

# PREDICTION OF MORTGAGE FORECLOSURES



# PROBLEM STATEMENT

Determine how to to assess the risk of mortgage loan application by categorizing into one of two categories:

- Foreclosure (Likely to foreclose)
- Non-Foreclosure (Not likely to foreclose)



# DATA SOURCES

Fannie Mae has made available on their website a subset of the single-family loans that were acquired by the agency since 2000.

The data set is available in .txt format by quarter and each quarter has the following two types of data sets:

- Acquisition Data (identifying data i.e. loan type, borrower credit score, original interest rate)
- Performance Data (monthly data i.e. current loan balance, delinquency status, loan age)



# DATA WRANGLING

The following steps were taken to clean up the data:

- Adding header names
- Removing rows with missing data for borrower credit score
- Filling in number of borrowers with either 1 or 2 depending on credit score data
- Combining the two data set files into one DataFrame



# EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) was conducted by visually exploring the data to gather insights and trends from summary statistics.

Below are the features that were initially reviewed:

- Borrower Credit Score, Co-Borrower Credit Score
- Zero Balance Code
- Original Interest Rate
- Original Unpaid Balance
- Original Loan to Value
- Original Debt-to-Income Ratio



# EXPLORATORY DATA ANALYSIS

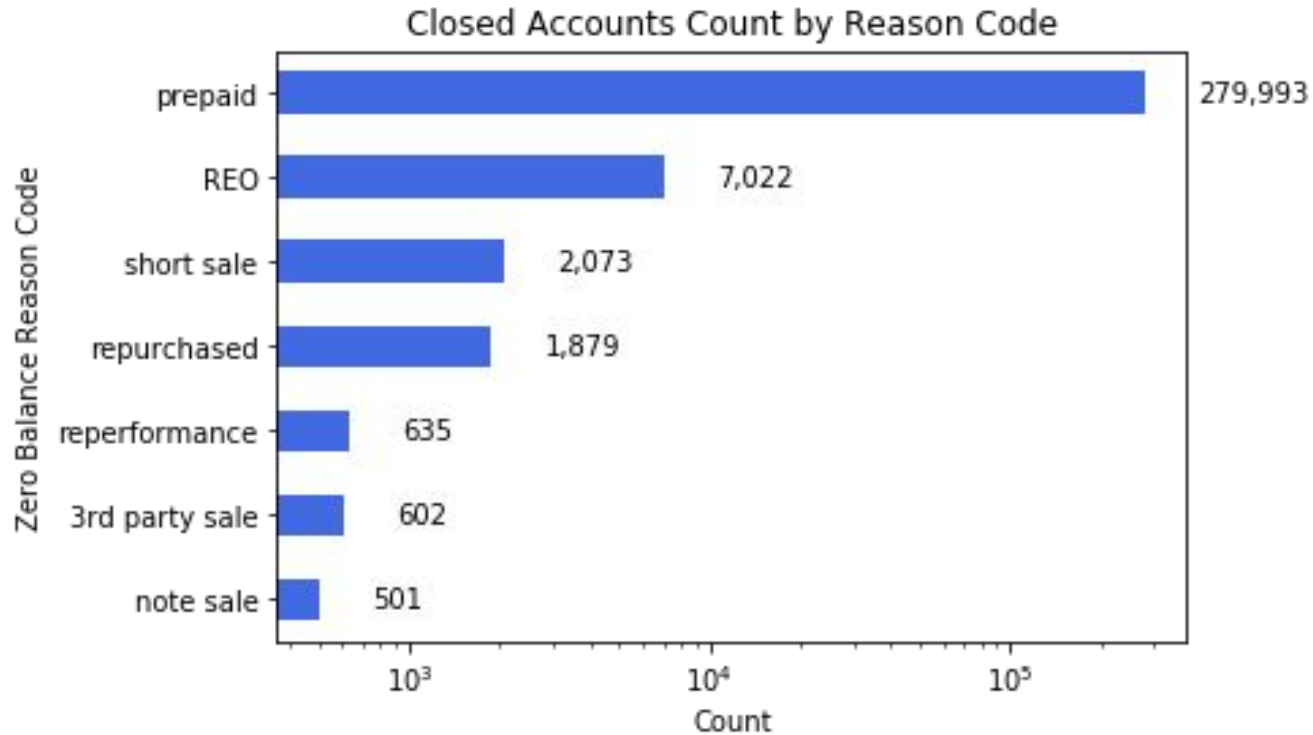
The loans originating in the 4th quarter of 2008 were used for this analysis.

