PREDICTING MORTGAGE RATES FROM GOVERNMENT DATA

Author: Cenk Ersoy (cenk.ersoy@gmail.com/cenker@microsoft.com)

Version: Dec/27/2019

1 - EXECUTIVE SUMMARY

The purpose of the study was to build a machine learning model to predict the rate spread of mortgage applications, based on the data from Home Mortgage Disclosure Act (HMDA) in the US.

The dataset can be obtained here: https://www.datasciencecapstone.org/competitions/18/mortgage-rates-from-government-data/page/56/

The training dataset included 200,000 data points (unique row_id) with 22 features. The dependent variable (variable to predict/label) was "rate spread".

The training dataset was processed and used to train a Supervised Learning Model based on CatBoost Regressor algorithm[(1)]. Then the trained model was used to predict the rate spread for a test (validation) dataset with 200,000 rows.

The trained model achieve a r-squared score of 0.76 on the test dataset. The key findings are given in the conclusion section.

The Jupyter Notebook for this study along with the dataset files can be found here: https://github.com/cenkersoy/MPP_DataScience

2 - DETAILS OF THE STUDY

The study included the following phases:

(1) Data acquisition
(2) Data cleanup
(3) Exploratory Data Analysis
(4) Feature Engineering/Selection
(5) Model training and optimization
(6) Prediction

2.1 - Data Acquisition:

The training and test data had been provided in Excel format. Data was read in and separate Python data frames were created for test and training datasets.

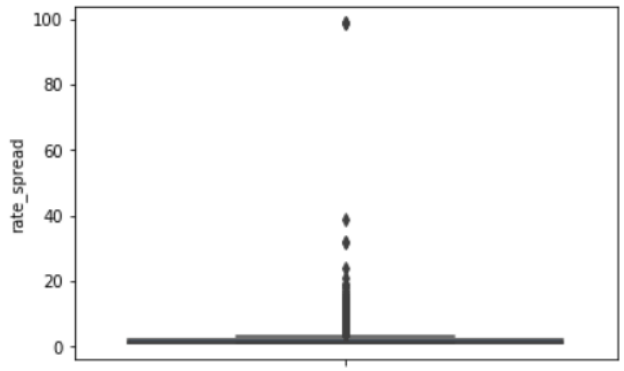
The training and test data sets included 9 numeric and 12 categorical features and 1 Boolean feature. The dependent variable (rate_spread) was integer.

Both the training and test datasets contained 200,000 rows.

2.2 - Data Cleanup:

There were no redundant values for row_id.

By using Box Plot analysis on the training dataset, it was noticed that there were a few outliers for the dependent value. Rows of the training dataset where rate_spread > 50 were removed.



Information was provided with the project that the presence of '-1' represented a missing value for the following columns: msa_md, state_code and county_code.

Imputation (replacement for missing data) was carried for both training and test datasets. First, those values are replaced with "not-a-number" in Python ('NaN'). The table below shows the percentages for the missing data in the training dataset:

	Missing Values	% of Total Values
applicant_income	10708	5.4
tract_to_msa_md_income_pct	2023	1.0
number_of_1_to_4_family_units	2016	1.0
number_of_owner-occupied_units	2012	1.0
population	1995	1.0
minority_population_pct	1995	1.0
ffiecmedian_family_income	1985	1.0
state_code	1338	0.7

The table below shows the percentages of the missing data in the test dataset:

	Missing Values	% of Total Values
applicant_income	10371	5.2
tract_to_msa_md_income_pct	1946	1.0
number_of_owner-occupied_units	1933	1.0
number_of_1_to_4_family_units	1933	1.0
minority_population_pct	1920	1.0
population	1918	1.0
ffiecmedian_family_income	1905	1.0
state_code	1229	0.6

Instead of removing the rows with missing data, all missing values were imputed using the following logic:

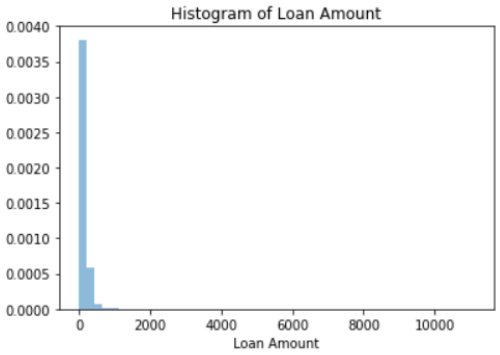
- For numerical features, the mean() of the column was used
- For categorical features, the mode() was used

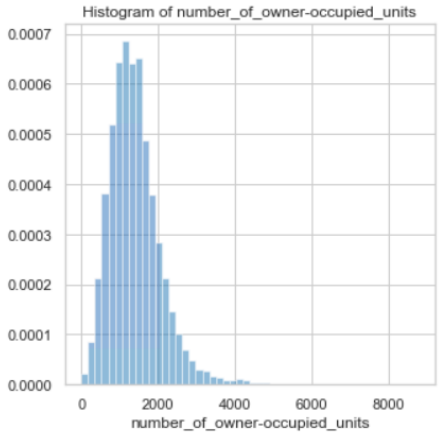
2.3 - Exploratory Data Analysis

Initial review of the statistics for the training dataset features revealed significant skew and a wide range of mean values for certain features. This indicated the need for normalization of the features so that some features do not dominate the training algorithm.

