Predicting the Likelihood of Loan Default

Saumit Bhave

TABLE OF CONTENTS

1. Introduction

1.1 Background

Access to credit is an important step towards financing many of the consumer goods that are necessary in today's modern society. Lenders have to take a variety of factors into consideration to determine if an applicant is creditworthy. Generally, a good measure is an applicant's credit score, which in the United States range from 300 - 850, with higher numbers indicating a more creditworthy applicant. This number takes into account payment history, credit utilization, length of credit history, new credit, and credit mix. Applicants with excellent credit scores can potentially get access to higher limits on credit and take advantage of lower interest rates. However, people with poor or non-existent credit history often do not get approved for loans through traditional financial institutions, and fall victim to predatory lending practices. These predatory loans with exorbitant interest rates are often the only option for low income applicants that need the money immediately. If they cannot pay the loan back on time, the high interest rate quickly compounds and only adds to their debt, creating a vicious cycle.

1.2 Problem Statement & Project Goals

The fundamental problem lenders face is assessing and minimizing risk when providing consumer loans to a diverse swath of the population. This problem becomes more difficult when it is necessary to look beyond a standardized metric such as a credit score to determine whether to lend to an applicant. It is a vitally important task to have a system that allows those with a lower score or financial missteps in the past to have access to credit without falling prey to shady practices or being shut out of the loan process. The lending company Home Credit aims to provide credit for the segment of the population underserved by traditional lenders. With the help of machine learning models and data collected by Home Credit, we will aim to predict the risk of default on a Home Credit loan using a variety of metrics beyond credit history and credit score.

An accurate model that uses alternative data to predict risk of default can have far reaching implications for the global unbanked population. Those who have been unable to acquire credit through traditional banking institutions will have an opportunity to receive loans if they are deemed trustworthy based on the parameters we will determine in the model.

2. Data

The dataset will be acquired through Home Credit and consist of 307,511 records and 122 fields. Each record in the dataset represents one loan, and each feature represents information about the applicant. The target variable 1 represents a loan default and a 0 represents a loan that was repaid. This will be a supervised binary classification problem with imbalanced classes, since the applicants in the data set that paid back their loan is disproportionately high to the number that defaulted. Upon initial exploration, we see that the data is a mix of categorical and numerical features. The categorical variables will need to be handled with encoding methods before applying any machine learning models. There are also a lot of variables that seem to be closely related (i.e. COMMONAREA_AVG, representing the common areas of a house, and LIVINGAREA_AVG, representing the living areas of a house). As part of the feature selection process, we will have to check for features that are highly correlated with each other.

3. Data Wrangling

3.1 Overview

Data wrangling for this project mainly consisted of removing rows with null or invalid values, performing missing value imputation, and outlier treatment. Since the goal is to predict whether a loan will be repaid or not, it is important to examine the proportion of defaulters in the initial dataset. Doing so reveals that 282,686 (91.93%) borrowers repaid their loan, and 24,285 (8.07%) defaulted. This is an imbalanced class problem, and after cleaning the data, the goal will be to have a ratio of 3:1 for the majority to minority class.

3.2 Dropping Null/Invalid Values

Examining null value counts reveals that 67/122 columns have missing values, with the greatest number of missing values at 214,865 rows shared by the following columns: COMMONAREA_MODE, COMMONAREA_AVG, COMMONAREA_MEDI. Before we drop rows with null values, we will remove any rows where gender (CODE_GENDER) is not male (‘M’) or female (‘F’). Next, we will examine if there are any anomalies in age of applicant (DAYS_BIRTH) and number of days he or she has been employed (‘DAYS_EMPLOYED). Since the two columns are represented by negative days (i.e. -1000 days), any positive values will need to be examined. DAYS_EMPLOYED has a max value of 365,243, indicating that there are positive values present in that columns. We will remove all rows where DAYS_EMPLOYED > 0. We will use similar analysis for positive values in DAYS_REGISTRATION and DAYS_ID_PUBLISH.

