,

Home Credit Default Risk

Ashley Tsao AT4880@NYU.EDU

Sage Harley SNH362@NYU.EDU

DS-301 Final Report

1. Background

The purpose of this ADS is to predict whether a customer will default on their home loans based on a variety of factors. The main reason is so that they can ensure that "clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful." The goal of this model is to help this institution find more people who might otherwise be neglected by traditional lenders. This makes our project on this ADS more interesting, since the entire point of this model originally was to ensure equal and fair access to loans even if they would otherwise have been unfavored.
There is likely going to be a tradeoff between higher income and more favored regions vs lower income and less favored regions, since sacrificing fairness in this Kaggle competition could lead to marginally higher accuracies. This could mean that the algorithms posted and the one we end up reviewing might be biased against certain groups. This is especially true if they did not take any measures to account for fairness or to penalize the algorithm if it does not perform fairly.

2. Input & Output

This data was collected by various financial institutions which reported client data to the Credit Bureau. It includes around 346 columns including the target column and ID of each customer. The data itself is split into 10 different csvs with one csv detailing what each column is. In particular, we focused on the application_train.csv and application_test.csv files. Application_train.csv and application_test.csv files contain 121 columns with features about the applicant, such as their gender, total income, credit amount of the loan requested, whether or not they own houses/flats and cars, and more. The application_train file includes an extra column, target, which is the output. The value is 0 for will repay loan on time and 1 for will have difficulty repaying loan.

Out of the 122 files in the application_train.csv file, 64 columns contained missing values. Here we have the first 20 columns containing missing values with the number of missing values and the percentage of missing values in the respective columns.

	Missing Values	% of Total Values
COMMONAREA_MEDI	214865	69.9
COMMONAREA_AVG	214865	69.9
COMMONAREA_MODE	214865	69.9
NONLIVINGAPARTMENTS_MEDI	213514	69.4
NONLIVINGAPARTMENTS_MODE	213514	69.4
NONLIVINGAPARTMENTS_AVG	213514	69.4
FONDKAPREMONT_MODE	210295	68.4
LIVINGAPARTMENTS_MODE	210199	68.4
LIVINGAPARTMENTS_MEDI	210199	68.4
LIVINGAPARTMENTS_AVG	210199	68.4
FLOORSMIN_MODE	208642	67.8
FLOORSMIN_MEDI	208642	67.8
FLOORSMIN_AVG	208642	67.8
YEARS_BUILD_MODE	204488	66.5
YEARS_BUILD_MEDI	204488	66.5
YEARS_BUILD_AVG	204488	66.5
OWN_CAR_AGE	202929	66.0
LANDAREA_AVG	182590	59.4
LANDAREA_MEDI	182590	59.4
LANDAREA_MODE	182590	59.4

Figure 1: Number and Percentage of Missing Values in Each Column

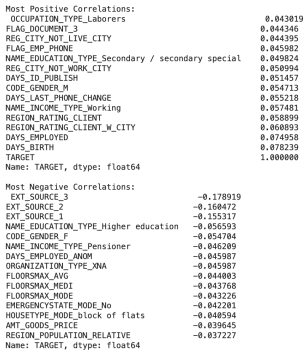
We also found the number of columns with each data type. There are 65 columns in the "float64" data type. There are 41 columns with the "int64" data type and there are 16 columns with the "object" data type.
There is a csv titled HomeCredit_columns_description which has information on every input variable. We did some more exploratory analysis to learn more about it. First, we found before that there were columns of "float32", "int64", and "object" data types. For the columns of "object" data types, we found the number of classes in each of those columns.

```
NAME_CONTRACT_TYPE              2
CODE_GENDER                     3
FLAG_OWN_CAR                    2
FLAG_OWN_REALTY                 2
NAME_TYPE_SUITE                 7
NAME_INCOME_TYPE                8
NAME_EDUCATION_TYPE             5
NAME_FAMILY_STATUS              6
NAME_HOUSING_TYPE               6
OCCUPATION_TYPE                18
WEEKDAY_APPR_PROCESS_START      7
ORGANIZATION_TYPE              58
FONDKAPREMONT_MODE              4
HOUSETYPE_MODE                  3
WALLSMATERIAL_MODE              7
EMERGENCYSTATE_MODE             2
dtype: int64
```

