Report: Predicting Mortgage Approvals from Government Data

Executive Summary

This document presents findings from data, which was adapted from the Federal Financial Institutions Examination Council's (FFIEC) satisfying the Home Mortgage Disclosure Act. In particular, the project aims to present the data analysis, machine learning modeling used, results and conclusion for predicting whether a mortgage application was accepted (meaning the loan was originated) or denied according to the given dataset and understanding the factors that increase that decision. The project utilizing distinct entries of around 50,000 of the applicants' particulars to consider whether demographics, location, property type, lender, and other factors are indicative of trends whether a mortgage application was accepted or denied.

The project is carried out in a streamline manner while digging into the underlying patterns in HMDA data. Firstly, the inputs in train values and the labels data are joined together and a separate test values data while verifying records based on lender, a unique identifier for each individual loan-making institution; secondly, data analysis is carried out to view dataset summary statistics, the distributions and relationships among different features; then based on the observations, classification and predictive models are built with benchmarking, feature selection to choose those influential characteristics, and hyper-parameter tuning to optimize the model. Leveraging visualization capability of Jupyter Notebook on Microsoft Azure Cloud, Python Matplotlib, Pandas, and scikit-learn machine learning libraries. In this report, the following three modelling concept was employed:

- What to do with missing values.

- Techniques used with imbalanced classification problems.
- An illustration on blending (average) ensemble methods using three different models (Extreme Gradient Boosting Classifier, Random Forest Classifier and CatBoostClassifier). The blending method gives us a boost in performance.

Finally, the following conclusions was drawn:

❖ Classification model for loan application across the United States with respect to demographics, location, property type, lender and other factors:

- A Decision tress model such as CatBoostClassifier, xgboost and RandomForestClassifier that achieves accuracy of 75.5% with cross validation performed.
- Most Important features were identified
- Fortunately, demographical or geolocation features form the deterministic values in classifying applicants' information for home loan. In fact, their vale brings positive effect to the model.

❖ Predictive model on whether a mortgage application was accepted or denied.:

- A Boosted Decision Tree (for classification) model that achieve MAE (Mean Absolute Error) of 1.5.
- Accuracy: 73.85% and F1 Score: 80.50%

Peek into the Data

The training values (features variable) and labels (target) dataset both contain applicants mortgage loan application information and attributes which this project aims to model. By combining the two datasets, there are 22 training features variables and 500000 rows in this dataset. Each row in the dataset represents a HMDA-reported loan application which covers one particular year. This implies that at least one row represent one

applicant loan application. Hence, the training values give us quantitative information about each data point and the target variable (accepted), a binary variable for each row of the test data set will be the variable we seek to predict. Another immediate observation is that the datasets most are categorical features versus numerical features, and missing values. Some numerical attributes are actually categorical in nature with no ordering (lender, msa_md, state_code, county_code). In addition, among 22 fields in applicants mortgage-application was accepted or denied, at least 7 fields contain missing information for modeling and can be replace with their median value, while 3 other fields (msa_md, state_code, county_code) contain values that are finite and non-finite, with no ordering and where -1 indicating a missing value. That being said, the following is a short description of each feature in the data set:

- **PROPERTY LOCATION**: msa_md (categorical), state_code (categorical), county_code (categorical)
- **LOAN INFORMATION**: lender (categorical), loan_amount (int), loan_type (categorical), property_type (categorical), loan_purpose (categorical), occupancy (categorical), preapproval (categorical)
- **APPLICANT INFORMATION**: applicant_income (int), applicant_ethnicity (categorical), applicant_race (categorical), applicant_sex (categorical), co_applicant (bool)
- **CENSUS INFORMATION**: population, minority_population_pct, ffiecmedian_family_income, tract_to_msa_md_income_pct, number_of_owner-occupied_units, number_of_1_to_4_family_units
- **INDEX AND TARGET VARIABLE**: row_id, accepted

Exploratory Data Analysis: Cleansing and Preparation

The home mortgage training data inputs together with label has 500000 data points (rows) with 22 variables (columns). In addition, there are 12 categorical variable, 1 boolean variable and 8 numerical variables in this dataset. Below diagram show top 5 rows of the combine data frame, an overview of our dataset content.

	loan_type	property_type	loan_purpose	occupancy	loan_amount	preapproval	msa_md	state_code	county_code	applicant_ethnicity	...	applicant_income
0	3	1	1	1	70.000	3	18	37	246	2	...	24.000
1	1	1	3	1	178.000	3	369	52	299	1	...	57.000
2	2	1	3	1	163.000	3	16	10	306	2	...	67.000
3	1	1	1	1	155.000	1	305	47	180	2	...	105.000
4	1	1	1	1	305.000	3	24	37	20	2	...	71.000

5 rows × 22 columns

Figure 1. An overview of the dataset

Replacing the null variables

This process is called imputation. In this case, since there are missing values and null variable in our dataset causing outlier, we replaced the null variables and missing data by median to substitute the missing values (see below diagram for detail replacement with substitute values).

```
Before Replacement:
Sum of Missing/Null values in each feature:
-----------------------------------
loan_type                          0
property_type                      0
loan_purpose                       0
occupancy                          0
loan_amount                        0
preapproval                        0
msa_md                             0
state_code                         0
county_code                        0
applicant_ethnicity                0
applicant_race                     0
applicant_sex                      0
applicant_income               39948
population                     22465
minority_population_pct        22466
ffiecmedian_family_income      22440
tract_to_msa_md_income_pct     22514
number_of_owner-occupied_units 22565
number_of_1_to_4_family_units  22530
lender                             0
co_applicant                       0
accepted                           0
dtype: int64
```

```
After Replacement:
Sum of Missing/Null values in each feature:
-----------------------------------
loan_type                      0
property_type                  0
loan_purpose                   0
occupancy                      0
loan_amount                    0
preapproval                    0
msa_md                         0
state_code                     0
county_code                    0
applicant_ethnicity            0
applicant_race                 0
applicant_sex                  0
applicant_income               0
population                     0
minority_population_pct        0
ffiecmedian_family_income      0
tract_to_msa_md_income_pct     0
number_of_owner-occupied_units 0
number_of_1_to_4_family_units  0
lender                         0
co_applicant                   0
accepted                       0
dtype: int64
```

Figure 2. Diagram showing missing values

Numeric Descriptive statistics

Descriptive or summary statistics for each numeric features are presented in the table below. For each feature, we can see the distinct count, standard deviation, maximum, minimum, mean, and median values. Looking at the values in the table, we can see that the range of values for each column differs quite a lot, so we can start to think about whether to apply normalization to the data. If we look back at our data summary, a low standard deviation suggested that most of numbers are close to the average and a high standard deviation implies that the numbers are spread out. Hence, the standard deviation is affected by outliers because it is based on the distance from the mean. The mean is also affected by outliers. Note that character columns were excluded and result were taken from total number of data entries (500,000 observations).

