Loan Default Prediction

Carla Rudder

Problem Statement

Banks offer loans to customers.

Customers either repay or the default on their loan.

Want to create a model to predict if a customer will default

Executive Summary

Banks want to better be able to **predict** which customer **will default** on their loan.

5960 loans were utilized to build a predictive model to determine if a customer would **default** on their loan.

12 variables observed in creating the model.

Findings – Indicators of Default

Debt to Income (DEBTINC) ratio was the number one indicator

Credit Line Age (CLAGE) was the second indicator

Number of **Derogatory (DEROG)** Credit Reports: DEROG >= 7 All default

Number of **Delinquent (DELINQ)** Credit Reports: DELINQ >= 6 All default

Chosen Model

- **Hyper Tuned Random Forest Model**
- 80% Accuracy
- Maximizing the recall – Minimizes the number of loans given to those who may default
- 80% Recall
- Same performance on Training and Test Set (F1- score =62%)

Business Implications

Risk Mitigation - Reduce the number of customers who default

Maintain Profitability- The bank will save money by having fewer defaulted loans

Bank will be able to identify the **characteristics** of the customer profile which **lead to default**

Customer Education - Opportunity for bank to provide financial education to their customer

Regulatory Compliance, Resource Allocation, Reputation Management, Credit Score Improvement, Maintain Competitive Advantage, Portfolio Diversity

Implementation

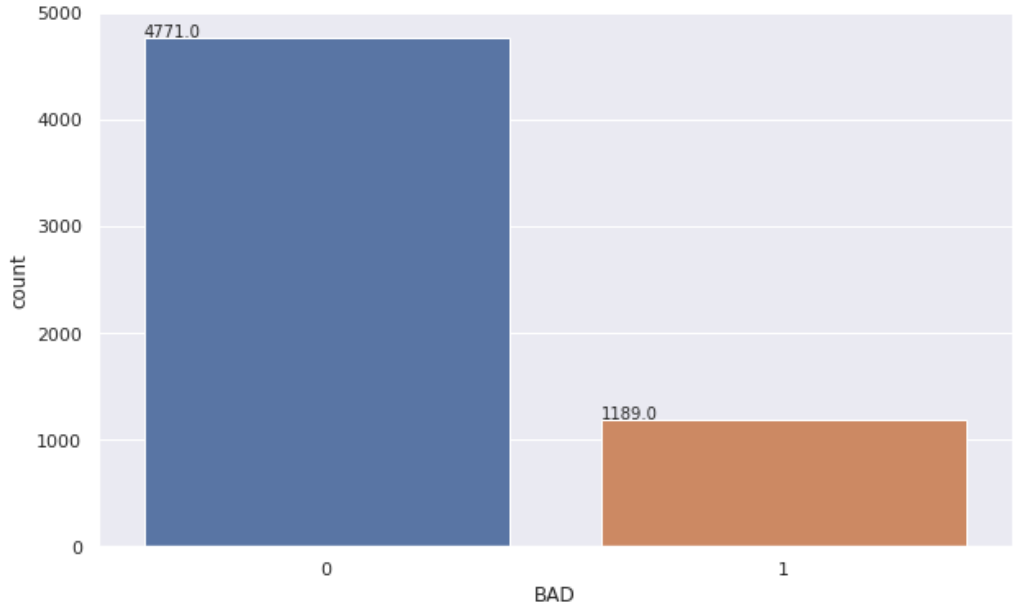
- **Ensure the Debt-to-Income ratio of potential customers is low**
- Ensure the customer has fewer than 6 derogatory reports (DEROG <6)
- Ensure the customer has fewer than 5 delinquent credit reports (DELINQ <5)
- Take greater care when offering a loan to customer who is self-employed
- Model integrated or stand alone, Training of loan officers, Continuous monitoring and maintenance, Feedback from loan officers

Process

Data Overview

Summary Statistics

1189 (20%) customers defaulted (1) on their loan



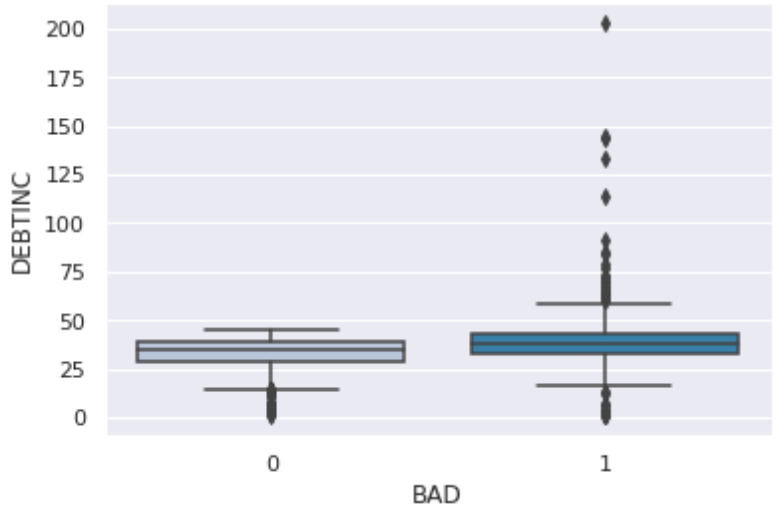
	LOAN	MORTDUE	VALUE	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
count	5960.000000	5442.000000	5848.000000	5445.000000	5252.000000	5380.000000	5652.000000	5450.000000	5738.000000	4693.000000
mean	18607.969799	73760.817200	101776.048741	8.922268	0.254570	0.449442	179.766275	1.186055	21.296096	33.779915
std	11207.480417	44457.609458	57385.775334	7.573982	0.846047	1.127266	85.810092	1.728675	10.138933	8.601746
min	1100.000000	2063.000000	8000.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.524499
25%	11100.000000	46276.000000	66075.500000	3.000000	0.000000	0.000000	115.116702	0.000000	15.000000	29.140031
50%	16300.000000	65019.000000	89235.500000	7.000000	0.000000	0.000000	173.466667	1.000000	20.000000	34.818262
75%	23300.000000	91488.000000	119824.250000	13.000000	0.000000	0.000000	231.562278	2.000000	26.000000	39.003141
max	89900.000000	399550.000000	855909.000000	41.000000	10.000000	15.000000	1168.233561	17.000000	71.000000	203.312149