**315k** total mortgage loan accounts

**93%** of accounts are closed

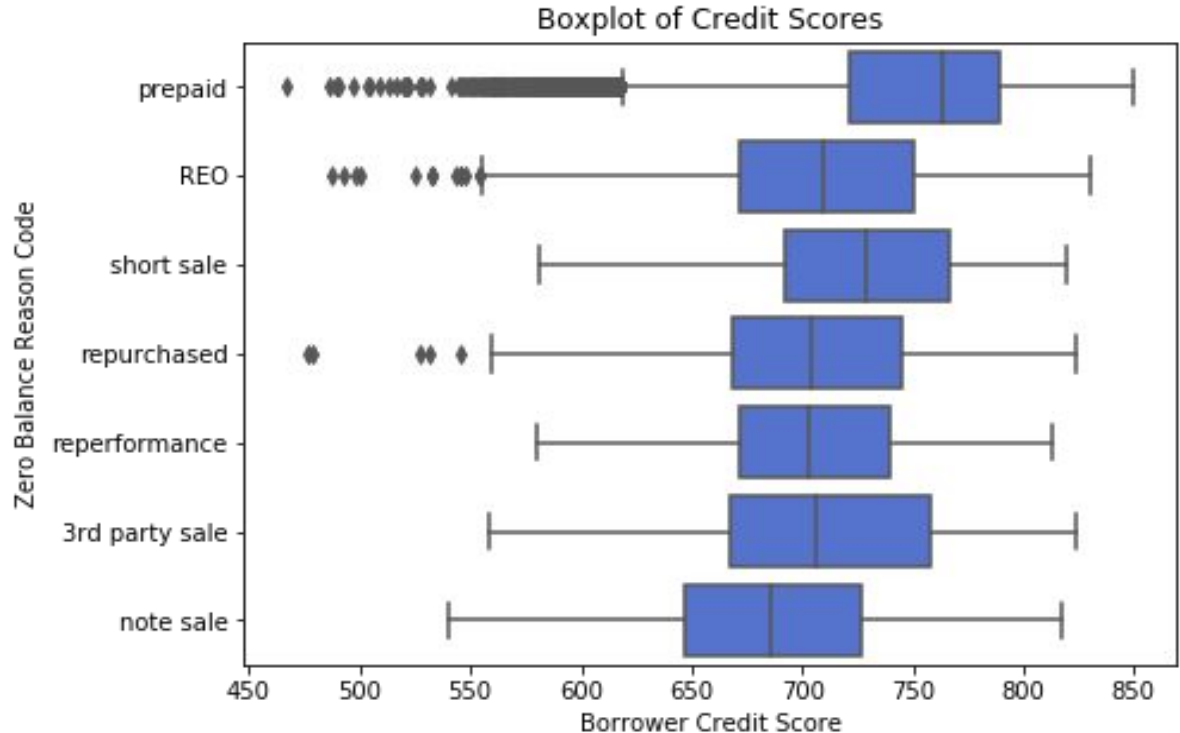


# EXPLORATORY DATA ANALYSIS



# EXPLORATORY DATA ANALYSIS

- With a larger portion of the data, the prepaid accounts have a credit scores with a larger variance and higher medians.
- Accounts closed coded as REO had significantly lower median.





# INFERENCEAL STATISTICS

**Inferential statistics** use random sampling from a population to make deductions about a data set.

**Bootstrap sampling**, which involves random sampling with replacement, was employed to test the null hypothesis.

In general, the **Null hypothesis** states that there is not anything significantly different between groups of data.



# INFERENCEAL STATISTICS

The null hypothesis used for this data set is that there is not a difference between credit scores of REO accounts and prepaid accounts.