After this initial data cleaning, we are down to 252,133 rows. To further reduce the size of the dataset while retaining a large percentage of remaining default TARGET rows, we will split the dataset into a default dataframe and non-default dataframe. We will drop all rows in the non-default dataframe where COMMONAREA_AVG is NaN. Next, we will combine the two separate dataframes, now containing 70,621 (76.38%) non-defaulters and 21,835 (23.61%) defaulters, which is close to our target ratio.

3.2 Missing Value Imputation

A large number of remaining columns with missing values will be imputed based on the specific columns. Columns containing numerical values will be imputed with the mean, mode, or zero depending on what is deemed appropriate. Categorical columns with missing values will be imputed with 'Other' or an existing label if that label makes up the majority of values. For example, 'block of flats' in the column HOUSETYPE_MODE is present in over 98% of rows, so missing rows will be imputed with 'block of flats'.

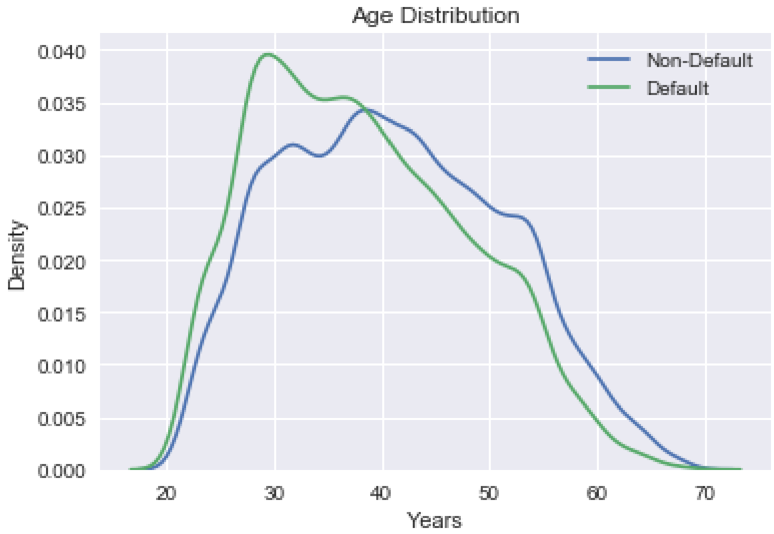
3.3 Outlier Treatment

The final step of data wrangling will be to perform outlier treatment. Each numerical category will be examined using the Pandas describe() method. For greater granularity, we will also look at percentile values for the bottom and top 1st, 5th, and 10th percentile. Depending on the extent of outlier values, we will drop rows on a certain percentile cutoff. Otherwise, we will drop rows on a value cutoff in the context of the column. For example, the max for number of children (CNT_CHILREN) is 19, while an overwhelming majority of rows have the values 0, 1, 2, and 3. Here we will drop rows where CNT_CHILDREN > 4. Sometimes, outliers will only be present on one end of the distribution. For example, we will drop the top percentile of AMT_ANNUITY. After performing outlier treatment and missing value imputation, we are left with a dataset containing 80603 rows, consisting of 61372 (76.14%) non-defaulters and 19231

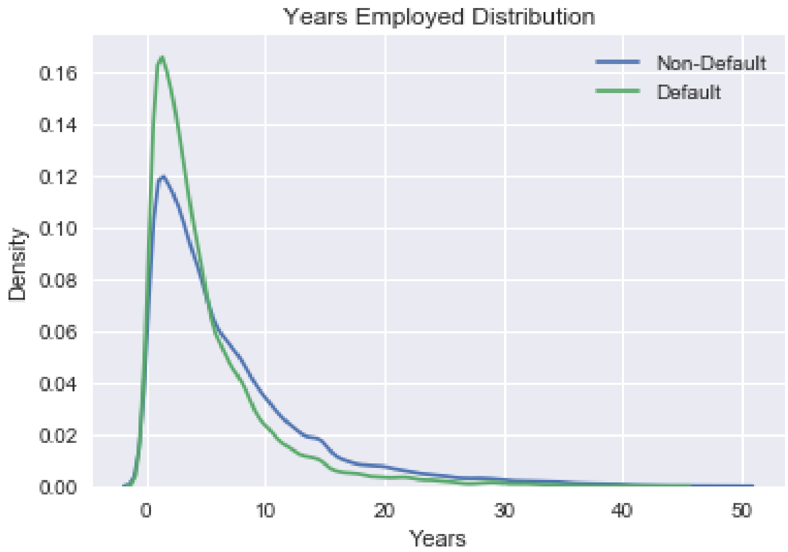
(23.86%) defaulters. We will create a csv file of this clean dataset for further exploratory data analysis.