Features/ Columns	Distinct Count	Std Dev	Max	Min	Mean	Median
loan_type	4	0.691	4	1	1.366	1
property_type	3	0.231	3	1	1.048	1
loan_purpose	3	0.948	3	1	2.067	2
occupancy	3	0.326	3	1	1.110	1
loan_amount	2997	590.642	100878	1	221.753	162
preapproval	3	0.543	3	1	2.765	3
msa_md	409	138.464	408	-1	181.607	192
state_code	53	15.983	52	-1	23.727	26
county_code	318	100.244	324	-1	144.542	131
applicant_ethnicity	4	0.511	4	1	2.036	2
applicant_race	7	1.025	7	1	4.787	5
applicant_sex	4	0.678	4	1	1.462	1
applicant_income	1897	147.474	10139	1	100.121	74
population	18202	2667.723	37097	14	5396.982	4975
minority_population_pct	91923	25.799	100	0.534	31.226	22.901
ffiecmedian_family_income	68868	14478.233	125248	17858	69158.876	67526
tract_to_msa_md_income_pct	54535	13.990	100	3	92.200	100
number_of_owner-occupied_units	6088	721.028	8771	4	1423.173	1327
number_of_1_to_4_family_units	7374	893.718	13623	1	1880.147	1753
lender	6111	1838.313	6508	0	3720.121	3731

Figure 3. Descriptive Summary Statistics. The mean and median values for each feature is significantly different and that the comparatively large standard deviation indicates that there is considerable variance in whether a mortgage application will be accepted or denied.

Histogram and Boxplots – understanding the distribution

Since applicant_income, loan_amount, loan_purpose, preapproval, property type, applicant race, co_applicant, and applicant ethnicity are features of interests. These features are important features in this analysis. By understanding the distribution of values for each feature, we can start to make judgements about how to treat the data, for instance whether we want to deal with outliers; or whether we want to normalize the data. Therefore, a histogram of the applicant_income and loan_amount feature shows that the loan application values are right- and left skewed – in other words, most applicants are at the lower end of the loan income range, as shown in Figure 4. Furthermore, this probably implies some features values in the dataset are either balance or not.

Some Observations:

- Log transformation was performed to make the feature closer to normal. The transfromation, the distribution of loan amount and applicant are positive skew.
- Balanced data: co-applicant, accepted. They need no further processing.
- Loan type, applicant ethnicity, and gender are monotonically decreasing. However, they are kept as they are and not trimmed to be "0 and more than 0" category, or otherwise some deterministic features might be lost in the transformation

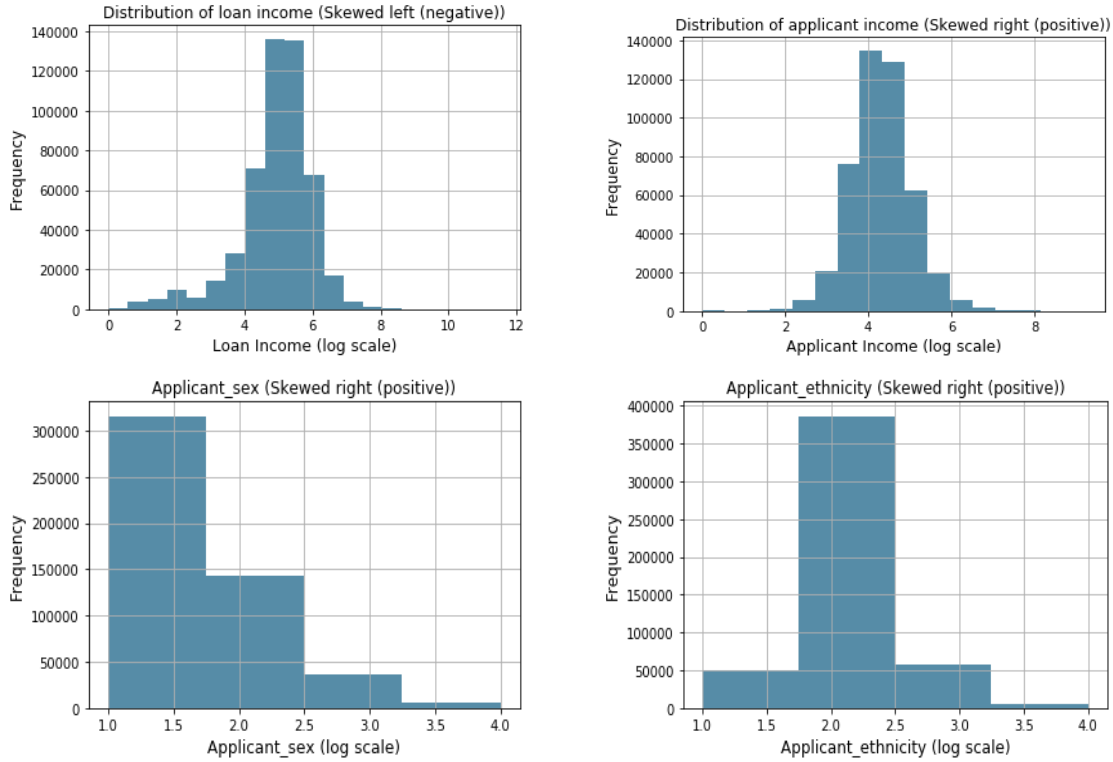


Figure 4: The distribution of few of the categorical and numerical features such as loan amount, applicant income, loan type and applicant ethnicity are presented as follows, except accepted which is the target variable.

