Machine Learning Nanodegree

Capstone Project

Predicting Loan Default in the Fannie Mae Single-Family Loan Performance Data

Felipe Ferreira

Udacity

Fall 2017

1 Definition

1.1 Project Overview

The Federal National Mortgage Association (FNMA), commonly referenced as Fannie Mae, is a corporation with the purpose to expand the secondary mortgage market by securitizing mortgages in the form of mortgage-backed securities (MBS). In a nutshell, Fannie Mae acquires mortgage loans from primary lenders such as Wells Fargo, Chase, and Quick Loans, among others. After the mortgages' acquisition, Fannie Mae groups them in mortgage pool, which, by a diversification effect, has lower risk than the risk of each mortgage individually. Until the 2008 financial crisis, these mortgage pools (MBS) sold by Fannie Mae were considered stable investments; however, the crisis showed that these pools were not as safe as people thought. In fact, during the crisis burst, many people defaulted on their mortgages, causing these securities prices to decreases significantly, thereby severely impacting the performance of these investments.

Following the crisis, Fannie Mae with the intention to promote a better understanding of the credit performance of Fannie Mae mortgage loans, started providing loan performance data on a portion of its single-family mortgage loans. The goal of this project is to predict from Fannie Mae single-family performance data, those borrowers who are most at risk of defaulting on their loans.

1.2 Problem Statement

The project's goal is to predict with high degree of accuracy whether a given mortgage will default or not based on the mortgage's characteristics at the time of its acquisition by Fannie Mae. Among the loan features, we will use to make our predictions; we have the debt-to-income ratio, borrower's credit score, and loan amount, and many others.

Our strategy is to use Ensemble Models - particularly, Random Forest models - to solve the defaulting prediction problem. We propose their use because they have been extensively used in Anomaly Detection problems and they are easier to interpret than more complex models such as Neural Networks and SVM.

The document is divided into the subsequent sections: Analysis, where we explore the data, its origin, features, and any possible problem we might encounter in the data and how we will solve these problems; we also dicuss in more details the algorithm we will use to tackle the problem and the baseline model for our problem. In the following section, Methodology, we examine the steps taken to preprocess the data, the model implementation and refinement, as well as the model evaluation and validation. Finally, we conclude the document with an analysis with the most important features for the final model, a reflection of how the whole project was developed and a discussion on any improvement that might be important for future development.

1.3 Metrics

The evaluation metric for the proposed model and the benchmark model is the Receiver Operating Characteristic (ROC) curve. Although sensitivity and specificity could be valid metrics, they are dependent on the right assessment of the appropriate trade-off between sensitivity and specificity. The ROC Curves provide the representation of the range of possible cut points with their associated true positive rate vs. false positive rate. And the area under the curve AUC offers the means to compare two or more models.

2 Analysis

2.1 Data Exploration

Fannie Mae has made available on its website[1] a portion of its single-family mortgage loans dataset. The Single Family Fixed Rate Mortgage dataset contains a subset of Fannie Mae's 30-year or less, fully amortizing, full documentation, single-family, conventional fixed-rate mortgages.

The loan performance dataset is divided into two files for each acquisition quarter. The **Acquisition file** includes static data at the time of the mortgage loan's origination and delivery to Fannie Mae. The **Performance file** contains the monthly performance data of each mortgage loan from the time of Fannie Maes acquisition up until its current status as of the previous quarter. For the project, we will use the data from the 2^o quarter of 2007.

Since we want to predict whether a loan will default given its characteristics at the time of its acquisition, we will merge the **Acquisition file** which contains the features to our model with the **Performance file** which contains our target variable (default or not default) using the *Loan ID* presented in both files as the key field for the merging. Moreover, since the dataset does not have a field directly informing whether a loan has defaulted or not, we will use the *Foreclosure Date* field as a proxy - if it is populated, it means that the loan has defaulted.