4. Data Exploration

With our cleaned and reduced dataset, we will start exploratory data analysis by comparing rates of default by gender. Among males, the rate of default is 28.95%, and for females, the rate of default is 20.72%, indicating that gender might play a role. Next, we will explore the age of loan holders in each target group, with the assumption that younger applicants with less financial capital might have higher rates of default. As the kernel density estimation plot shows, younger applicants do indeed have higher rates of default than older loan holders.



Similar analysis for years employed shows that applicants with less years of employment have higher rates of loan default.



Further data exploration consists of examining distributions of monetary values such as total income, amount of credit borrowed, amount of annuity, and price of goods the loan was taken out for. Additionally, we will explore other features that may strain the finances of an average applicant, such as whether they own a car or real estate and the number of children they have. Finally, we will also examine the occupations of each applicant as well as their education level and see if there is a relationship to the target variable.

5. Feature Selection

5.1 Correlation to Response Variable

After exploratory data analysis, the next step will be to apply statistical methods to narrow down the feature set to only features that are deemed most relevant to predicting whether an applicant will default on his or her loan. We will start by finding out the top 20 positive and negative correlations in the feature set to the target variable. The top four features with the highest positive correlation all contain location based information about the applicant such as whether they live in the city they work. Some noteworthy features with a high negative correlation are age of the applicant and years of employment. This corroborates with our findings while doing exploratory data analysis. Next, we will use Weight of Evidence (WOE) and Information Value (IV) for variable selection.

5.2 Weight of Evidence & Information Value

WOE and IV are used to evaluate and analyze variables for a binary classification problem. These techniques are often used for variable selection in credit modeling scenarios, such as predicting the risk of default. Weight of Evidence and Information Value is calculated with the formula shown below:

$$WOE = \ln\left(\frac{\text{Event\%}}{\text{Non Event\%}}\right)$$

$$IV = \sum(\text{Event\%} - \text{Non Event\%}) * (\text{WOE})$$

Event % is calculated as the percent of loan defaulters and Non Event % is calculated as the percent of non-defaulters. After calculating the Weight of Evidence and Information Value for each variable, we will proceed with feature selection. The criteria for selection will be an IV value greater than 0.1 and less than 0.7. Additionally, we will consider features that fall outside this threshold but have a reasonable correlation with the target variable. Note that categorical variables do not have an IV value calculated and will need to be examined on an individual basis. Once this analysis is done, we are down to 43 independent variables.

5.3 Multicollinearity Analysis

Our final step will be to perform multicollinearity analysis. Due to the highly related nature of a lot of independent variables, it is reasonable to suspect that a large number of independent variables will be highly correlated with each other. The metric we will use to determine multicollinearity is Variance Inflation Factor (VIF). We will calculate the VIF by treating each independent variable as the dependent variable and fitting a linear regression for each feature. We will eliminate features that have a VIF greater than 5. Once the initial VIF scores have been calculated, we will determine a few features to eliminate and run VIF analysis again. This new set of scores will then be examined and new features will be eliminated. This iterative process will continue until all remaining features have a VIF less than 5. This yields a final dataset consisting of 80,603 records and 23 features.

6. Modeling

6.1 Encoding Categorical Variables

After feature selection, there are two remaining categorical variables that will need to be encoded before applying any machined learning models. The first is education type (amount of schooling applicant has) and the second is occupation type of the applicant. Education type has five unique values, making it a good candidate for One Hot Encoding. Occupation type has no clear ordinal positioning between different occupations, which also makes it a good candidate for One Hot Encoding. We will use the pandas function get_dummies() to encode both categorical variables with k-1 dummies created out of k categories. The dataset is ready for modeling.