$\overline{m}_P$  = population mean of prepaid loans

$\overline{m}_R$  = population mean of REO loans

$\overline{x}_P$  = sample mean of prepaid loans

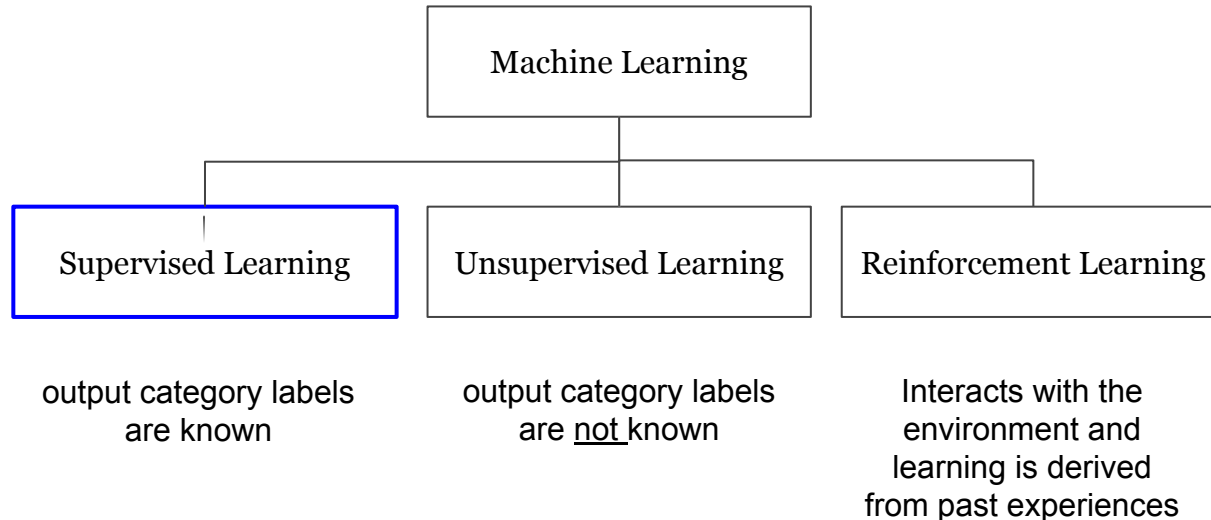
$\overline{x}_R$  = sample mean of REO loans

$$p \text{ value} = \sum \left[ (\overline{m}_P - \overline{m}_R) \geq (\overline{x}_P - \overline{x}_R) \right] / 10,000$$



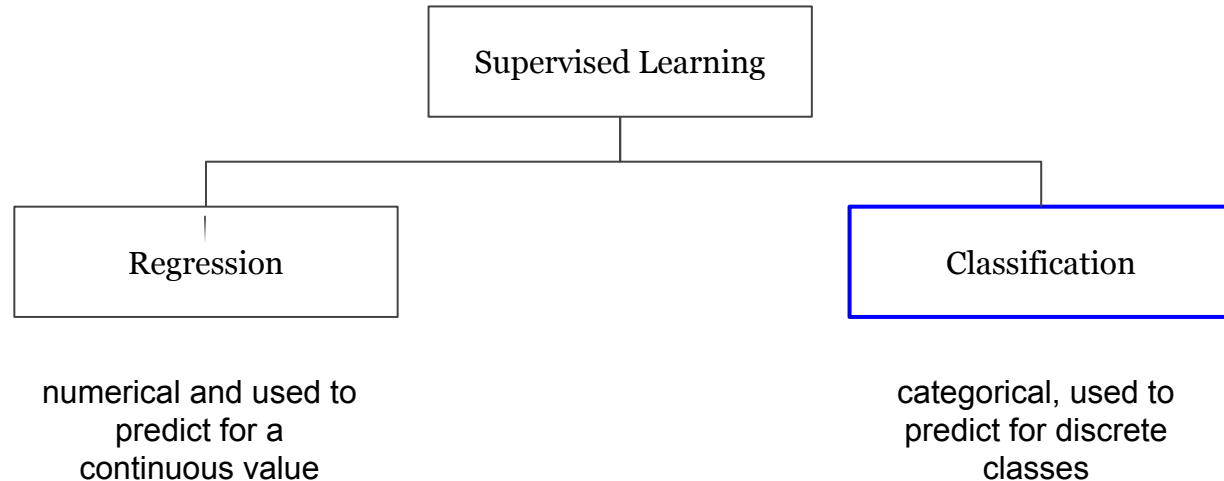
# MACHINE LEARNING

Machine learning is a branch of artificial intelligence that uses algorithms to learn without explicitly being programmed.



# MACHINE LEARNING

Of the two types of supervised learning, classification is the appropriate choice for placing each loan into two categories.