The final dataset has 287,286 samples and 25 features:

- Sales Channel - Type: Categorical
- Seller Name - Type: Categorical
- Original Interest Rate - Type: Numeric
- Original Unpaid Principal - Type: Numeric
- Original Loan Term - Type: Numeric
- Original Loan-to-Value Ratio - Type: Numeric

[1] `http://www.fanniemae.com/portal/funding-the-market/data/loan-performance-data.html`

- Original Combined Loan-to-Value Ratio - Type: Numeric
- Number of Borrowers - Type: Numeric
- Debt-to-Income Ratio - Type: Numeric
- Borrower Credit Score - Type: Numeric
- First-Time Home Buyer Indicator - Boolean
- Loan Purpose - Type: Categorical
- Property Type - Type: Categorical
- Number of Units - Type: Numeric
- Occupancy Status - Type: Categorical
- Property State - Type: Categorical
- ZIP Code (first 3 digits) - Type: Categorical
- Relocation Mortgage Indicator - Type: Boolean
- Month of Mortgage Acquisition by Fannie Mae - Type: Numeric
- Year of Mortgage Acquisition by Fannie Mae - Type: Numeric
- Month of the First Payment - Type: Numeric
- Year of the First Payment - Type: Numeric

In addition, in the next section we present box-plots for the feature we think might be important to predict the loan default. We also present, in the annex Data Example, a few examples presented in the final dataset.

2.1.1 Categorical Features

Given we have a large number of categorical features, and each has several categories. We plan to use Label Enconding, instead of One-Hot-Encoding. Otherwise, we will have to deal with a large and sparce input matrix.

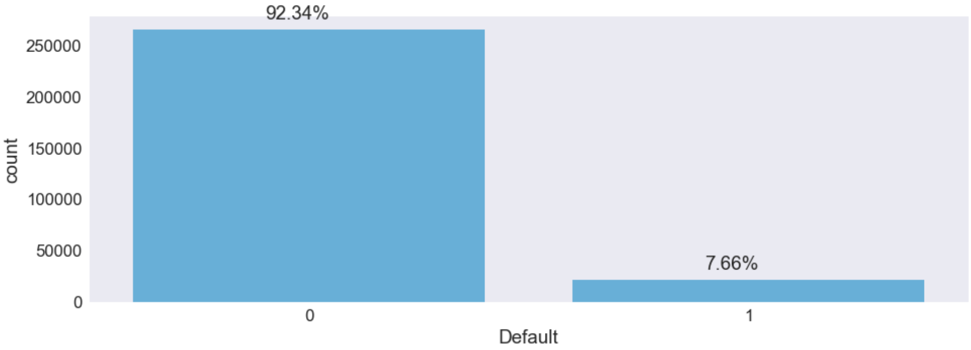


Figure 1: Imbalanced Dataset

2.2 Exploratory Visualization

2.2.1 Imbalanced Dataset

From Figure 1, we see that the number of loan that defaulted since their acquisition date are only 7.66% of the samples. Thus, it is clear we are dealing with an imbalanced dataset (when the number of observations in one class is much larger than those belonging to other classes). This behavior was alreasy expected, since loan default prediction belongs a class of problems we can call **Anomaly Detection**. Belonging to this class, we also have identification of diseases ans fraudulent transactions in banks.

2.2.2 Possible Important Features

Let's look at the most important features when we analyse loan's characteristics:

- **Original Loan-to-Value** - The loan-to-value ratio is a financial term used by lenders to express the ratio of a loan to the value of an asset purchased. The higher the LTV ratio, the riskier is the loan;

- **Debt-to-Income Ratio** - The debt-to-income ratio is the percentage of a consumer's monthly gross income that goes toward paying debts. The higher the DTI ratio, the riskier is the loan;

- **Credit Score** - The borrower credit score at the time of the loan acquisition by Fannie Mae;

- **Original Interest Loan** - The loan interest rate;

From Figure 2,we see that our intuition was correct. Defaulters have, on average, higher Loan-to-Value ratio, higher Debt-to-Income ratio, lower Credit Score at the time of the loan acquisition. Interestingly, the interest rate boxplot indicates that the loan rate might not have a preditive power over which loan will defaulted.