6.2 Modeling Approach

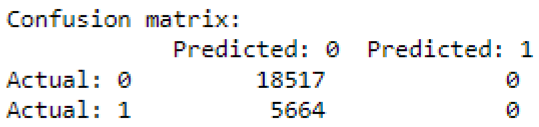
The dataset that will be used for modeling is imbalanced, with a majority class outnumbering the minority class with a ratio of approximately 3:1. For this situation, accuracy is not the best metric for model performance since predicting all applicants as non-defaulters will lead to a baseline accuracy score of 76.14%. Instead, we will use precision and recall (sensitivity), which are defined below:

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives} \qquad precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Precision and recall are useful metrics for an imbalanced binary classification problem because both do not take into account the model’s ability to predict true negatives – we are only concerned with the model’s ability to predict the minority class. Another key metric we will use is the area under the curve (AUC) for the receiving operating characteristic curve (ROC), which plots the true positive rate, or sensitivity, against the False Positive Rate at different thresholds. The ROC AUC can be used to compare overall model performance across different models. We will use a train test split of 70/30, and apply our models to the training set. We will primarily use supervised classification algorithms.

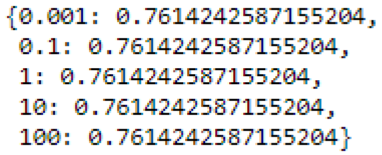
6.3 Logistic Regression

We start with a baseline logistic regression model with a threshold of 0.5. This has poor predictive accuracy. As the confusion matrix below shows, the model accurately classifies 18,517 non defaulters and misclassifies 5,664 defaulters as non-defaulters. This baseline model predicts all applicants to have paid back their loan.



```
Confusion matrix:
            Predicted: 0  Predicted: 1
Actual: 0          18517             0
Actual: 1           5664             0
```

Precision and recall for Target = 1 (applicant defaulted on loan) is zero because this model has not predicted any true positives. Let’s try logistic regression again with different values for C, which determines the strength of regularization. The accuracy scores for different values of C are shown below:

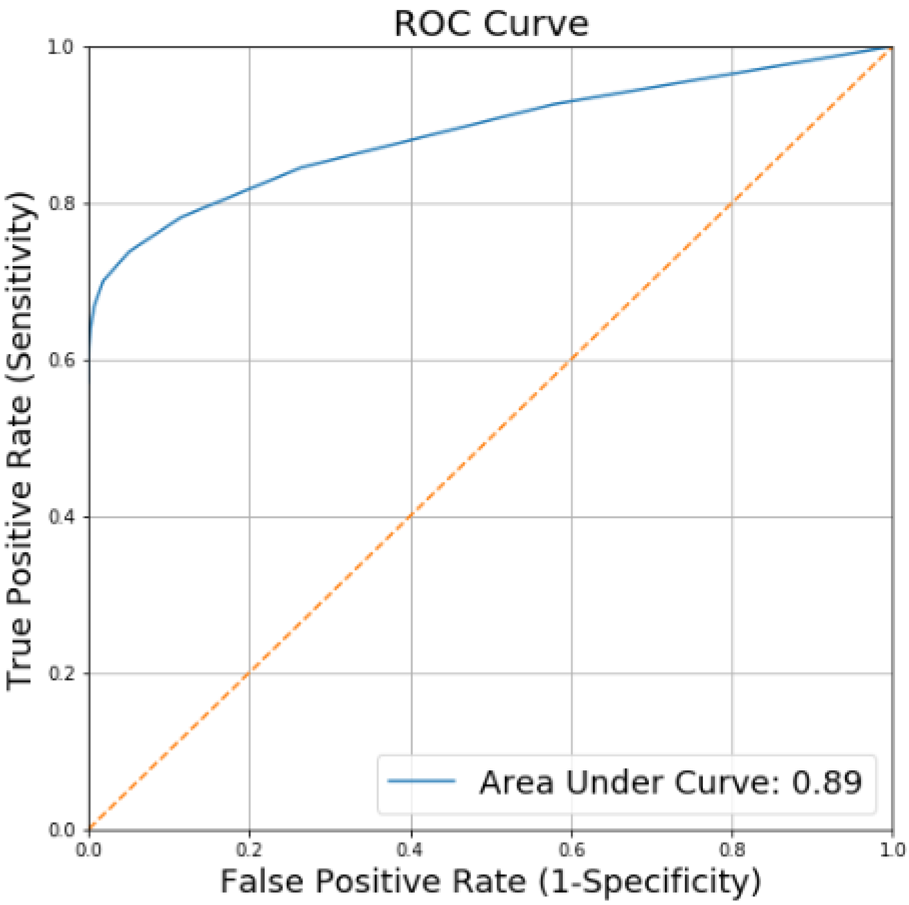


```
{0.001: 0.7614242587155204,
 0.1: 0.7614242587155204,
 1: 0.7614242587155204,
 10: 0.7614242587155204,
 100: 0.7614242587155204}
```