	row_id	loan_amount	applicant_income	population	minority_population_pct	ffiecmedian_family_income	tract_to_msa_md_income_pct
count	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000
mean	99999.500000	142.574940	73.617902	5391.099099	34.238640	64595.355801	89.283022
std	57735.171256	142.559487	102.828490	2655.683596	27.791227	12661.211303	14.982867
min	0.000000	1.000000	1.000000	7.000000	0.326000	17860.000000	6.193000
25%	49999.750000	67.000000	40.000000	3730.000000	11.047000	56763.000000	81.846750
50%	99999.500000	116.000000	59.000000	4986.000000	26.357000	63609.000000	98.618500
75%	149999.250000	179.000000	80.000000	6450.000000	51.641000	71176.000000	100.000000
max	199999.000000	11104.000000	10042.000000	34126.000000	100.000000	125095.000000	100.000000

Below are some of the histograms for the numerical features that illustrate the skew.

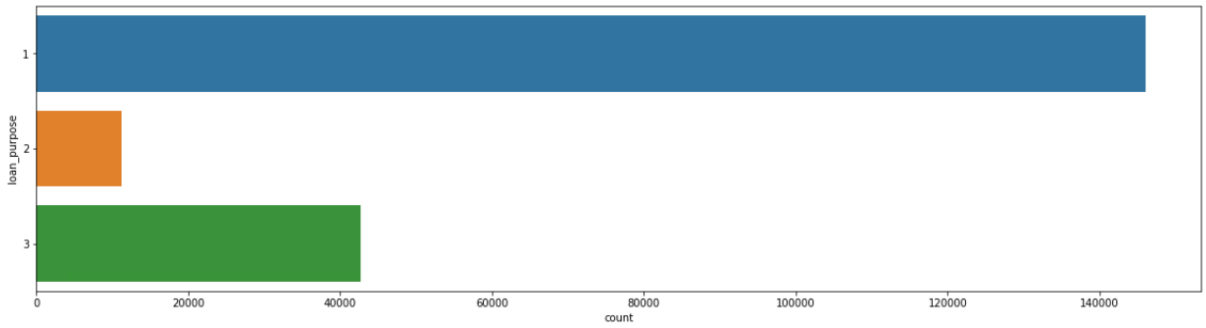




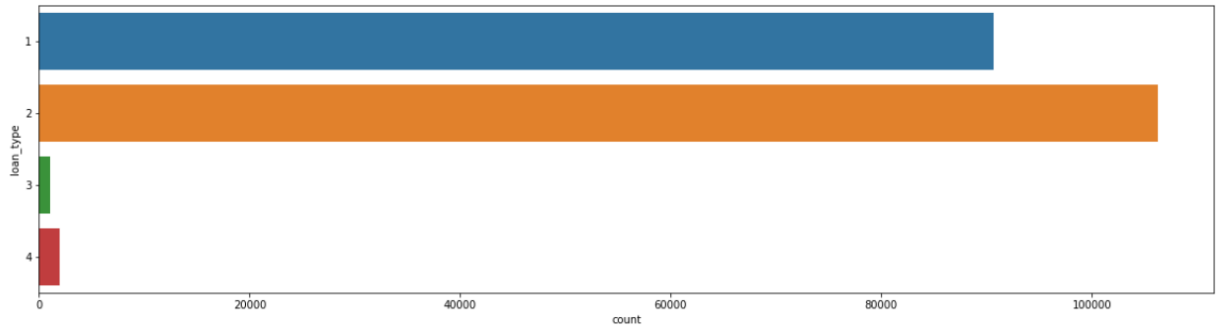
Analysis of the frequency of categorical variables resulted in some interesting findings. These findings are detailed here and summarized again in the “Key Findings and Conclusion” section.

(a) Majority of the mortgages were for home purchase, FHA-insured and for 1-4 room family homes.