Figure 2: Number of Classes in "Object" Columns

Since a majority of these columns have a small number of classes, we did Label Encoding for any categorical variables with only 2 categories and One-Hot Encoding for any categorical variables with more than 2 categories to deal with these entries. 3 columns were label encoded and after one-hot encoding, the training features shape is (307511, 243) and the testing features shape is (48744, 239). Since there were more columns in the training data, we removed the ones that weren't also in the testing data.

To continue with exploratory data analysis, we found the most positive and most negative correlations in the application_train dataset. We see that the column with the highest correlation is the "DAYS_BIRTH" column. This is a bit strange because the column represents the age in days of the client at the time of the loan in negative days. We see that the correlation is positive while the value of this feature is actually negative, meaning that as the client gets older, they are less likely to default on their loan. This doesn't make much sense so we took the absolute value of the feature and then the correlation became -0.078. As the client gets older, they tend to repay their loans on time more indicating a negative linear relationship with the target. To visualize this, we produced a plot with the distribution as well as the average failure to repay loans by age bracket. We can see that



```
Most Positive Correlations:
 OCCUPATION_TYPE_Laborers                              0.043019
FLAG_DOCUMENT_3                                        0.044346
REG_CITY_NOT_LIVE_CITY                                 0.044395
FLAG_EMP_PHONE                                         0.045982
NAME_EDUCATION_TYPE_Secondary / secondary special      0.049824
REG_CITY_NOT_WORK_CITY                                 0.050994
DAYS_ID_PUBLISH                                        0.051457
CODE_GENDER_M                                          0.054713
DAYS_LAST_PHONE_CHANGE                                 0.055218
NAME_INCOME_TYPE_Working                               0.057481
REGION_RATING_CLIENT                                   0.058899
REGION_RATING_CLIENT_W_CITY                            0.060893
DAYS_EMPLOYED                                          0.074958
DAYS_BIRTH                                             0.078239
TARGET                                                 1.000000
Name: TARGET, dtype: float64

Most Negative Correlations:
 EXT_SOURCE_3                                         -0.178919
EXT_SOURCE_2                                          -0.160472
EXT_SOURCE_1                                          -0.155317
NAME_EDUCATION_TYPE_Higher education                  -0.056593
CODE_GENDER_F                                         -0.054704
NAME_INCOME_TYPE_Pensioner                            -0.046209
DAYS_EMPLOYED_ANOM                                    -0.045987
ORGANIZATION_TYPE_XNA                                 -0.045987
FLOORSMAX_AVG                                         -0.044003
FLOORSMAX_MEDI                                        -0.043768
FLOORSMAX_MODE                                        -0.043226
EMERGENCYSTATE_MODE_No                                -0.042201
HOUSETYPE_MODE_block of flats                         -0.040594
AMT_GOODS_PRICE                                       -0.039645
REGION_POPULATION_RELATIVE                            -0.037227
Name: TARGET, dtype: float64
```

Figure 3: Most Positive and Negative Correlations

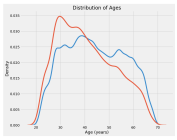


Figure 4: Distribution of Ages (Red line = Target is 1, Blue line = Target is 0)

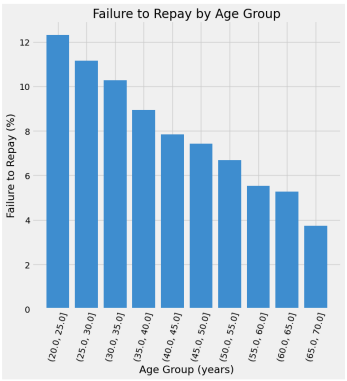


Figure 5: Failure to Repay Loan by Age Group

Next, we looked at the most negative correlations. These were "EXT_SOURCE_3", "EXT_SOURCE_2", and "EXT_SOURCE_1." The documentation file provided says these columns represent a "normalized score from external data source". To learn more about these variables, we found their correlations and created a heat map.

