Model Report

Detailed Analysis & Steps

Within this business problem, there are two relevant datasets that both play a role in addressing the problem. The first training dataset is based on previous records with the target variable of “loan\_status” already applied to each instance. This data will be used in the predictive modeling process. The other holdout dataset contains the same data but without the target variable in “loan\_status”. The purpose of this dataset is to use the predictions based on the first dataset and try to correctly predict whether each instance in the holdout data identifies as a defaulted loan. The breakdown of each dataset can be seen below:

| File Name | Record Count | Column Count | Numeric Columns | Character Columns |
| --- | --- | --- | --- | --- |
| loan\_train.csv | 29777 | 52 | 31 | 21 |
| loan\_holdout.csv | 12761 | 51 | 31 | 20 |

Field Summary

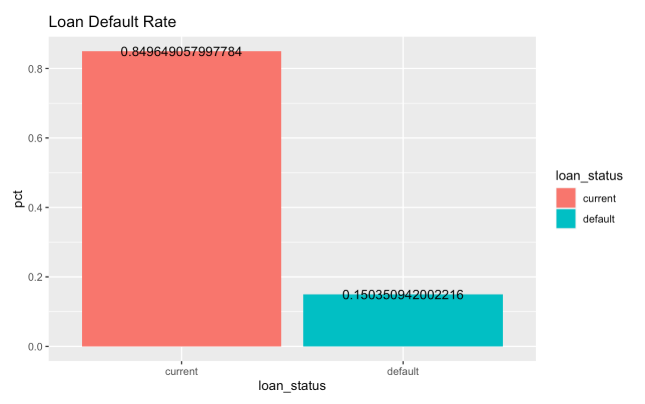
This table displays all the variables in the dataset, including factors such as the data type, number of instances, number of uniques, number of missing values, and the percentages of each. This tables helps us to get a better understanding of the framework of the dataset used to develop these predictive models.