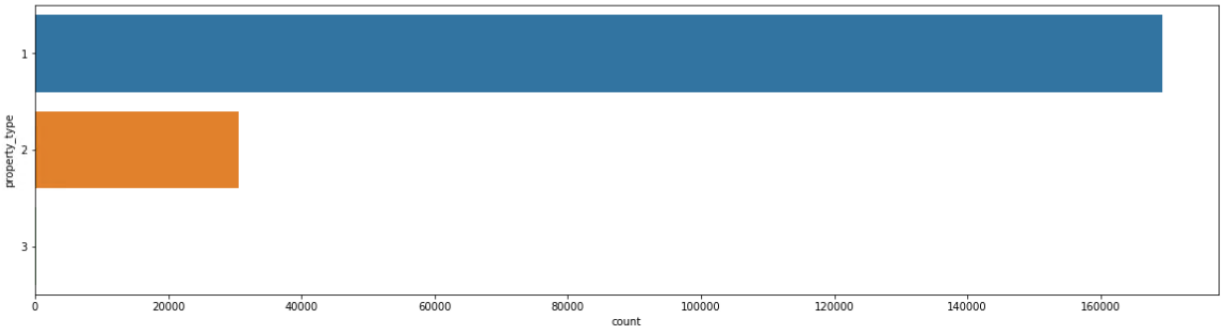
```
# Checking the frequency distribution of 'loan purpose'
# 1 -- Home purchase
# 2 -- Home improvement
# 3 -- Refinancing

# Let's view the distribution
pyplot.figure(figsize=(20, 5))
sns.countplot(y="loan_purpose", data=train);
```



```
: # Checking the frequency distribution of 'loan type'
  # 1 -- Conventional (any loan other than FHA, VA, FSA, or RHS loans)
  # 2 -- FHA-insured (Federal Housing Administration)
  # 3 -- VA-guaranteed (Veterans Administration)
  # 4 -- FSA/RHS (Farm Service Agency or Rural Housing Service)

  # Let's view the distribution
  pyplot.figure(figsize=(20, 5))
  sns.countplot(y="loan_type", data=train);
```



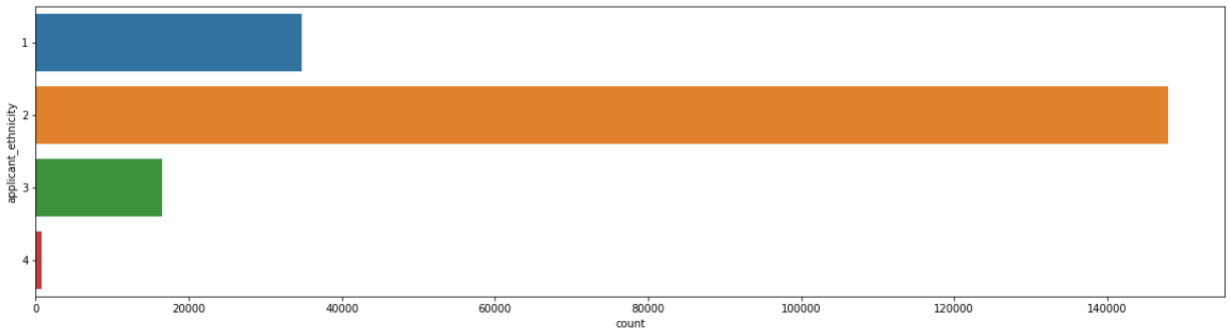

```
: # Checking the frequency distribution of 'property type'
  # 1 -- One to four-family (other than manufactured housing)
  # 2 -- Manufactured housing
  # 3 -- Multifamily

  # Let's view the distribution
  pyplot.figure(figsize=(20, 5))
  sns.countplot(y="property_type", data=train);
```



(b) Unfortunately, race and ethnicity seem to have influence on the mortgage approvals.

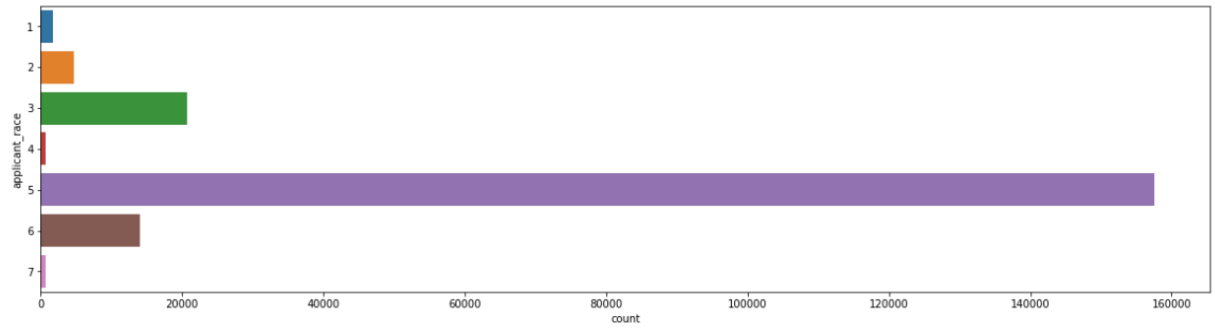
Most of the mortgages were granted to “non-Hispanic or Latinos”.

```
# Checking the frequency distribution of 'applicant ethnicity'
# 1 -- Hispanic or Latino
# 2 -- Not Hispanic or Latino
# 3 -- Information not provided by applicant in mail, Internet, or telephone pplication
# 4 -- Not applicable
# 5 -- No co-applicant

# Let's view the distribution
pyplot.figure(figsize=(20, 5))
sns.countplot(y="applicant_ethnicity", data=train);
```



The mortgage recipients were overwhelmingly white:

```
# Checking the frequency distribution of 'applicant race'
# 1 -- American Indian or Alaska Native
# 2 -- Asian
# 3 -- Black or African American
# 4 -- Native Hawaiian or Other Pacific Islander
# 5 -- White
# 6 -- Information not provided by applicant in mail, Internet, or telephone application
# 7 -- Not applicable
# 8 -- No co-applicant

# Let's view the distribution
pyplot.figure(figsize=(20, 5))
sns.countplot(y="applicant_race", data=train);
```

