Capstone #2:
Home Credit Default Risk

Miles Craig
Website: MilesIn3D.com
LinkedIn: LinkedIn.com/in/MilesCraig
GitHub: GitHub.com/MilesIn3D

Outline

1. Introduction
2. Data
3. Exploratory Data Analysis (EDA)
4. Finalizing the Data
5. Machine Learning
6. Summary and Next Steps

1

Introduction

Problem Statement

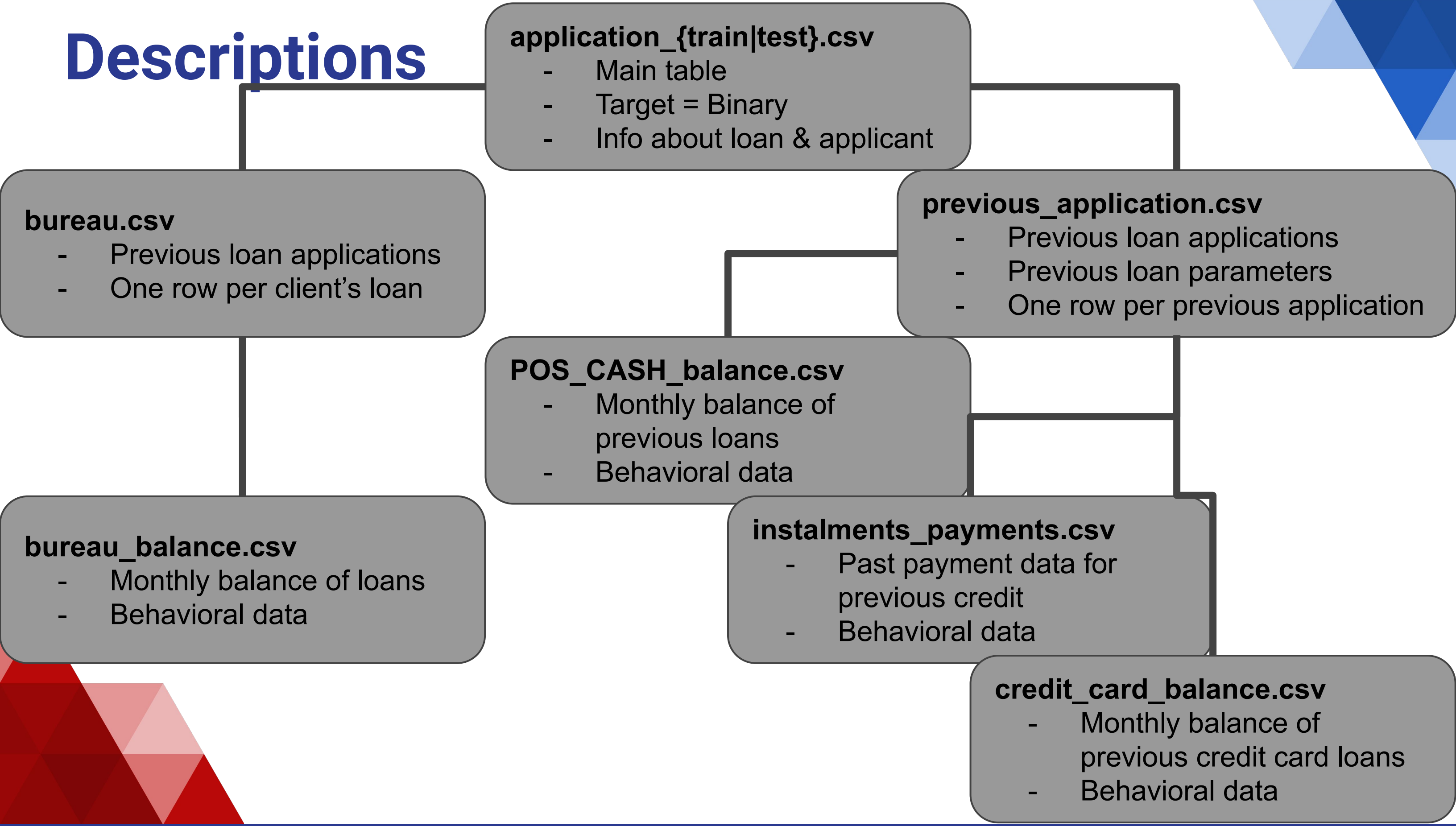
- **Current Situation**: Many people struggle getting loans due to insufficient or non-existent credit histories