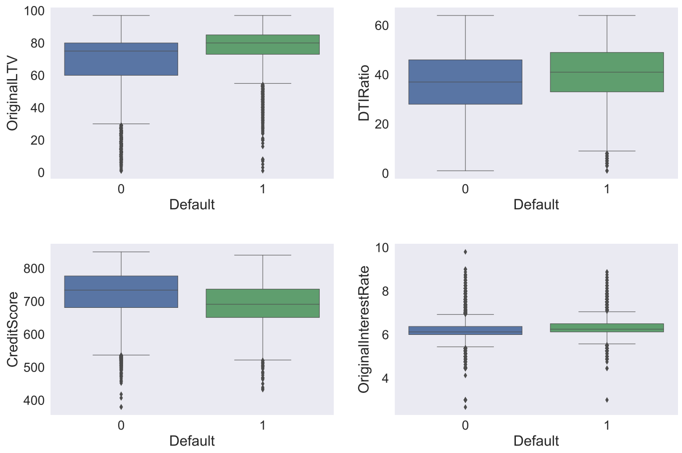


Figure 2: Possible Important Features

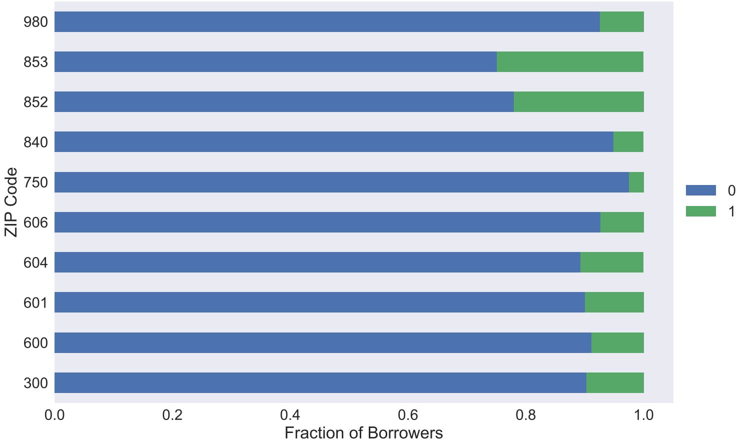


Figure 3: Defaulting Distribution on the 10 most common ZIP Codes

Besides these features, Figure 3 also shows that the ZIP Code may be a significant factor when predicting whether a loan will default. Given the differences in the proportion of defaulters by ZIP Code.

2.2.3 Missing Values

Figure 4 shows the features with missing values. With large number of missing values, we will drop the MortInsurancePerc, the CoCreditScore, and the MortInsuranceType features entirely. Besides these features, we will also drop the feature ProductType, since it has only one value.

For the other four features - OriginalCLTV, NumBorrowers, DTIRatio, and CreditScore - with missing values, we will not drop them altogether but only those rows with the missing values. We have chosen this approach because the fraction of missing values is relatively

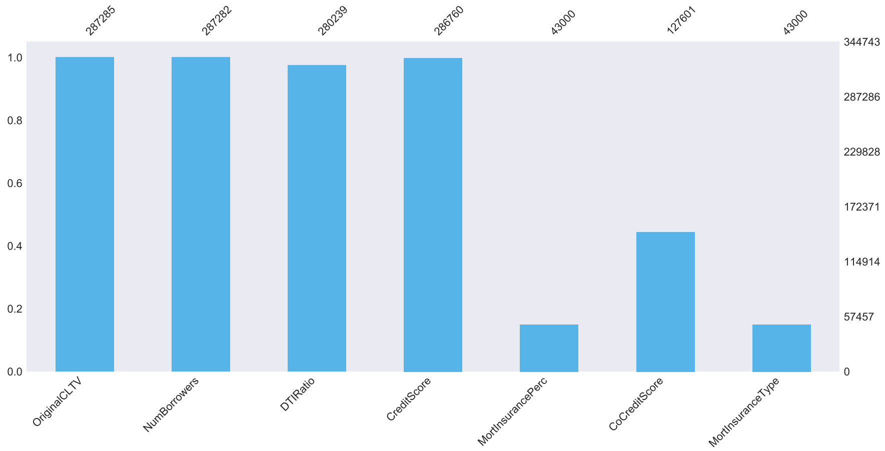


Figure 4: Features with Missing Values

small and as discussed before, these features might be important to our model.

2.3 Algorithms and Techniques

The project attempts to predict whether a mortgage loan will default based on its characteristic at the time of the loan acquisition by Fannie Mae.