Accuracy score remains the same regardless of the C value. Accuracy is the same as our baseline model (0.7614), indicating that all predicted values were also Target = 0 (applicant paid back loan). There does not seem to be a strong linear relationship between the target and independent variables. We will try decision tree based models, which can learn non-linear relationships.

6.4 Random Forest

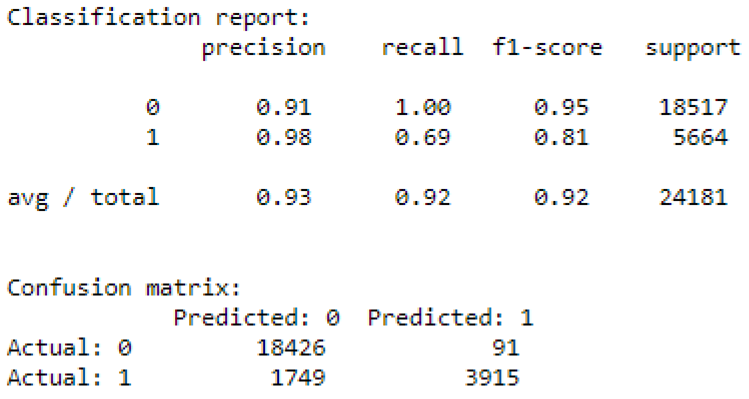
Instead of using a single decision tree, which can overfit and have high variance, we will use a random forest classifier. Random Forest is an ensemble of decision trees, which combines multiple decision trees to reduce variance and improve prediction accuracy. Let’s start with a baseline random forest model. The baseline model yields an ROC AUC score of 0.89. The ROC curve is plotted below:



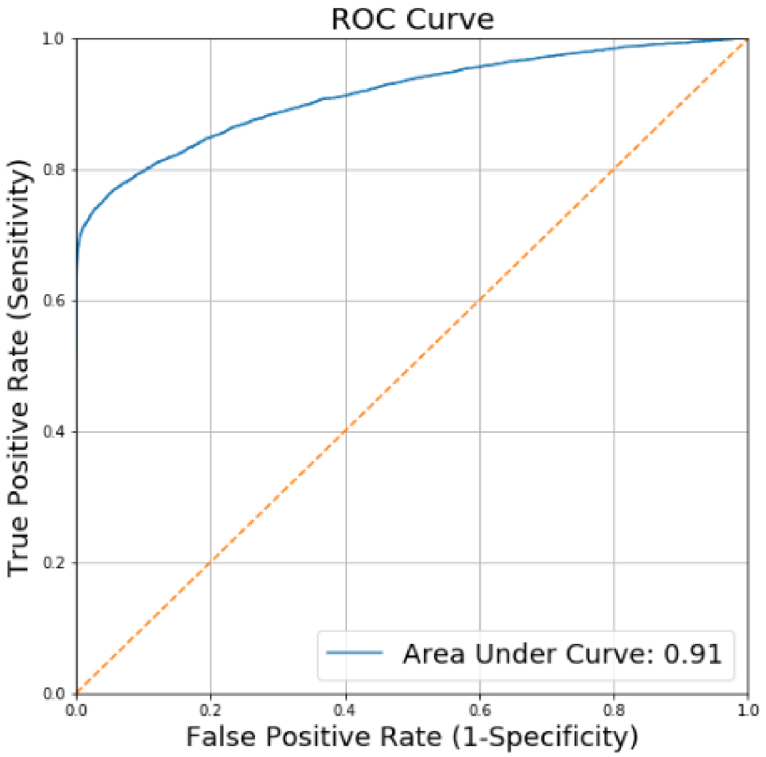
The baseline prediction performs much better than logistic regression. Let’s see if we can use hyperparameter tuning to further improve performance. We will adjust the following hyperparameters:

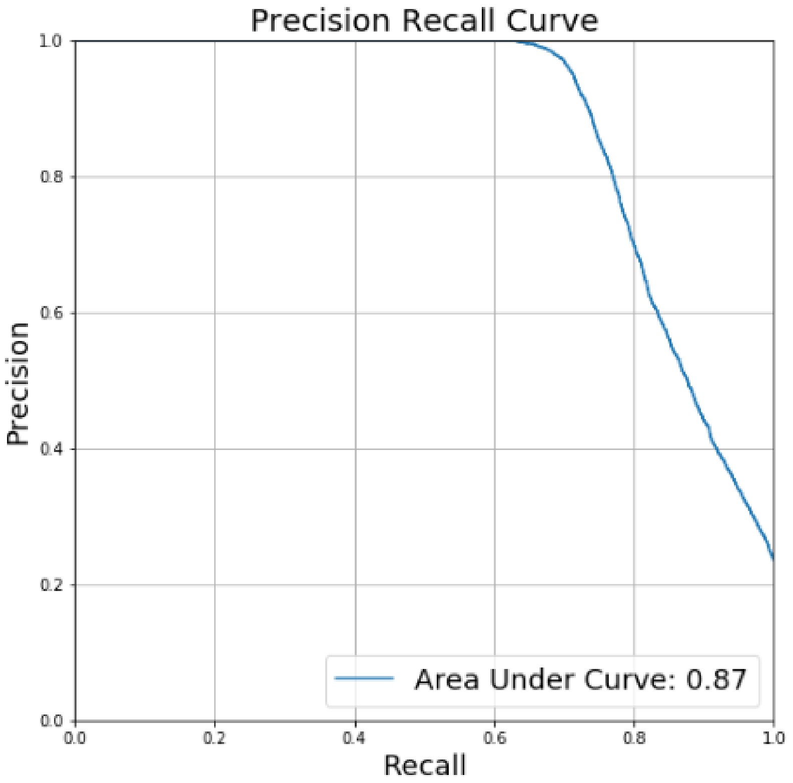
- bootstrap: method for sampling data points (with or without replacement)
- max_depth: max levels in each decision tree
- max_features: max features considered for splitting a node
- min_samples_leaf: min number of data points allowed in a leaf node
- min_samples_split: min number of data points put in a node before node is split
- n_estimators: number of trees in the forest

Since it will be too computationally expensive to exhaustively try every single parameter combination, we will use RandomizedSearchCV to take a random subset of possible parameter combinations. After running randomized search using 5 fold cross-validation, we will use the best estimator to fit the training set. Doing so results in an ROC AUC score of 0.91, which is an improvement over our baseline. As the classification report and confusion matrix below show, the model had high precision and decent recall for predicting loan defaulters.



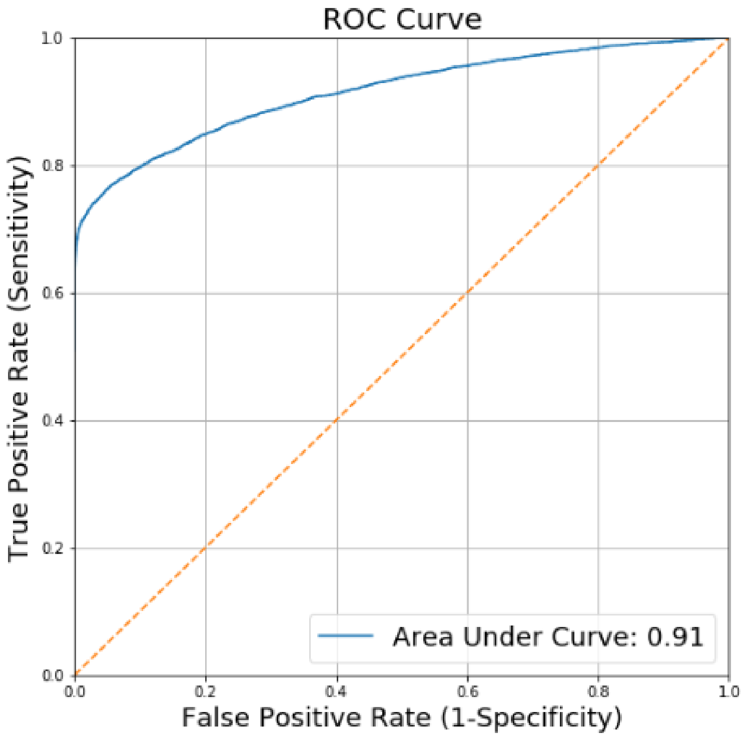
```
Classification report:
              precision    recall  f1-score   support

          0       0.91      1.00      0.95     18517
          1       0.98      0.69      0.81      5664

avg / total       0.93      0.92      0.92     24181


Confusion matrix:
            Predicted: 0  Predicted: 1
Actual: 0          18426            91
Actual: 1           1749          3915
```