- **Goal**: Broaden financial inclusion for the unbanked population by making use of a variety of alternative data to predict their clients' repayment abilities

2

Data

Descriptions



application_{train|test}.csv

- Main table
- Target = Binary
- Info about loan & applicant

bureau.csv

- Previous loan applications
- One row per client’s loan

previous_application.csv

- Previous loan applications
- Previous loan parameters
- One row per previous application

POS_CASH_balance.csv

- Monthly balance of previous loans
- Behavioral data

bureau_balance.csv

- Monthly balance of loans
- Behavioral data

instalments_payments.csv

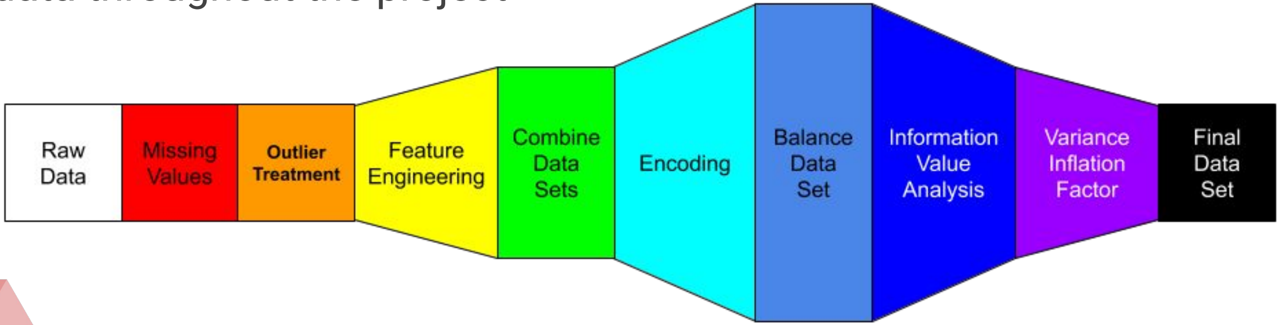
- Past payment data for previous credit
- Behavioral data

credit_card_balance.csv

- Monthly balance of previous credit card loans
- Behavioral data

Overview

- Many steps are used in the cleaning process the data before modeling it
- Feature Engineering and Encoding added new columns to the data
- Information Value Analysis and Variance Inflation Factor reduced the number of columns
- The below diagram is a graphical representation of number of columns of data throughout the project



Missing Values

Data sets can have missing values for a myriad of reasons:

- A form was only partially filled out
- A particular column of data wasn’t stored until a certain point in time

It is important to handle missing values at the beginning, before Variable Selection methods, like IV and VIF. A simple approach is to remove all rows where there is a missing value, but in this case, that would be ~90% of the rows. So each column of data was handled independently.

- For Categorical data, an additional category was created, ‘unknown’
- For Numerical data, the average, median or mode replaced the NaN value

Outlier Treatment

- Outliers are defined as values that are greater than 3 standard deviations away from the mean in either the positive or negative direction.

