ROC aspect represents the curve of probabilities and the AUC aspect is ”area under the curve”. Ideally, we want this score to be as close to 1 as possible where 0.5 has no better performance than randomly guessing. This score considers all true positives, false positives, true negatives, and false negatives that our model classifies and compares against the training set that we have established prior to our training.

Overall, our work this quarter so far has been focused on creating additional features that

would allow for our models to gain more insight into the true accurate credit score that we can ultimately award a consumer. Through the addition of each individual category's mean, median, standard deviation, etc, these features would be able to provide our models with a deeper understanding of the behavior of the category features overall. Rather that giving our model direct information / relationships for when a respective consumer may receive a lower credit scoring by using all the features, we can use these category metrics to reason that by spending more in a category poses as more of a risk. The addition of our extraneous features have been beneficial in our ROC-AUC scores.

The continuation of this project is focused on filtering the respective features that we have built so far and to create the new final credit scoring for a respective consumer.