| Variable Name | Data Type | Feature Type | Count | # Distinct | % Distinct | # Missing | % Missing |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | dbl | ID | 29774 | 29775 | 99.99% | 3 | 0.01% |
| member\_id | dbl | ID | 29774 | 29775 | 99.99% | 3 | 0.01% |
| loan\_amnt | dbl | Numeric | 29774 | 828 | 2.78% | 3 | 0.01% |
| funded\_amnt | dbl | Numeric | 29774 | 982 | 3.30% | 3 | 0.01% |
| funded\_amnt\_inv | dbl | Numeric | 29774 | 6863 | 23.05% | 3 | 0.01% |
| term | chr | Categorical | 29774 | 3 | 0.01% | 3 | 0.01% |
| int\_rate | chr | Percentage | 29774 | 391 | 1.31% | 3 | 0.01% |
| installment | dbl | Numeric | 29774 | 13256 | 44.52% | 3 | 0.01% |
| grade | chr | Categorical | 29774 | 8 | 0.03% | 3 | 0.01% |
| sub\_grade | chr | Categorical | 29774 | 36 | 0.12% | 3 | 0.01% |
| emp\_title | chr | Categorical | 27960 | 22144 | 74.37% | 1817 | 6.10% |
| emp\_length | chr | Categorical | 29774 | 13 | 0.04% | 3 | 0.01% |
| home\_ownership | chr | Categorical | 29774 | 6 | 0.02% | 3 | 0.01% |
| annual\_inc | dbl | Numeric | 29773 | 4287 | 14.40% | 4 | 0.01% |
| verification\_status | chr | Categorical | 29774 | 4 | 0.01% | 3 | 0.01% |
| issue\_d | chr | Categorical | 29774 | 56 | 0.19% | 3 | 0.01% |
| loan\_status | chr | target | 29777 | 2 | 0.01% | 0 | 0.00% |
| pymnt\_plan | chr | Categorical | 29774 | 3 | 0.01% | 3 | 0.01% |
| url | chr | Categorical | 29774 | 29775 | 99.99% | 3 | 0.01% |
| desc | chr | Categorical | 20345 | 20311 | 68.21% | 9432 | 31.68% |
| purpose | chr | Categorical | 29774 | 15 | 0.05% | 3 | 0.01% |
| title | chr | Categorical | 29764 | 15201 | 51.05% | 13 | 0.04% |
| zip\_code | chr | Categorical | 29774 | 820 | 2.75% | 3 | 0.01% |
| addr\_state | chr | Categorical | 29774 | 51 | 0.17% | 3 | 0.01% |
| dti | dbl | Numeric | 29774 | 2847 | 9.56% | 3 | 0.01% |
| delinq\_2yrs | dbl | Numeric | 29754 | 12 | 0.04% | 23 | 0.08% |
| earliest\_cr\_line | chr | Categorical | 29754 | 517 | 1.74% | 23 | 0.08% |
| fico\_range\_low | dbl | Numeric | 29774 | 44 | 0.15% | 3 | 0.01% |
| fico\_range\_high | dbl | Numeric | 29774 | 44 | 0.15% | 3 | 0.01% |
| inq\_last\_6mths | dbl | Numeric | 29754 | 28 | 0.09% | 23 | 0.08% |
| mths\_since\_last\_delinq | dbl | Numeric | 10870 | 91 | 0.31% | 18907 | 63.50% |
| mths\_since\_last\_record | dbl | Numeric | 2569 | 108 | 0.36% | 27208 | 91.37% |
| open\_acc | dbl | Numeric | 29754 | 45 | 0.15% | 23 | 0.08% |
| pub\_rec | dbl | Numeric | 29754 | 7 | 0.02% | 23 | 0.08% |
| revol\_bal | dbl | Numeric | 29774 | 18400 | 61.79% | 3 | 0.01% |
| revol\_util | chr | Categorical | 29710 | 1095 | 3.68% | 67 | 0.23% |
| total\_acc | dbl | Numeric | 29754 | 79 | 0.27% | 23 | 0.08% |
| out\_prncp | dbl | Numeric | 29774 | 384 | 1.29% | 3 | 0.01% |
| out\_prncp\_inv | dbl | Numeric | 29774 | 385 | 1.29% | 3 | 0.01% |
| total\_rec\_late\_fee | dbl | Numeric | 29774 | 1605 | 5.39% | 3 | 0.01% |
| last\_ptmnt\_d | chr | Categorical | 29710 | 107 | 0.36% | 67 | 0.23% |
| last\_pymnt\_amnt | dbl | Numeric | 29774 | 26904 | 90.35% | 3 | 0.01% |
| next\_pymnt\_d | chr | Timestamp | 2352 | 97 | 0.33% | 27425 | 92.10% |
| last\_credit\_pull\_d | chr | Timestamp | 29772 | 110 | 0.37% | 5 | 0.02% |
| collections\_12\_mths\_ex\_med | dbl | Binary | 29673 | 2 | 0.01% | 104 | 0.35% |
| policy\_code | dbl | Binary | 29774 | 2 | 0.01% | 3 | 0.01% |
| application\_type | chr | Binary | 29774 | 2 | 0.01% | 3 | 0.01% |
| acc\_now\_delinq | dbl | Numeric | 29754 | 3 | 0.01% | 23 | 0.08% |
| chargeoff\_within\_12\_mths | dbl | Binary | 29673 | 2 | 0.01% | 104 | 0.35% |
| delinq\_amnt | dbl | Numeric | 29754 | 4 | 0.01% | 23 | 0.08% |
| pub\_rec\_bankruptcies | dbl | Numeric | 28811 | 4 | 0.01% | 966 | 3.24% |
| tax\_lines | dbl | Numeric | 29698 | 3 | 0.01% | 79 | 0.27% |

Target Variable Summary

The target variable in this case is “loan\_default” and is a binary variable that returns either “current” or “default” which means the loan was defaulted. The breakdown of this variable in the training dataset is as follows with the count of each instance and a visual representation in the form of a bar graph:

| loan\_status | Count | PCT |
| --- | --- | --- |
| current | 23500 | 84.96% |
| default | 4477 | 15.04% |

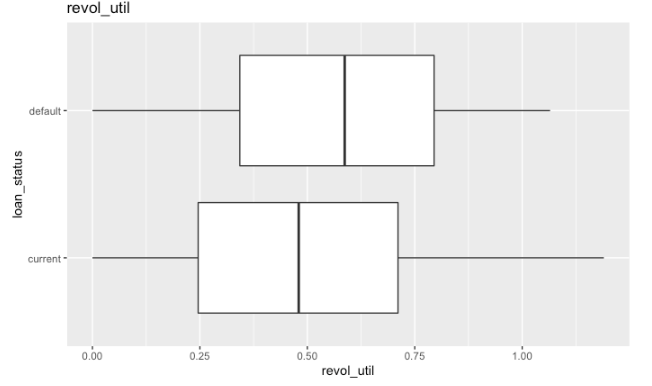
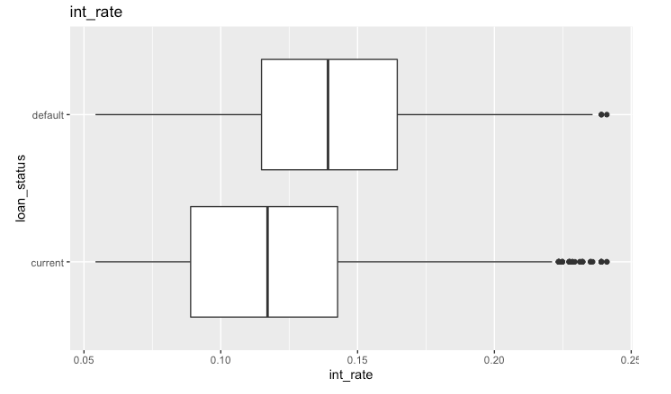


