Akshay Srinivasan and Rashed Rifat
Professor George Wood
DS-UA 202: Responsible Data Science

Final Report

Introduction and Background

The members of our group are Akshay Srinivasan and Rashed Rifat. The Automated Decision System (ADS) that we have chosen to build a nutritional label for a solution to a Kaggle contest posted, entitled "Give Me Some Credit." This ADS predicts the probability that someone will experience financial distress in the next two years. Specifically, the goal of this ADS is to help borrowers make the best financial decisions. We choose to focus on this ADS due to the above-average consequences that result from the disparate impact on the minority groups' access to financial products within the model. Ultimately, developing a nutritional label for this ADS will allow us to judge how effective this model is for our intended use, both in terms of fairness and accuracy.

The data for the project can be publicly found on Kaggle; more specifically, the data can be downloaded here. Similarly, the code is available in the form of Jupyter Notebook (.ipnyb) at this link. For semantic purposes, we refer to this original code as the raw code from hereon. We have modified the original code such that it can be run within Google Colab, which can be found here. These modifications consist of modifying the method via which the data was loaded - they do not affect the model logic - rather, they ensure that path dependencies for data are successfully uploaded. All of our code is publically available for download and should be ready to execute with minor modifications. Note that you will have to have uploaded the project data onto your Google Drive to run the Notebook as is.

Input and Output

In this section, we will explain how the data was collected, selected, and used by the ADS. There is not any metadata on how this data was collected or selected. However, given that this information is being released from a bank, we can hypothesize that this data was lifted directly from consumer data stored at the bank. As the description of this competition describes it as improving a credit scoring algorithm, we can reasonably extrapolate that this data is being used in real models at the bank. Note that these are simply estimates based on the metadata we have - there is not sufficient detail to verify or reject these assumptions.

We now examine the pre-processing done by the decision maker with regards to the description, datatype, and null counts for each input feature of the ADS.

Table 1: Description and Datatype of Each Input Feature

Variable Name	Description	Type
SeriousDlqin2yrs	**Person experienced 90 days past due delinquency or worse**	**Y/N**
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	percentage
age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
DebtRatio	Monthly debt payments, alimony,living costs divided by monthy gross income	percentage
MonthlyIncome	Monthly income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

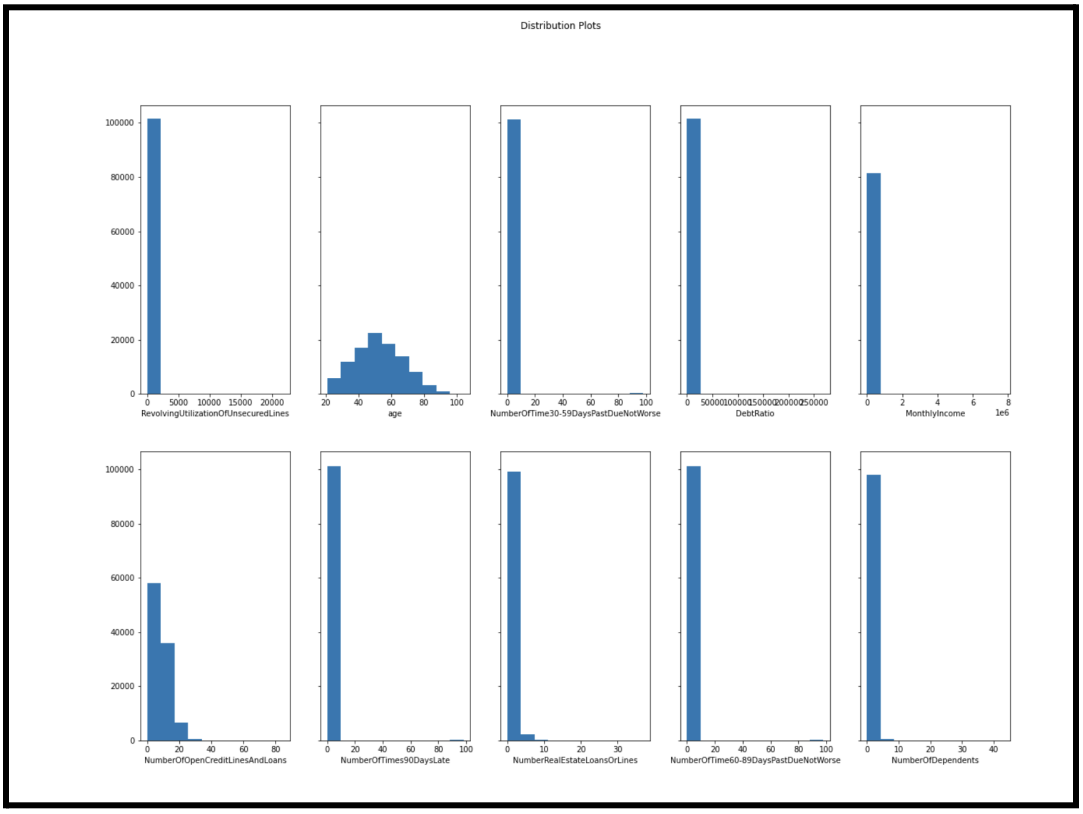
Table 2: General Data Profiling of Each Input Feature