# MACHINE LEARNING

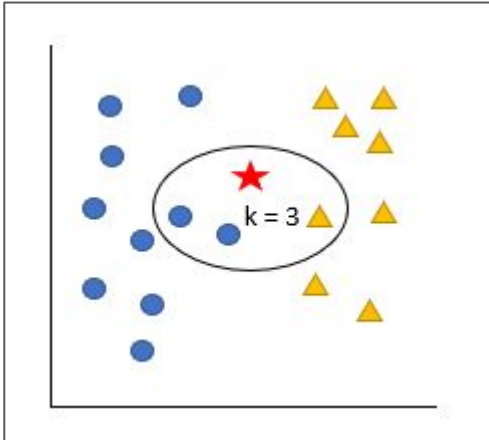
Three classification models were considered and metrics compared for this analysis.

- K Nearest Neighbors - predict the class of new data by using its proximity to labeled data
- Logistic Regression - uses the logistic function, an S-shaped curve that can take any value between 0 and 1, for binary classification
- Random Forest - aggregates multiple decision trees (makes classification based on a series of rules from the features of the data set) and uses the majority vote to determine the class

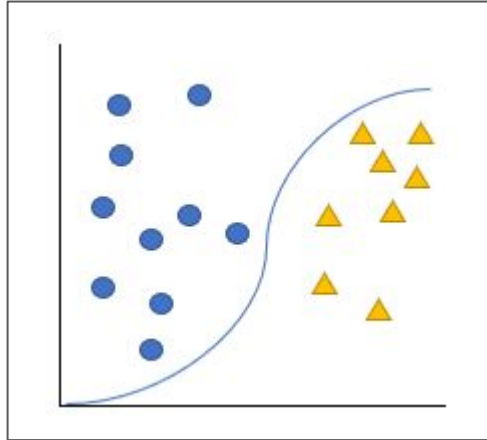


# MACHINE LEARNING

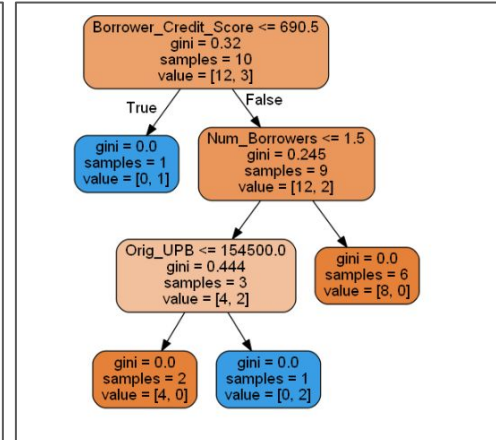
K Nearest Neighbors



Logistic Regression



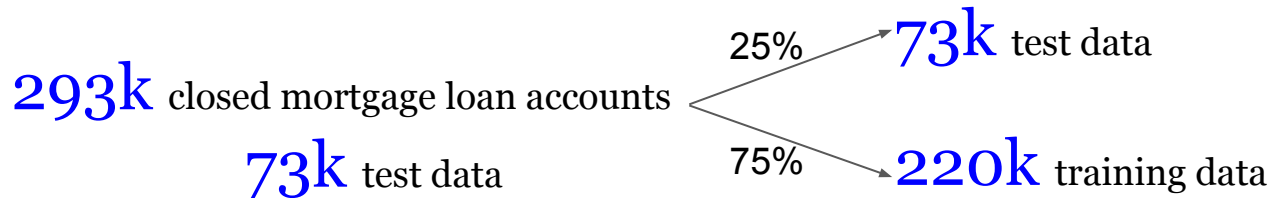
Random Forest



# MACHINE LEARNING

In supervised machine learning, the data set is split into two separate subsets, training and test.

The training subset is used to learn the data and make predictions on the test data set. 75% of the data was assigned to the training set and the remaining 25% assigned for the testing.



# MACHINE LEARNING

The F1 score was used to measure and compare each of the three algorithms' test data results. F1 is a composite score of the precision and recall scores.

**Precision** measures the proportion of positive identifications that are correct:  $\text{True Positive} / (\text{True Positive} + \text{False Positive})$

**Recall** measures the proportion of actual positives correctly identified:  $\text{True Positive} / (\text{True Positive} + \text{False Negative})$

**F1 Score** is the weighted average of the precision and recall scores:  $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$





# MACHINE LEARNING

The features included in the models were reduced to 11 after reviewing the summary from the function `pandas_profiling.ProfileReport()`.