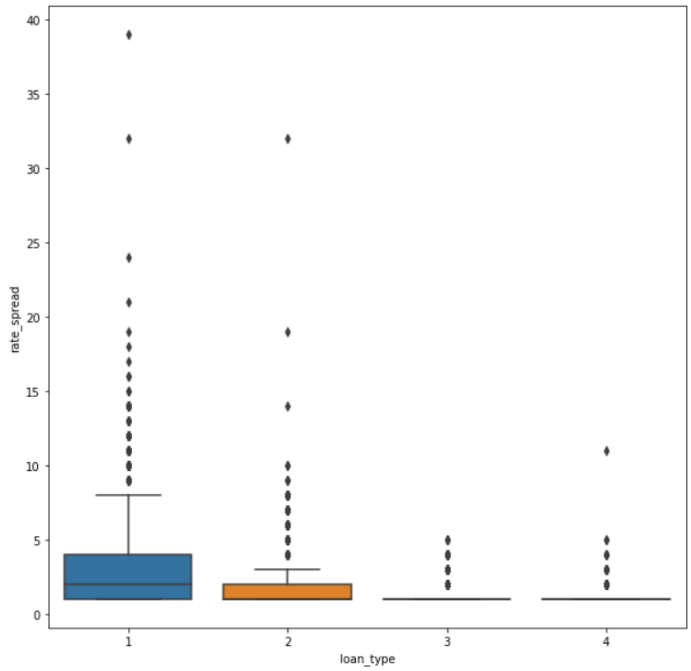


(c) Also, the average rate spread and its standard deviation varied for certain cases as explained below.

The average rate spread was significantly higher for conventional loans.

```
# Checking the frequency distribution of 'loan type'
# 1 -- Conventional (any loan other than FHA, VA, FSA, or RHS loans)
# 2 -- FHA-insured (Federal Housing Administration)
# 3 -- VA-guaranteed (Veterans Administration)
# 4 -- FSA/RHS (Farm Service Agency or Rural Housing Service)

pyplot.figure(figsize=(10, 10))
sns.boxplot('loan_type','rate_spread',data=train)
```

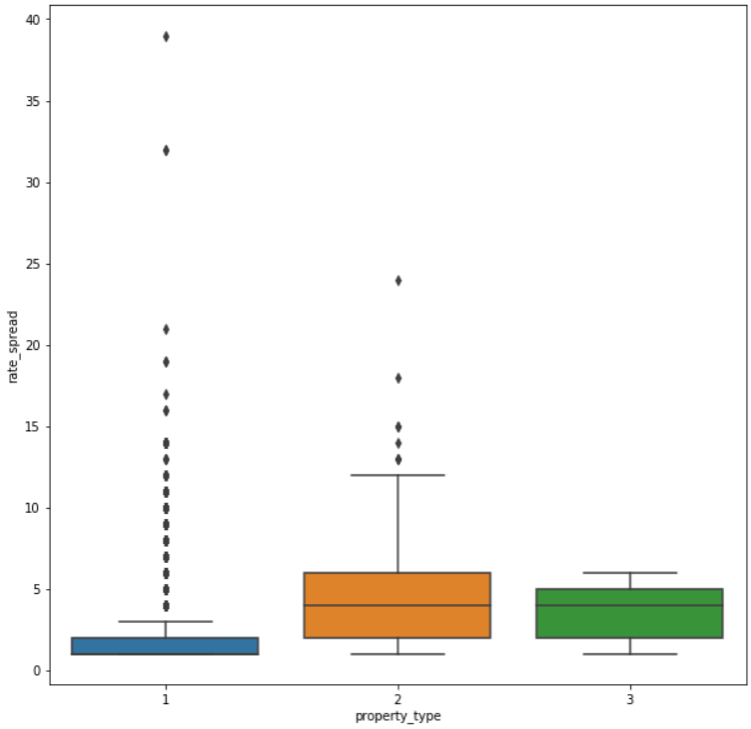
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f75c512dd8>
```



1-4 family home mortgages enjoyed lower rate spread.

```
# Checking the frequency distribution of 'property type'
# 1 -- One to four-family (other than manufactured housing)
# 2 -- Manufactured housing
# 3 -- Multifamily