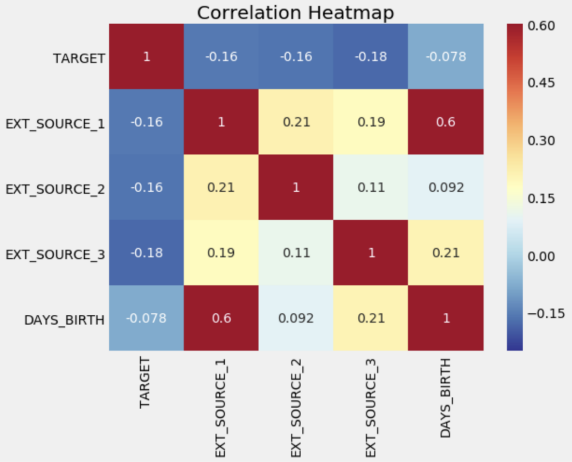


Figure 6: Negative Correlation Heat Map

We can see that the "EXT_SOURCE" features are negatively correlated with the target. This can indicate that when the value of the "EXT_SOURCE" increases, it means that the client is more likely to repay the loan. We even see that "DAYS_BIRTH" has a positive correlation with "EXT_SOURCE_1", which could mean that one of the factors in that source is age. To have more visualization about this, we produced the distribution plot.

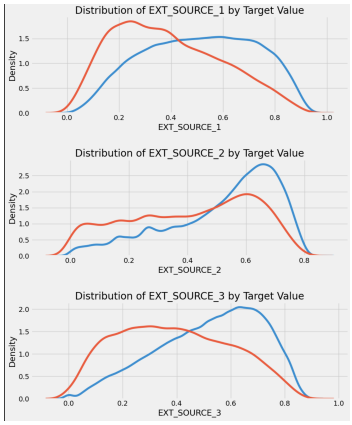


Figure 7: Distribution Plot for "EXT_SOURCE"

Lastly, we made a pair plot of the "EXT_SOURCE" variables and "DAYS_BIRTH" variable. This is a helpful tool since we can see the relationships between multiple pairs of variables as well as distributions of single variables.

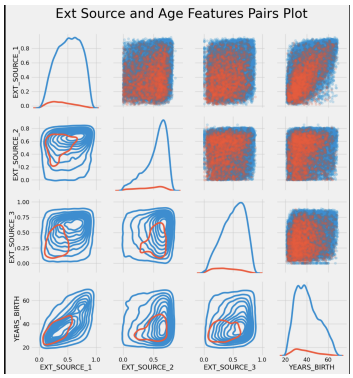


Figure 8: Pair Plot

In this plot, the red indicates loans that weren't repaid and blue are paid. We see different relationships within the data. There is a slight positive linear relationship between the "EXT_SOURCE_1" and the "DAYS_BIRTH" (or equivalently YEARS_BIRTH), furthermore indicating that this feature takes into account the age of the client.
The application_train file includes an extra column, target, which represents whether or not they got their loan. The value is 0 for will repay loan on time and 1 for will have difficulty repaying loan.
The output of the system is a binary classification of each customer as being predicted to default or being predicted to not default and the scores of each model for this Kaggle competition are based on the area under the ROC curve. We can interpret this as the model predicting customers to default or not default on the test set, and then essentially seeing a metric for accuracy to the true values of the test set.

3. Implementation and validation

The dataset came already pretty clean and complete. To verify this, we checked the data types of the columns and checked the column descriptions to ensure they logically matched the column title. For missing values, it was an option an option to drop columns with high percentage of missing values but we wouldn't have known ahead of time which columns could be helpful or important. Thus, we kept all the columns during preprocessing and filled in the missing values using imputation when building the machine learning model.
This system implemented for the Home Credit Default Risk competition on Kaggle is centered on creating predictive models to determine the likelihood of an applicant defaulting on a loan. The process involves begins with feature engineering. In this step, they create and select additional variables that are derived from the raw data in order to improve the model's performance. In particular, their feature engineer strategies include polynomial features and domain knowledge features. Polynomial features include creating new features by taking powers and interaction terms of existing variables in order to

capture more complex relationships in the data. Domain knowledge features generated features based on logical financial insights like the ratios of credit to income, annuity to income, and the relation of days employed to age. This could provide more context on an applicant's financial burden and stability.