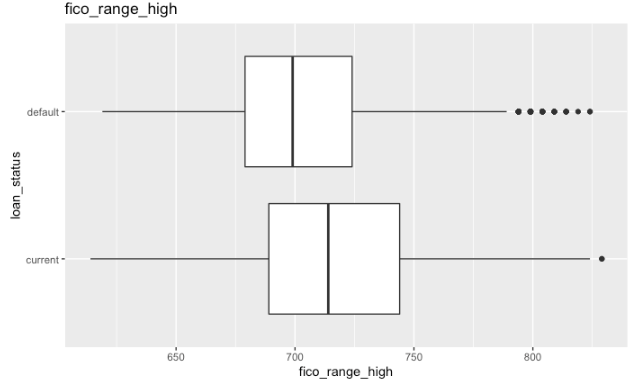
Numeric Variable Analysis

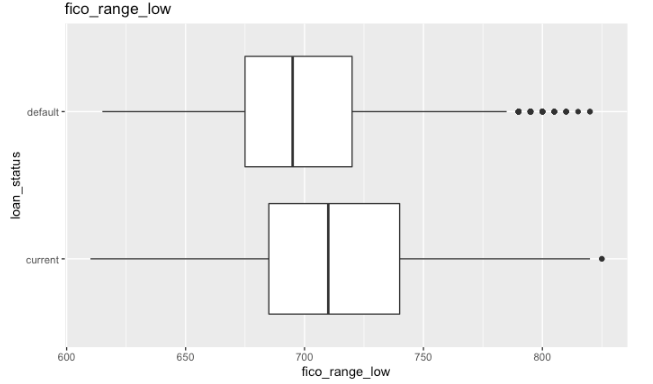
This dataset contains both numeric and character variables and it is important to look at each type to determine which can be considered significant to correctly predicting whether a loan will default. First we can take a look at the numeric variables in the dataset with the following table and some summary statistics for each:

| Variable | Count | Distinct | # Missing | Mean | Min | Max |
| --- | --- | --- | --- | --- | --- | --- |
| loan\_amnt | 29774 | 828 | 3 | 11109.00 | 500.00 | 35000.00 |
| funded\_amnt | 29774 | 982 | 3 | 10844.00 | 500.00 | 35000.00 |
| funded\_amnt\_inv | 29774 | 6863 | 3 | 10150.00 | 0.00 | 35000.00 |
| installment | 29774 | 13256 | 3 | 323.81 | 15.67 | 1305.19 |
| annual\_inc | 29773 | 4287 | 4 | 69201.00 | 2000.00 | 6000000.00 |
| dti | 29774 | 2847 | 3 | 13.38 | 0.00 | 29.99 |
| delinq\_2yrs | 29754 | 12 | 23 | 0.16 | 0.00 | 13.00 |
| fico\_range\_low | 29774 | 44 | 3 | 713.10 | 610.00 | 825.00 |
| fico\_range\_high | 29774 | 44 | 3 | 717.10 | 614.00 | 829.00 |
| inq\_last\_6mths | 29754 | 28 | 23 | 1.08 | 0.00 | 33.00 |
| mths\_since\_last\_delinq | 10870 | 91 | 18907 | 34.72 | 0.00 | 120.00 |
| mths\_since\_last\_record | 2569 | 108 | 27208 | 59.23 | 0.00 | 129.00 |
| open\_acc | 29754 | 45 | 23 | 9.34 | 1.00 | 47.00 |
| pub\_rec | 29754 | 7 | 23 | 0.06 | 0.00 | 5.00 |
| revol\_bal | 29774 | 18400 | 3 | 14310.00 | 0.00 | 1207359.00 |
| total\_acc | 29754 | 1095 | 23 | 22.08 | 1.00 | 81.00 |
| out\_prncp | 29774 | 384 | 3 | 11.80 | 0.00 | 3126.60 |
| out\_prncp\_inv | 29774 | 385 | 3 | 11.76 | 0.00 | 3123.44 |
| total\_rec\_late\_fee | 29774 | 1605 | 3 | 1.51 | 0.00 | 180.20 |
| last\_pymt\_amnt | 29774 | 107 | 3 | 2615.00 | 0.00 | 36115.00 |
| delinq\_amnt | 29754 | 4 | 23 | 0.20 | 0.00 | 6053.00 |
| int\_rate | 29774 | 391 | 3 | 0.12 | 0.05 | 0.24 |
| revol\_util | 29710 | 1095 | 67 | 0.49 | 0.00 | 1.19 |