- 'Borrower\_Credit\_Score' (Borrower Credit Score)
- 'Coborrower\_Credit\_Score' (Co-borrower Credit Score)
- 'Orig\_Debt\_Inc\_Ratio' (Original Debt to Income Ratio)
- 'Orig\_LTV' (Original Loan to Value)
- 'Orig\_UPB' (Original Unpaid Balance)
- 'First\_Time\_Home\_Buyer\_Ind' (First Time Home Buyer Indicator)
- 'Loan\_Purpose' (Loan Purpose)
- 'Property\_Type' (Property Type)
- 'Num\_Borrowers' (Number of Borrowers)
- 'Occupancy\_Type' (Occupancy Type)
- 'Zip\_Code' (Zip Code)



# MACHINE LEARNING

The **Random Forest** classifier was the first to be executed and measured.

```
1 # Random forest classifier
2 from sklearn.ensemble import RandomForestClassifier
3 def random_forest(n_estimator, max_feature, min_samples_split):
4     global y_test, y_predict, model
5     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=0)
6     model = RandomForestClassifier(n_estimators=n_estimator, random_state=0, max_features=max_feature,
7                                   min_samples_split=min_samples_split)
8     model.fit(X_train, y_train)
9     y_predict = model.predict(X_test)
10    return y_test, y_predict, model
```

```
1 # Initial random forest run of model using default hyperparameters
2 random_forest(100, int(np.sqrt(len(features))), 2)
3 print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	0.97	1.00	0.98	70885
1	0.68	0.02	0.04	2292
avg / total	0.96	0.97	0.95	73177



# MACHINE LEARNING

As shown in the previous slide, the total F1 score for the Random Forest classifier was quite high. However, when taking a closer look, there was an imbalance between the two classifications. 0 represents non-foreclosures, whereas 1 represent foreclosures. Since there were so many more non-foreclosures, the composite score was very high, however the f1 score for the foreclosures was unacceptably low at a mere 4%.

	precision	recall	f1-score	support
0	0.97	1.00	0.98	70885
1	0.68	0.02	0.04	2292
avg / total	0.96	0.97	0.95	73177



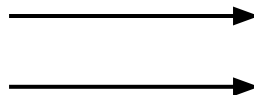
# MACHINE LEARNING

To mitigate the imbalance, resampling in the form of random undersampling was performed to reduce the count of non-foreclosure equal to the number of foreclosures.

```
# Resample data for undersampling
from imblearn.under_sampling import NearMiss
nr = NearMiss()
X, y = nr.fit_sample(X, y)
```

Original Data

9.7k foreclosure loans  
283k non-foreclosure loans



Resampled Data

9.7k foreclosure loans  
9.7k non-foreclosure loans



# MACHINE LEARNING

F1-score for foreclosures improved after undersampling the going from from 4% to 79%.

## Before Sampling

	precision	recall	f1-score	support
0	0.97	1.00	0.98	70885
1	0.68	0.02	0.04	2292
avg / total	0.96	0.97	0.95	73177

## After Sampling

	precision	recall	f1-score	support
0	0.79	0.77	0.78	2402
1	0.78	0.79	0.79	2430
avg / total	0.78	0.78	0.78	4832



# MACHINE LEARNING

**K Nearest Neighbors** was executed next and the F1 score for foreclosures was calculated as 75%.

```
1 # k nearest neighbors
2 from sklearn.neighbors import KNeighborsClassifier
3 X_train, X_test, y_train, y_test = train_test_split(
4     X, y, test_size = 0.25, random_state=0)
5
6 knn = KNeighborsClassifier(n_neighbors=5)
7
8 knn.fit(X_train, y_train)
9
10 y_predict = knn.predict(X_test)
11 print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	0.72	0.89	0.80	2402
1	0.86	0.66	0.75	2430
avg / total	0.79	0.78	0.77	4832