pyplot.figure(figsize=(10,10))
sns.boxplot('property_type','rate_spread',data=train)
```

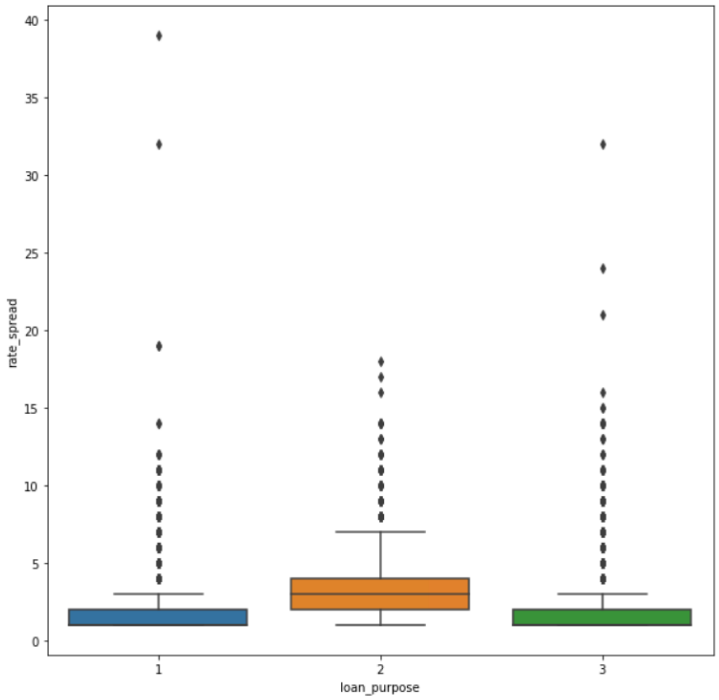
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f75ef96d68>
```



Average rate spread was much higher for home improvements.

```
# Checking the frequency distribution of 'loan purpose'
# 1 -- Home purchase
# 2 -- Home improvement
# 3 -- Refinancing

pyplot.figure(figsize=(10,10))
sns.boxplot('loan_purpose','rate_spread',data=train)
```

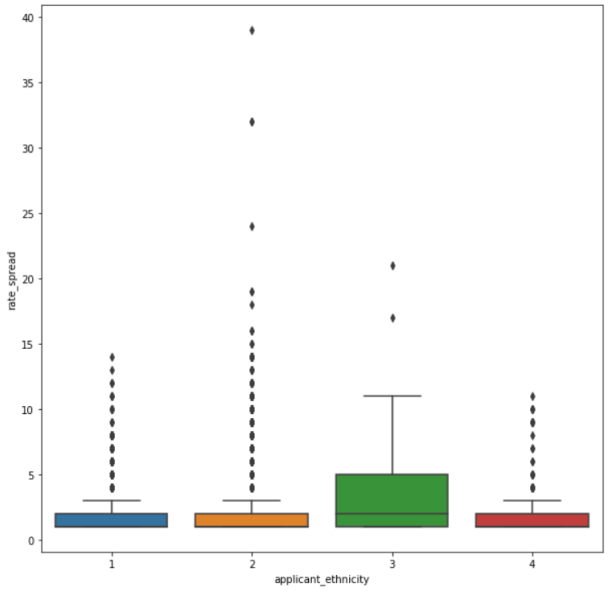
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f75f0544e0>
```



Once again, ethnicity and race came into play with alarming results. If an applicant does not indicate ethnicity, then the average rate spread is very high and with large standard deviation.

```
# Checking the frequency distribution of 'applicant ethnicity'
# 1 -- Hispanic or Latino
# 2 -- Not Hispanic or Latino
# 3 -- Information not provided by applicant in mail, Internet, or telephone pplication
# 4 -- Not applicable
# 5 -- No co-applicant

pyplot.figure(figsize=(10,10))
sns.boxplot('applicant_ethnicity','rate_spread',data=train)
```

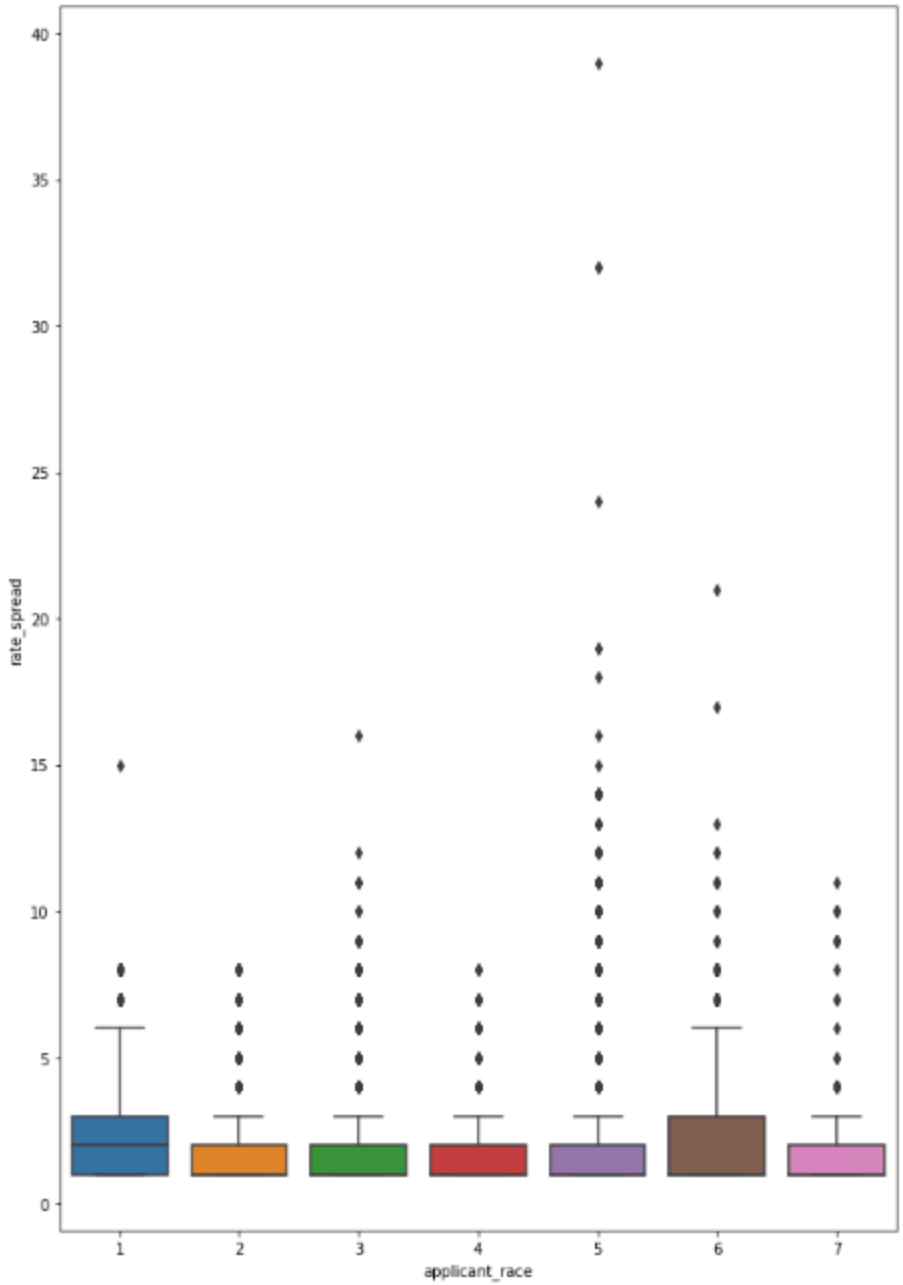
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f75eb8da20>
```