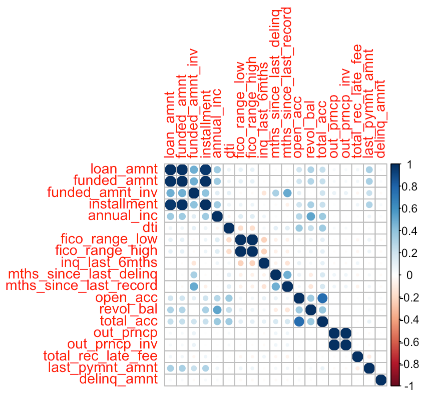
After looking through the numeric variables in the dataset, comparison can be done on each of the variables based on the target variable. Boxplots can be constructed to display separability between each variable and the target. The larger discretion between the target, the more significant the variable may be in a predictive model. Here are a few example of variables that display this instance and may be important to include in the modeling process:





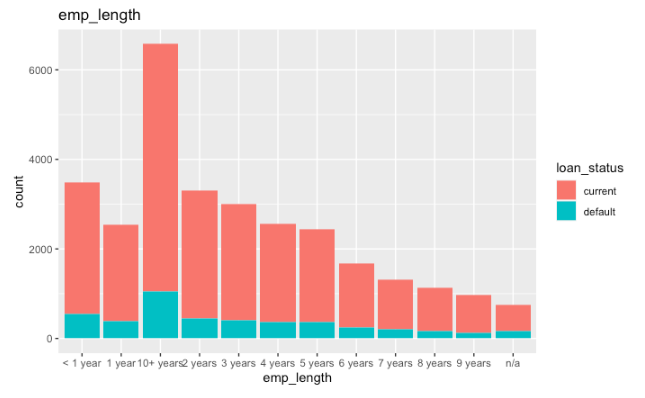


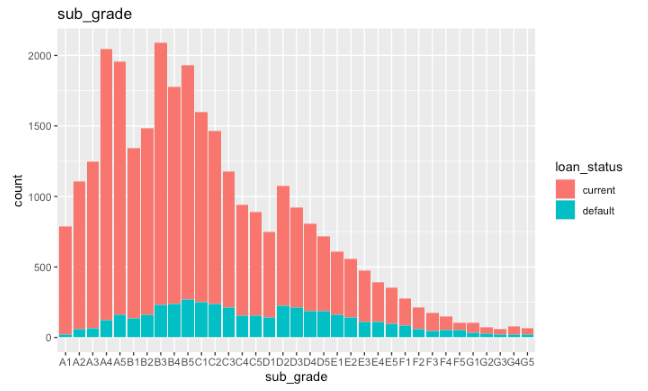
These are a few of the variables that might render significant to the model. With the numeric variables, a correlation matrix can also be constructed to determine correlation among the numeric variables. As seen below, the correlation matrix displays the significance between each variable. For example, “installment” and “loan\_amnt” have a strong correlation, while “inq\_last\_6mths” and “fico\_range\_low” have a negative correlation. This is insightful information that can be used as well when determining which variables to include in a predictive model.

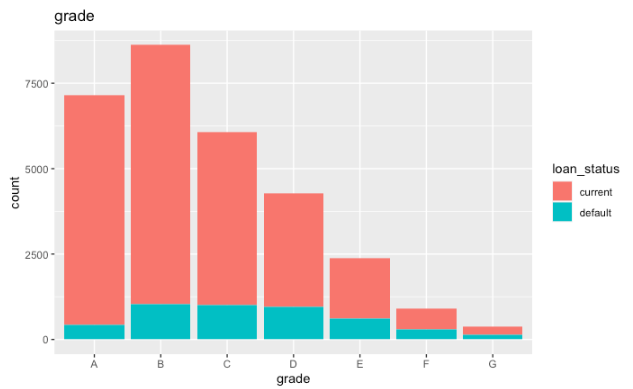


Character Variable Analysis

The other side of the data are the character and binary variables, these variables can be of use as well to the predictive modeling process, and can be analyzed in a similar way as the numerics relative to their correlation to the target variable of “loan\_status”. Bar charts displaying the count of each variable and its components as well as the relation to the count of the target within each component can be constructed to analyze any tendencies among the character variables. A few examples are listed as follows that might give some insight into how each are important in the predictive modeling process:







Data Transformations

As seen in the table identifying each variable and the number of missing values, each variable has a different number of missing values and these need to be addressed in some manner. The first option would be to eliminate the variable entirely if the completion rate for the variable is above 5%. In this case in the holdout dataset, the variables that can be eliminated from the start due to the significant number of missing values includes:

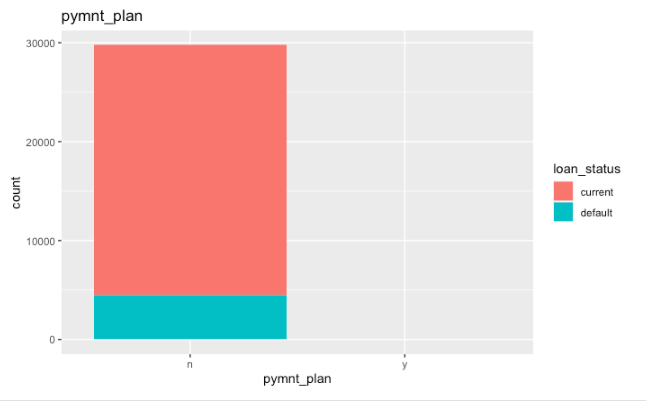
* emp\_title
* desc
* mths\_since\_last\_delinq
* mths\_since\_last\_record
* next\_pymnt\_d

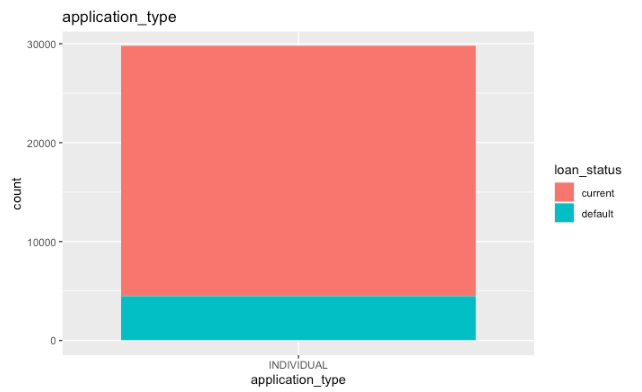
These variables are necessary to remove because imputing them would not be an accurate representation of them and could possibly be misleading if they were to be used for analysis. In the purpose of this project, these variables are significant, but could possibly be for other purposes. Also necessary in the data preparation process is dropping variables that don’t render useful in a predictive modeling process. These could include unique id’s, geographical data, or character variables with extremely high variability and uniqueness. In this case, these variables should be removed from the beginning as well and identify as:

* id
* member\_id
* url
* zip\_code
* title
* emp\_title
* addr\_state

A final step that can be done after doing some additional exploratory analysis is dropping variables that don't have any separability in the entire dataset. This means that the variable has the same value for every instance in the dataset, and if included, will not be able to be used in a predictive model. In this dataset, these variables are as follows and can be verified by these bar charts:

* pymnt\_plan
* application\_type





With the variables now leff in the dataset, it is still important to replace the missing values. When constructing the recipes for the predictive models, there are several functions that can replace missing values for both numeric, character, and binary variables. These are as follows:

* “step\_impute\_median()” for numeric variables
* “step\_dummy()” for nominal variables
* “step\_impute \_mode()” for nominal predictors
* “step\_normalize()” to normalize the numeric variables
* “step\_novel()” for nominal predictors”

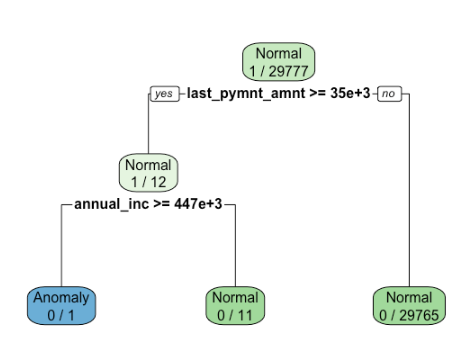
Train/Test Split

A training and testing split is also done on the dataset for predictive modeling purposes so that the results of the training data can be compared to something, hence the test data. This is a critical step of every predictive modeling process and such is the case here.

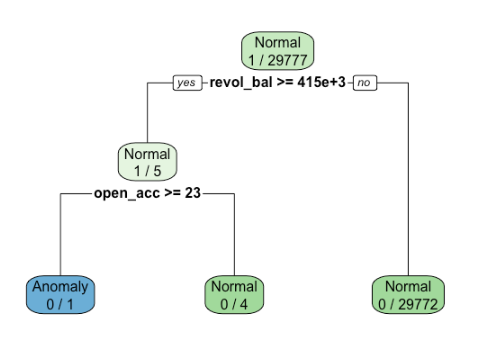
Anomaly Detection

The process of anomaly detection can be a useful resource and analytical method as well in this instance of predicting loan status. An anomaly is something that strays from what is the norm or an expected instance. The goal here would be to identify instances that don’t conform to this normal status and profile as anomalous. For this situation, I will identify some anomalous records in the dataset. The top in this dataset are as follows defined by their rules followed by the decision trees that are the basis for each rule.

