Loan Status Analysis

Max Groves

Introduction

I used a set of Lending Club Data that included current loan status and information on the customers, and loans

The Metric for performance is the accuracy of predicting bad loans and the features most important to making those predictions

I used three different models:

1. KNN
2. Logistic Regression
3. Decision Tree

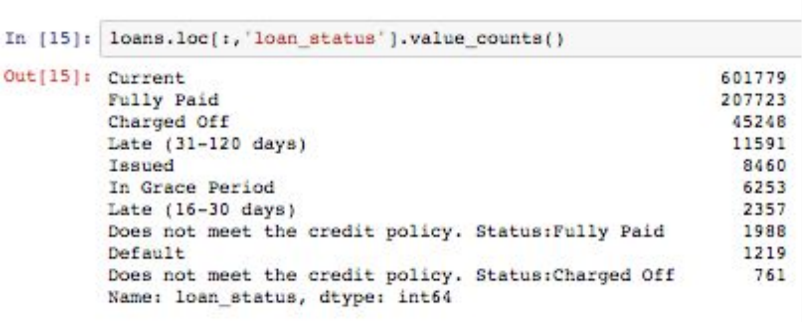
Picking the Correct Feature Columns

The original data set had over 70 columns. Some of these columns had information that we would not have at the origination of the loan

While going through the EDA I paired down the list to loan amount, interest rate, size of payment, dummy variables for housing status, term, annual income and number of open accounts because these are statistics you have upon originating the loan

The significance of having these variables and making a good model would be that we can maximize our collections efforts on identified loans and take preventative measures to maximize our ROI

Changing Loan Status



```
In [15]: loans.loc[:,'loan_status'].value_counts()

Out[15]: Current                                                601779
         Fully Paid                                             207723
         Charged Off                                             45248
         Late (31-120 days)                                      11591
         Issued                                                   8460
         In Grace Period                                          6253
         Late (16-30 days)                                        2357
         Does not meet the credit policy. Status:Fully Paid       1988
         Default                                                  1219
         Does not meet the credit policy. Status:Charged Off       761
         Name: loan_status, dtype: int64
```

I wanted to simplify the target variable so I took the loan statuses on the left and mapped them to a Bad Status indicator

Charged Off, Late (both buckets) and Does not meet the credit policy. Status: Charged Off

Logistic Regression

With 2 possible end states I decided to use a logistic regression model as my first go around. I split my data into a training and a test set

The Accuracy of the model ended up being 93.06% - This initially looked pretty good BUT after looking at the actual predictions it seemed the model was just predicting Good loan status every time

The outcome of this model was the NULL model.

I expanded the model to try and fit on ALL the data set to see if there was some type of bias in test set selection but that model returned the same result!

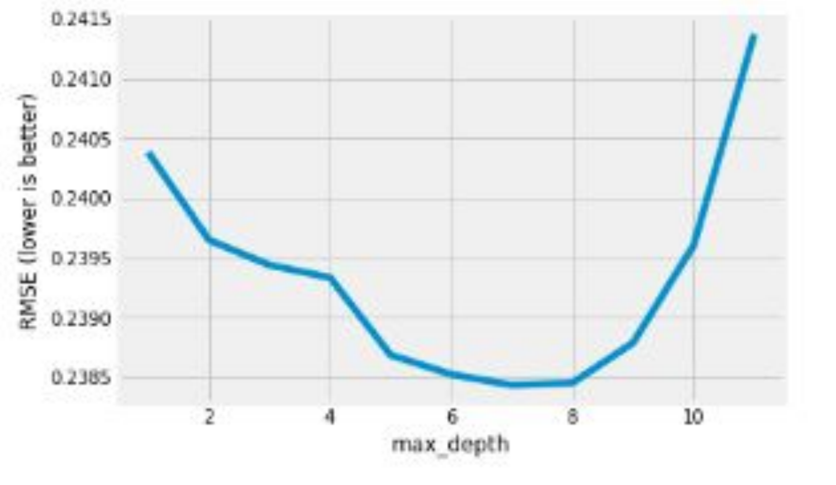
While the final model isn't great for end stat prediction it can still identify which loans have (relatively) high chance of having a bad status

Decision Tree Model

I used the same feature columns as I did for the Logistic Regression Model

In order to select a the most accurate model I built a loop and recorded the RMSE values of each depth (graph on the right)