As ensemble models, particularly Random Forest algorithms, have been extensive and successfully used to solve this type of problem, we will use these type of machine learning algorithms to solve our problem.

Random Forests are a combination of decision tree predictors where each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest.

The combination of simpler algorithms (weak predictors) into a much stronger model is a technique known as Ensemble Models. Random Forests are an excellent tool for making predictions considering Random Forests are a combination of decision tree predictors they are easier to train where each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest.

Decision trees usually have high variance or high bias. Random Forests attempts to mitigate these issues by averaging the individual decision trees outputs aiming to find a natural balance between the two extremes. Considering that Random Forests have few parameters to tune and can be used merely with default parameter settings, they are easier models to train.

Random Forests model trains many decision trees. Where each tree is built as follows:

- If the number of examples in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.

- If there are M features, a number $m << M$ is specified such that at each node, m features are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
- Each tree is grown to the minimum leaf size.

Random Forest algorithms have the following hyperparameters:

- **Number of Estimators (Decision Trees)**: the maximum number of decision trees the model should build before taking the voting or averages of predictions. The higher number of trees better is the model performance; however, this increasing performance may slow the the algorithm;
- **Minimum Leaf Size**: Leaf is the end node in a decision tree. Small number of leafs makes the model susceptible to noise data, which in turn might affect the quality of the prediction;
- **Maximum Number of Features -** $\boldsymbol{m}$: The maximum number of features the algorithm is allowed to assess in an individual decision tree;

2.4 Benchmark

Our benchmark model will be a simple decision tree without any parameter optimisation. Because we can see Random Forests as an extension of Decision Trees, we think the latter might be a good benchmark for the former.

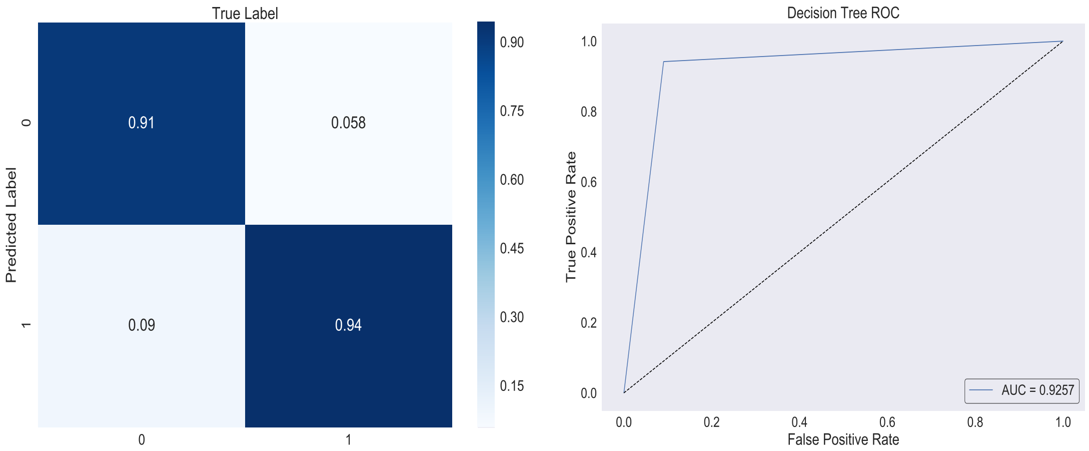
From rom the confusion matrix below, we see the model predict correctly 91% (true negatives) of all non-defaulters and 94% (true positives) of the defaulters. In term of importance for Fannie Mae, the number of false negatives is an important metric, since Fannie Mae profitabily is affected when a defaulter is misclassified. In our case, false negatives were approximately 9.1% of all defaulters.

The ROC curve (Receiver Operating Characteristics) shows the number of true positives vs. the number of false positives labeled by the algorithm for a number of classification threshold values. In the case of a perfect classifier, the area-under-the-curve (AUC) would be 1. The black dashed curve represents a random classifier (with no predicitive power) and has AUC equal to 0.5. In our case, it is clear that the model model has a great predictive power with AUC = 0.92. And this is number we want to overcome with the Random Forest model.