	column_name	col_data_type	col_memory	%_of_dtype_mem	%_of_total_memory	dtype_count	dtype_total	dtype_%_total_mem	non_null_values	%_of_non_nulls	...	%_of_nulls	unique_values_count	count	mean	std	min	25%	50%	75%	max
0	Unnamed: 0	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	101503	101503	50752.00	29301.54	1.0	25376.50	50752.00	76127.50	101503.0
1	SeriousDlqin2yrs	float64	0.774406	20.0	8.333333	5	3.872032	41.666119	0	0.0	...	100.0	0	101503	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	RevolvingUtilizationOfUnsecuredLines	float64	0.774406	20.0	8.333333	5	3.872032	41.666119	101503	100.0	...	0.0	85716	101503	5.31	196.16	0.0	0.03	0.15	0.56	21821.0
3	age	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	82	101503	52.41	14.78	21.0	41.00	52.00	63.00	104.0
4	NumberOfTime30-59DaysPastDueNotWorse	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	16	101503	0.45	4.54	0.0	0.00	0.00	0.00	98.0
5	DebtRatio	float64	0.774406	20.0	8.333333	5	3.872032	41.666119	101503	100.0	...	0.0	79878	101503	344.48	1632.60	0.0	0.17	0.36	0.85	268326.0
6	MonthlyIncome	float64	0.774406	20.0	8.333333	5	3.872032	41.666119	81400	80.194674	...	19.805326	11976	101503	6855.04	36508.60	0.0	3408.00	5400.00	8200.00	7727000.0
7	NumberOfOpenCreditLinesAndLoans	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	56	101503	8.45	5.14	0.0	5.00	8.00	11.00	85.0
8	NumberOfTimes90DaysLate	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	18	101503	0.30	4.52	0.0	0.00	0.00	0.00	98.0
9	NumberRealEstateLoansOrLines	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	24	101503	1.01	1.11	0.0	0.00	1.00	2.00	37.0
10	NumberOfTime60-89DaysPastDueNotWorse	int64	0.774406	14.285714	8.333333	7	5.420845	58.332567	101503	100.0	...	0.0	12	101503	0.27	4.50	0.0	0.00	0.00	0.00	98.0
11	NumberOfDependents	float64	0.774406	20.0	8.333333	5	3.872032	41.666119	98877	97.412884	...	2.587116	13	101503	0.77	1.14	0.0	0.00	0.00	1.00	43.0

12 rows × 21 columns

Based on the table above and the value distributions of each input feature, we can make a deluge of observations.. First, the features *Monthly Income* and *Number Of Dependents* contain 29731 (19.82%) and 3924 (2.61%) null values respectively. Second, the feature *Revolving Utilization of Unsecured Lines*, represents a ratio of the aggregate money owed to the total credit limit of a borrower. As conveyed in the distribution plot, the values represent a right-skewed distribution. This implies that the proportion of people defaulting should increase along with the value of R*evolving Utilization of Unsecured Lines increase.* Moreover, the decision maker noted that since the minimum value of this feature is 13, the proportion of defaulters is smaller than the case of borrowers who owed money without exceeding the credit limit. From this, the decision maker removed samples in which the value of R*evolving Utilization of Unsecured Lines was greater than or equal to 13.*

Figure 1: Distribution Plot of Each Input Feature

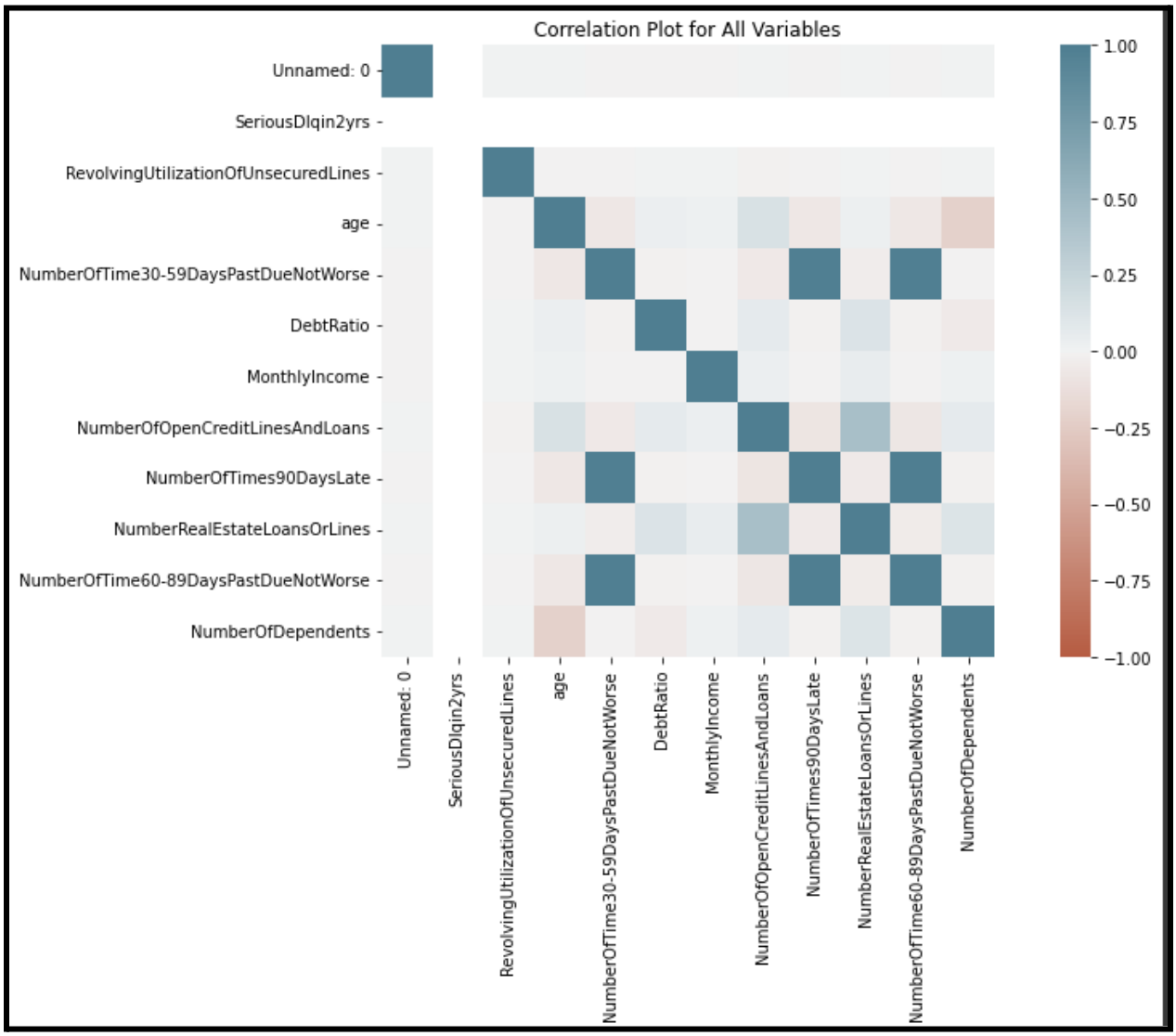


Observing the various distributions of the input features overall, we cannot reasonably gather any information from these distributions (other than age, number of credit lines, and loans open) as most values fall within one bin. This suggests that most of these plots have a large number of outliers, which would lead to the binning of a significant number of data (as the scale compresses to fit all points on the graph). To investigate these distributions further, we examine these distributions individually. For the sake of the brevity of this report, we have included these graphs within the appendix. Since wealth is highly concentrated, most of the data went through an outlier transformation, where outliers were removed from the dataset and placed into a subset. We define outliers as being three standard deviations above the mean, with the exception of monthly income. It is important to note that these outliers were removed from the initial data simply for data visualization purposes. We then plotted both the distribution of the normalized data as well as the outliers, for a clearer view of the data distribution.