Average rate spread was much higher for “American Indian or Alaska Natives”. If an applicant does not indicate race on their form, the standard deviation of rate spread is very high.

```
# Checking the frequency distribution of 'applicant race'
# 1 -- American Indian or Alaska Native
# 2 -- Asian
# 3 -- Black or African American
# 4 -- Native Hawaiian or Other Pacific Islander
# 5 -- White
# 6 -- Information not provided by applicant in mail, Internet, or telephone application
# 7 -- Not applicable
# 8 -- No co-applicant

pyplot.figure(figsize=(10,15))
sns.boxplot('applicant_race','rate_spread',data=train)
```

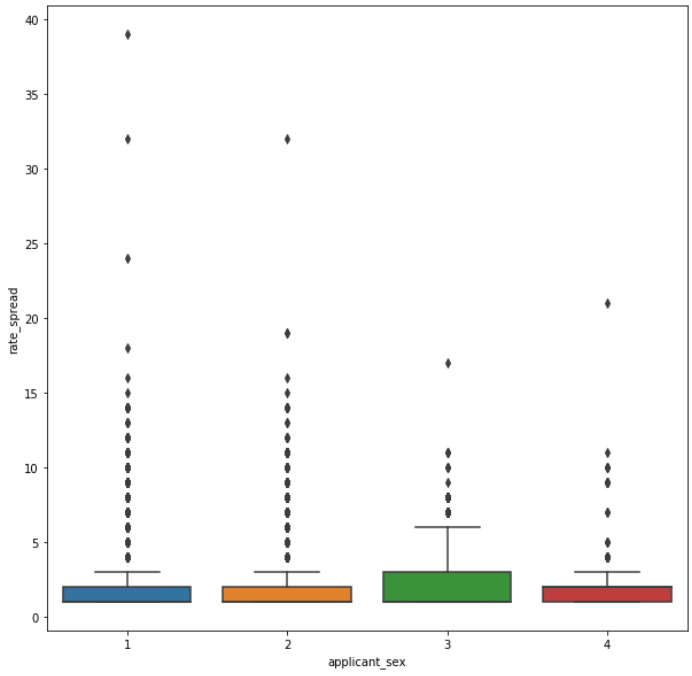
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f75c5342e8>
```



Similarly, if an applicant does not indicate “gender” in their application, the standard deviation of the rate spread is higher.

```
# Checking the frequency distribution of 'applicant sex'
# 1 -- Male
# 2 -- Female
# 3 -- Information not provided by applicant in mail, Internet, or telephone appli
# 4 or 5 -- Not applicable