Gearing towards a more targeted comparison with the train label feature – loan acceptance rates across applicant ethnicity and gender flags – their distribution tells useful. Figure 5 show applicants where applicant_ethnicity = 4 have a higher loan acceptance rate on average than where applicant_ethnicity = 1. That is, higher loan acceptance rates across ethnicity = 50822 and lower loan acceptance rates across ethnicity = 5819. Hence, the proportion of higher loan acceptance rates across ethnicity to lower loan acceptance rates across ethnicity = 873.38%. On the same Figure 5, also show applicants where applicant_sex = 1 have a higher loan acceptance rate on average than where applicant_sex = 2. Higher loan acceptance rates across gender = 315806 and lower loan acceptance rates across gender = 142876. Therefore, the proportion of higher

loan acceptance rates across gender to lower loan acceptance rates across gender = 221.04%. Note that the "loan acceptance rate" is the average value of accepted.

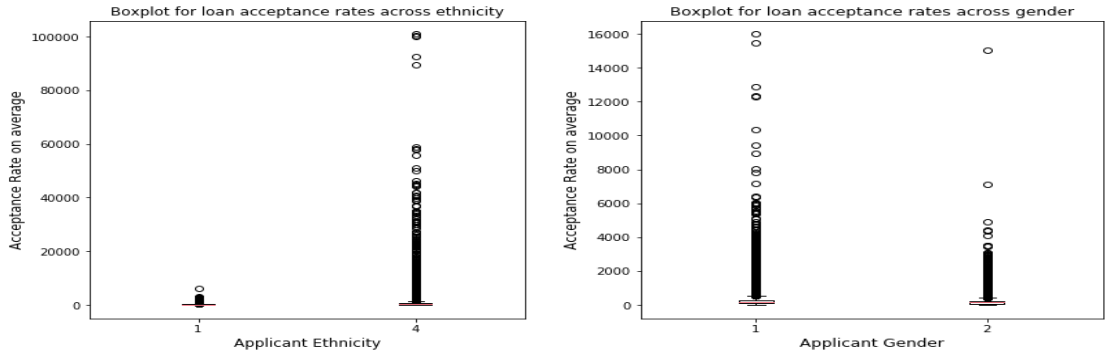


Figure 5: Distribution of loan acceptance rates across ethnicity and gender. The boxplot show the presence of outliers. This can be attributed to the disparity whether loan mortgage application will be accepted or not across ethnicity and gender.

The relationship between applicant income and loan amount

The figure below shows a higher applicant income is associated with a higher loan amount, on average.

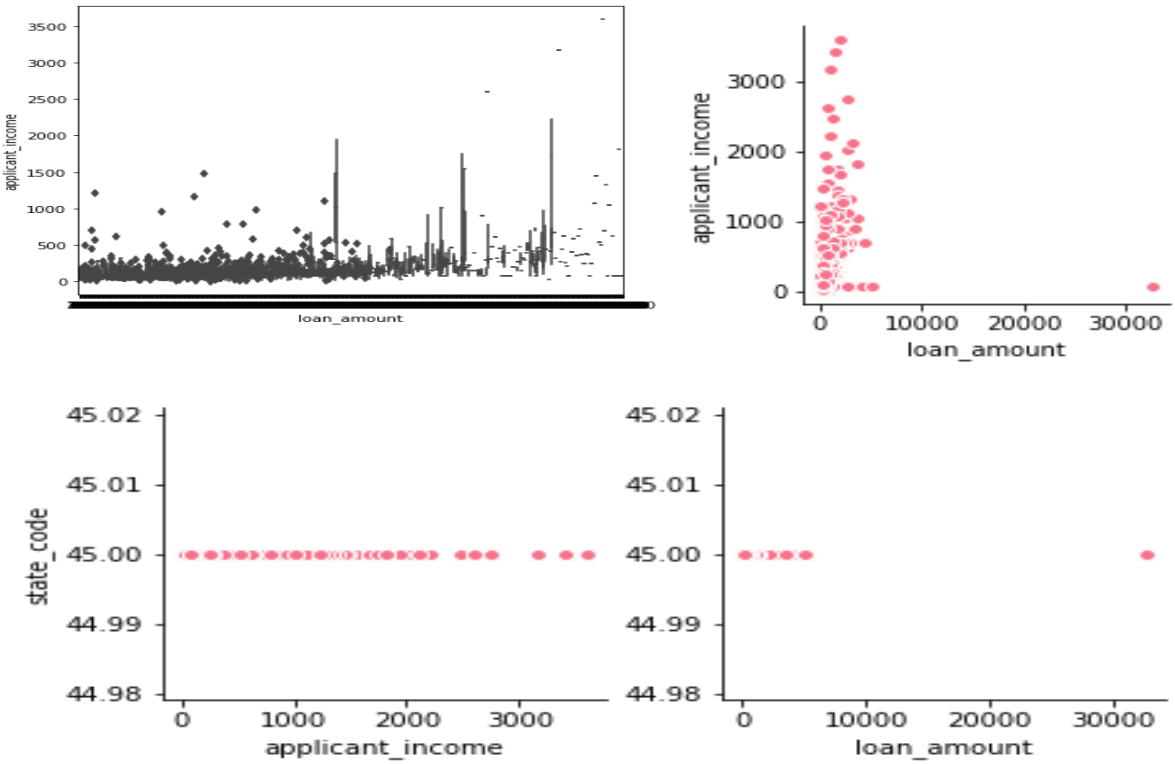


Figure 6: Understanding the distribution of applicant income and loan amount. As shown in the above diagram, applicants in state 45, show the relationship between applicant income and loan amount.

Loan acceptance across counties

As can be seen in Figure 7 where state_code == 48, the average rate of loan acceptance across counties varies substantially, ranging from around 30% to around 70%. The data is skewed right (positive). Hence, the distribution of loan acceptance across counties show that state with higher applicant incomes are more likely to be granted mortgage loan acceptance than state with lower applicant incomes.