From initial examination of the *Age* distribution plot, we note that the values are normally distributed. Moreover, by looking directly at the dataset we see that more young people are defaulting. We will analyze this with respect to fairness in the Outcomes section of this report. With regards to the *Debt Ratio* input, approximately 2.5% of borrowers owe 3490x of what they own in assets. For the borrowers in this set, only 185 borrowers contain values of either 0 or 1 for their monthly incomes. From the distribution plot of Monthly Income, the overall modal value of monthly income in the non-outlier data is $12,000, whereas the mean in the data before

pre-processing is $5,400. The authors did not note this skew and imputed the mean of the outlier data for the observations with null values. Lastly, for the null values of the *Number of Dependents* input feature, the decision maker imputed the overall modal value of 0. For visualizations of these distribution plots, look at the appendix which can be found at the end of the report. Next, we will use a correlation plot to depict the relationship between each input feature.

Figure #2: Correlation Plot for All Variables



From this figure, we can see that the number of times a borrower has been 30-59 days late has a perfect positive relationship with the number of times a borrower has been 60-89 days late and the number of times a borrower has been over 90 days late. Through examination of the dataset, we can see that the number of times a borrower is 90 days late, number of times a borrower is 60-89 days late, and the number of times a borrower is 30-59 days late all share the

same values of 96 and 98. While it may seem illogical for a borrower to be 30 days late 96 times during a 2-year time period, it is possible if the borrower has multiple lines of credit for which they default.

Lastly, as discussed earlier, the output of this ADS is the probability that someone will experience financial distress in the next two years. However, the model that is trained is a binary classifier - we simply derive the probabilities from it. We will interpret this probability from the objectivist interpretation, namely the relative frequency in the long run.

Implementation and Validation

We note that the authors of this model considered multiple models before decciding on their "best" model (we note the evaluation criterion below). We analyze the final model that the authors choose as this would have been the model that would have been used within a real-world scenario. The model described within this system is relatively simple and well documented. Due to the robust documentation of the code by the authors we can discuss the strategy employed by the decision maker of this solution. The authors began by training a multitude of models where they examined both the scores and classification of each particular. Models trained included a baseline Logistic Regression model, a Logistic Regression with a LBGGS solver, a Random Forest Classifier, a XGBClassifier, and a GridSearchCV. Agains, we note that these models are not discussed in the context of this report - the final or "best" model is. Models were ranked based on their roc_auc_score curve, with the XGB Classifer model being ranked first. Note that the pursuit of pure accuracy has negative effects on fairness metrics, we will explore below.

The final model for this submission was the XGBClassifier model from the XGBoost library. The documentation for this library can be found here. This library operates the Gradient Boosting framework, which "is a machine learning technique used in regression and classification tasks… that gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is a weak learner, the resulting algorithm is called gradient-boosted trees… A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function" (Wikipedia, Gradient Boosting).

In other words, the XGBoost is an ensemble model, or a conglomeration of many other models (priamarly decision stress) along with a variation of a technique called bagging. Bagging, or "bootstrap aggregation," chooses samples with replacement (bootstrapping) and amalgamates them by calculating their average (aggregation). Unlike bagging, boosting converts weak learners into strong learners by concentrating on the weakest individual models of the ensemble model. XGBoost, also known as "eXtreme gradient Boosting," is approximately ten times faster than a traditional Gradient Boosting method because of its use of cache awareness and parallel computing. Moreover, XGBoost optimizes trees through a distinct split-finding algorithm, and while using regularization to reduce overfitting. The documentation for the specific model,

XGBClassifier, can be found here. This library uses the scikit-learn API implementation for the XGBoost classification. The documentation from scikit-learn XGBoost Classification can be found here.

Outcomes

First, using the figure below, we will analyze the effectiveness of the ADS by using a confusion matrix, distribution of predictions, and the ROC Curve. Note that these figures were generated by the authors of the ADS and then were independently verified within this report. Based on the distribution plot of the ROC Curve we can see that the XGBClassifier has the largest area of 0.87 (as opposed to the other models mentioned beforehand). The confusion matrix denotes the number of True Negatives, False Positives, False Negatives and True Positives. We note that there are approximately 6 times as many False Negatives relative to False Positives, an indication that we should examine the rates further. We also observe that the model correctly predicts that most of the borrowers will not experience financial distress in the next two years - in fact, the majority of the data is composed of borrowers that are considered safe. This has further implications on the False Negative Rate, which we examine below.

Finally, we also examine the Distribution of Predictions which demonstrates the decision boundary that denotes whether the model classifies a particular observation as a positive or a negative instance.