3 Methodology

3.1 Data Processing

For the pre-processing phase, we will apply the following steps:



(a) Confusion Matrix

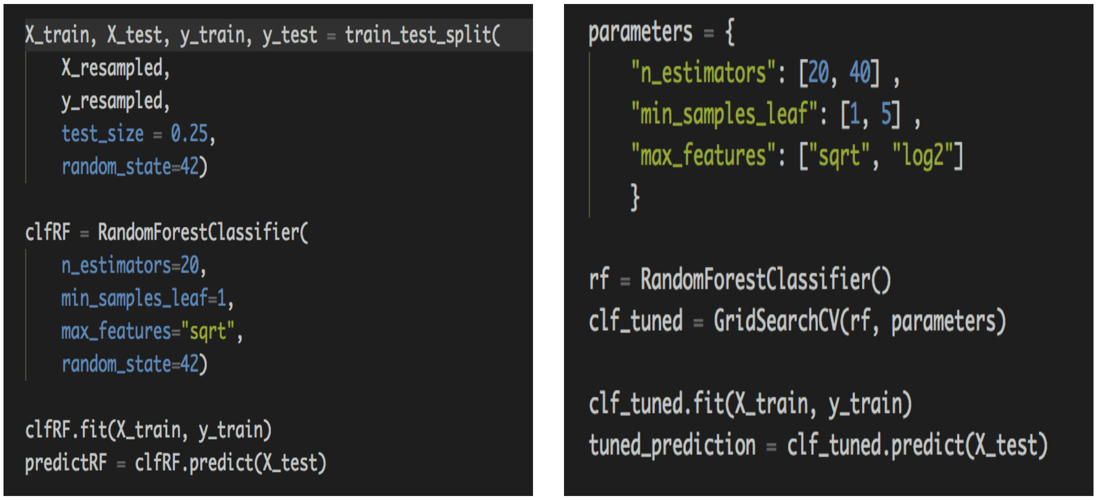
(b) ROC Curve

Figure 5: Decision Tree - Baseline Model

- Split the **OriginalDate** and the **FirstPayment** features in their year and months, in case these may have some predictive power;
- Missing Values
  - Drop the **MortInsurancePerc**, the **CoCreditScore**, and the **MortInsuranceType** features entirely, because they have a large number of missing values;
  - Drop rows with missing values in any other feature;
- Drop the **ProductType** feature because it has only one value;
- Use label enconding in all categorical features. We could have used one-hot enconding; however we might have ended up with a large sparce matrix, since we have many categorical features with some of them with many categories.
- To deal with the imbalanced dataset, we will apply a technique implementend in the library *imbalanced-learn* [2]called SMOTE: Synthetic Minority Over-sampling Technique [3] that works by interpolating new instances from the existing ones of the minority class.

[2] https://pypi.python.org/pypi/imbalanced-learn

[3] https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/chawla2002.html



```
X_train, X_test, y_train, y_test = train_test_split(
    X_resampled,
    y_resampled,
    test_size = 0.25,
    random_state=42)

clfRF = RandomForestClassifier(
    n_estimators=20,
    min_samples_leaf=1,
    max_features="sqrt",
    random_state=42)

clfRF.fit(X_train, y_train)
predictRF = clfRF.predict(X_test)
```

(a) Implementation Code

```
parameters = {
    "n_estimators": [20, 40] ,
    "min_samples_leaf": [1, 5] ,
    "max_features": ["sqrt", "log2"]
    }

rf = RandomForestClassifier()
clf_tuned = GridSearchCV(rf, parameters)

clf_tuned.fit(X_train, y_train)
tuned_prediction = clf_tuned.predict(X_test)
```

(b) Refinement Code

Figure 6: Tuned Random Forest Model

3.2 Implementation

The model was implementend using the library scikit-learn [4]. First, we split the preprocessed data into datasets - the training set, with 75% of the samples, and the testing set, with 25% of the samples. Following, we instantiate a RandomForestClassifier, with 20 decision trees, minimum samples leaf equals to 1 and maximum features equals to square root of the total number of features, Finally, we fit the model with the training data and make our predictions with the testing dataset.