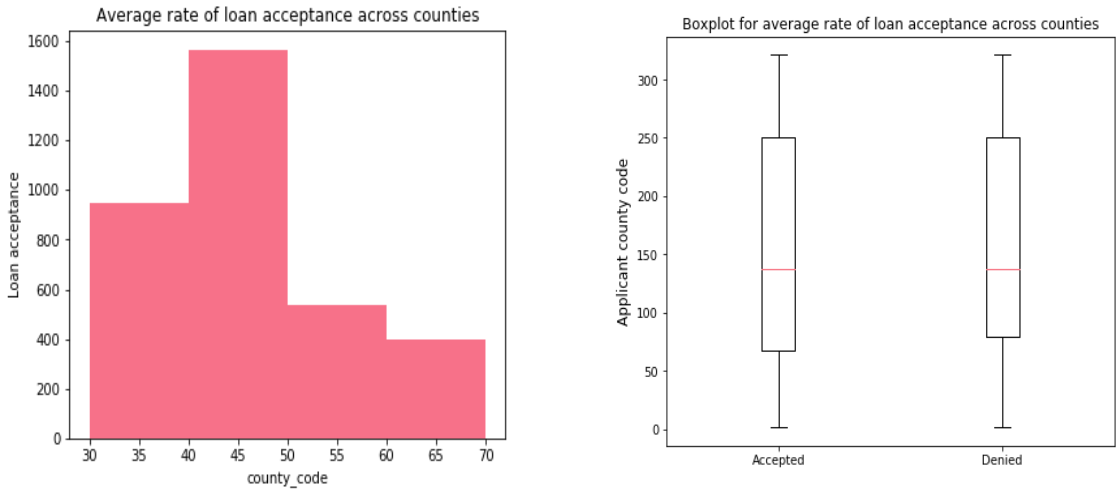


Figure 7: Distribution of Loan acceptance across counties where state_code == 48, limiting just to state 48 and ignoring where county is missing (missing value being -1). Loan accepted = 7659 and denied = 8052. Hence, as can be seen in the boxplot diagram the average rate of loan accepted to denied is 95.12%.

Loan types across states

For each of the four loan types, where state_code = 2, the average loan acceptance rate is 0.46 whereas where state_code = 4, the average loan acceptance rate is 0.58. The proportion of loan types across states is 79.38%. Thus, the loan acceptance rate in state 2 is lower than for the corresponding loan type in state 4 (Figure 8).

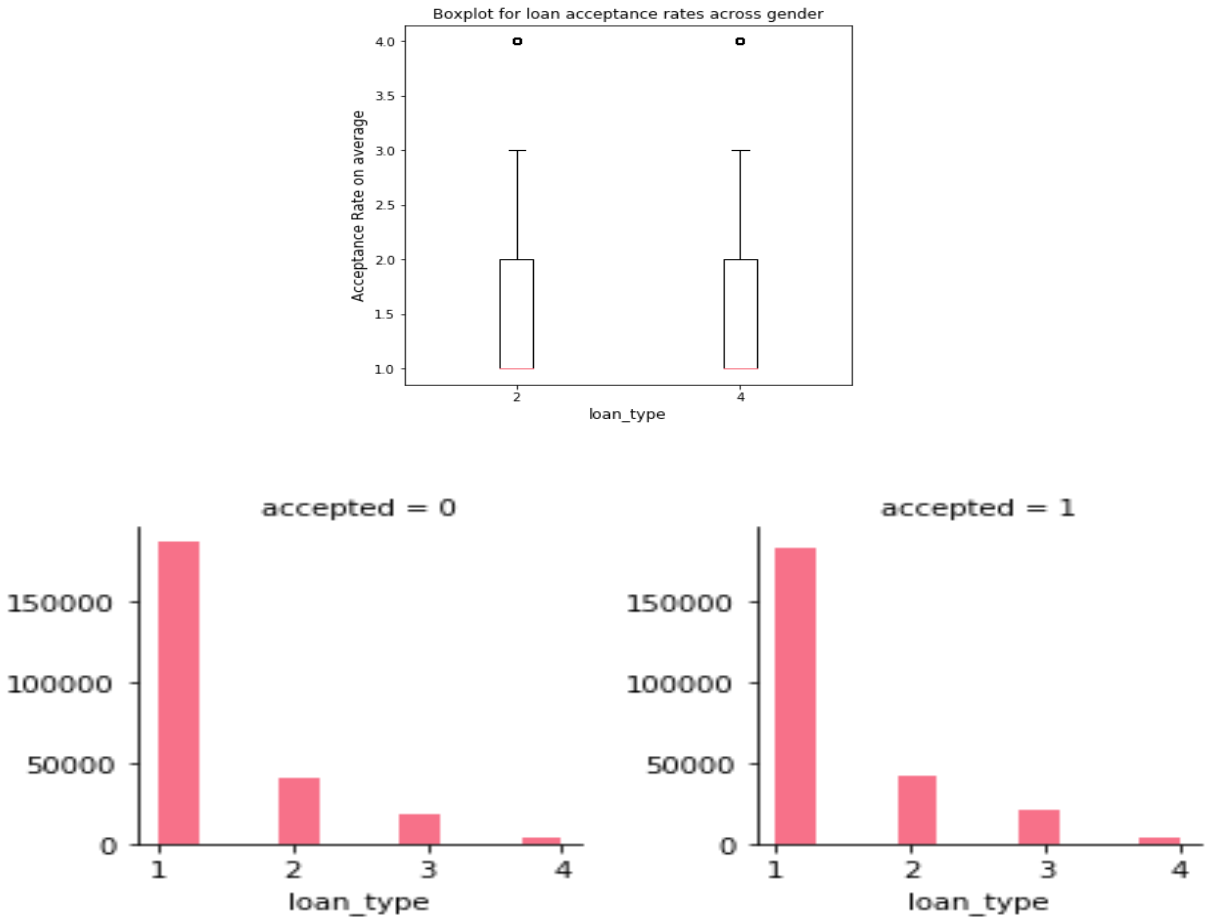


Figure 8: The distribution of loan acceptance rate across loan type.

Correlation and Apparent Relationships

As shown below in Figure 9 and 10, an attempt was made to identify the association between features in the dataset – in particular, between some features in loan info, applicant info, and census information.