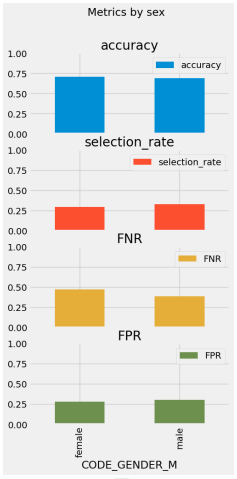
In addition, several machine learning model's were utilized to predict the probability of loan defaults. Our ADS with logistic regression as a baseline model. Logistic regression is a linear model for binary classification that can predict probabilities. It's fast, interpretable and is a good starting point. To implement this, the ADS applied it to the dataset after preprocessing steps that included imputation of missing values and scaling of features to a uniform range using MinMaxScaler. The regularization parameter C was adjusted to reduce overfitting.

Then the ADS used a Random Forest Classifier to improve on the previous baseline model's performance. Random Forest is an ensemble of decision trees, typically providing high accuracy and robustness by averaging multiple deep decision trees, trained on different parts of the same training set. The random forest model was trained using all features from the dataset including the existing and engineered. It was configured with multiple trees (n_estimators=100) and allowed to run in parallel (n_jobs=1) for faster computation. This model provides insights into feature importance, which can help understand which features contribute the most to predicting the target variable.

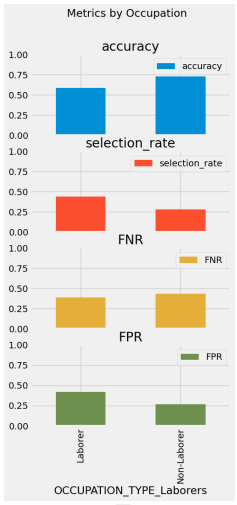
The models were validated and tested with the primary metrics of AUC-ROC and Feature importance analysis. The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) is really useful for binary classification problems and is robust against imbalanced class distributions. It also measures the ability of a classifier to distinguish between classes. On the other hand, feature importance analysis is particularly useful for random forest. The importance provided for each feature can help in understanding the predictive power of the features. By comparing the model's performance with the baseline models across different feature sets, we can determine if the engineering efforts are providing improvements. In summary, the ADS is validated and its effectiveness is confirmed through a combination of internal and external evaluations. Together, the AUC-ROC and feature importance analysis allowed for robust evaluation of the model's ability to correctly predict outcomes.

4. Outcomes

Based on the 2 below plots of subpopulations, we can see that the model performs significantly better for laborers rather than non-laborers, part of this is from the following top correlations of our data:



(a) Differences in Fairness metrics by Gender



(b) Differences in Fairness metrics by Occupation Type test

Figure 9: Figures for comparison of Fairness metrics

Table 1: Segment of the Most Positive Correlations

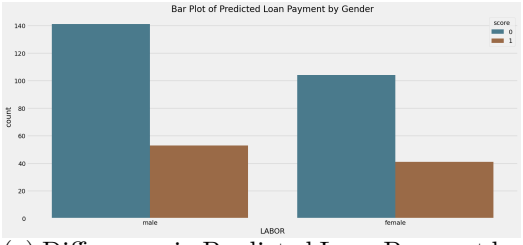
Feature	Correlation
OCCUPATION_TYPE_Laborers	0.043019
FLAG_DOCUMENT_3	0.044346
REG_CITY_NOT_LIVE_CITY	0.044395
FLAG_EMP_PHONE	0.045982
NAME_EDUCATION_TYPE_Secondary / secondary special	0.049824
REG_CITY_NOT_WORK_CITY	0.050994
DAYS_ID_PUBLISH	0.051457
CODE_GENDER_M	0.054713