The code snippet in Figure 6(a) shows each of these implementation steps:

3.3 Refinement

The model parameters were tuned with the sciit-learn function **GridSearchCV** [5]. See model parameters tuning in Figure 6(b).

3.4 Model Evaluation and Validation

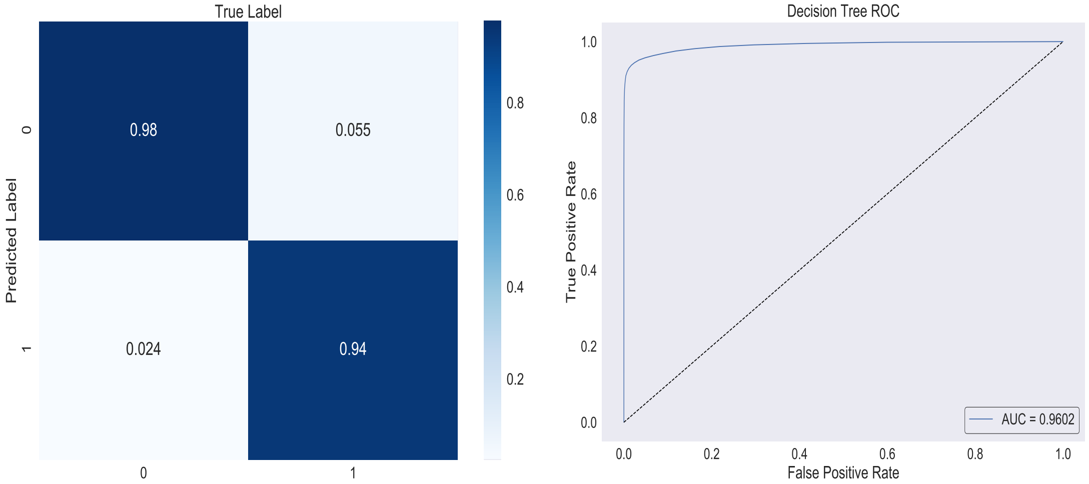
Below, we present the confusion matrix and the ROC curve for both models - the untuned Random Forest model and the tuned (using Grid Search) Random Forest. We also present the the ROC score for a 5-fold cross validation in the table below to the model's robustness, i.e. its effectiveness (accurracy) on new data and. With an average AUC Score of 0.9780

[4] http://scikit-learn.org/stable/index.html

[5] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Model	False Negative Rate	AUC Score
Baseline Model	5.58%	0.9257
Untuned Random Forest	5.50%	0.9602
Tuned (GridSearch) Random Forest	5.53%	0.9618
Average 5-Fold Cross Validation	-	0.9780

Table 1: AUC Score - Models



(a) Confusion Matrix

(b) ROC Curve

Figure 7: Untuned Random Forest Model

on the 5-fold cross validation, we can say the model's accuracy did not deteriorate when exposed to new data, which shows its robustness.

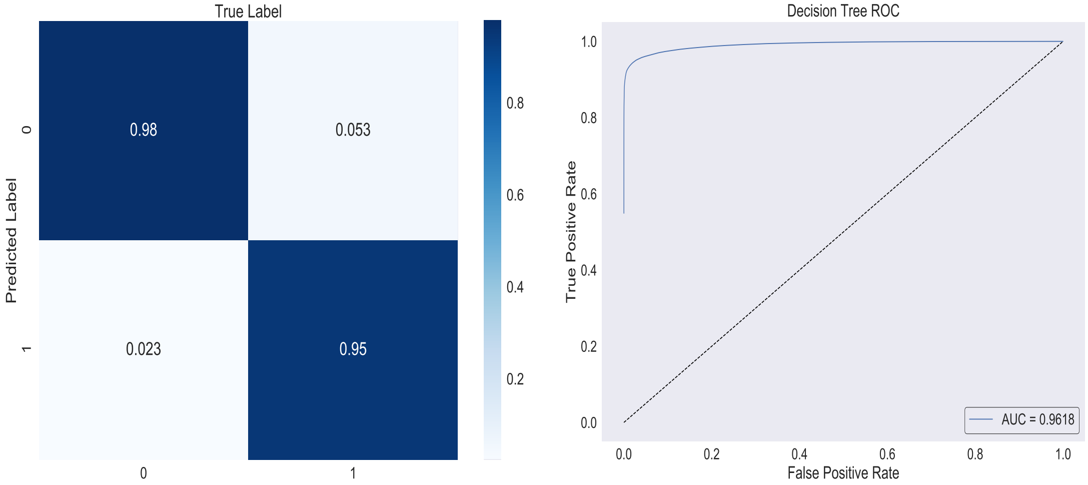
Though there is still room for improvement, as we discuss in the Improvement section. The final results clearly show the tuned model yielded better result than the baseline model.