Numeric and Categorical Relationships

The following scatter-plot matrix was generated to compare numeric features relationship with one another. The key features in this matrix are show below:

Figure 9: scatter-plot matrix was generated initially to compare numeric features with one another

Looking at the plots above, in the bottom row or the right-most column of this matrix shows an apparent relationship between loan acceptance rate and other numeric features. Specifically, as the rate of mortgage application accepted increases, so does loan_amount, applicant_income, lenders, total population in tract, and FFIEC Median family income in dollars increases. Though many points on the scatter plot fall into the long tail, therefore outlier region, the dense area clearly shows a positive relationship

between lender, FFIEC Median family income, applicant income and loan acceptance rate. Also, as expected, there seems to be a strong negative correlation between loan type, property type, and applicant income (Figure 10).

	loan_type	property_type	loan_purpose	loan_amount	preapproval	applicant_ethnicity	applicant_race	applicant_sex	applicant_income
loan_type	1.0	-0.065	-0.12	-0.019	-0.13	-0.05	-0.017	-0.072	-0.1
property_type	-0.065	1.0	-0.11	0.14	0.046	0.13	0.063	0.11	-0.061
loan_purpose	-0.12	-0.11	1.0	-0.0028	0.49	0.023	0.022	0.014	0.0095
loan_amount	-0.019	0.14	-0.0028	1.0	-0.0018	0.099	0.037	0.062	0.16
preapproval	-0.13	0.046	0.49	-0.0018	1.0	0.017	0.019	0.019	0.017
applicant_ethnicity	-0.05	0.13	0.023	0.099	0.017	1.0	0.28	0.5	0.038
applicant_race	-0.017	0.063	0.022	0.037	0.019	0.28	1.0	0.27	0.0065
applicant_sex	-0.072	0.11	0.014	0.062	0.019	0.5	0.27	1.0	-0.049
applicant_income	-0.1	-0.061	0.0095	0.16	0.017	0.038	0.0065	-0.049	1.0

Figure 10: scatter-plot matrix was generated initially to compare numeric features with categorical features.

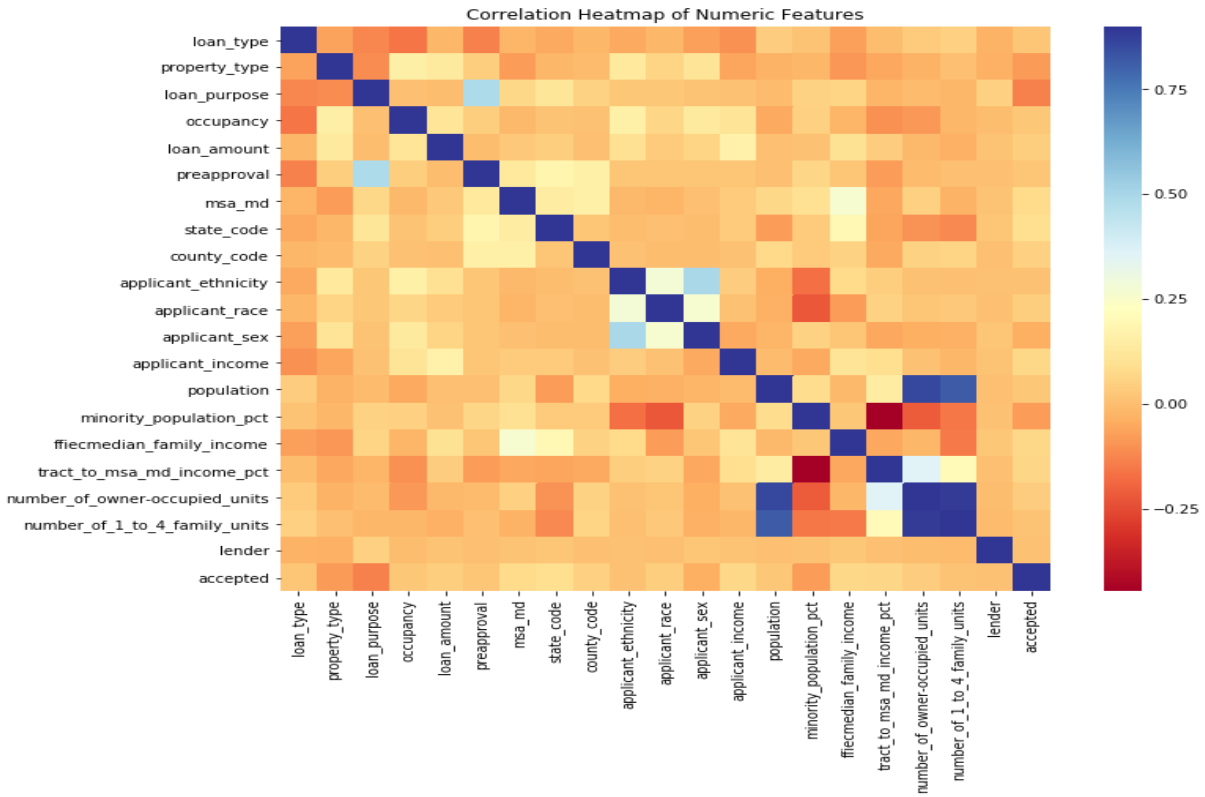


Figure 11: Heatmaps for correlation of numeric features. This show features that are correlated to each other. Though, correlation was generated initially to compare numeric features with categorical features.

Analysing the target variable

we extended the analysis by creating a heatmap that shows the correlation between the features and target (accepted). Looking at this, the trending is not linear, this implies a log scale transformation might be needed in the subsequent treatment. Hence, as shown below, the correlation heatmaps of numerical features increased linearity in the relationships between the loan granted, applied for, or purchased and the other numeric features (Figure 11).

Data Wrangling and Transformation

By exploring data above, some decided data wrangling process would be:

- Make every input, except categorical features
- Segment the various categorical features, dropped thereafter)
- Change loan_amount, lender, population to log scale

Classification Model for Predicting a Loan Acceptance Rate

The following diagram show an overview of machine learning pre-processing workflow implementation.