Exploratory Data Analysis

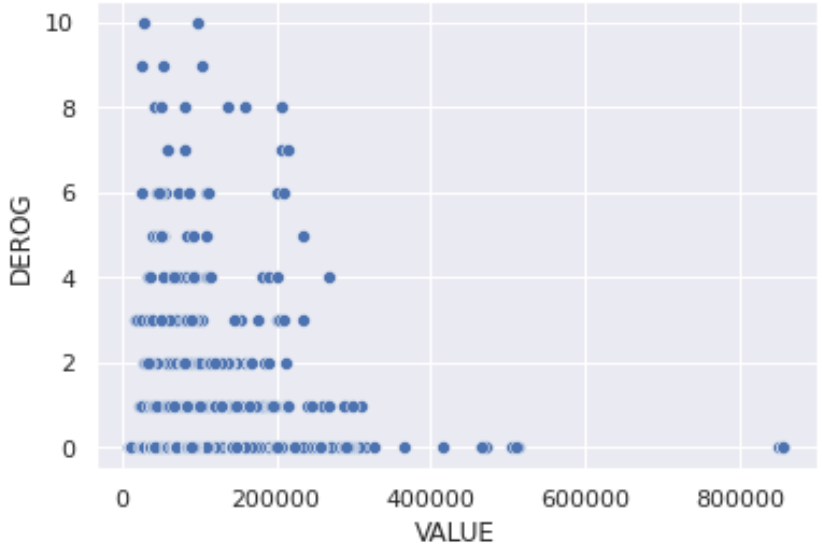
Visualized the data

Univariate and Bivariate analysis

Greater debt to income ratio **does** seem to result in more defaults



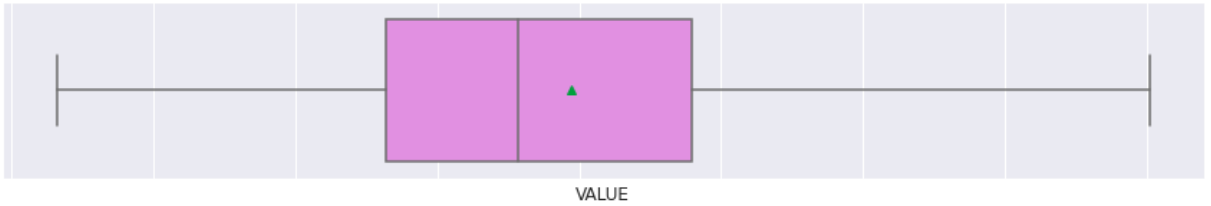
The larger the loan the lower number of derogatory records.