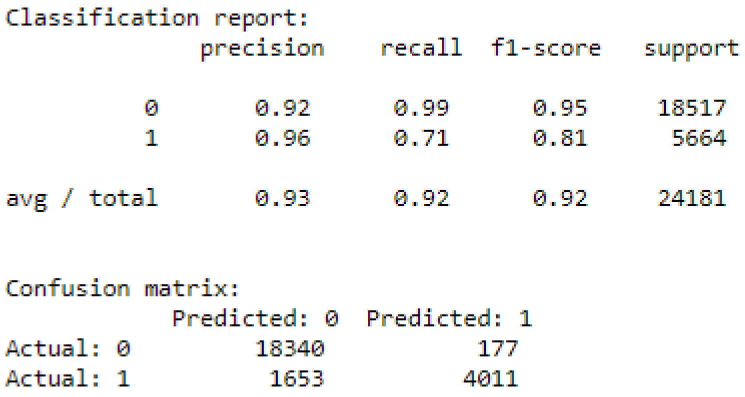


6.5 XGBoost

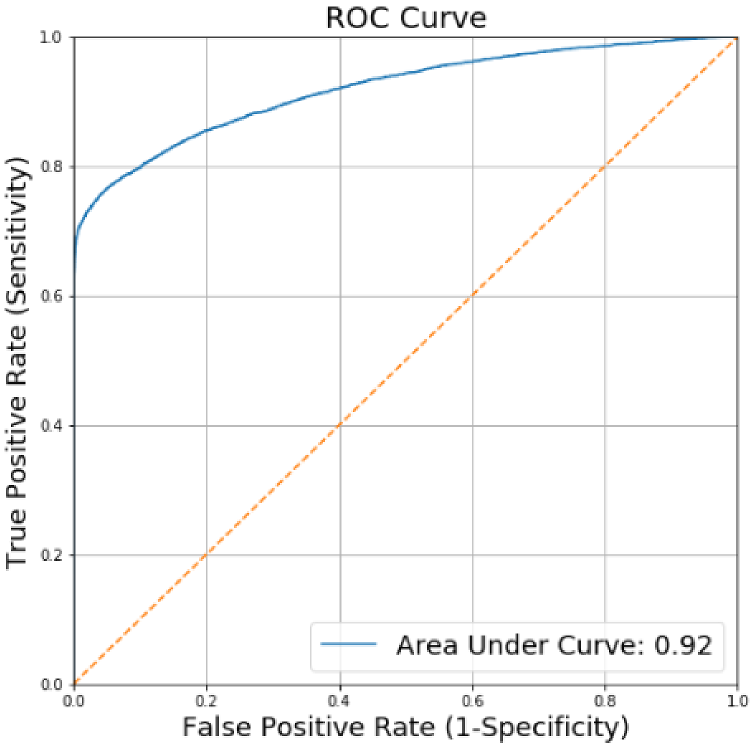
Next, we will use Extreme Gradient Boosting (XGBoost), which is another tree ensemble model that works on the principle of boosting. Boosting combines weak learners and takes into account the outcome of the previous instance to weigh outcomes of the next instance. The baseline ROC AUC using an XGBClassifier model yields a score of 0.91. This is the best out of the box performance thus far. The ROC AUC plot is shown below:

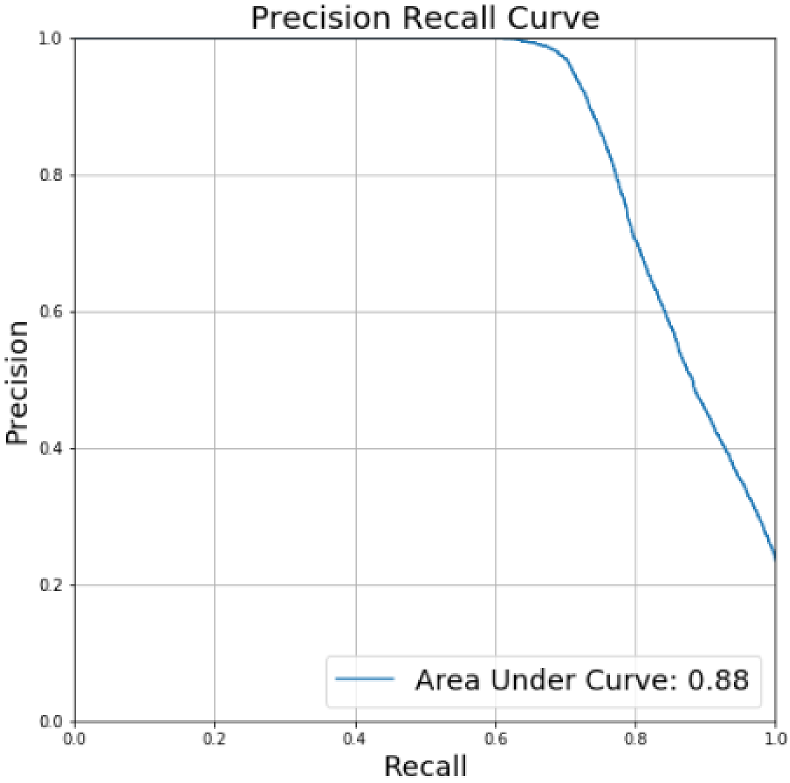


We will try to tune the parameters to see if there is an improvement in performance. We will tune max_depth, min_child_weight, n_estimators, and learning_rate and use GridSearchCV with three folds. The ROC AUC score for the best estimator is 0.92, which is the best performing model. The Precision Recall AUC score is 0.88, indicating a high proportion of true positives are being correctly classified by the model. The classification report, confusion matrix, and respective plots for ROC AUC curve and Precision Recall Curve are shown below:



```
Classification report:
             precision    recall  f1-score   support

          0       0.92      0.99      0.95     18517
          1       0.96      0.71      0.81      5664

avg / total       0.93      0.92      0.92     24181


Confusion matrix:
            Predicted: 0  Predicted: 1
Actual: 0          18340           177
Actual: 1           1653          4011
```


ROC Curve
True Positive Rate (Sensitivity)
False Positive Rate (1-Specificity)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Area Under Curve: 0.92


Precision Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Area Under Curve: 0.88

7. Conclusion

7.1 Client Recommendations

We have applied different models with varying results of predictive power. As shown above, tree based algorithms produce the best ROC AUC as well as the best precision recall on the test data. We recommend that Home Credit should apply both a Random Forest and XGBoost model to a new applicant and average the result. Due to the imbalanced nature of the problem, it is difficult to create discrete decision boundaries between defaulters and non-defaulters – there will be applicants in both categories with similar profiles. Therefore, Home Credit should not rely solely on the model score to assess whether a new applicant should be approved for a loan. Additionally, instead of only evaluating the binary output of the model (default or non-default), it will be essential to also evaluate the predicted probability of default for each new applicant.

Home Credit should use the model results as a supplement to human review. The results should form one component of a holistic profile review that takes into account factors that cannot be captured by the model. This model can also be useful for reducing manpower and manual review. Based on client configuration, the model can be used to automatically approve loans where the predicted probability of default is low. This threshold can be set by the client based on business requirements.

7.2 Future Work

We have applied various machine learning techniques but by no means have we performed an exhaustive study. In the future, we can try to apply other learning models and do more exhaustive hyperparameter tuning. We can also try to incorporate other data Home Credit may have or data from external agencies to build a more robust model.