- This only applied to numerical columns, not categorical columns like flag or boolean types with values of 0 and 1

Feature Engineering

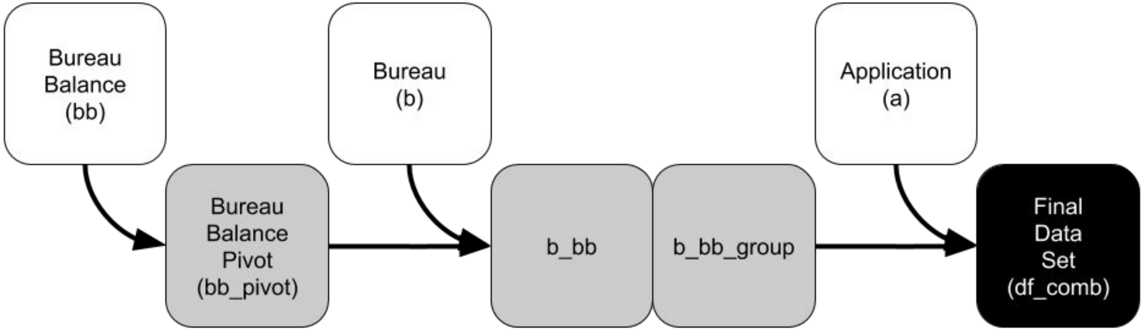
Feature Engineering is a technique used to expand upon the original data set

Here are four examples of columns created through FE:

- **FLAG_CHILDREN** = 1 if CNT_Children > 0; else 0
- **FLAG_INCOME** = 1 if AMT_INCOME_TOTAL > 0; else 0
- **FLAG_INCOME_6figs** = 1 if AMT_INCOME_TOTAL >= 100000; else 0
- **CNT_DOC** = SUM(FLAG_DOC)

Combine Datasets

Three datasets were combined to create the final dataset for modeling. The Bureau Balance table had to first be pivoted so that each row had a unique index. When merged with the Bureau table, this created a new dataset that didn't have unique indexes. Each index was grouped by either counting, averaging, etc. Then this dataset was combined with the Application table to create the final dataset.



3

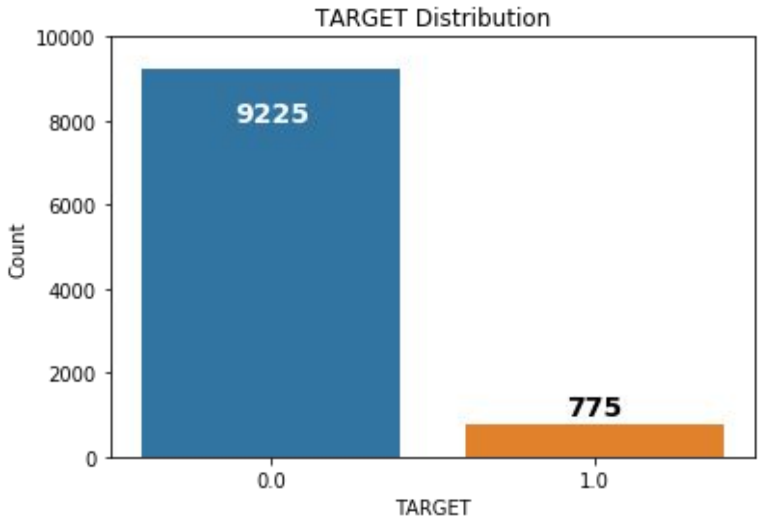
Exploratory Data Analysis

Target Distribution

The target data is a 1 or 0:

- **1**: client with payment difficulties
- **0**: all other cases

This plot shows that the dataset is unbalanced, roughly 92% to 8%. This will be handled later in the project.

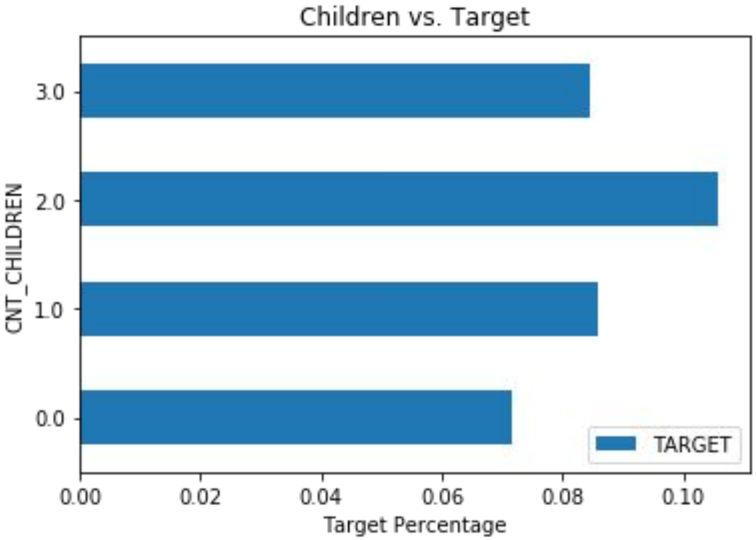


Children

This plot compares the number of children an applicant has with the percentage of 1s and 0s.