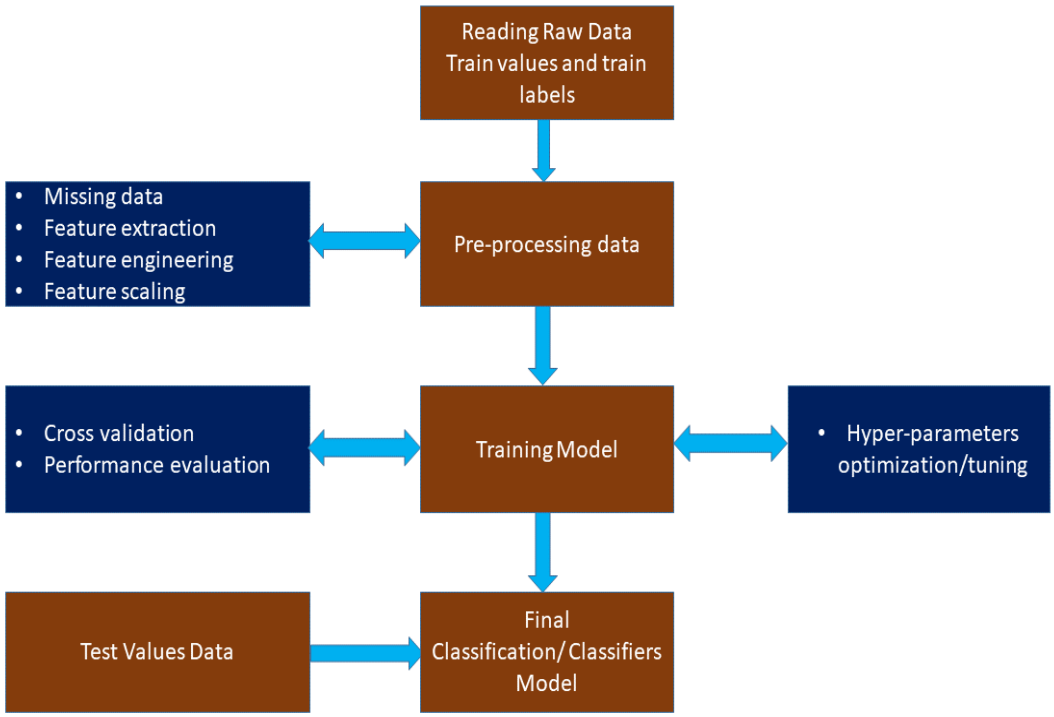


Figure 12: ML Overview including cleansing and wrangling

Pre-processing Machine Learning Recipes

This section discuss how we pre-process the data in order to ensure it is a suitable for machine learning modeling. The following show the steps we took for preprocess the data:

- We load the given dataset from a CSV
- Handling and dealing with missing values. This is where we identify what, if, any missing data we have and how to deal with it. For example, we replace missing values in features such as: applicant income, population, minority population pct, ffiecmedian family income, tract to msa md income pct, number of owner-occupied units, number of 1 to 4 family units with their mean and median value or by the average of the neighbouring values.

- We treat categorical values, by converting them into a numerical representation that can be modelled.
- We split the dataset into the input and output variables for machine learning.
- We normalise the data, for example by ensuring the data is, for example all on the scale (such as within two defined values); normally distributed; has a zero-mean, etc. This is necessary to make the ML models to work, and can also help speed up the time it takes for the models to run.
- We summarize the data to show the change.

In this project, we look to remove outliers, which are values that were erroneous and were over-influence the model, and normalize the data. The transforms are calculated in such a way that they can be applied to the training data and any samples of data in the future. The scikit-learn documentation has some information on how to use various different pre-processing methods.

Applying Ordinal Encoding to Categorical

We need to convert some features into categorical group to make processing simpler. The columns applicant_race, applicant_ethnicity, preapproval, occupancy, property_type, loan_purpose and loan_type represent categorical features. However, because they are integers, they are initially parsed as continuous numbers. It is also required to encode features like co_applicant with a string category since XGBoost (like all the other machine learning algorithms in Python) requires every feature vector to include only digits.

Split the data

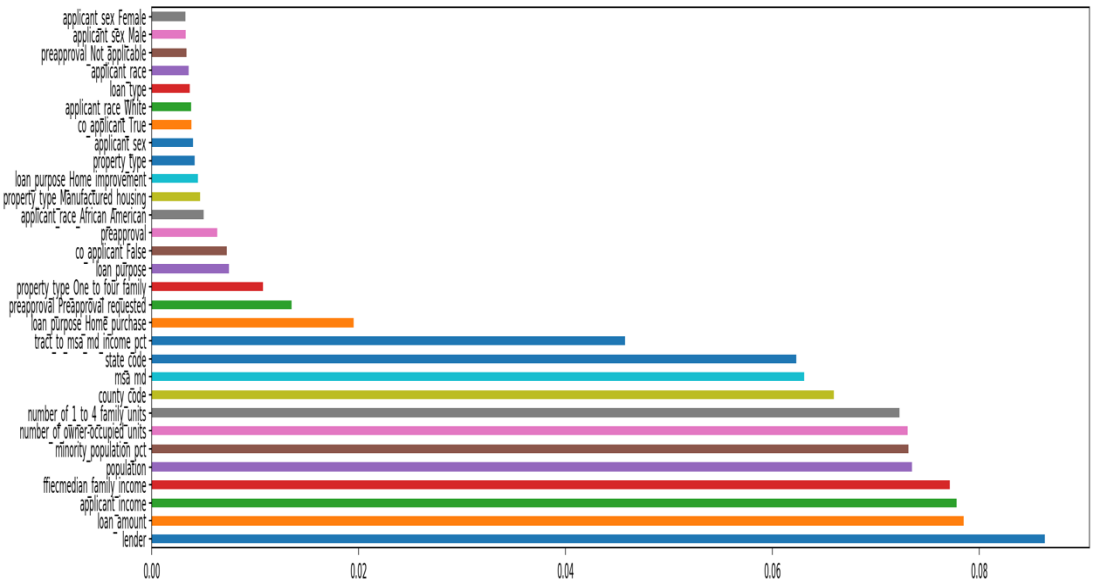
The original given dataset was split up into training and testing data. The training data is used to build the model. For instance, using the train set to find the optimal coefficients in a linear classifier and regression model. The testing data is used to see

how well the model performs on unseen data, as it would in a real-world scenario. The test data was left completely unseen until we test and evaluate the model performance.