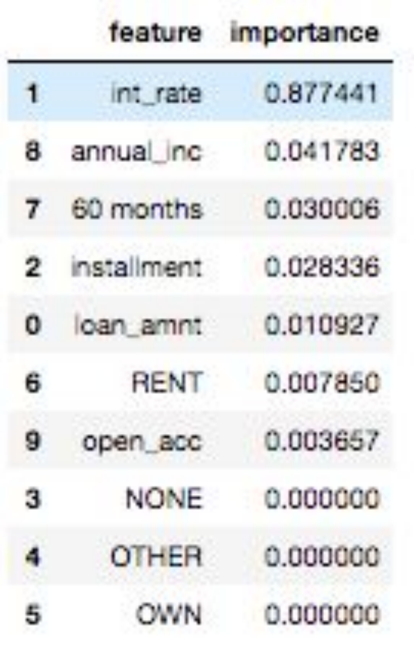
I found my best model at a Max depth of 7, and fitted a model to that



Decision Tree Results

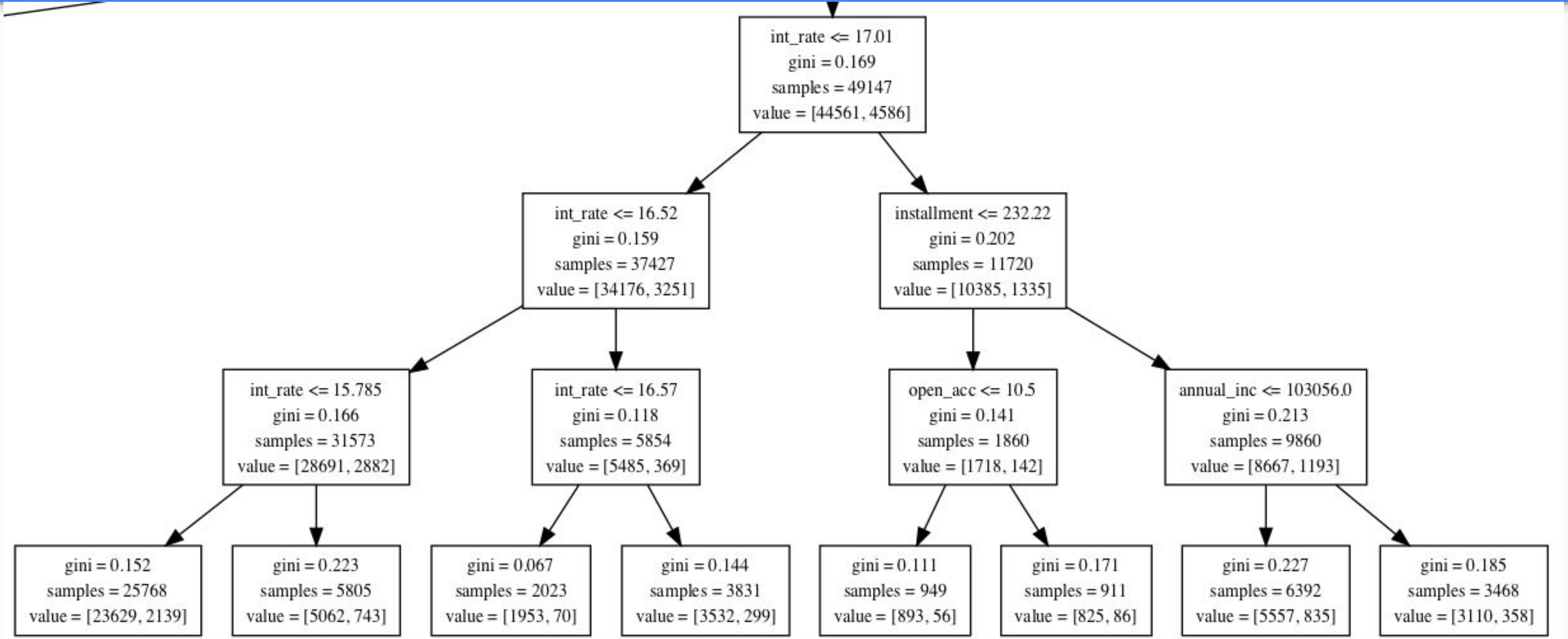
Once fitting my model I wanted to see which factors were most important so I grabbed the feature importances