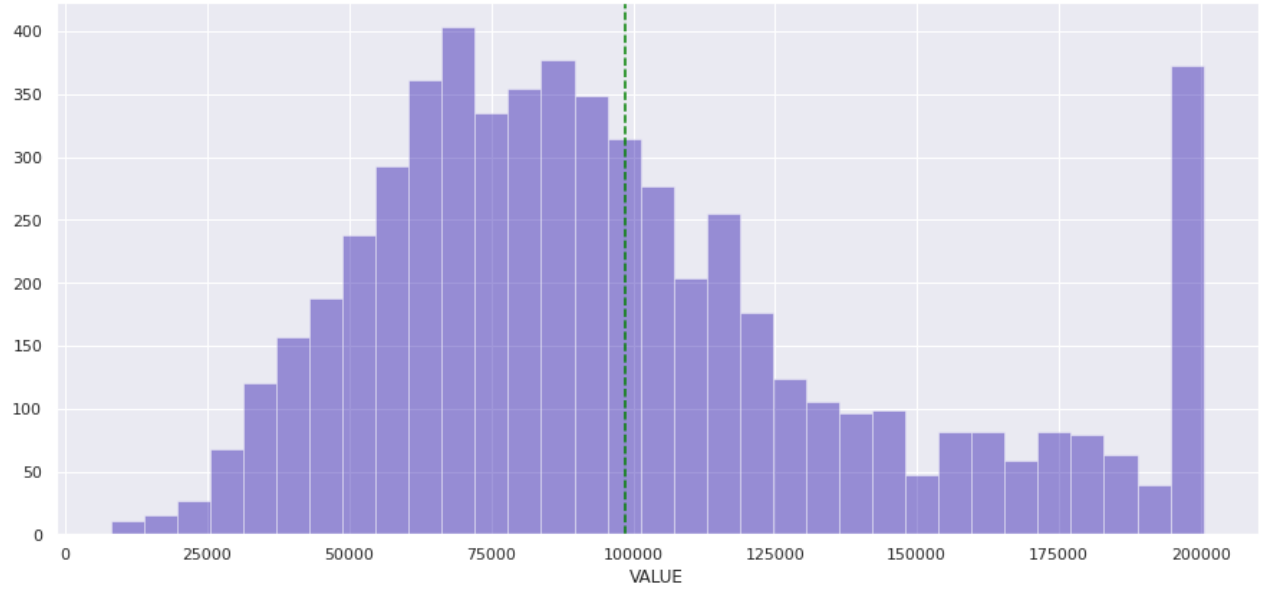


Data Cleaning:

Correcting outliers

Imputing missing data



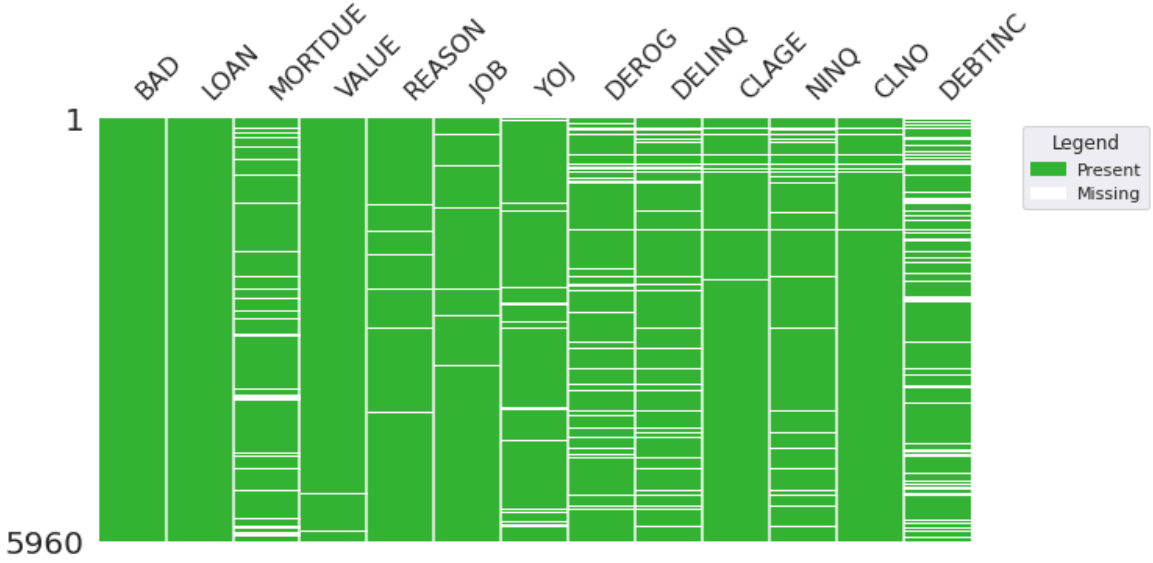


11 columns had missing values

# of missing values ranged from 112 to 1267

Numeric values replaced with the **median**

Categorical values replaced with the **mode**



All variables had outliers

Outliers < Q1 were replaced with Lower Whisker

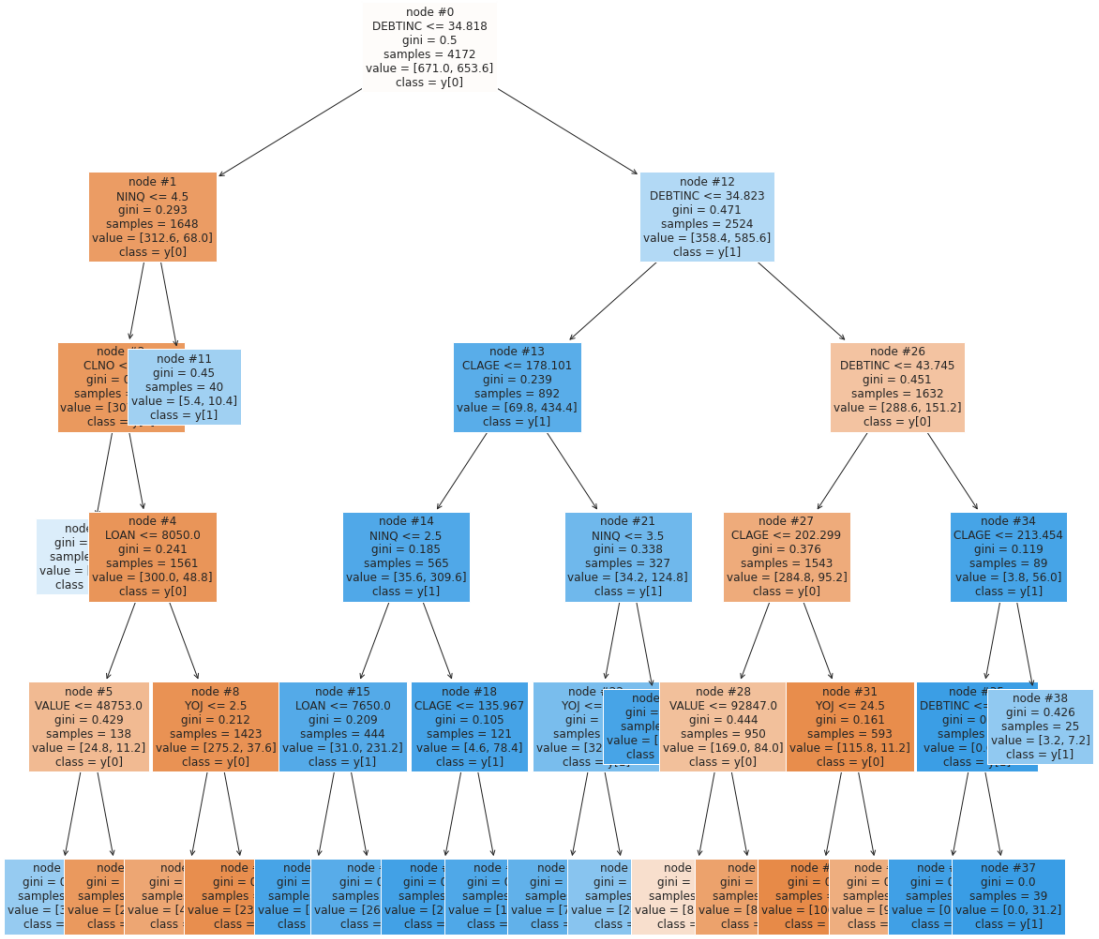
Outliers > Q3 were replaced with Upper Whisker

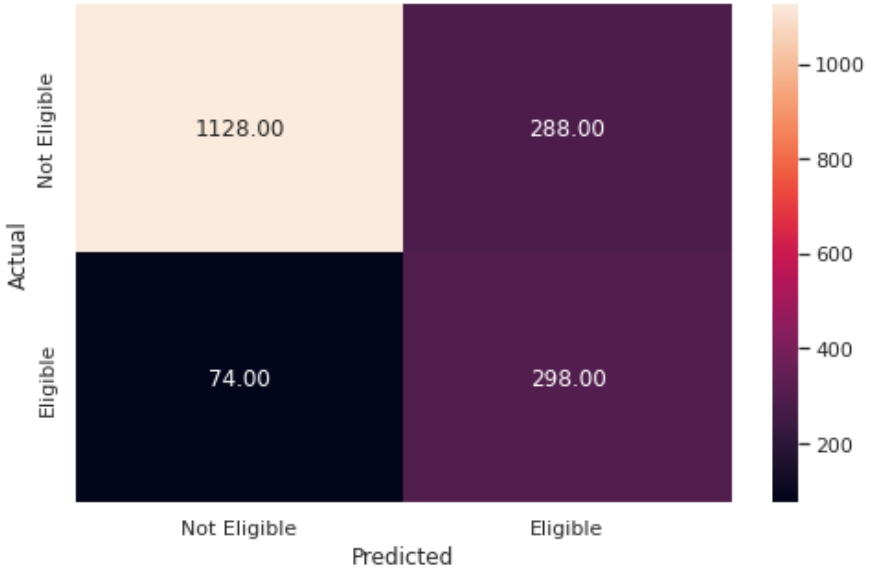
Created and tuned models

Decision Tree

Random Forest

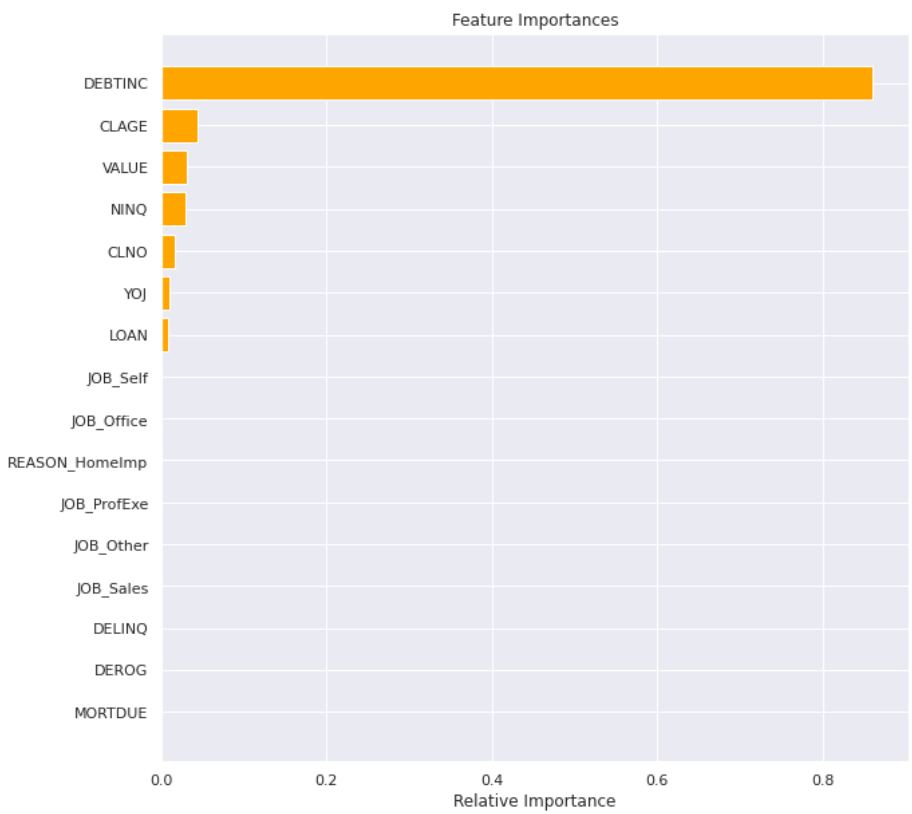
Model is attempting to **find** those **that will default** (1) on their loan, which will be our True Positive (TP), and therefore non-defaulters (0) will be our True Negative(TN).