pyplot.figure(figsize=(10,10))
sns.boxplot('applicant_sex','rate_spread',data=train)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f75373e400>
```

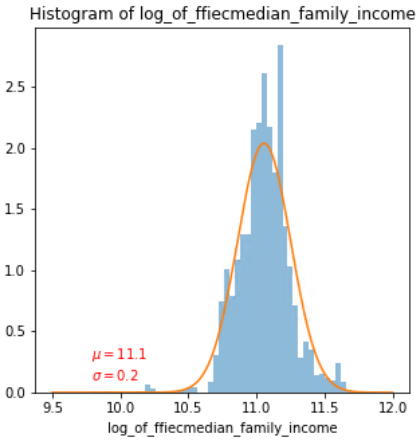


2.4 - Feature Engineering

Since the histogram of the numeric features had high skews, mathematical transformations were applied to those features:

Numeric Feature	Transformation Applied
loan_amount	Log Transform
applicant_income	Log Transform
population	Log Transform
minority_population_pct	Square root
ffiecmedian_family_income	Log Transform
tract_to_msd_md_income_pct	Log Transform
number_of_owner-occupied-units	Log Transform
number_of_1_to_4_family_units	Log Transform

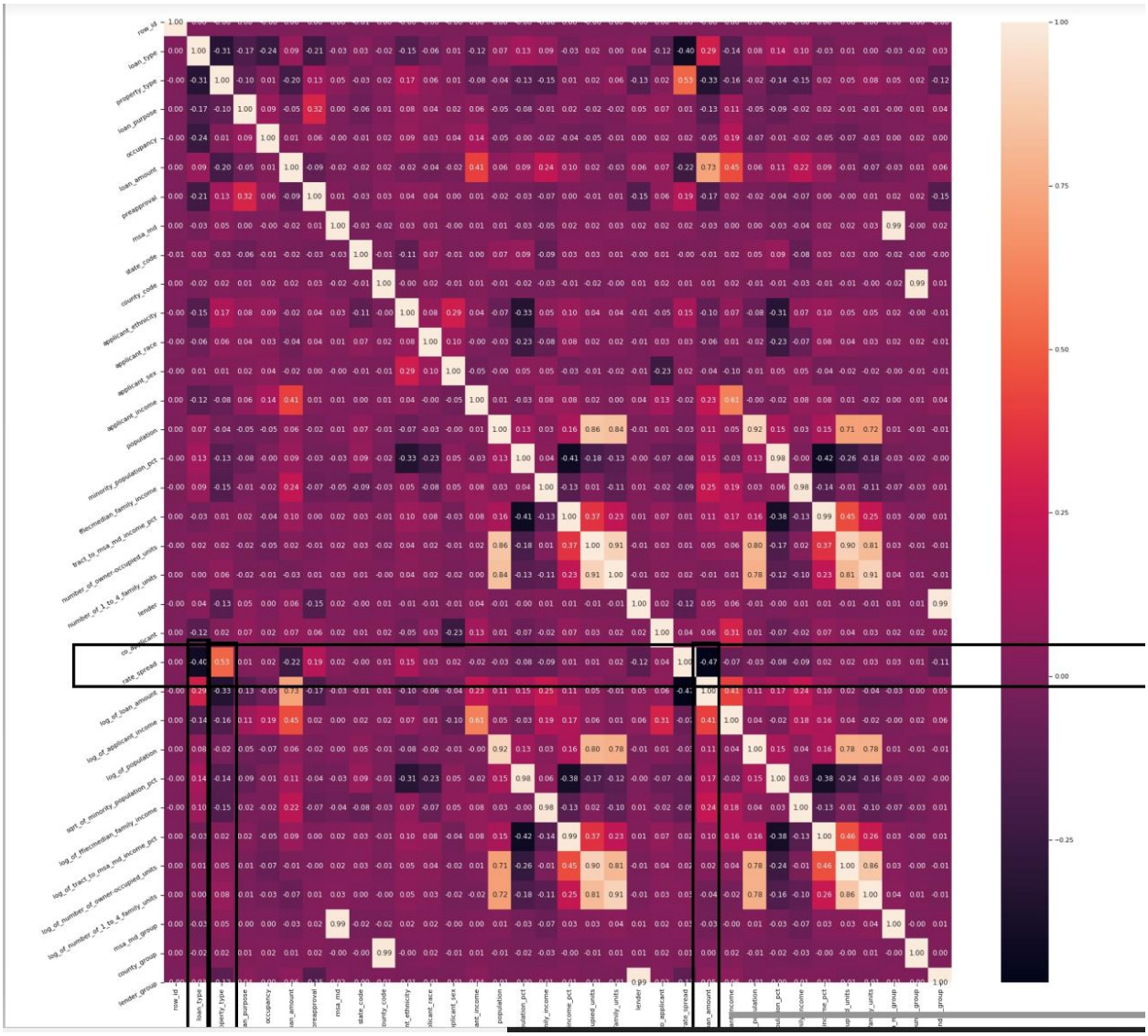
The histograms of the transformed features had Gaussian distributions. An example is below.



Also, the range of the transformed numeric features were all within 0-20.

The following categorical features were binned: ‘msa_md’ , ‘county_code’, ‘lender’

The resulting correlation matrix is given below:



The dependent variable “rate_spread” was mostly correlated to the following features:

- Loan_type
- Property_type
- Log_of_loan_amount

The author decided to use all available categorical features and the transformed numeric features for model training.

2.5 - Model training and optimization

CatBoostRegressor[1] was chosen as the algorithm for training since it handles categorical features without conversion. Also, the algorithm provided successful results in many Kaggle competitions.

Cross validation was used with fold=3. The initial run of the model with the following parameters resulted in a r-squared score of 0.7478 with standard deviation of 0.002

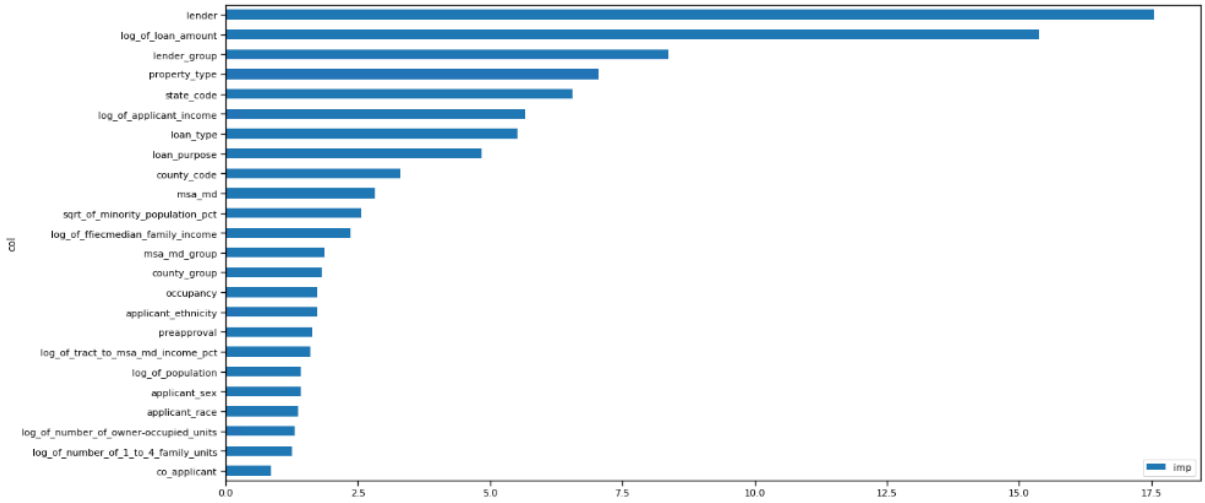
```
model=CatBoostRegressor(iterations=200, depth=6, learning_rate=0.1,
loss_function='RMSE')
```

GridSearch with Cross Validation was utilized to compute the optimal hyperparameters. The optimal model had the following hyperparameter:

```
{'depth': 12, 'iterations': 800, 'learning_rate': 0.1}
```

With the optimal hyperparameters given above the R2 score increased from 0.7478 to 0.7721.

While it is possible to improve on the model by expanding the count and the range of parameters for GridSearch, the author decided to post the results. The feature importance ratings obtained are listed in the table below:



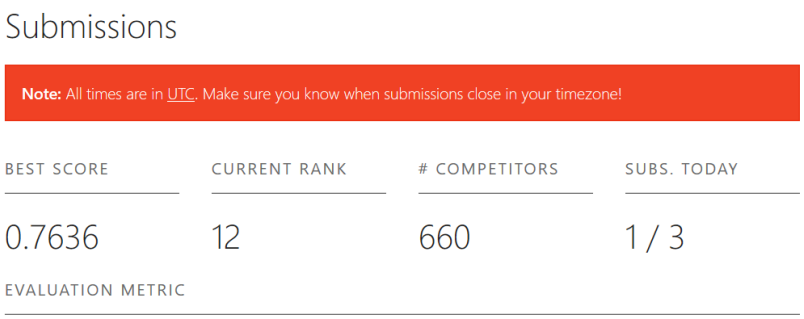
It seems that the selected lender, amount of loan, the location of the property (state/county code) as well as the type & purpose of the mortgage are important.

If less computation is required, then the model can be re-trained with those features or feature reduction methods such as Principle Component Analysis (PCA) or Linear Discriminant Analysis (LDA) could be utilized.

2.6 – Prediction

Finally, the optimal model was used on the test (validation) dataset.

As of Nov/18/2019, this submission ranked 12th in the leader scoreboard:



3 – KEY FINDINGS AND CONCLUSION

Analysis of the training dataset revealed the following:

- Majority of the mortgages were for home purchase, FHA-insured and for 1-4 room family homes.
- Gender, race and ethnicity seem to have some influence on the mortgage approvals. Most mortgages were awarded to whites and average rate spread was higher for certain ethnic groups. If an applicant did not specify race or gender, the standard deviation was higher for rate spread.

The CatBoostRegressor can be used after applying feature transformations and binning of certain categorical features to predict rate spread for the mortgage applications mentioned in the executive summary. As it stands, the model has an average R2 of 0.77 based on grid search & cross validation results.

In the current model, lender, loan amount, location of the property, type of property and purpose of the loan seem to be important features in determining the rate spread.

4 - RESOURCES

(1) https://catboost.ai/docs/concepts/python-reference_catboostregressor.html
(2) https://www.datasciencecapstone.org/competitions/18/mortgage-rates-from-government-data/page/56/
(3) https://www.ffiec.gov/hmda/default.htm