4 Conclusion

4.1 Free Form Visualization

The importance of each feature is shown in the Figure X.

To test the performance of using the most important features, the tuned model was run with only the 10 top important features. The AUC score of this 10-top-tuned model was 0.9556, which is significatly lower than the AUC score of the all-features-tuned model, 0.9618.



(a) Confusion Matrix

(b) ROC Curve

Figure 8: Tuned Random Forest Model

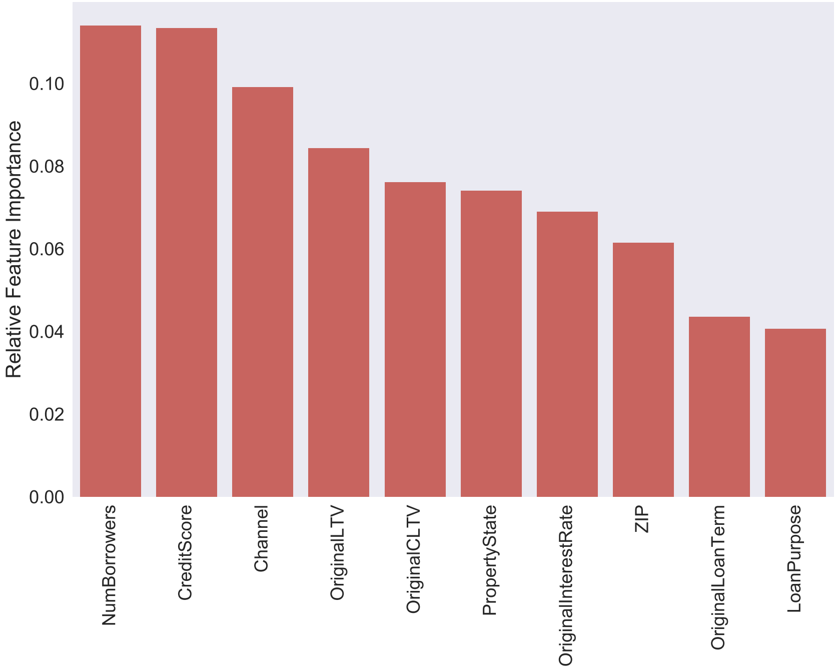
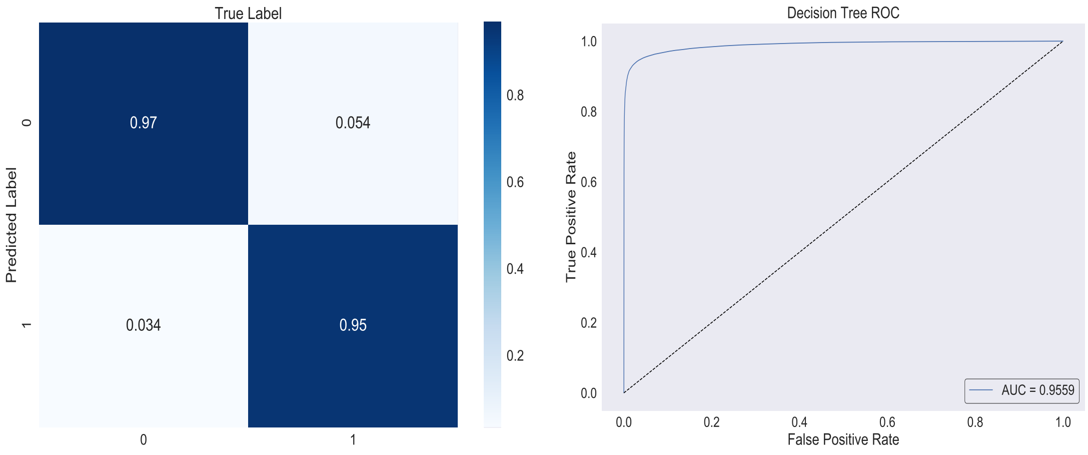