Table 2: Segment of the Most Negative Correlations

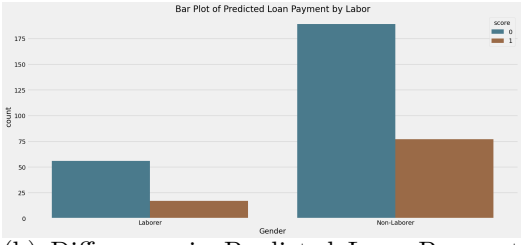
Feature	Correlation
EXT_SOURCE_3	-0.178919
EXT_SOURCE_2	-0.160472
EXT_SOURCE_1	-0.155317
NAME_EDUCATION_TYPE_Higher education	-0.056593
CODE_GENDER_F	-0.054704

These tables indicate the way in which being male and being a laborer could be significantly related to getting a loan from this company. The model itself does not make any attempt to adjust for fairness and I chose these accuracy metrics because they

emphasize the False Negatives and False Positives where people might otherwise have been able to get a loan because they could actually pay it, and where people really should not have gotten a loan when the model claimed they could get one. What we notice immediately is that because of the highly unbalanced dataset, the model is far more likely to label someone as a False Negative than as a True Positive. Since a vast majority of the data is not able to pay back their loans, the model is extremely biased towards being more risky and being more likely to falsely label someone as being able to pay even if they would have otherwise been unable to pay. This is also highly ironic because the whole goal of this Kaggle competition in particular was to be able to provide loans to those who were otherwise shunned off from the banks, and now it is doing the opposite of what the banks are presumably doing. This model does in some ways help non-laborers in particular but those who follow in the Laborer occupation often end up still being disregarded by the model. The correlations above do emphasize the way in which Laborers SHOULD have been more likely to have receieved loans given their positive correlation with the target value, but likely were devalued because the Laborer occupation is associated with lower incomes, working in regions where the bank might have rated more lowly, and because Laborers might still be at an age where they have kids and much fewer assets than other occupations.