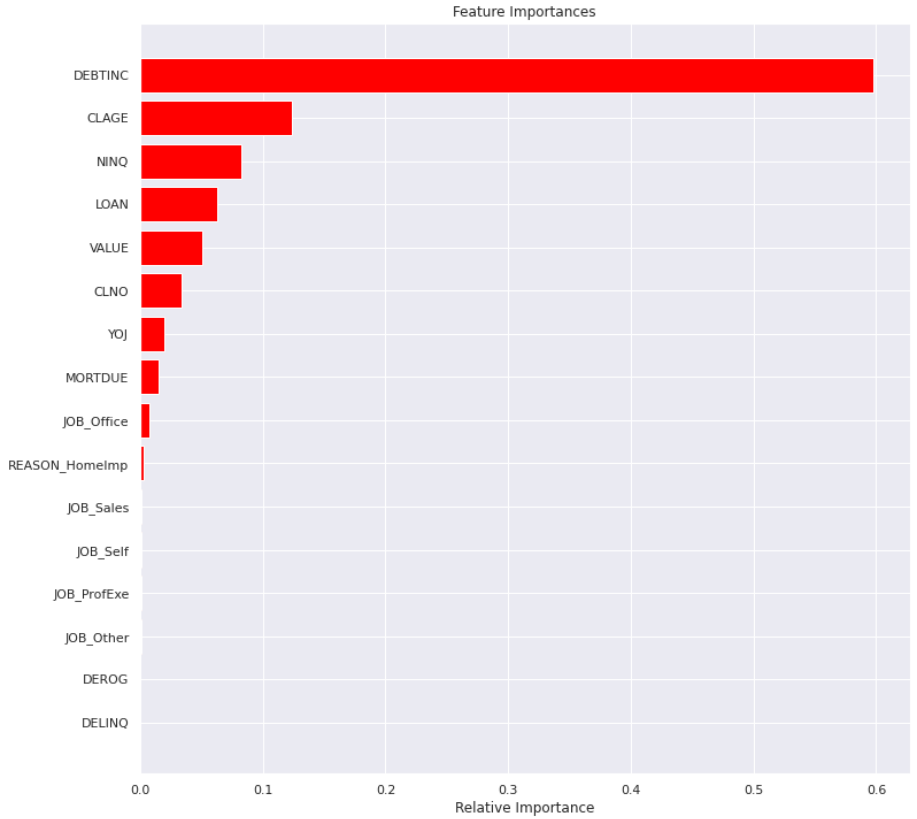




Model_Name	Train_f1	Train_recall	Test_f1	Test_recall	Test_precision
d_tree_base	100	100	60	56	64
d_tree_base	68	79	67	75	60
random_forest	100	100	69	59	85
random_forest	100	100	68	55	87
random_forest	62	83	62	80	51

Variables Indicating Default - Models





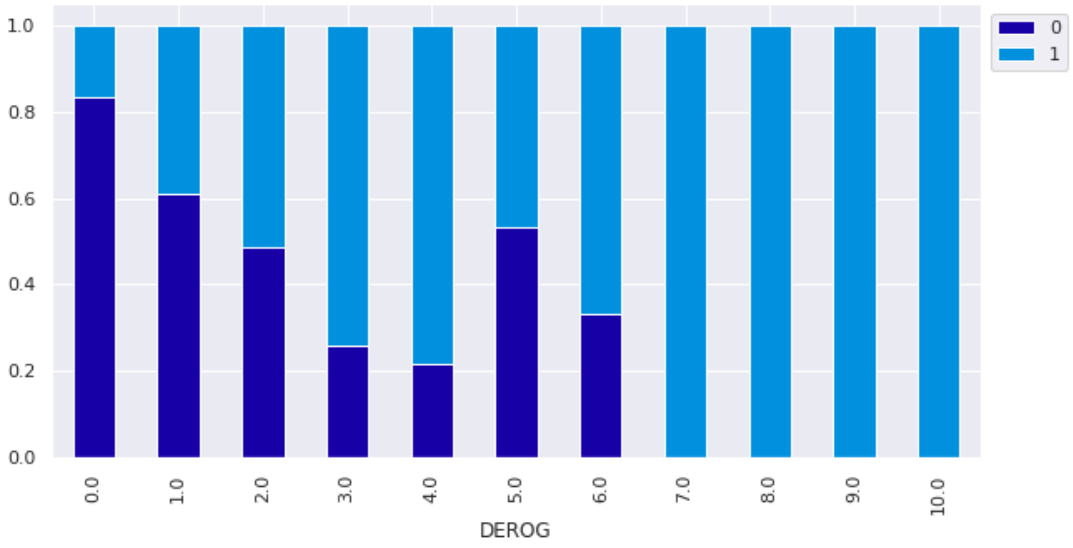
Debt to Income (DEBTINC) ratio was the number one indicator