Applicants without children are the least likely to have payment difficulties.

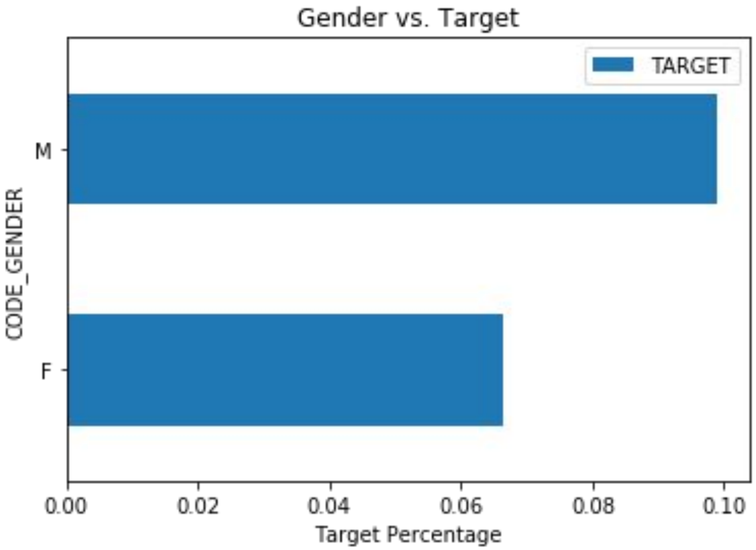
Applicants with 2 children are most likely, ~50% more likely.



Gender

This plot compares the gender of an applicant with the percentage of 1s and 0s.

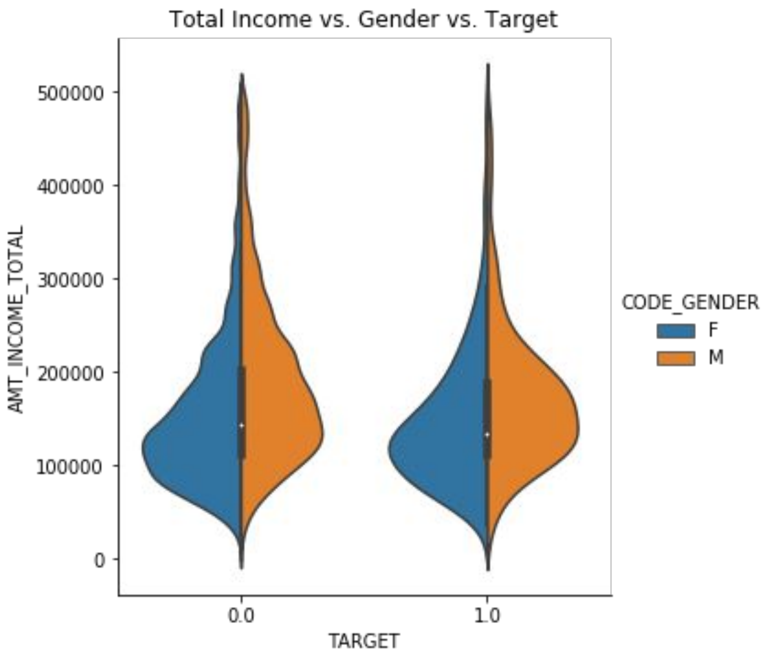
Men are ~50% more likely to have payment issues when compared to women. This is quite a significant difference.



Total Income

This plot shows the distribution of total income for 1s and 0s.