# MACHINE LEARNING

Finally, **Logistic Regression** was executed and the F1 score for foreclosures was calculated as 66%.

```
1 # Logistic regression
2 from sklearn.linear_model import LogisticRegression
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
4                                                    random_state=0)
5 logreg = LogisticRegression()
6 logreg.fit(X_train, y_train)
7 y_predict = logreg.predict(X_test)
8 print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	0.65	0.59	0.62	2402
1	0.63	0.69	0.66	2430
avg / total	0.64	0.64	0.64	4832



# MACHINE LEARNING

The **Random Forest** model having the highest F1 score of 79% for foreclosure loans was chosen as the best choice for this classification problem.

Three of the hyperparameters were tuned to achieve a slightly better F1 score:

- **n\_estimators**=[10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000], the number of trees in forest
- **max\_features**=[2, 3, 4, 5, 6, 7, 8, 9, 10, 11], number of features to consider when looking for the best split
- **min\_sample\_split**= [2, 4, 8, 16, 32, 64, 128, 256], minimum number of samples required to split an internal node





# MACHINE LEARNING

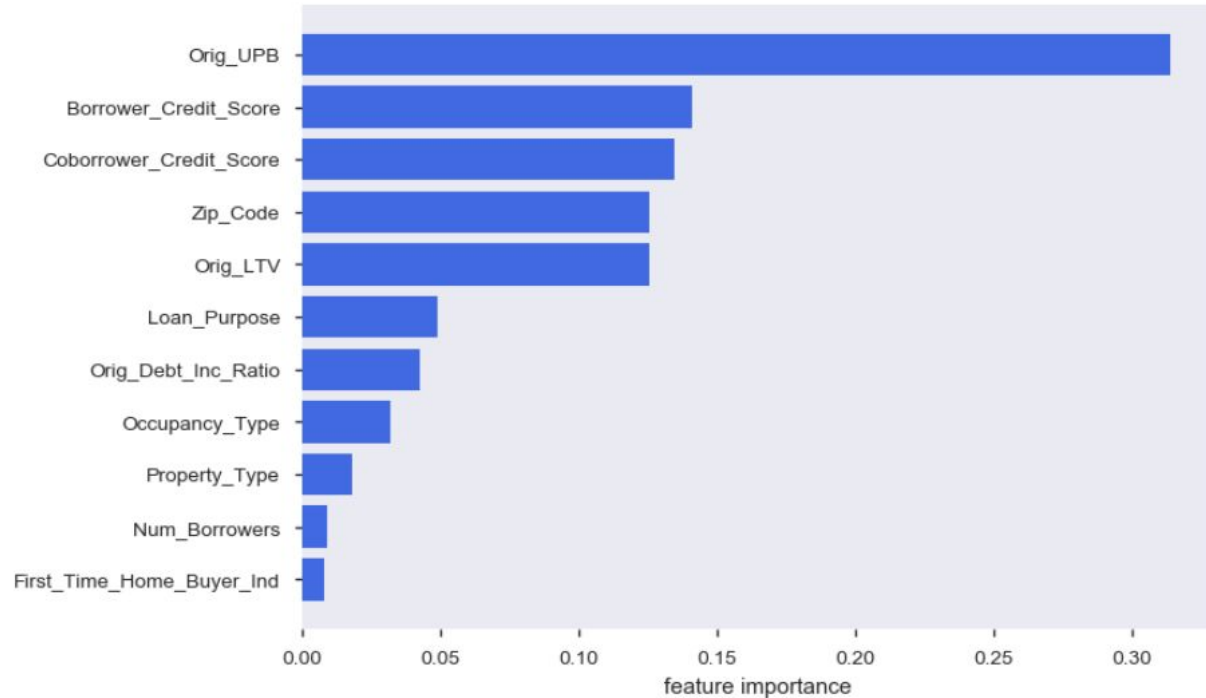
The final Random Forest F1 score for predicting loan foreclosures (with `n_samples=400`, `max_features=2`, and `min_sample_split=64`) was calculated as 80%.

	precision	recall	f1-score	support
0	0.81	0.75	0.78	2402
1	0.77	0.82	0.80	2430
avg / total	0.79	0.79	0.79	4832



# MACHINE LEARNING

Percent of importance by feature for Random Forest model



# CONCLUSION

- Using the random forest classifier on the Fannie Mae data set provided a fairly good model, with an F1 score of 80%, to predict the loans that may foreclose with the sample of 19k records.
- This model may be improved by reducing the number of features and further tuning of the hyperparameters.
- Evaluation of data from additional year's quarters may improve upon the model's accuracy scores.