Credit Line Age (CLAGE) was the second indicator

Variables indicating Default – from EDA

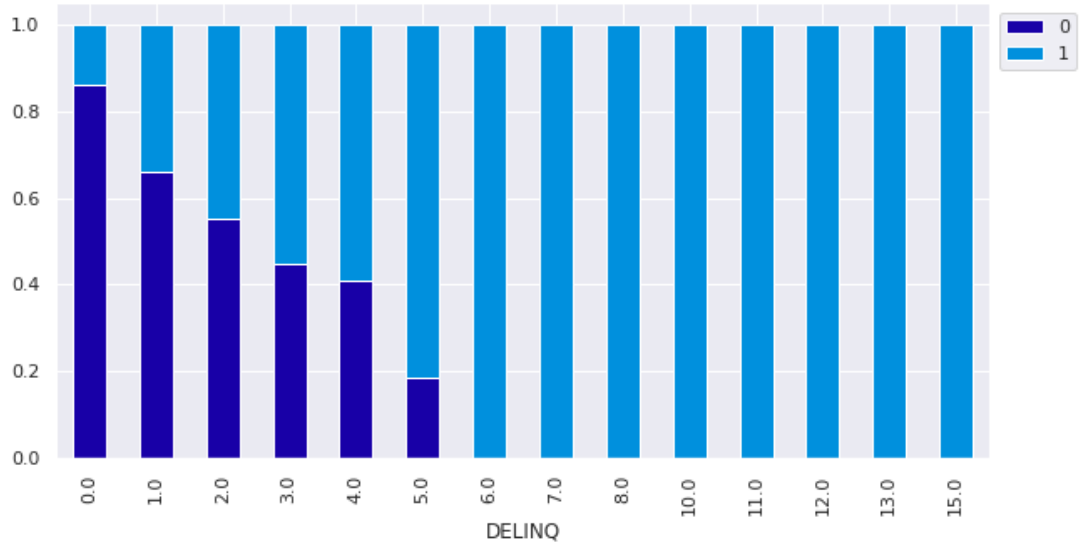
Number of Derogatory Credit Reports

DEROG >= 7 All default



Number of Delinquent Credit Reports

DELINQ >= 6 All default



Results – Decision Tree Model

Results

Tuned Decision Tree:

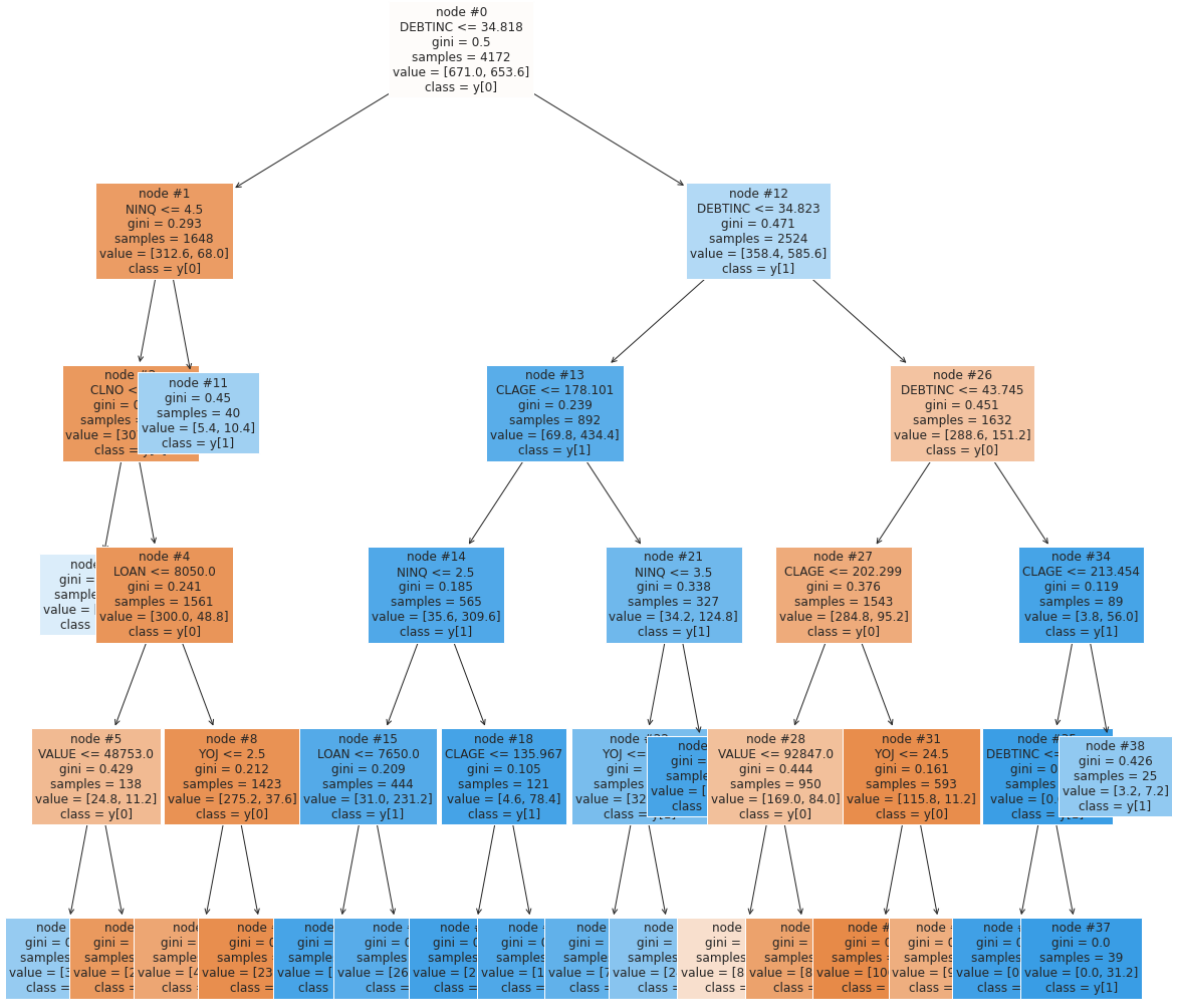
Blue (1) – Defaulter, Orange (0) – Non-defaulter

