**Project Report Draft 1**

Responsible Data Science

Project Partners: Kieren Gill, Mimi Chen

## Background: general information about chosen ADS

*A. What is the purpose of this ADS? What are its stated goals?*

This ADS is a model that borrowers can use to help them make financial decisions. The ADS works to improve credit scoring algorithms by predicting “the probability that somebody will experience financial distress in the next two years” (Kaggle). With this model, borrowers can better understand what they need to do to improve their credit scores. The fairness of the model is extremely important in this situation because those who belong to underrepresented or historically disadvantaged groups may receive unfair results. The developer of this ADS minimizes the impacts of gender, religion, nationality, and age.

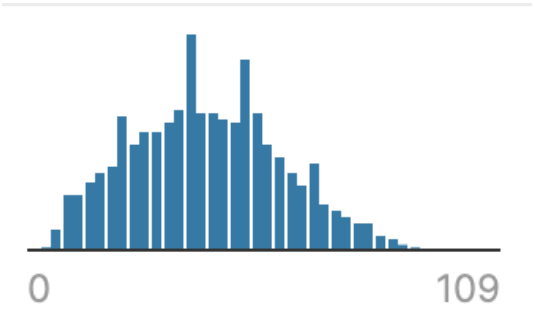
*B. If the ADS has multiple goals, explain any trade-offs that these goals may introduce.*

The stakeholders in this ADS are the bank and the borrowers. There is an inherent tradeoff when it comes to correcting the model for bias. For example, if the model’s false-negative rate is minimized, the banks will benefit by having fewer people default or experience financial distress. If there is a high false-positive rate, borrowers who would have had a chance to receive a loan may not be entitled to do so anymore.

## Input and Output

*A. Describe the data used by this ADS. How was this data collected or selected?*

We have data on 250,000 borrowers. We trained the ADS with data from 150,000 borrowers.