Feature Importance Selections

Feature importance is defined by the absolute value of its corresponding coefficient. In this project, data was normalized for all of the feature values so that it will be valid to compare them. The following diagram show feature importance according to the k highest scores, using the ANOVA (f_classif) scoring function for the feature selection. From the feature importance though, we observed that some features affected the model accuracy positively. This is shown below;

- **Generated features**: Loan purpose, preapproval, property type, and applicant race are relevant features from score.
- **Property Location feature**: state_code, msa_md and county_code
- **Applicant Information**: applicant_income, applicant_ethnicity, applicant_race and gender (applicant_sex).



	Specs	Score
29	loan_purpose_Home_purchase	13106.289826
2	loan_purpose	8811.125900
36	preapproval_Preapproval_not_requested	6742.605950
30	loan_purpose_Home_improvement	6301.742257
27	property_type_Manufactured_housing	6265.408954
44	applicant_race_African_American	5445.969410
46	applicant_race_White	5299.462495
21	co_applicant_False	5165.009020
20	co_applicant_True	5165.009020
35	preapproval_Preapproval_requested	5118.845798
26	property_type_One_to_four_family	4923.917086
31	loan_purpose_Refinancing	4263.821084
7	state_code	4132.043896
1	property_type	3269.627173
6	msa_md	3243.944737
14	minority_population_pct	2894.482407
47	applicant_race_Information_not_provided	2835.577988
39	applicant_ethnicity_Not_Hispanic_Latino	2805.688604
40	applicant_ethnicity_Information_not_provided	2680.454224
12	applicant_income	2508.806862
15	ffiecmedian_family_income	2476.012908
16	tract_to_msa_md_income_pct	2115.145632
52	applicant_sex_Not_applicable	2015.988499
48	applicant_race_Not_applicable	2006.500231
41	applicant_ethnicity_Not_applicable	1975.823769
49	applicant_sex_Male	1458.673870
8	county_code	1387.923177
51	applicant_sex_Information_not_provided	1365.187622
38	applicant_ethnicity_Hispanic_Latino	1202.659876
4	loan_amount	1077.393464

Figure 13: Feature Importance

Choose a Baseline algorithm

We create a baseline model to benchmark against other estimators. In this case, Decision tree classifier is a fairly simple algorithm compared to more complicate classifier options.

We fit a decision tree using default parameters to get a baseline idea of the performance. Decision forest is used as the kernel for the classifier, and hyper-parameters are tuned for optimized result.

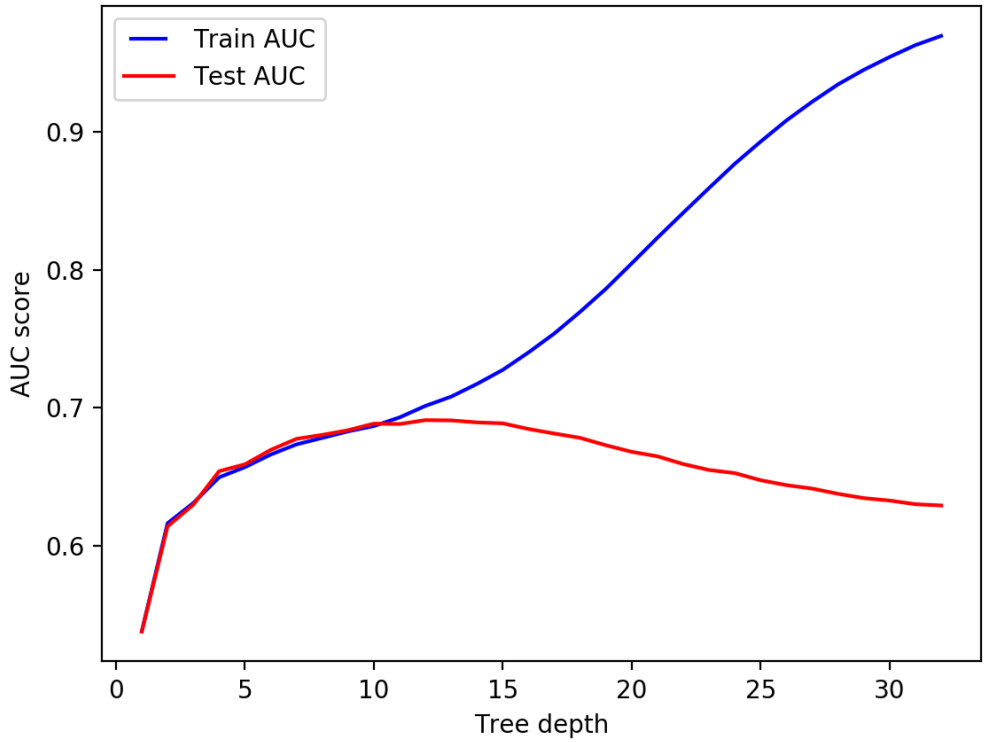


Figure14: Diagram show the Decision tree as benchmark for our model and how they impact the model in term of over-fitting and under-fitting. We see from the diagram above (Figure 14) that our model overfits for large depth values. The tree perfectly predicts all the train data; however, it fails to generalize the findings for new data.

Fitting the Model and Predict scores against test set

We fit the model we developed to the training data and make predictions. That is, we assessed how good the model is and then generate a set of predictions on our unseen features. Also, we fit a new estimator and use cross validation score to get a score based on a defined metric. In order to find the optimal value of coefficient for logistic regression. In this project, we apply stratified 5-fold validation and then look at the ROC AUC against different values of the parameter of the coefficient for logistic regression. One of the

important metrics of model quality is the Area Under the Curve (AUC). ROC AUC varies from 0 to 1. The closer ROC AUC is to 1, the better the quality of our classification model.

Evaluation Selection Metrics for Classification Problem

We choose an evaluation that suit this project and then we compare our predictions with the actual result and measure them in some way. Classification metrics include:

- Accuracy: This assess how often the model selects the best class. In this case, this was used on our balanced classes (i.e. there are a similar number of instances of each class we are trying to predict). There are some limits to this metric but we used metrics that tell us how our model performs in more detail (such as f1 score, roc_auc, and recall).
- Evaluation ROC also reveals that even with great extent of feature pruning, accuracy is not affected.