- First split is made on DEBTINC (Debt-to-income ratio) - high ability to predict defaulter (Higher DEBTINC indicates the applicant is more likely to default)

This is consistent with our observations in EDA

Next high priority splits are

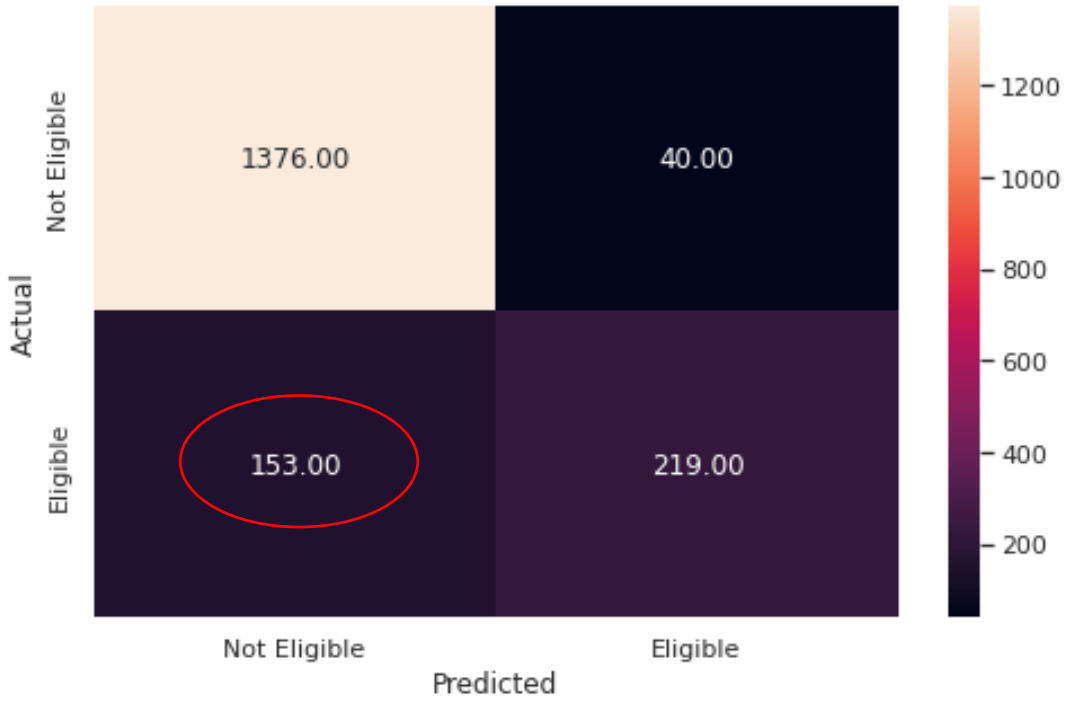
- NINQ(Number of recent credit inquiries) - Where a higher number is more likely to default
- CLAGE (Age of the oldest credit line in months) - Where a lower credit age is more likely to defualt