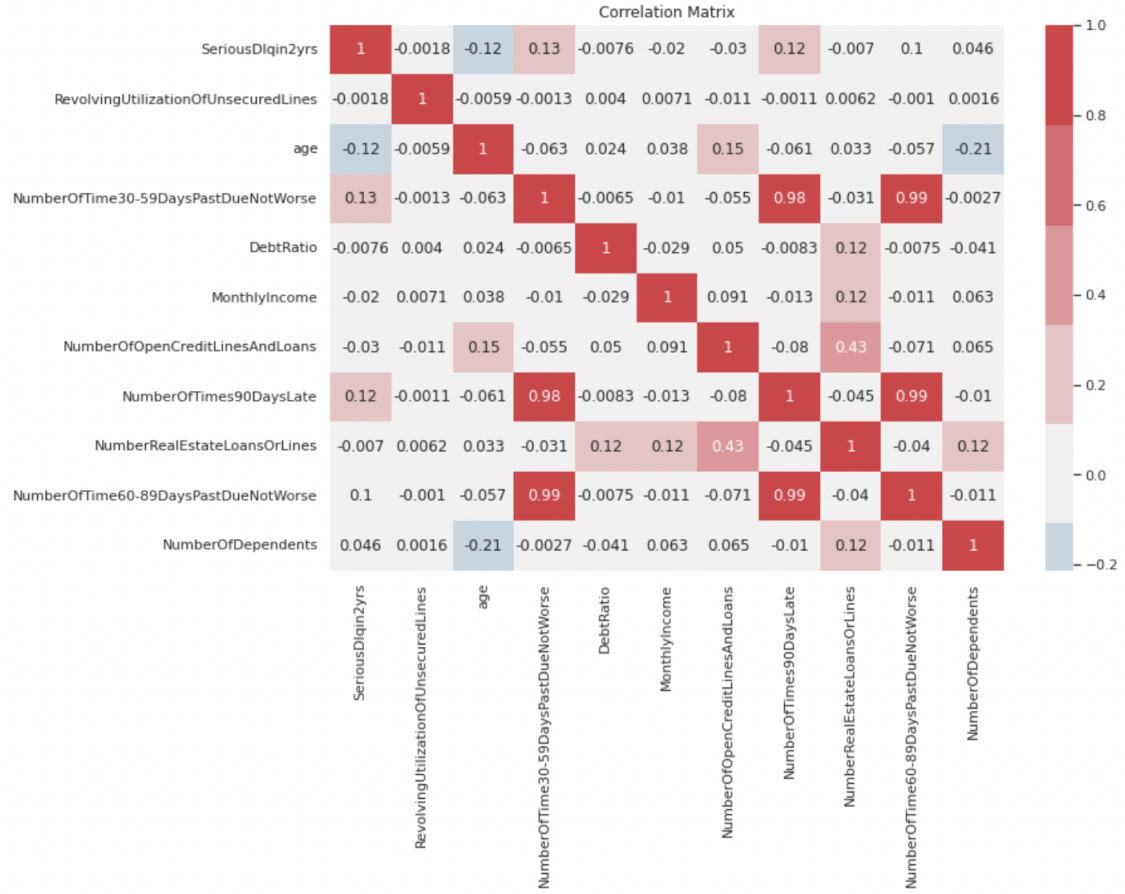

Although Kaggle did not provide information on where this data was collected from, his distribution shows the ages of the borrowers in the training dataset. This could be reflective of the population because older people are less likely to borrow money as opposed to people in the early stages of their career (for loans, mortgages, etc).

*B. For each input feature, describe its datatype and give information on missing values and on the value distribution. Show pairwise correlations between features if appropriate. Run any other reasonable profiling of the input that you find interesting and appropriate.*

Input data type for the 11 input features:

```
RangeIndex: 150000 entries, 0 to 149999
Data columns (total 11 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   SeriousDlqin2yrs                      150000 non-null  int64
 1   RevolvingUtilizationOfUnsecuredLines  150000 non-null  float64
 2   age                                   150000 non-null  int64
 3   NumberOfTime30-59DaysPastDueNotWorse  150000 non-null  int64
 4   DebtRatio                             150000 non-null  float64
 5   MonthlyIncome                         120269 non-null  float64
 6   NumberOfOpenCreditLinesAndLoans       150000 non-null  int64
 7   NumberOfTimes90DaysLate               150000 non-null  int64
 8   NumberRealEstateLoansOrLines          150000 non-null  int64
 9   NumberOfTime60-89DaysPastDueNotWorse  150000 non-null  int64
 10  NumberOfDependents                    146076 non-null  float64
```

Pairwise Correlation through a heatmap:




Missing values: Only 2 features have NaN values (Monthly income and number of dependents). This could be due to customers not wanting to declare these personal information.

We have 29k observations with >= 1 feature having NaN values, corresponding to 20% of the dataset, which is quite significant. We will have to impute these NaN values rather than dropping observations with NaN. MonthlyIncome and NumberOfDependents have right tail skews, so median imputation is preferred to be robust to outliers.

*C. What is the output of the system (e.g., is it a class label, a score, a probability, or some other type of output), and how do we interpret it?*

The output of the system is a probability assigned to each individual. This probability is interpreted as the probability that the individual will experience financial distress in the next two years.

**Implementation and Validation**

*A. Describe data cleaning and any other pre-processing*

**Exploratory Data Analysis**

Dropping NaNs:

Many learning algorithms cannot handle missing values, hence we need to handle missing values by dropping these observations, or imputing them. Imputing will increase the bias in our model, but dropping too many observations leads to a much smaller dataset for training.

Checking for outliers:

Many of the financial features have outliers (extreme outliers = > 3* interquartile range), with a right-tail skew. Hence, we should either use models that are robust to outliers (e.g. tree-based models), or transform the features accordingly (e.g. Box-Cox transformation).

Checking for multicollinearity:

Detecting multicollinearity: 30, 60 and 90 days past due are highly correlated with each other (close to 1). This may impact performance if the learning algorithm used assumes independence in dependent variables. We can either use models that are robust to multicollinearity (e.g. tree-based models), or use feature selection / regularization methods to use just one of these features.

Handling duplicates:

Getting counts of number of rows by the number of duplicates: Most severe = 1 observations with 12 instances. Overall, the training set has not too many duplicates, hence will not likely bias the model much. These duplicates are only a problem if they came from non-random error, e.g. duplicate entry into database. In a real scenario, we should dig deeper to identify if these are errors or coincidences. In this situation, we will not remove these duplicates and assume they are correct.

**Data Preprocessing**

Based on the EDA, we have found the need for imputation and scaling. Hence, we assemble a pipeline of imputation (median) and scaling (Box-Cox via PowerTransformer), testing out different classifiers to compare model performance.

*B. Give high-level information about the implementation of the system*

We train several baseline models using different learning algorithms to get a sense of what learning algorithm best performs during k-fold cross validation. In the interest of time, we perform model selection with minimal hyperparameter tuning, and select the model with best performance on cross-validation score. Ideally, this selection process happens after hyperparameter tuning for all of the models.

Based on the baseline model performances, XGBoost algorithm performed the best (highest AUC with lowest standard deviation across 3 folds). To improve performance, we tune the model hyperparameters using randomized search, which does random search over the range of hyperparameters. This is faster than grid search, which exhaustively searches over all possible hyperparameters in the range provided.