Figure #3: Confusion Matrix, Distribution of Predictions, and ROC Curve

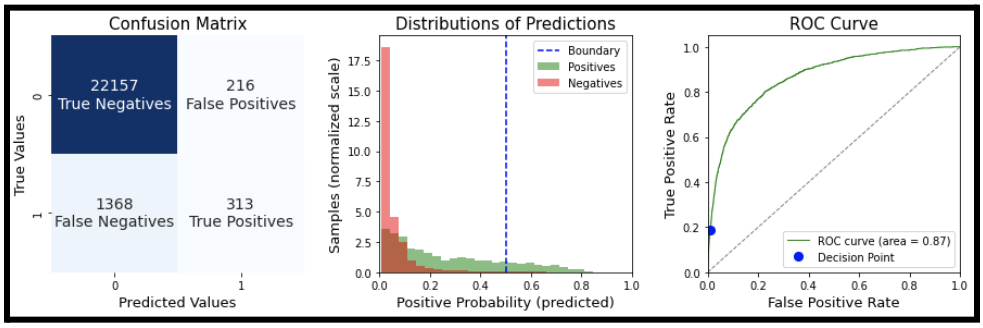


Figure #4: Fairness Metrics

```
({'FN': 1368, 'FP': 216, 'TN': 22157, 'TP': 313},
 {'F1 Score': 0.2832579185520362,
  'False Negative Rate/Miss Rate': 0.8138013087447947,
  'False Positive Rate': 0.009654494256469852,
  'Positive Predictive Value/Precision': 0.5916824196597353,
  'True Negative Rate/Specificity': 0.9903455057435302,
  'True Positive Rate/Recall': 0.18619869125520525})
```

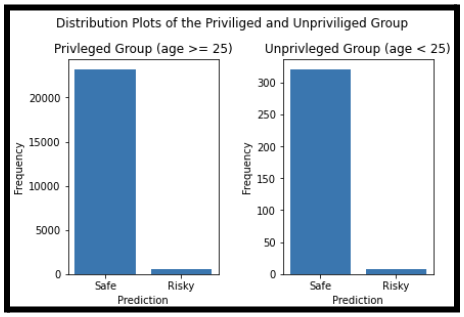
Next, we use the figure above (from our model analysis) to examine the efficacy of the ADS based on the following metrics: F1 Score, False Negative Rate, False Positive Rate, Positive Predictive Rate, Precision, Specificity, and Recall. We know that high precision, or the ratio of correctly predicted positive instances to the total predictive positive instances, implies a

low false positive rate. This is surprisingly not the case based on the data above - the model demonstrated precision of only 0.59 and a false positive rate of 0.01. Furthermore, the metrics and the input data corroborates the fact that the probability of a borrower not experiencing financial distress in the next 2 years is highly prevalent, through a high false negative rate and a low false positive rate. This could have been a result of the proportion of the safe and risky borrowers within the training data. This model also contains an exceptionally high specificity. From this, we can infer that the author of the ADS optimized for predicting a true negative instance given the fact that most observations are negative instances. However, this has significant trade offs as we can see from the rates above. This model exhibits low recall which implies that it is not able to find a significant proportion of the positive cases in the data. Using these metrics of Precision and Recall we calculated the F1 score, which is relatively low.

We now proceed by examining the fairness of the ADS by comparing its performance across different subpopulations. For the purposes of this analysis, our protected attribute age. We denote the privileged group as a borrower with an age greater than or equal to 25 and the unprivileged group as any borrower less than 25 years of age . Using the AIF360 framework, we calculated the two fairness metrics of mean difference and disparate impact on the training data using the BinaryLabelDatasetMetric class.

The mean difference of the ADS was 0.00236, a comparatively low value. This implies that statistical parity exists for the model, or members of both the unprivileged and privileged groups have the same probability of being predicted as a positive instance (in this case, experiencing financial ruin within the next two years). Second, the disparate impact of this ADS was 1.10729. We know that disparate impact compares the fraction of borrowers that are labeled as a positive instance between the unprivileged and privileged groups, so we can interpret the value greater than 1 as achieving group fairness. Moreover, disparate impact for this ADS demonstrates that the unprivileged group has more favorable outcomes than the privileged group. This could be due to the fact that there are far fewer observations in the privileged group than the unprivileged group, as demonstrated in the figure below. Since this ADS didn't violate disparate impact, we use a Disparate Impact Remover or Prejudice Remover.

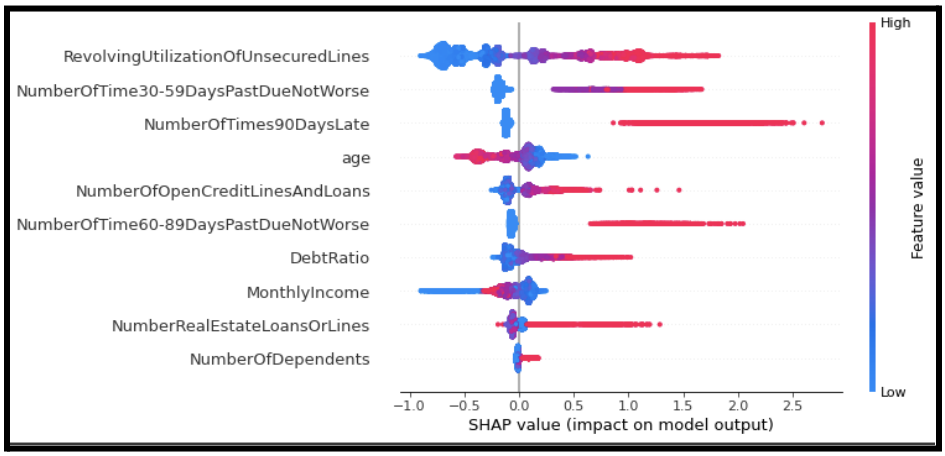
Figure #5: Distribution Plots of the Privileged Group and Unprivileged Group



Now that we've analyzed this model in terms of fairness, we will use the SHAP framework in order to examine the model's feature importance. As discussed earlier, the XGBClassifier is a

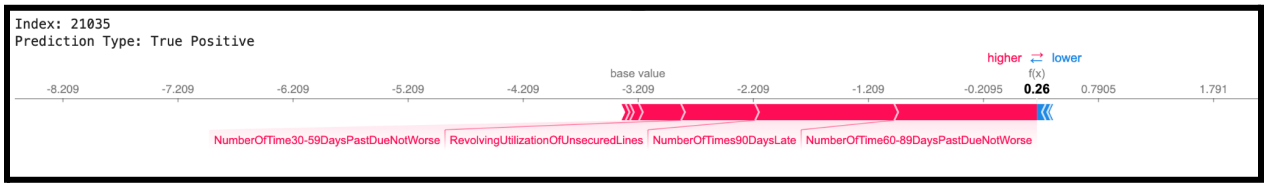
generic class that implements the gradient boosted tree classifier, which makes it suitable for the SHAP framework. Within the explainer, we set the check_additivity flag to be false. The check_additivity flag checks fail if for one of the samples the sum of the SHAP values does not match the model output. Although the test and train datasets are of the same shape, we fail one of these checks with an epsilon less than 0.001, which is acceptable. In the figure below we employ the summary plot to summarize the most important features of the model.