Predicting Probabilities

In this project, two types of predictive probabilities were used:

- **False Positive:** That is, predict an event when there was no event.
- **False Negative:** That is, predict no event when in fact there was an event.

By predicting probabilities and calibrating a threshold, a balance of these two concerns were chosen for the model. A common way we compare models that predict probabilities for binary classification problems is through a precision and recall measurement. These measurements are useful in applied machine learning for evaluating binary classification models. Precision is a ratio of the number of true positives divided by the sum of the true positives and false positives. It describes how good a model is at predicting the positive class. Precision is referred to as the positive predictive value.

Positive Predictive Power (Precision) = True Positives / (True Positives + False Positives)

Recall is calculated as the ratio of the number of true positives divided by the sum of the true positives and the false negatives. This describes how good the model is at predicting the positive class when the actual outcome is positive. The true positive rate is also referred to as sensitivity. Recall is the same as sensitivity.

Recall/Sensitivity (true positive rate) = True Positives / (True Positives + False Negatives)

Reviewing both precision and recall is useful in this project since there is an imbalance in the observations between features. Specifically, there are many examples of no event (features such as: applicant_race, applicant_sex, applicant_ethnicity) and examples of an event (features such as : msa_md, state_code, and county_code). The reason for this is that typically the large number of class 0 examples means we are less interested in the skill of the model at predicting class 0 correctly, e.g. high true negatives. A precision-recall curve is a plot of the precision (y-axis) and the recall (x-axis) for different thresholds, much like the ROC curve. The no-skill line is defined by the total number of positive cases divide by the total number of positive and negative cases. Three composites scores that we used to summarised the precision and recall:

- **F1 score** : that calculates the harmonic mean of the precision and recall (harmonic mean because the precision and recall are ratios).
- **Average precision:** that summarizes the weighted increase in precision with each change in recall for the thresholds in the precision-recall curve.
- **Area Under Curve:** like the AUC, summarizes the integral or an approximation of the area under the precision-recall curve.

In terms of this project model selection, F1 score summarizes model skill for a specific probability threshold, whereas average precision and area under curve summarize the skill of the model across thresholds. This makes precision-recall and a plot of precision vs. recall and summary measures useful tools for binary classification problems that have an imbalance in the observations for each class. The precision-recall curve plot is then created showing the precision/recall for each threshold compared to a no skill model.

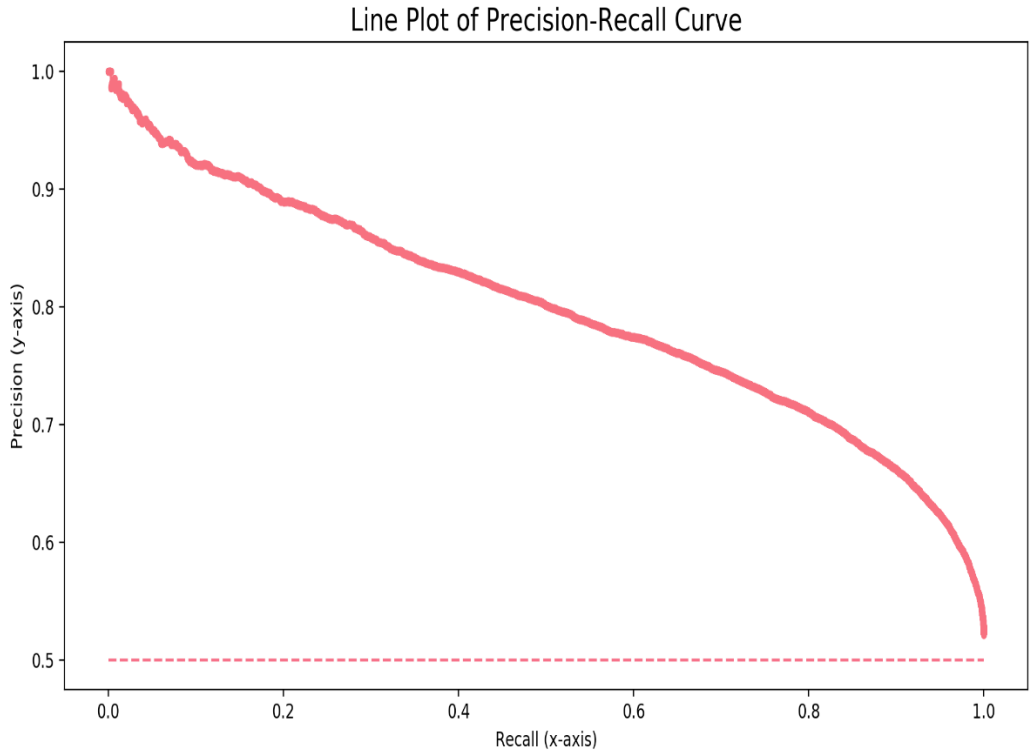


Figure 15: Plot of the true positive rate (Recall-x-axis) versus the positive predictive value (Precision-y-axis). The F1 score (0.751), area under curve (AUC = 0.798) and average precision (AP = 0.798) scores. The plot shows the true positive rate (Recall-x-axis) versus the positive predictive value (Precision-y-axis) for several different features threshold values between 0.0 and 1.0. In order words, we plot the predictive value rate versus the hit rate.

Conclusion

In conclusion, after the exploratory data analysis and machine learning, we build a predictive model that help to evaluate whether mortgage was accept/reject for future loan applications. Furthermore, we build an optimization algorithm to identify which loan product should you offer to the accepted requests. For the prediction, multiple prediction models were used with consideration to their performance. Then, this project selects the best model or combine models. More so, a bootstrap aggregating algorithm (such as random forests) and a boosting algorithm (such as xgboost) was used. For the optimization algorithm, this project focus on the data related to the past granted loans and how they performed.

References:

1. Precision-Recall Curve: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

2. LightGBM: https://github.com/Microsoft/LightGBM/

3. CatBoost is a high-performance open source library for gradient boosting on decision trees [https://catboost.ai/]

4. XGBoost https://xgboost.readthedocs.io/en/latest/

5. Loan Application: https://www.sciencedirect.com/topics/computer-science/loan-application