*C. How was the ADS validated? How do we know that it meets its stated goal(s)?*

We evaluate model performance using ROC AUC, which is the area under the receiver operating characteristic curve.

From the model classification report, we see that its recall for the positive class is high at 84%. This is preferred, as per business use cases where type II errors are more costly than type I errors. Overall, its ROC is 86.6% on the held out test set, which is similar to the average 86.4% on cross-validation set, showing that the model can perform well with out-of-sample data (i.e. is not overfitted).

**Outcomes**

*A. Analyze the effectiveness (accuracy) of the ADS by comparing its performance* across different subpopulations.

The XGBoost area under the curve of receiving operating characteristics for the test results was 0.86083. The AUC-ROC tells us whether the model is able to distinguish between classes, and the higher the score the better the performance of the machine learning model is at distinguishing positive and negative classes. We also want to analyze the performance of the model across different age groups to ensure fairness.

*B. Select one or several fairness or diversity measures, justify your choice of these measures for the ADS in question, and quantify the fairness or diversity of this ADS.*

We will analyze how the model ensures individual and group fairness through its predictions. The developer noted that features that are deemed unfair such as gender, religion, nationality, and age will be minimized. In addition, looking at the feature importance table below, the key attributes that determine how likely a borrower will default on a loan repayment are RevolvingUtilizationOfUnsecuredLines and NumberOfTime30-59DaysPastDueNotWorse. These measures a fair because they do not discriminate against underrepresented groups. In contrast, attributes like MonthlyIncome and age are of less importance.

| | features | feature_importance |
|---|---|---|
| 0 | RevolvingUtilizationOfUnsecuredLines | 0.260455 |
| 2 | NumberOfTime30-59DaysPastDueNotWorse | 0.208461 |
| 6 | NumberOfTimes90DaysLate | 0.194711 |
| 8 | NumberOfTime60-89DaysPastDueNotWorse | 0.123161 |
| 7 | NumberRealEstateLoansOrLines | 0.064353 |
| 1 | age | 0.040401 |
| 5 | NumberOfOpenCreditLinesAndLoans | 0.038258 |
| 3 | DebtRatio | 0.025377 |
| 4 | MonthlyIncome | 0.023473 |
| 9 | NumberOfDependents | 0.021350 |

*C. Develop additional methods for analyzing ADS performance: think about stability, robustness, performance on difficult or otherwise important examples (in the style of LIME), or any other property that you believe is important to check for this ADS.*

One way we can analyze the ADS is model explainability. Its initial goal was to build a model that borrowers can use to help them make better financial decisions by analyzing the key attributes that affect their predicted risk of defaulting on a loan. From the features importance section, we can see that the ADS provides borrowers with actionable, quantitative insights on

what to prioritize in order to increase their chances of getting a loan. This table provides insight to the borrowers in a way that is digestible and informative.

**Summary**

*A. Do you believe that the data was appropriate for this ADS?*

Yes, we believe that the data was appropriate for this ADS because it provides information that is relevant to the ADS' goals. This ADS is a model that borrowers can use to help them make financial decisions, so information in the input dataset pertaining to income, spending, and borrowing habits, are all related to an individual's financial position.

*B. Do you believe the implementation is robust, accurate, and fair? Discuss your choice of accuracy and fairness measures, and explain which stakeholders may find these measures appropriate.*

We think that the implementation is robust and accurate, but not necessarily fair. ROC AUC is a better measure than mean accuracy because the dataset is imbalanced, and the implementation is accurate because the ROC is 86.6% on the held out test set, which is similar to the average 86.4% on cross-validation set. However, it might not necessarily be fair because it does not control for 'age' which is a feature that could lead to biases and should have been a protected attribute.

*C. Would you be comfortable deploying this ADS in the public sector, or in the industry? Why so or why not?*

We would not be comfortable deploying this ADS in either the public sector or in the industry solely because it has not been adjusted for fairness. Although it is possible that the model is not biased against certain age groups, we do not know for sure unless we run a disparate impact analysis with the protected attribute to see if there are biases present within the model. If there are significant biases beyond a certain threshold, we should correct for these biases using preprocessing techniques such as AIF360's disparate impact remover before training the model using the dataset.

*D. What improvements do you recommend to the data collection, processing, or analysis methodology?*

As mentioned above, we should run disparate impact testing with certain features (namely age) to see if there are biases present in the model. If there are biases beyond a certain threshold, we should correct for these biases using preprocessing techniques such as AIF360's disparate impact remover before training the model using the dataset.