Figure #6: Summary Plot of Feature Importance



From the figure above, we can see that the features of the number of times a borrower is late (30-59, 60-89, 90+), debt ratio, and number of real estate loans or lines of credit contain mainly a positive impact on the prediction of a positive instance. Revolving utilization of unsecured lines and age both have a positive and negative impact on the model output. Monthly income has a negative impact and the number of dependents seems to have a little effect on the predictions. To get a better understanding of feature importance, we look at the SHAP explanation for a true positive, true negative, false positive, and false negative entry. Figures 7-10 display how the SHAP values of all features sum up to explain why the prediction was different from the baseline.

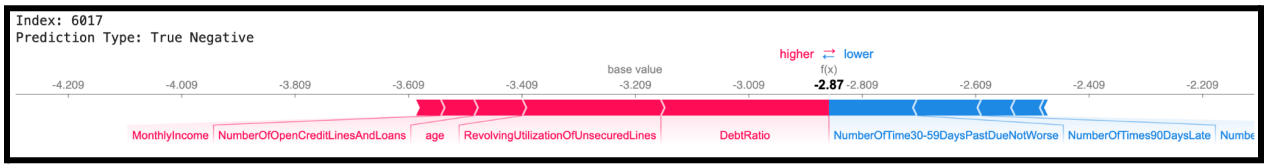
Figure #7: SHAP Explainer of True Positive Entry



For this particular true positive entry, the model predicted 0.26, whereas the base value is -3.209. The feature values causing the largest increased predictions are revolving utilization of unsecured lines, the number of times borrowers are 90 days late, and the number of times borrowers are 60-89 days late. This could be harmful to the predictive power of the ADS because

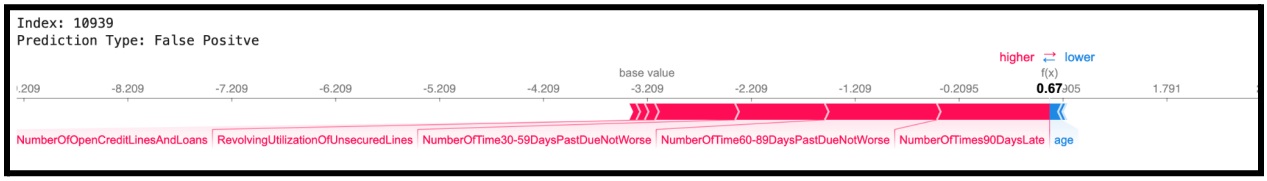
as we noted earlier many values of the number of times borrowers are late variables are illogical. However, given that the values are accurate this does make intuitive sense.

Figure #8: SHAP Explainer of True Negative Entry



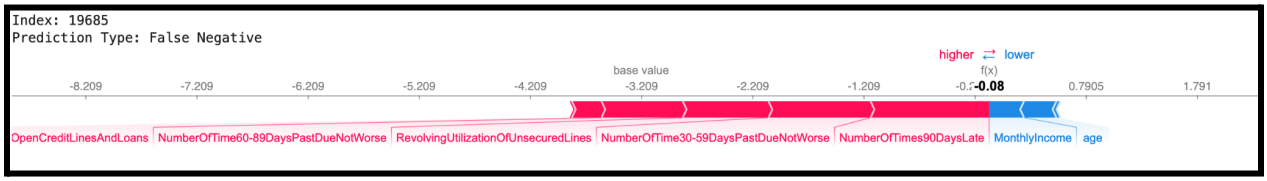
For this particular true negative entry, the model predicted -2.87, whereas the base value is -3.209. As one would expect, the feature values causing the largest increase in predictions in the true positive entry, played a key role in decreasing the predictions for the true negative entry. However, unlike the true positive entry the debt ratio played a key role in increasing the prediction of a positive instance.

Figure #9: SHAP Explainer of False Negative Entry



For this particular false negative entry, the model predicted 0.67, whereas the base value is -3.209. The feature values causing the largest increased predictions are the same as above. We must note that 0.67 is relatively close to being a negative instance, which means that any minor movement in the feature value could have caused the ADS to change this entry from a positive to a negative instance. Moreover, if the values for the number of times a borrower is late is inaccurate, then this can play a significant role in determining whether an observation is a positive instance.

Figure #10: SHAP Explainer of False Positive Entry



For this particular false negative entry, the model predicted -0.08, whereas the base value is -3.209. Similar to the False Positive Entry, we see that the model predicted a value close to 0 which implies that any minor change in the feature values could change the classification from positive to negative.

Summary

For the most part, we believe that the data was appropriate for this ADS as most of the input features objectively measured whether a hypothetical borrower will experience financial distress in the next 2 years. This is because the variables with highest feature importance according to the SHAP explainer didn't contain null values. We also believe that the dataset itself was robust as it contained enough observations (101503) to represent all types of borrowers. Furthermore, many of the input features weren't directly correlated to a potential protected attribute. Age was the one input feature which could have possibly been discriminatory, but as shown by our analysis using the AIF360, we found no violations of disparate impact. Also, as noted in prior sections some of the data for the feature of number of times past due date were illogical which would have resulted in many misclassifications.

Based on our examination of various fairness metrics, AIF360 framework, and the SHAP explainer we believe that the implementation is adequate in terms of fairness. Conditioned on the data given, it achieves group fairness based on the protected attribute of age. We know that a tradeoff exists between fairness and accuracy. The high false negative rate of 0.81 conveys this by demonstrating how the bias leans towards being a negative instance. This is beneficial for prospective borrowers who are being classified by this ADS as they are given the benefit of the doubt. Most likely, the employers themselves wouldn't find this ADS useful due to its high miss rate.

