Credit Risk Dataset Analysis and Prediction

Stephen Kim
UC San Diego
La Jolla, CA, 92037
sskim@ucsd.edu, A15400652

***Abstract*—Finances play an important role in many aspects of an individual's or company's lifetime. In making financial decisions, often, the costs are spread across many months or years. Usually, income comes in increments, so payments are, often, also made in increments. In some seasons, one's financial situations are better than others. Credit is one of the ways that banks have provided to ease payment frustration in worse seasons or to lump costs in good faith that the bank will be repaid at a later date. The important question then is Can whom the bank gives credit to, in the form of a loan or credit line, repay the borrowed finances? On the other side, the important question is What factors are most important to prove the trustworthiness to receive credit? Using credit data, I answer both of these questions using machine-learning models and will create a model that will return the probabilities and classify the credit risk of an individual. Github Project Repository Link : Link.**

1 Introduction

The Kaggle Code Competition *Give Me Some Credit*[1] gives credit data and asks competitors to create a model that can help users determine their Credit Risk. Credit Risk is the probability that a user will be a financial risk for a credit association within the next two years.
When making financial decisions that involve credit such as taking out a loan Credit Risk plays an important factor in determining whether an individual is eligible. Being eligible means that the financial institution can trust the individual to pay back the borrowed finances.
This project will analyze the competition data and observe the distributions and correlations between the different features. Afterward, I will create machine-learning models to determine the probabilities and classify an individual's Credit Risk.

2 Data Analysis

2.1 Data Features

As stated earlier, we are using the *Give Me Some Credit* competition dataset. This dataset has a total of 11 features, 1 of which is the classification feature. Table 1 shows each of the features as well as gives a short description them.

[1] Give Me Some Code: https://www.kaggle.com/competitions/GiveMeSomeCredit

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Table 1: Give Me Some Credit Data Features

Feature	Description
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits
age	Age of borrower in years
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years
DebtRatio	Monthly debt payments, alimony,living costs divided by monthy gross income
MonthlyIncome	Monthly income
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)

Note: Descriptions are taken from Dataset Dictionary provided by competition

2.2 Data Overview

The data taken from the *Give Me Some Credit* competition has 150000 rows with 11 columns representing 10 features and 1 classification result. The competition also provided test data, but the analysis will only consist of the training data. The test data will be used later when predicting the probabilities and classifying the data.

2.3 Data Cleaning

The first step taken to clean the data is to see which features had Nan values. The features that had Nan values were the *MonthlyIncome* and the *NumberOfDependents* features. These features are not so easy to replace, so it is better to just omit the rows with Nan values. This brings us to a resulting 120269 data points to analyze. There are more cleaning steps as it seems that the data itself has not been extensively vetted as there are some values that do not belong. Also, some of the distributions are entirely left-skewed, so functions are applied to lessen that effect.

- For *RevolvingUtilizationOfUnsecuredLines* there are some outliers that left-skew the data by an enormous amount, so to bring the data more centered, this formula is applied to the feature:

$$RevolvingUtilizationOfUnsecuredLines <= \frac{\sigma^2}{\mu} + \mu$$

- For *age*, only ages above 0 are considered.
- For *NumberOfTime30-59DaysPastDueNotWorse*, only entries less than or equal to 25 are allowed since we are looking at a span of 2 years.
- For *DebtRatio*, filtering by entries less than 20000 (%) are considered so that we can aim to avoid an entirely left-skewed distribution.

After cleaning and processing all of the features, there are a resulting 119948 rows left in the data.

2.4 Feature Analysis

2.4.1 Data Distributions

Now, I will begin analyzing the different features. First, Figure 1 shows the distributions of our data split between the two classes. The two classes, in this case, are whether or not an individual was a Credit Risk in the past two years - 0 corresponding to true and 1 corresponding to false.

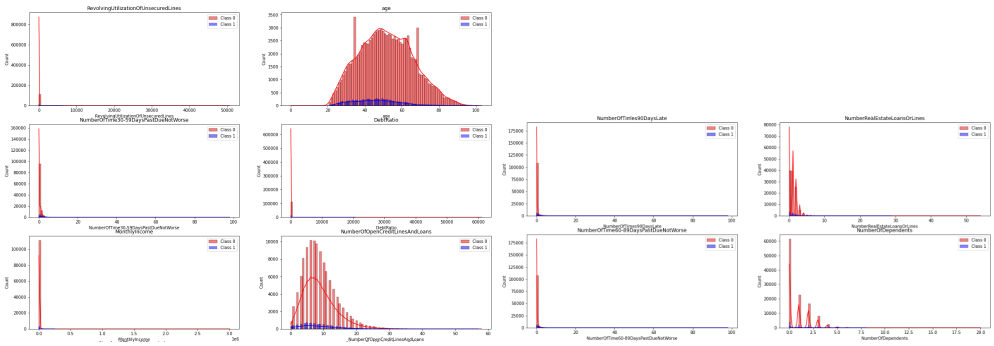

Figure 1: Feature Distributions with Class Separation

After the cleaning steps, it can be observed there is some separation between the two classes in our data. Given that the number of positives for Credit Risk is lower, it can also be seen that certain features share similar distribution given that there was a smaller sample size.