Figure 9: 10 most important features



(a) Confusion Matrix

(b) ROC Curve

Figure 10: 10-top tuned Random Forest Model

4.2 Reflection

The project is overall a well-defined classification problem. The projects goal was to predict with some degree of accuracy whether a given mortgage will default or not based on the mortgages characteristics at the time of its acquisition by Fannie Mae. The most time-consuming part of the project was pre-processing phase (missing values and categorical features treatment), which befits to the idea that the most laborious part in any data science project is data wrangling.

Another challenge was the model training. The notebook used to train the model was not power enough to either test several parameters or to use more advanced models, such as XGBoost algorithms.

Nonetheless, the results were satisfactory with the average AUC score of X in an 5-fold cross-validation.

4.3 Improvement

In the general, the tuned Random Forest model worked well, and showed an improved result over the Decision Tree (baseline) model.

To further improve the final model, we could use more power processor or even a GPU to accelerate the training and the parameter tuning. Also, we could use a XGBoost Classifier [6], which has shown on diverse winning solution on Kaggle superior performance than simple Decision Trees or Random Forest algorithms.

[6] http://xgboost.readthedocs.io/en/latest/model.html

5 Annex - Data Example

LoanID	100001007633	100001666223	100002547982
Channel	R	C	B
SellerName	OTHER	SUNTRUST MORTGAGE INC.	FLAGSTAR CAPITAL MARKETS CORPORATION
OriginalInterestRate	5.8750	6.0000	6.3750
OriginalUnpaidPrinc	190000	87000	318000
OriginalLoanTerm	360	240	360
OriginalDate	02/2007	03/2007	04/2007
FirstPayment	04/2007	05/2007	06/2007
OriginalLTV	80	80	72
OriginalCLTV	80.0000	80.0000	72.0000
NumBorrowers	2.0000	2.0000	1.0000
DTIRatio	27.0000	21.0000	34.0000
CreditScore	802.0000	737.0000	706.0000
FTHomeBuyer	N	Y	N
LoanPurpose	P	P	R
PropertyType	SF	SF	SF
NumUnits	1	1	1
OccStatus	P	P	P
PropertyState	MN	PA	CA
ZIP	554	150	923
MortInsurancePerc	NaN	NaN	NaN
ProductType	FRM	FRM	FRM
CoCreditScore	817.0000	743.0000	NaN
MortInsuranceType	NaN	NaN	NaN
RelocationMortgageIndicator	N	N	N
Default	0	0	1

LoanID	100135503492	100136433815	100136621701	100138879587
Channel	C	B	C	R
SellerName	CITIMORTGAGE, INC.	PNC BANK, N.A.	BANK OF AMERICA, N.A.	OTHER
OriginalInterestRate	6.2500	5.8750	6.5000	6.2500
OriginalUnpaidPrinc	133000	150000	229000	150000
OriginalLoanTerm	360	360	360	360
OriginalDate	05/2007	02/2007	03/2007	05/2007
FirstPayment	07/2007	04/2007	05/2007	07/2007
OriginalLTV	74	66	80	56
OriginalCLTV	74.0000	66.0000	95.0000	62.0000
NumBorrowers	2.0000	1.0000	2.0000	2.0000
DTIRatio	43.0000	45.0000	62.0000	58.0000
CreditScore	622.0000	732.0000	684.0000	756.0000
FTHomeBuyer	N	N	N	N
LoanPurpose	C	R	P	C
PropertyType	SF	CO	PU	SF
NumUnits	1	1	1	1
OccStatus	P	P	P	P
PropertyState	FL	VA	AZ	MI
ZIP	338	201	852	496
MortInsurancePerc	NaN	NaN	NaN	NaN
ProductType	FRM	FRM	FRM	FRM
CoCreditScore	629.0000	NaN	677.0000	753.0000
MortInsuranceType	NaN	NaN	NaN	NaN
RelocationMortgageIndicator	N	N	N	N
Default	0	1	1	0