We would also like to commend the decision maker for following methods of data privacy. We can state this because we cannot identify any particular prospective borrowed used in the data

We would not be comfortable deploying this ADS in the public sector and industry at this particular iteration due to its high miss (false negative) rate. However, this belief would change if the following improvements were made in the data collection, processing, and analysis methodology. First, the data could have been improved by having more instances of borrowers under the age of 25, to see if statistical parity was truly achieved with both younger and older borrowers. Second, if the original dataset included race we would have been able to verify whether group fairness was achieved based on that particular protected attribute. Third, independently verifying every value of each observation such that no illogical values existed would considerably improve the accuracy of the classification. Fourth, if the ADS were to be modified such that the miss rate was significantly lower than the employers would feel much more comfortable in deploying this ADS to make decisions.

Appendix

Table 1: Null Counts for Each Input Feature

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101503 entries, 0 to 101502
Data columns (total 12 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   Unnamed: 0                            101503 non-null  int64
 1   SeriousDlqin2yrs                      0 non-null       float64
 2   RevolvingUtilizationOfUnsecuredLines  101503 non-null  float64
 3   age                                   101503 non-null  int64
 4   NumberOfTime30-59DaysPastDueNotWorse  101503 non-null  int64
 5   DebtRatio                             101503 non-null  float64
 6   MonthlyIncome                         81400 non-null   float64
 7   NumberOfOpenCreditLinesAndLoans       101503 non-null  int64
 8   NumberOfTimes90DaysLate               101503 non-null  int64
 9   NumberRealEstateLoansOrLines          101503 non-null  int64
 10  NumberOfTime60-89DaysPastDueNotWorse  101503 non-null  int64
 11  NumberOfDependents                    98877 non-null   float64
dtypes: float64(5), int64(7)
memory usage: 9.3 MB