2.4.2 Data Correlations

Next, I will be looking at the correlations among the various features. The correlation matrix is shown in Figure 2. We can see that the different features have a minor correlation with our classification feature. Also, we can to see that some of our features are directly related to one another. This makes sense as these features relate to each other in the number of payments late in different increments. If a data point is in the largest category, it surely will be in the others before it. These features are *NumberOfTime30- 59DaysPastDueNotWorse*, *NumberOfTimes90DaysLate*, and *NumberOfTime60- 89DaysPastDueNotWorse*.

From here, we can partly see which features will begin to play the largest role in determining if an individual will be at Credit Risk. For instance, *age* has a negative correlation with Credit Risk. This gives us the informal inference that younger individuals are more at Credit Risk, which makes

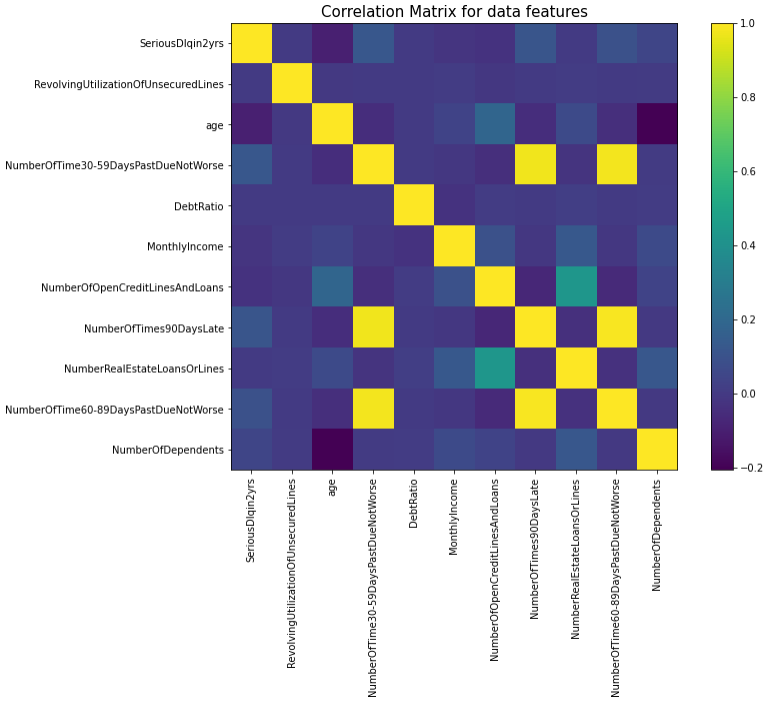

Figure 2: Correlation Matrix

sense since usually younger individual's careers aren't nearly as successful as their older counterparts so their financial wealth is not stable.

With a little hint of what features were important for classification, I then went into creating machine-learning models to classify Credit Risk.