Men with high income tend to have less payment issues. Also, men tend to have higher income per category, 1 or 0.

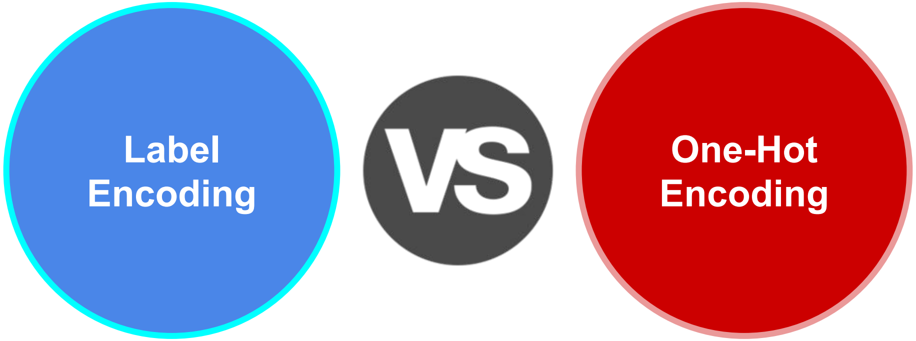


4

Finalizing the Data

Encode the Data

Almost all datasets include categorical type information. This could be colors, sizes, groups, etc. But for the data to be modeled with Machine Learning Algorithms, the categories have to be converted into numerical values. This can be done with encoding.



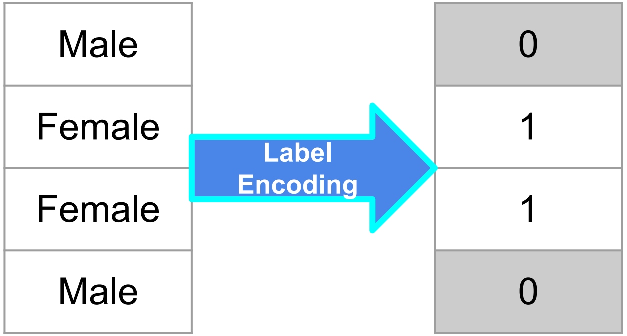
Label Encoding

VS

One-Hot Encoding

Label Encoding

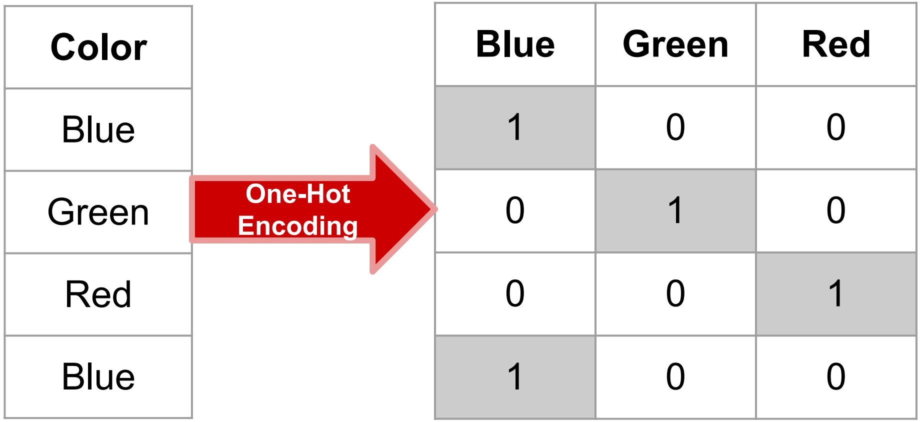
- Assign each unique category in a categorical variable with an integer
- No new columns are created
- Only applied to columns with 2 categories OR where categories have order to them
  - Male vs. Female
  - Sizes - small, medium, large



Male	0
Female	1
Female	1
Male	0

One-Hot Encoding

- Creates a new column for each unique category in a categorical variable
- Each observation receives a 1 in the column for its corresponding category and a 0 in all other new columns
- Applied to columns with 3+ categories and categorical data where order doesn't matter
    - Colors - red, blue, green