My model output was that Interest Rate was the most important factor, with annual income 60 month term, installment (payment amount), loan amount, the Rent dummy and open accounts also having some importance. This mostly seems like a reasonable result



	feature	importance
1	int_rate	0.877441
8	annual_inc	0.041783
7	60 months	0.030006
2	installment	0.028336
0	loan_amnt	0.010927
6	RENT	0.007850
9	open_acc	0.003657
3	NONE	0.000000
4	OTHER	0.000000
5	OWN	0.000000

Decision Tree Picture (Just a portion)



int_rate <= 17.01
gini = 0.169
samples = 49147
value = [44561, 4586]

int_rate <= 16.52
gini = 0.159
samples = 37427
value = [34176, 3251]

installment <= 232.22
gini = 0.202
samples = 11720
value = [10385, 1335]

int_rate <= 15.785
gini = 0.166
samples = 31573
value = [28691, 2882]

int_rate <= 16.57
gini = 0.118
samples = 5854
value = [5485, 369]

open_acc <= 10.5
gini = 0.141
samples = 1860
value = [1718, 142]

annual_inc <= 103056.0
gini = 0.213
samples = 9860
value = [8667, 1193]

gini = 0.152
samples = 25768
value = [23629, 2139]

gini = 0.223
samples = 5805
value = [5062, 743]

gini = 0.067
samples = 2023
value = [1953, 70]

gini = 0.144
samples = 3831
value = [3532, 299]

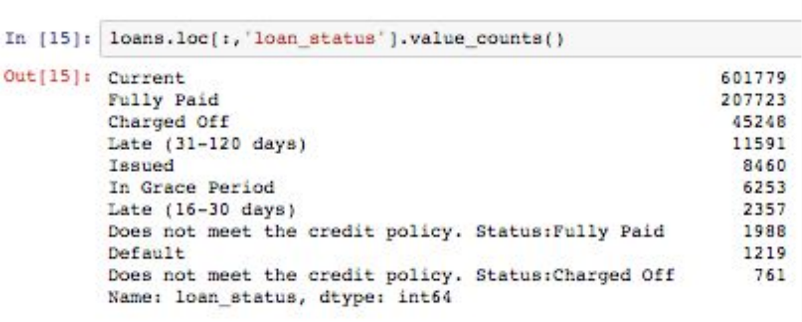
gini = 0.111
samples = 949
value = [893, 56]

gini = 0.171
samples = 911
value = [825, 86]

gini = 0.227
samples = 6392
value = [5557, 835]

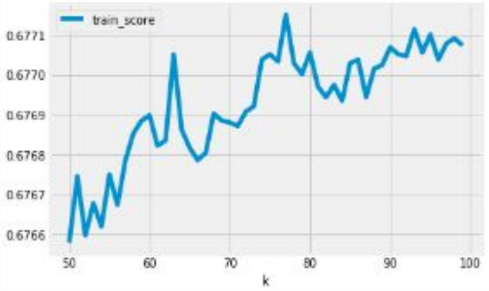
gini = 0.185
samples = 3468
value = [3110, 358]

Side Note: KNN



```
In [15]:  loans.loc[:,'loan_status'].value_counts()

Out[15]:  Current                                                601779
          Fully Paid                                             207723
          Charged Off                                             45248
          Late (31-120 days)                                      11591
          Issued                                                   8460
          In Grace Period                                          6253
          Late (16-30 days)                                        2357
          Does not meet the credit policy. Status:Fully Paid       1988
          Default                                                  1219
          Does not meet the credit policy. Status:Charged Off       761
          Name: loan_status, dtype: int64
```



I tried using a KNN model on the original loan status as KNN can also account for multiple end states (All the states are in the table on the left)

The models topped out at around 68% accuracy

Summary

After looking at the outcome of my 2 main models it seemed like the outcome of my models was best served in looking at which factors are most important in predicting bad loans.

Having Interest Rate be the most important factor makes sense as people who have higher interest rates are usually riskier customers, and will have to pay back more interest for similar purchases than others

Possible next steps for this analysis would have been looking at collections of the users with high interest rates, and possibly running a test to see if offering these users different terms would lead to less of them getting to bad status

Thanks!