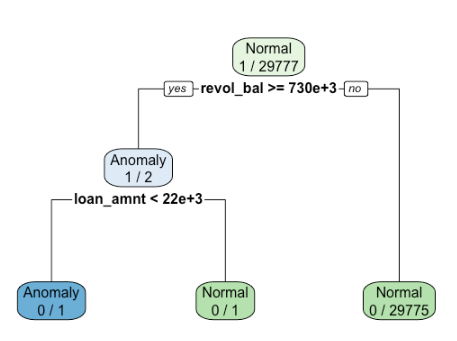
| Top 5 Anomalous Records | | |
| --- | --- | --- |
|  | | |
| Rule # | Rule | Cover |
| 1 | IF last\_pymnt\_amnt >= 34537 & annual\_inc >= 446500 | 0% |
| 2 | IF revol\_bal >= 415492 & open\_acc >= 23 | 0% |
| 3 | IF revol\_bal >= 730487 & loan\_amnt < 21875 | 0% |
| 4 | IF annual\_inc >= 683840 & mths\_since\_last\_delinq < 2.5 | 0% |
| 5 | IF annual\_inc is 850000 to 880000 | 0% |



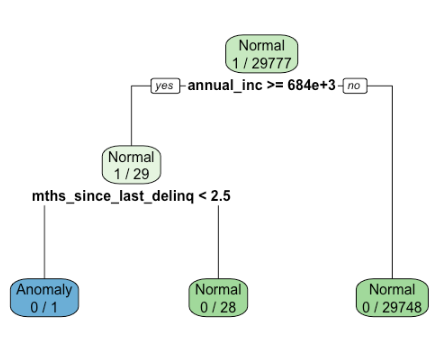
# 1 - Telling by the first rule and the decision tree above, this instance is true if the last payment amount is greater than or equal to 34,537 and annual income is greater than 446,500.



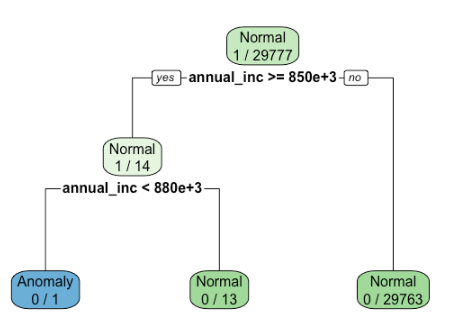
# 2 - Telling by the second rule and the decision tree above, this instance is true if the revolving line utilization rate is greater than 415,492 and the number of open credit lines is greater than or equal to 23.



# 3 - Telling by the third rule and the decision tree above, this instance is true if the revolving line utilization rate is greater than 730,487 and the loan amount is less than 21,875.



# 4 - Telling by the fourth rule and the decision tree above, this instance is true if self reported annual income is greater than 683,840 and the number of months since the borrower’s last delinquency is less than 2.5.



$ 5 - Telling by the fifth rule and the decision tree above, this instance is true if annual income is between 850,000 to 880,000.

Understanding these anomalies offers more insight into the variable relation analysis and how each might be useful in different situations. Now that all of the exploratory analysis is complete, the predictive models can start to be constructed, starting with the logistic regression model.

Model Building

Logistic Regression Model

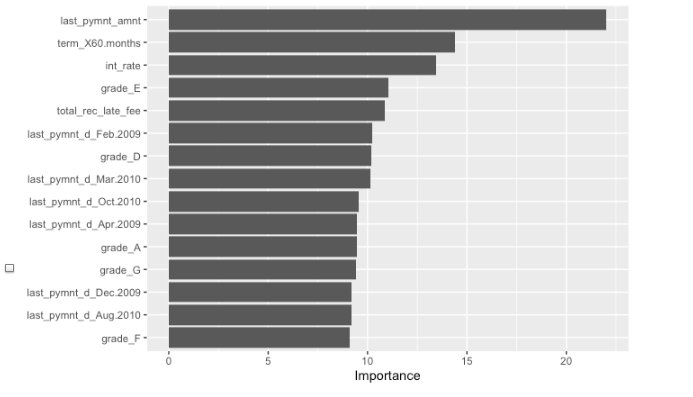
To begin the model building process, a logistic regression model is a great place to start, as it is one of the more simple classification models and is easy to comprehend. The first step is identifying which variables are still included in the dataset and to run a full baseline model with all of these variables. A subset of the variable chart from earlier is presented below. The date variables have also been removed for this recipe:

| Logistic Regression Recipe | | | |
| --- | --- | --- | --- |
| Variable | Status | Variable | Status |
| id | Excluded | fico\_range\_low | Included |
| member\_id | Excluded | fico\_range\_high | Excluded |
| loan\_amnt | Included | inq\_last\_6mths | Included |
| funded\_amnt | Excluded | mths\_since\_last\_delinq | Excluded |
| funded\_amnt\_inv | Included | mths\_since\_last\_record | Excluded |
| term | Included | open\_acc | Excluded |
| int\_rate | Included | pub\_rec | Excluded |
| installment | Excluded | revol\_bal | Excluded |
| grade | Excluded | revol\_util | Included |
| sub\_grade | Included | total\_acc | Excluded |
| emp\_title | Excluded | out\_prncp | Included |
| emp\_length | Included | out\_prncp\_inv | Included |
| home\_ownership | Excluded | total\_rec\_late\_fee | Included |
| annual\_inc | Included | last\_pymnt\_d | Included |
| verification\_status | Included | last\_pymnt\_amnt | Included |
| issue\_d | Excluded | next\_pymnt\_d | Excluded |
| loan\_status | Target | last\_credit\_pull\_d | Included |
| pymnt\_plan | Excluded | collections\_12\_mths\_ex\_med | Excluded |
| url | Excluded | policy\_code | Excluded |
| desc | Excluded | application\_type | Excluded |
| purpose | Included | acc\_now\_delinq | Excluded |
| title | Excluded | chargeoff\_within\_12\_mths | Excluded |
| zip\_code | Excluded | delinq\_amnt | Excluded |
| addr\_state | Excluded | pub\_rec\_bankruptcies | Excluded |
| dti | Excluded | tax\_lines | Included |
| delinq\_2yrs | Included | int\_rate | Included |
| earliest\_cr\_line | Excluded | revol\_util | Included |

Logistic Model Steps

1. The first step is to construct the recipe for the model, and in this case, will be the full model with the variables included as above in the chart.
2. Next will be to impute missing values as mentioned earlier in the report.
3. After missing values are imputed, the workflow can be defined and the model can be fitted to it.
4. In a logistic model, intercepts of each variable will be defined along with their p-value, determining how significant each variable is to the model.
5. The final step would be to evaluate the metrics and try to better the model by removing or adding variables and setting hyperparameters

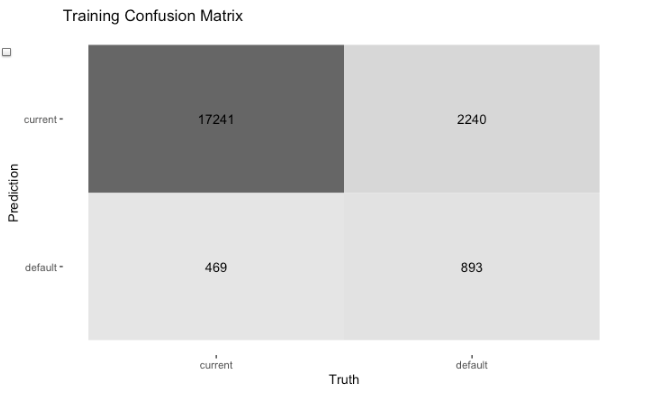
Looking at the most important variables to this model as well as the p-values to determine significance, we can better understand how to try and build a better logistic model:

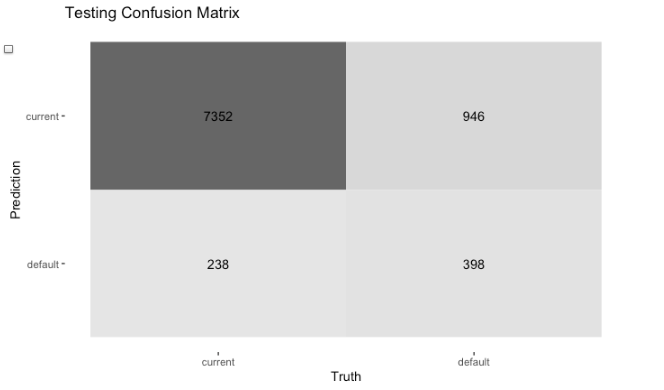


Full Logistic Model Metrics

To judge the model's performance, we can look at the metrics to determine how it performs. A table including various metrics and the confusion matrices on both the training and testing datasets used to calculate the metrics.