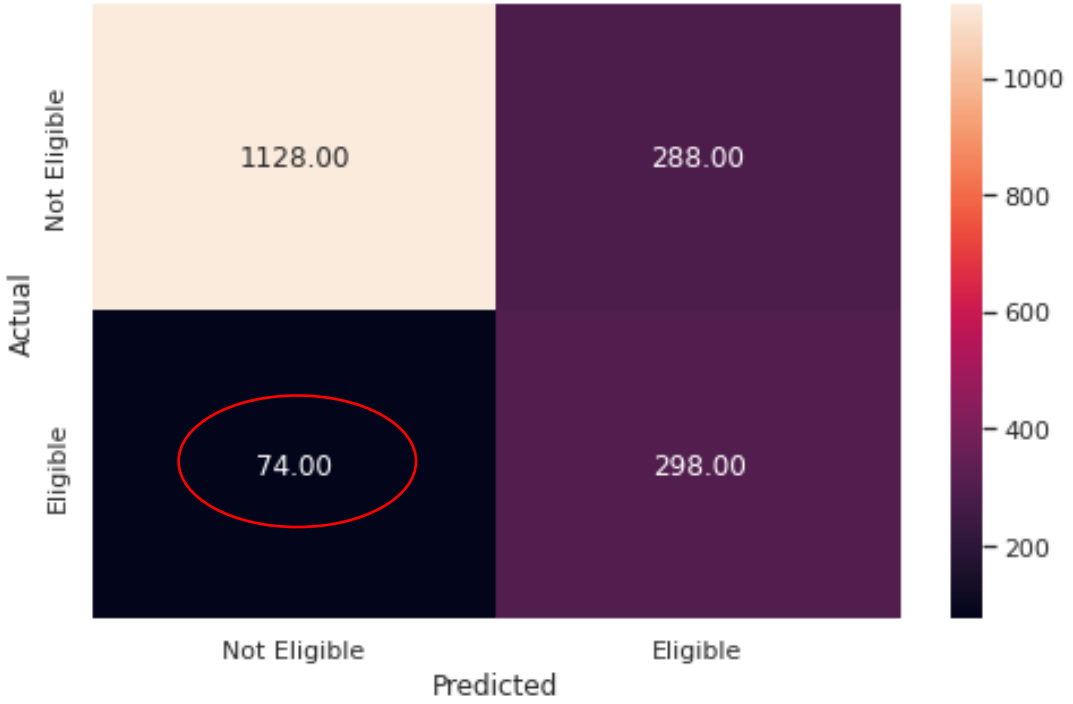
Results - Hyper Random Forest

Goal: Reduce the False Negatives

Predict a person will not default and they default



Reduced the number of False Negatives



Model Comparison – Goal is to **maximize** the **recall**

Tuned Decision Tree Model

- Accuracy of 85%,
- an F1-score of 67%,
- a recall of 75%

Hyper Tuned Random Forest Model

- Accuracy of 80%,
- an F1-score of 62%,
- a recall of 80%

Problem Statement

Using a classification model, the bank should be able to predict which features or combination of features of a customer would likely default on their loan.

Recommendations

80%

Accuracy

- Given our focus on optimizing recall while maintaining a reasonable level of accuracy, the **Hyper Tuned Random Forest Model** emerges as the preferred choice. Its higher recall rate ensures that we minimize the chances of missing positive cases, aligning well with our objective. The slight decrease in F1-score from 67% to 62% is an acceptable trade-off in light of achieving a more robust recall rate.
- Considering the importance of capturing true positive cases, we can confidently recommend the **Hyper Tuned Random Forest Model** as the more suitable option for our scenario.

Recommendations

- **Debt to income ratio is a very powerful characterisitic in predicting defaulters.**

The bank can use debt to income ratio as an initial indicator when evaluating a loan.

- **Business/Customer Focus**

Make customers with a high dept to income ratio aware of potential difficulties of paying off a loan.

Offer financial counseling on how to lower their debt-to-income ratio to qualify for future loans.

Recommendations

- **Credit Age is another characteristic in predicting defaulters.**

The amount of time someone has had credit the better the bank can gauge how well they will repay their loans.

- **Business/Customer Focus**

Make customers aware of how their credit age is a benefit for them.

Offer secure credit cards or small loans to assist in building credit.

Recommendations

- **Number of Derogatory and Delinquent Credit Reports is another tool in predicting defaulters.**

1. Persons with greater than or equal to 7 Derogatory reports ALL defaulted on their loans.
2. Persons with greater than or equal to 6 Delinquent credit reports ALL defaulted on their loans.

- **Business/Customer Focus**

Make customers aware of how derogatory and delinquent credit reports affect their ability to secure a loan.

Offer financial counseling on how to prevent having these reports in the future and how to improve their current credit reports

Overall - For Future Modeling

More data should be used to create a stronger model.

More care should be taken when gathering data.

There were errors/ missing data that could have assisted with improving the overall model.

DEBTINC had the most missing values and had some possible inaccuracies in the data.

This feature was found to be the MOST important feature for predicting if a person would default.

Appendices – Imbalance /Class Distribution

```
#check for imbalance
bad_counts = trainingData['BAD'].value_counts()

print(bad_counts)
```

```
BAD
0    3339
1     832
Name: count, dtype: int64
```

```
# check class distrubution
class_distribution = trainingData['BAD'].value_counts(normalize=True)

print(class_distribution)
```

```
BAD
0    0.800527
1    0.199473
Name: proportion, dtype: float64
```

Class Distribution

- There is a class bias
- To address class imbalance, we need to use techniques like oversampling the minority class, undersampling the majority class, or using synthetic data generation techniques (e.g., SMOTE).