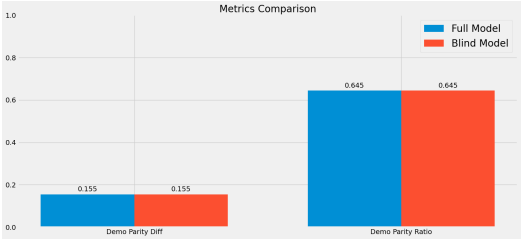


(a) Differences in Predicted Loan Payment by Gender

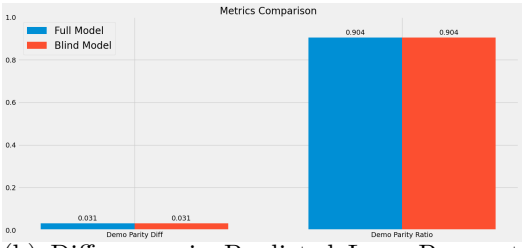


(b) Differences in Predicted Loan Payment by Occupation Type

Figure 10: Figures for comparison of Fairness metrics



(a) Differences in Predicted Loan Payment by Gender

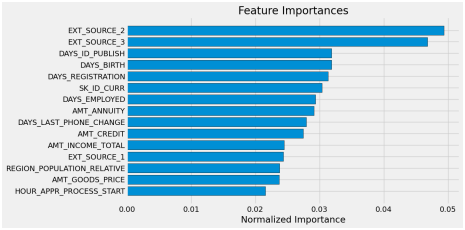


(b) Differences in Predicted Loan Payment by Occupation Type

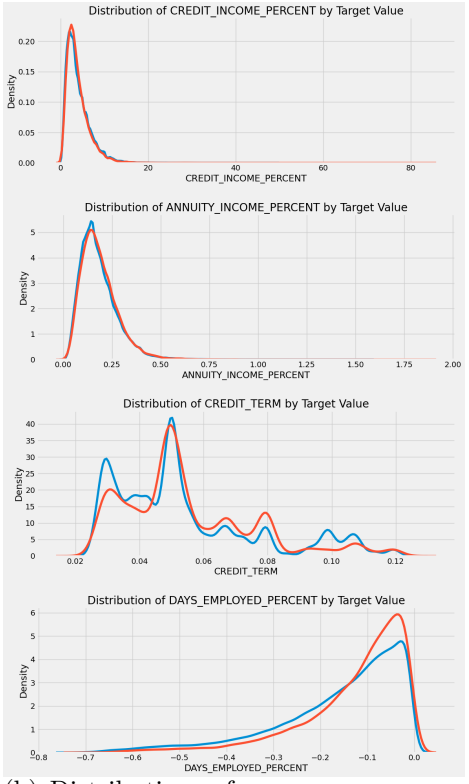
Figure 11: More Figures for comparison of Fairness metrics

The plots above highlight major differences between 2 very key demographics in the model: gender and laborers vs non-laborers. We find that though on a gender level it remains relatively fair, there is an enormous disparity by occupation type. Laborers are

found to be significantly less likely to receive a loan from this company than occupations other than Laborers, and this remains true even in the various types of models that this ADS employs. For reference again the ADS deployed a few types of random forests and a logistic regression model. All of which followed this same level of bias. There is interestingly more males being predicted as not being able to pay their loan. One could argue there is almost a very very slight bias towards women in this particular model. The Demographic parity difference and ratios are here to highlight the way in which Gender is relatively balanced but for Laborers there is a sizeable disparity. These metrics were chosen in particular because of it's clear and highly understandable criterion for fairness. And for a clearly outlined visualization of the discrimination that is occurring in those model against laborers.



(a) Feature Importances for the model



(b) Distributions of some domain continuous features for the model

Figure 12: Figures for comparison of Fairness metrics

Above we have plots emphasizing the feature importance of the top features for the model as well as the distributions of some of the continuous domain specific features in our data. What this is here to emphasize is that the distributions of these features can often be very ugly and highly skewed, meaning our model can predict based on cases near the median but extremities will be much harder to predict.

5. Summary

I believe this data was appropriate because the huge amount of data that this model is analyzing and the relatively high quality of the data. It also contains features of interest for fairness like gender, occupation types, income levels, region ratings, etc. For this ADS this makes perfect sense since the goal of the Kaggle competition was to emphasize helping underprivileged communities and people have access to loans when they otherwise might not have been able to have access to loans. I think the model accomplishes a relatively good job at giving groups and people who otherwise would have been looked over for loans a chance, but there is still much more work they could have done to improve fairness. But again improving fairness might have reduced the accuracy of the model, which means they would have been further down on the charts and having a lower chance of winning the prize money.

Yes, I believe it is relatively robust and fair. The model does a decent job of maintaining fairness between some demographics like gender and even in the Laborer vs Non-Laborer demographic differences it was not the biggest differences that could have otherwise been possible if it was just the bank's associates looking at this data. Based on our accuracy metrics however the model definitely has a pretty substantial amount of FPRs and FNRs in the demographics analyzed, in both men/women and laborers/non-laborers.

I would be comfortable deploying this ADS in the public sector given it is relatively fair and even in the case of Laborers it isn't necessarily the most unfair algorithm that one could imagine. Especially if you imagine this algorithm being deployed ALONGSIDE some associate or alongside government officials looking to improve loan accessibility, I think this algorithm could actually help a lot. It is not the most cautionary model, preferring to take risks and label people as being able to pay loans, but that can otherwise be an advantage to those interest groups who might want a more loose and loan accessible economy.

I would probably recommend that the model itself should be readjusted to be more fair in the same way that we have done in this class. Given that the goal is to improve fairness in the Kaggle competition it should also be the goal that we improve inter-group fairness and parity to ensure that the algorithm is truly of assistance to everyone and to those underrepresented folk who otherwise would never have a chance at getting a loan for their home, their families, or any other instances.

Contributions

- Ashley: Ran the code from the ADS, completed the input/output as well as implementation/validation sections
- Sage: made the fairness and accuracy plots, completed summary and outcomes sections