```

Figure 1: Value Distribution of Age

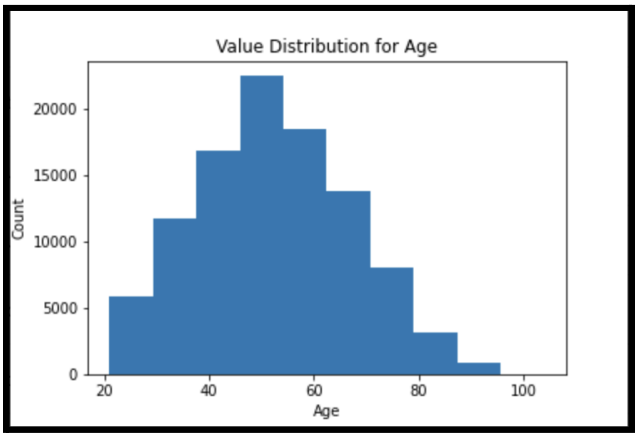


Figure 2: Value Distribution of Monthly Income

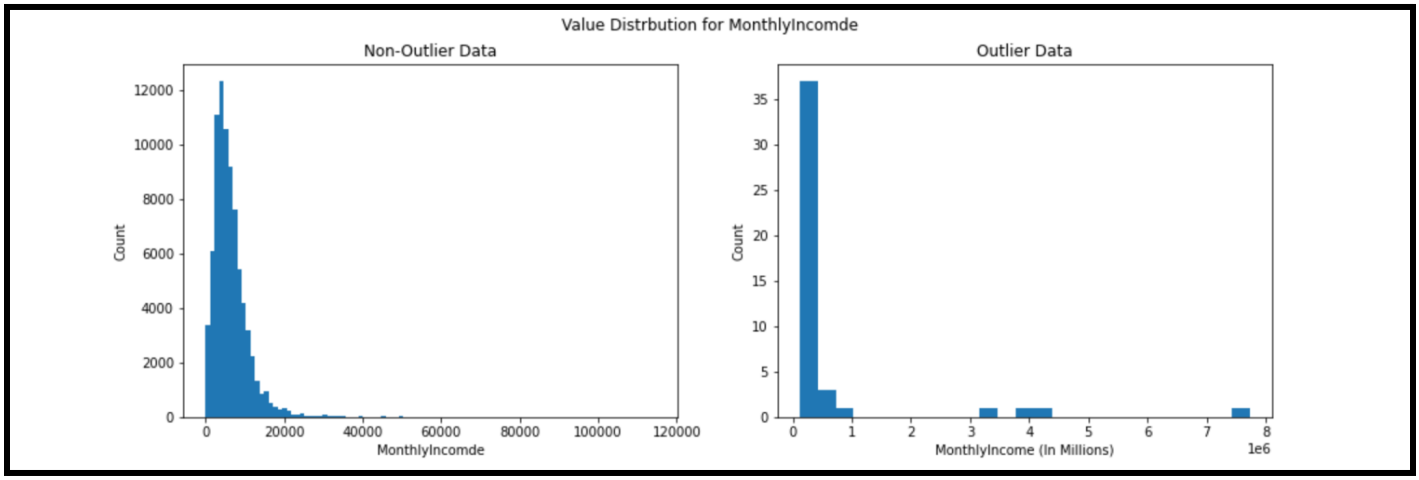


Figure 3: Value Distribution of Debt Ratio

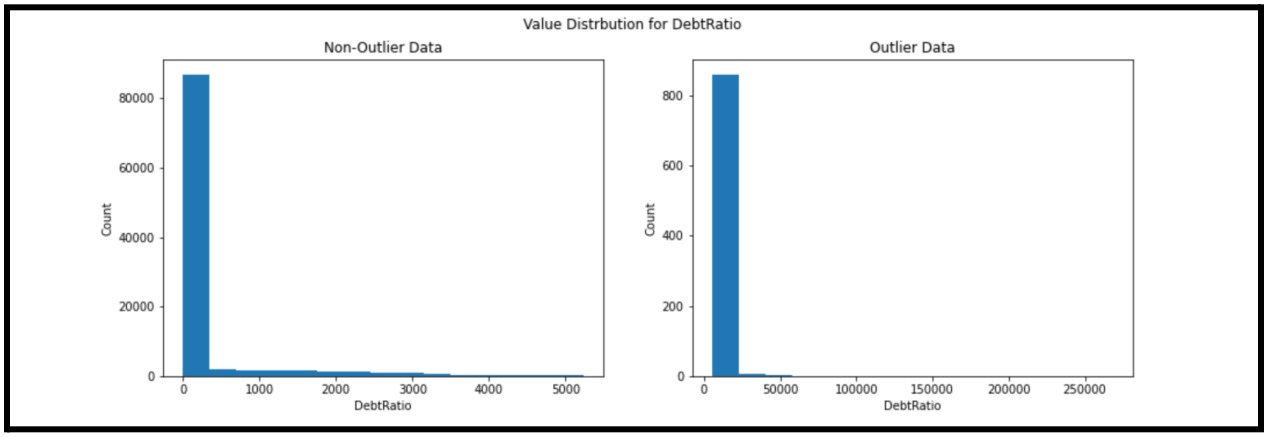


Figure 4: Value Distribution of Number of Dependents

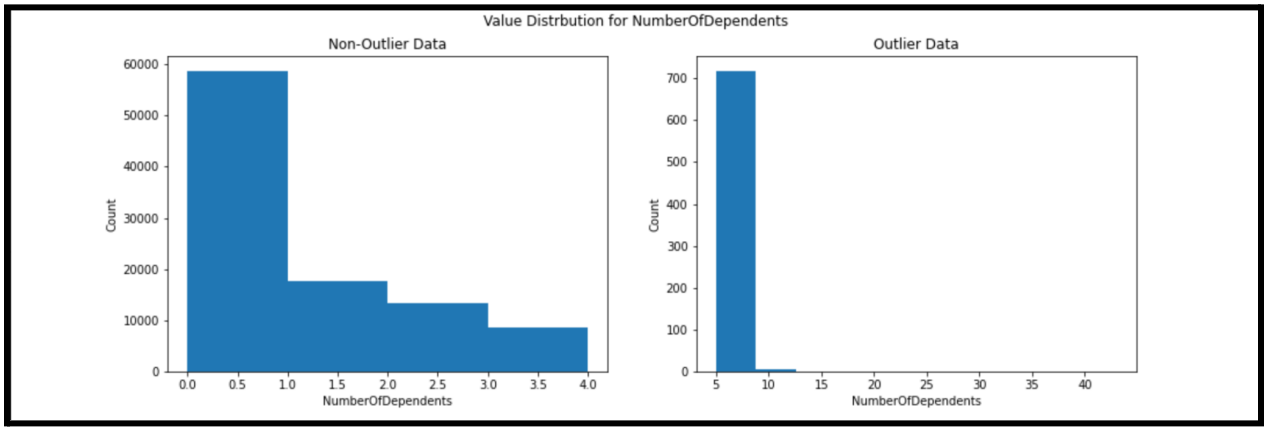


Figure 5: Value Distribution by Number of Open Credit Lines and Loans

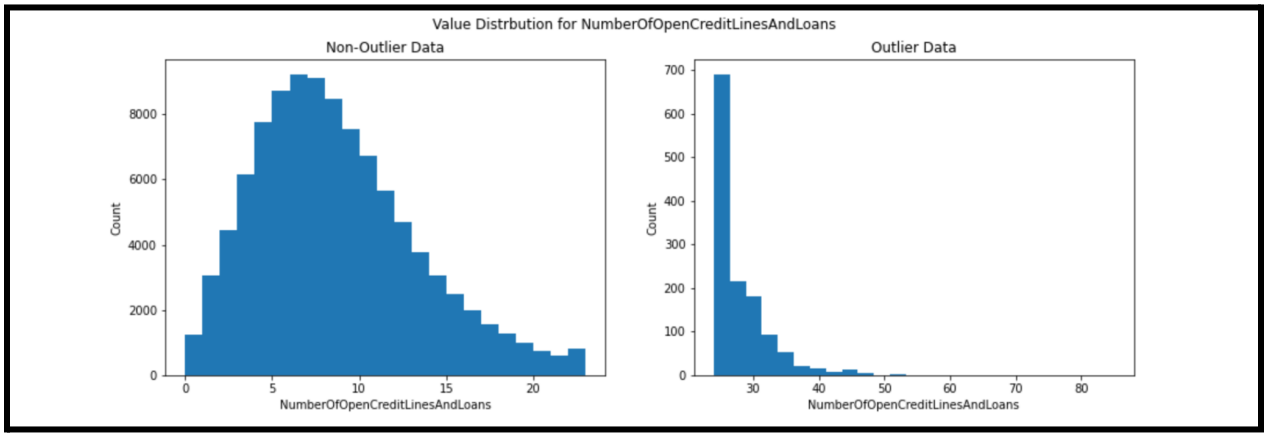


Figure 6: Value Distribution of Number of Time 30-59 Days Past Due Date But Not Worse in the Last Two Years

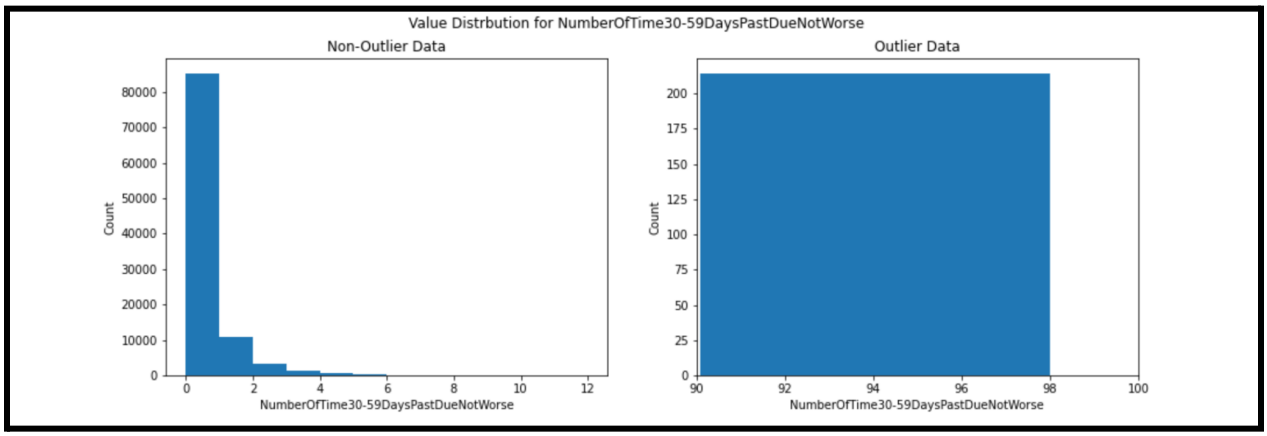


Figure 7: Value Distribution of Number of Times 60-89 Days Past Due Date But Not Worse in the Last Two Years

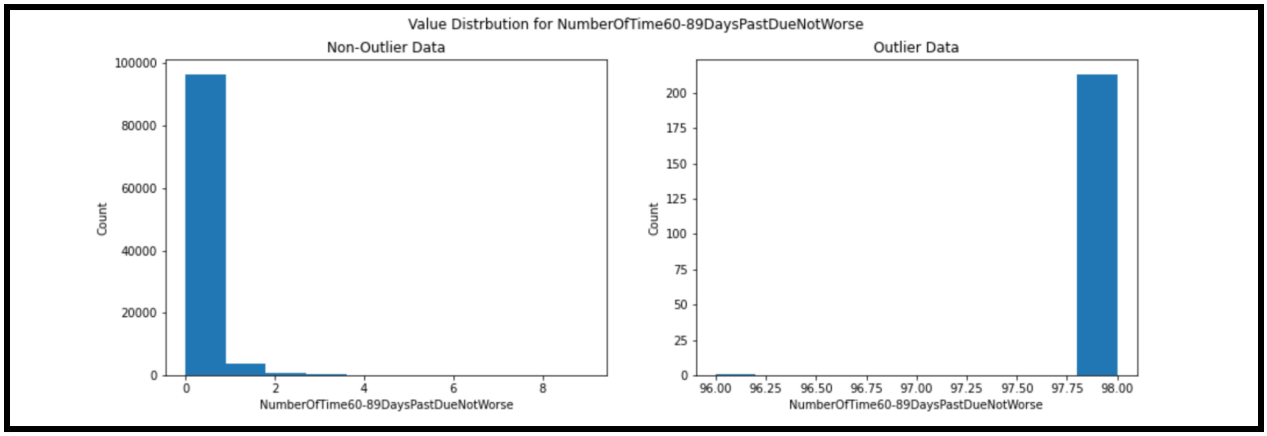


Figure 8: Value Distribution of Number of Times 90 Days Past Due Date But Not Worse in the Last Two Years

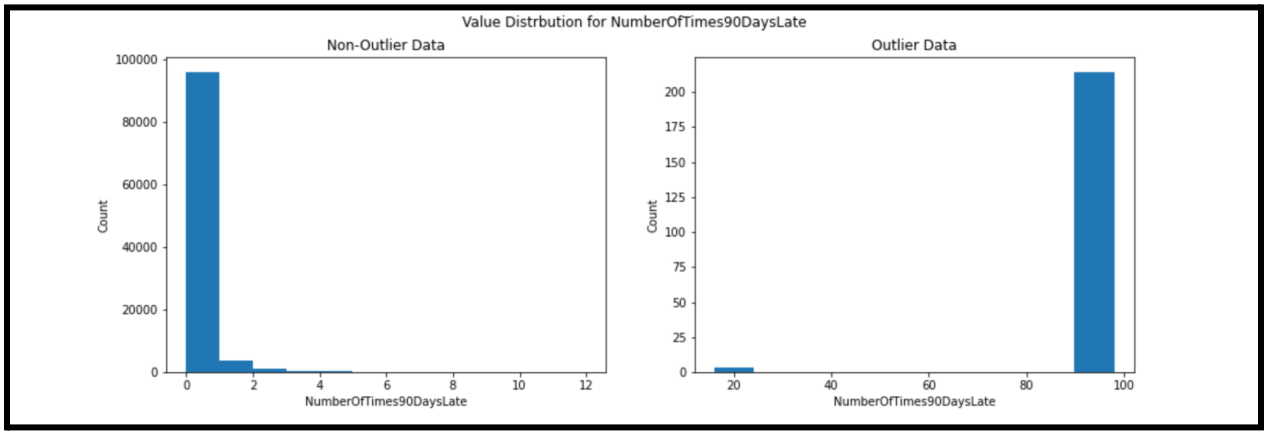


Figure 9: Value Distribution of Number of Real Estate Loans of Lines

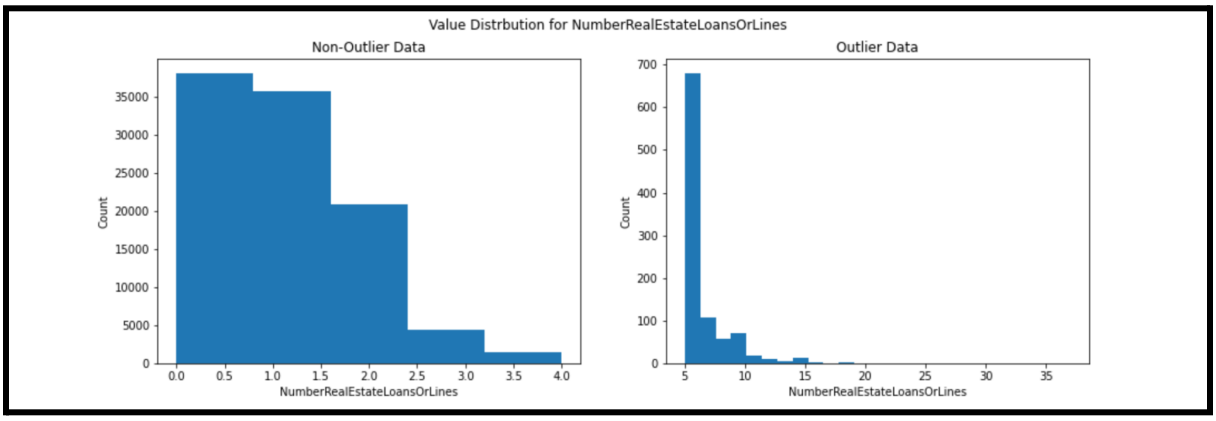


Figure 10: Value Distribution of Revolving Utilization of Unsecured Lines