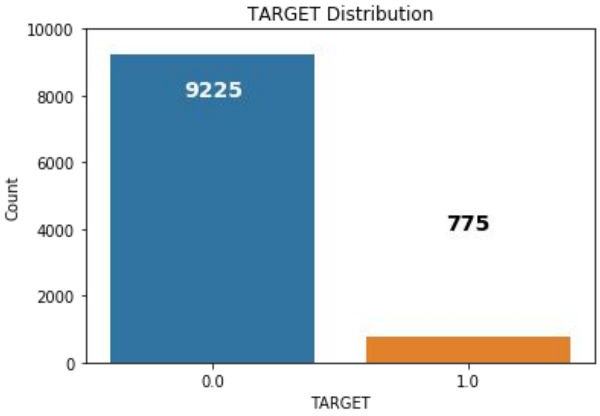
Color
Blue
Green
Red
Blue

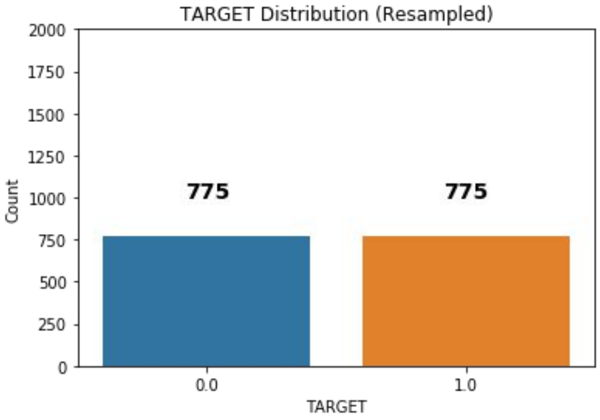
One-Hot Encoding

Blue	Green	Red
1	0	0
0	1	0
0	0	1
1	0	0

Balance the Data

The raw data, shown in the left plot is very unbalanced. For Machine Learning Algorithms, it is important to balance the data. This was done by undersampling the rows where TARGET is 0. This caused the final dataset to be 50/50 0s and 1s, shown in the right plot.





Split the Data

The final dataset needed to be split into sub datasets:

- **Train**: The models will be trained on this data
- **Test**: The models will be tested and scored on this data
- **X**: The independent variables
- **Y**: The dependent variable (TARGET)

	X				Y
	Index	Feature_1	Feature_2	Feature_3	TARGET
Train	1	0	1	1	0
	2	1	1	0	0
Test	3	1	0	1	1

Variable Selection

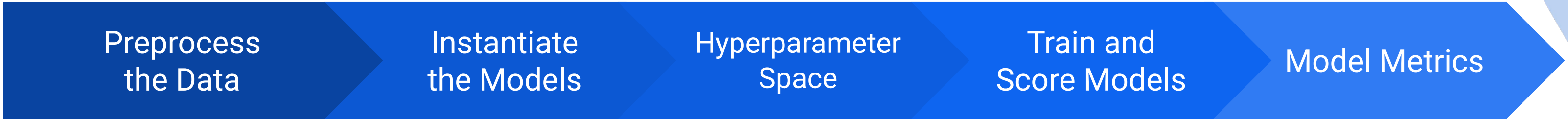
Variable Selection is a process that reduces the number of columns of the dataset. After encoding the data, it has 366 columns of information. There are two methods to select variables:

- **Information Value Analysis (IV)**: A data exploration technique that helps determine which data columns in a dataset have predictive power or influence on the value of a specified dependent variable.
  - 53 columns remain
- **Variance Inflation Factor (VIF)**: The ratio of variance in a model with multiple terms, divided by the variance of a model with one term alone. It quantifies the severity of multicollinearity in an ordinary least squares regression analysis.
  - 41 columns remain

5

Machine Learning

Machine Learning Steps



Preprocess the Data

Instantiate the Models

Hyperparameter Space

Train and Score Models

Model Metrics

Two pipelines were created to handle the different types of data. One for handling numerical data and one for handling categorical data. Then these were combined into a Column Transformer.