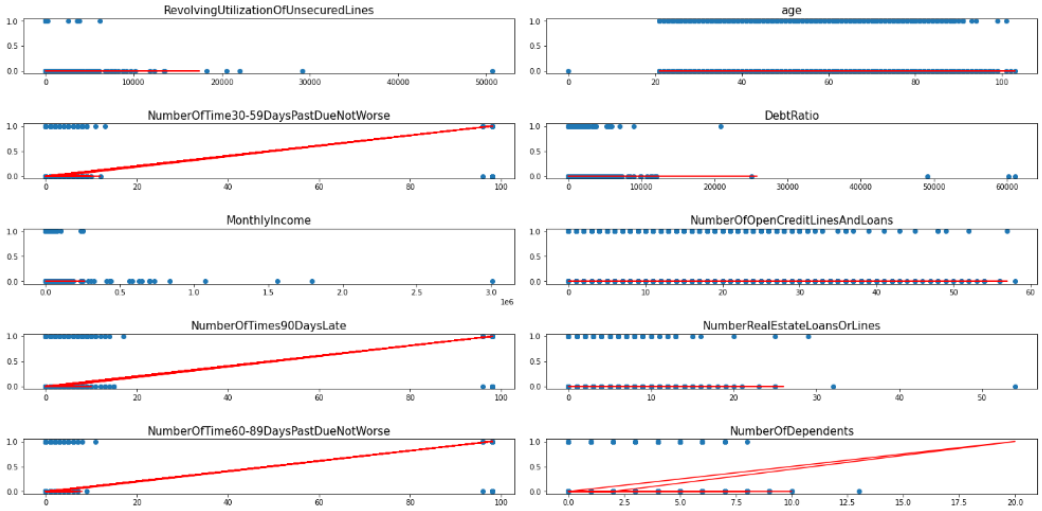

Figure 3: Single Feature Logistic Regression

3 Models

3.1 Logistic Regression

The first type of model I will create is Logistic Regression. Logistic Regression is a classification method that estimates the parameters of the different classes and then classifies different inputs based on predictive analysis. Before I tie all of the features together, I looked at the results of each feature on its own. These results are shown in Figure 3. From visual inspection, we can see that the three number of days late intervals and NumberOfDependents features give some results. The other features have no relationship. These are the features that had some correlation (although close to 0), so this result makes sense. We can also look at the chi-squared test to see another approach to determining significance. We first start with the null hypothesis "There is no statistical significance between feature X_i and an individual's Credit Risk (SeriousDlqin2yrs)", this will be shown in Table 2.

Table 2: Chi^2 Values

Feature	Value
RevolvingUtilizationOfUnsecuredLines	0.0
age	0.0
NumberOfTime30-59DaysPastDueNotWorse	0.0
DebtRatio	0.0
MonthlyIncome	0.0
NumberOfOpenCreditLinesAndLoans	3.028e-45
NumberOfTimes90DaysLate	0.0
NumberRealEstateLoansOrLines	3.035e-1
NumberOfTime60-89DaysPastDueNotWorse	0.0
NumberOfDependents	1.711e-82

From this, we can see that none of the features exhibit statistical significance to reject the null hypothesis, although *NumberRealEstateLoansOrLines* almost reached the threshold. Our correlation values are pretty weak, so the lack of statistical significance makes sense.

3.1.1 Model 1

Now, we are going to use a simple model to perform classification. Model 1 will be a summation of all of the features without any weight modification. In sum:

$$Model1 = \sum_{feature:X_i} X_i$$

After fitting the model and generating our results, I created a confusion matrix to show the accuracy of the model, as shown in Figure 4. From what we can see, the model performs quite poorly when identifying individuals at Credit Risk, so this model may not be the best for our classification problem.

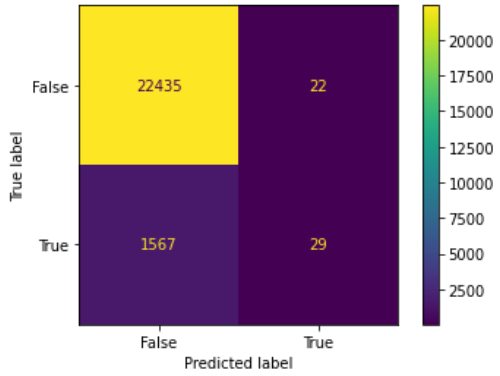

Figure 4: Model 1 Confusion Matrix

3.1.2 Model 2

In our data processing, we saw that some of the features had tremendously large outliers. After cleaning those features, they still had significant outliers relatively. In an effort to try to capture those outliers still, the next model will take the logarithm of the following features: *RevolvingUtilizationOfUnsecuredLines* and *MonthlyIncome*. There will be a small epsilon addition beforehand to prevent Nan values. Besides the logarithm, the remainder of the model will remain the same. Model 2's confusion matrix is shown in Figure 5.

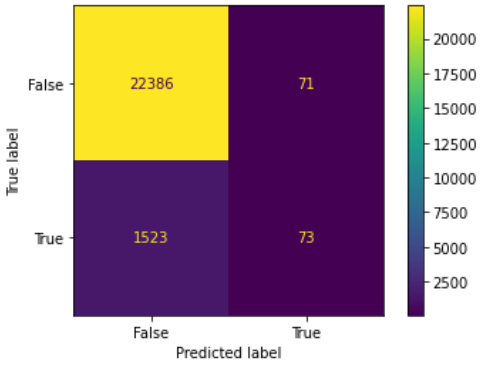

Figure 5: Model 2 Confusion Matrix

There's a little bit of improvement between the two models, but it is not nearly enough for what we need to answer our problem.

3.2 Support Vector Machine

Logistic Regression is a machine-learning approach based on probabilities. It decision boundaries don't seem to be enough for this problem, so the next model I will try is with Support Vectors. In this approach, I perform similar models but in the end, achieve similar results as in Model 2 for Logistic Regression. This is interesting as with Support Vector Machines, there are data manipulations that have to be done in order to output a result in a reasonable amount of time. Before getting into the underlying question I have, I want to try a couple of universal solutions to machine-learning problems before I come to a conclusion.

3.3 Random Forests

Random Forests are one of the machine learning models that can be seen as fairly universal when solving a problem. Often, Random Forests give the best result when attempting to solve a complex classification problem. This is because they are an ensemble of binary decision trees, hence the name forests. Surprisingly, the same result as model 2 was reached, as well.

3.4 Neural Network (MLP)

The last model I will create is a neural network. This model consists of a 10-node input layer, 50-node hidden layer, and 1 node output for binary classification. I trained the model for 10 epochs with a batch size of 100. The loss graph is shown in Figure 6.

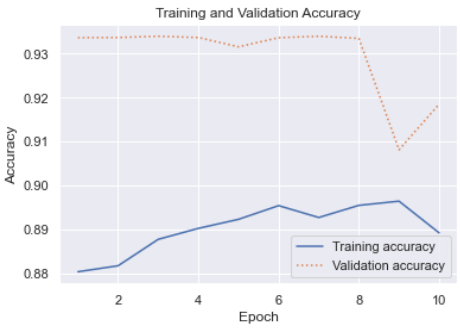

Figure 6: MLP Loss Graph

Loss was calculated using binary cross entropy and for the optimizer, I used the adam optimizer with an initial learning rate of 0.01. Finally, we

can see that the results are different from what the previous models obtained. This is shown in Figure 7.

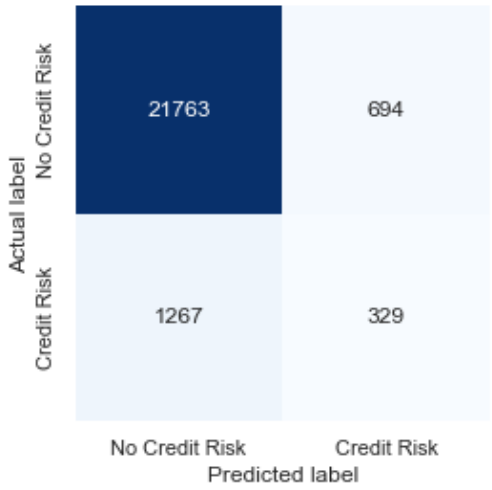

Figure 7: MLP Confusion Matrix

4 Current State & Future Work

As shown by the results of the models that were created, they perform poorly, especially under the false-negative category. The interesting fact is that multiple models converged to a similar result. What I believe to be the issue is then the data. We saw that there was little statistical significance between our features and our classification result. This leads me to believe that the relationship was either too complex or needed more processing for a relationship to appear. The weights of the features are shown in Table 3. The feature's weight is the strength of its factor in an individual's Credit Risk.

Table 3: Feature Weight

Feature	Value
RevolvingUtilizationOfUnsecuredLines	8.08e-1
age	-1.99e-2
NumberOfTime30-59DaysPastDueNotWorse	4.15e-1
DebtRatio	-9.29e-5
MonthlyIncome	-2.77e-2
NumberOfOpenCreditLinesAndLoans	3.36e-4
NumberOfTimes90DaysLate	3.69e-1
NumberRealEstateLoansOrLines	2.74e-2
NumberOfTime60-89DaysPastDueNotWorse	-7.51e-1
NumberOfDependents	6.75e-2

The main statistical means I used to evaluate the "cleanliness" of the data were mean and standard deviation. I briefly looked at skewness but did not extensively account for it. I also left kurtosis out of the picture. Upon starting this project, both of those features were relatively unknown to me, but in another iteration, these factors will be accounted for and could possibly give a better result.

5 Conclusion

In summary, from the *Give Me Some Credit* competition data set, I analyzed the individual features and did some data processing in an attempt to make a statistical relationship shine and remove outliers. Then, I analyzed the resulting distributions for significance, but unfortunately, no strong relationship came out of it. Hoping that a combination of features would result in a significant relationship, I began creating machine learning models. The types of models I used were Logistic Regression, Support Vector Machines, Random Forests, and Neural Networks. Most of the models converged to a similar result that performed poorly with false negatives and overall mainly performed well on true negatives. We can see each of the feature's weights, which relate directly to their importance, but this result is not entirely true since we did not establish a statistical relationship.

For future work, the models can still stand, but the data processing would need to be changed in a way that accounts for other statistical features. If a statistical relationship can be established with the features, then the models could achieve a better result.

References

[1] Give Me Some Credit. https://www.kaggle.com/competitions/GiveMeSomeCredit. 2011.

[2] Keras (Neural Network). https://keras.io/.

[3] Scikit-learn (Logistic Regression, SVM, Random Forest). https://scikit-learn.org/stable/.

[4] Matplotlib (Plots). https://matplotlib.org/

All of the code is available at the GitHub repository: https://github.com/kimsternator/ECE-225-Project.