Each model was instantiated using the different libraries in sklearn. Some needed parameters set to specific values when created.

Each model has its unique parameters. The hyperparameter space is where a list of values are given for each parameter so that the model can be attempted with different parameter combinations. The more parameters attempted, the longer the process takes.

A function was created to do a multitude of things:

1. Create the CV Object
2. Fit the Model on the Training Data
3. Predict the Output with the Test Data
4. Score the Model with the Test Data

A metrics table was created to store the metrics of each of the different models. This table includes information like accuracy, precision, recall, area under the curve (auc), and time to train.

Model Metrics

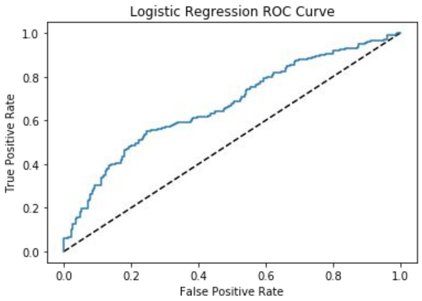
Seven different models were selected to be trained and tested. Below is a table with all the results of the different models related scores utilizing the same dataset. AUC is one of the best ways to compare models. Notice that the best model is the Logistic Regression, with an AUC score of 67, followed by XGBoost and the Random Forest.

Metrics	Dummy Model	Logistic Regression	K Nearest Neighbors	Decision Tree	Random Forest	Naive Bayes	XGBoost
accuracy	49.200	61.500	55.500	58.300	59.100	54.800	60.600
precision	0.000	62.900	57.000	57.600	62.100	54.600	63.100
recall	0.000	58.900	50.000	67.400	50.000	64.800	54.200
auc	50.000	67.000	58.500	60.700	64.300	60.700	65.300
time to train	0.008	6.619	5.088	0.798	5.373	0.009	72.646

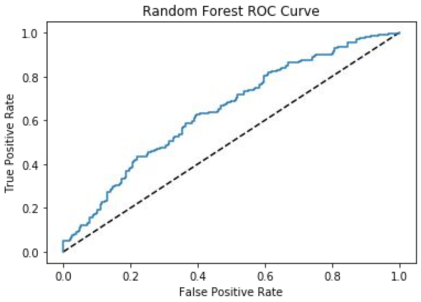
Top Model Performers

Below are the top three ROC curves that are used to calculate the Area Under the Curve (AUC). The larger the AUC, the better the model performance.

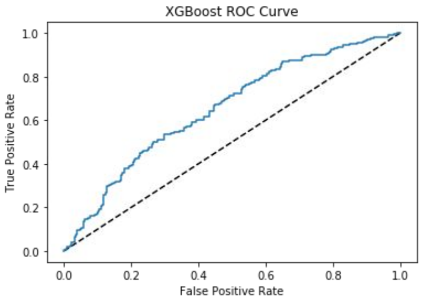
67.0



64.3



65.3



6

Summary & Next Steps

Summary

There is a lot of financial data for each applicant. It can be used to better serve new applicants who are looking to expand their financial footprint in a responsible fashion. This alternative data can be taken advantage of to empower these new clients. The data sets were wrangled and combined to create a final dataset. Seven different models were tested on the final data set. Each one utilized a range of parameters to find the best version of each. According to the Area Under the Curve (AUC), the Logistic Regression had the highest score, 67 out of 100.

Next Steps

Additional Datasets

- In this project, three out of the seven datasets were utilized. This showed the value in combining multiple datasets into one before modeling the information. In a future project, the remaining datasets could be included in the final dataset to further increase the accuracy of the models.

Feature Engineering

- Another step that could be taken would be to increase the number of Feature Engineered columns. This could be done with further insight into the background of the raw data, how it was collected, and how it's applied currently in the field.

TPOT

- TPOT is a Python library that automates the entire Machine Learning pipeline and returns the best performing Machine Learning model. This library could be used to increase the scope and the speed of the ML process, and eventually could return a better final model.