| Logistic Full Model Metrics | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Partition | Accuracy | ROC AUC | LogLoss | Precision | Recall | F1 |
| Training | 0.9524061 | 0.9805575 | 0.1244199 | 0.8644876 | 0.8104054 | 0.8365733 |
| Testing | 0.9452653 | 0.9703724 | 0.1609859 | 0.8363493 | 0.7909226 | 0.8130019 |





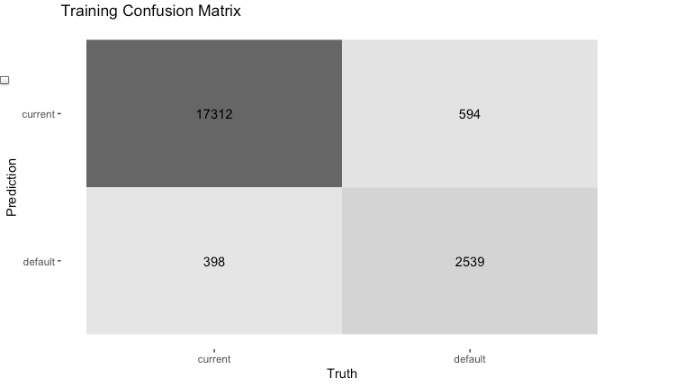
Reduced Logistic Model Metrics

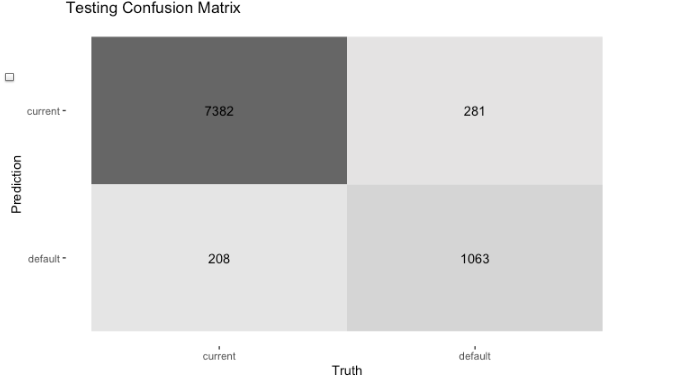
Once the metrics are calculated on the full model, the insignificant variables can be removed to try and make the model perform better and easier to comprehend in a business sense. In this case, the variables that render insignificant due to a high p-value are as follows, and can be removed from the model:

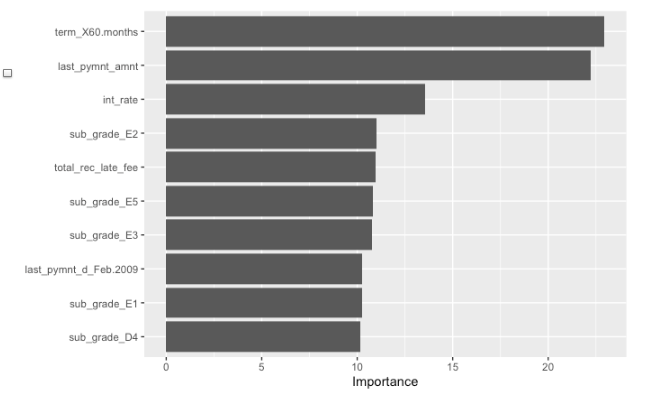
* funded\_amnt
* pub\_rec\_bankruptcies
* pub\_rec
* revol\_bal
* acc\_now\_delinq
* collections\_12\_mths\_ex\_med
* chargeoff\_within\_12\_mths
* installment
* policy\_code
* fico\_range\_high
* open\_acc
* total\_acc
* dti
* grade
* home\_ownership

Removing these variables yields the following metrics and confusion matrices:

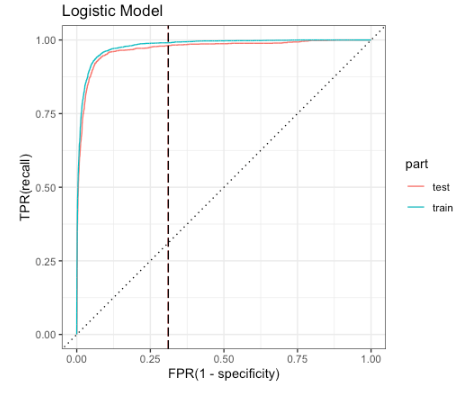
| Logistic Reduced Model Metrics | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Partition | Accuracy | ROC AUC | LogLoss | Precision | Recall | F1 |
| Training | 0.9521182 | 0.9804152 | 0.1250168 | 0.8644876 | 0.8104054 | 0.8365733 |
| Testing | 0.9449295 | 0.9702392 | 0.1615682 | 0.8363493 | 0.7909226 | 0.8130019 |







Based upon the 5% operating threshold, the ROC curve below displays the AUC for the operating range.



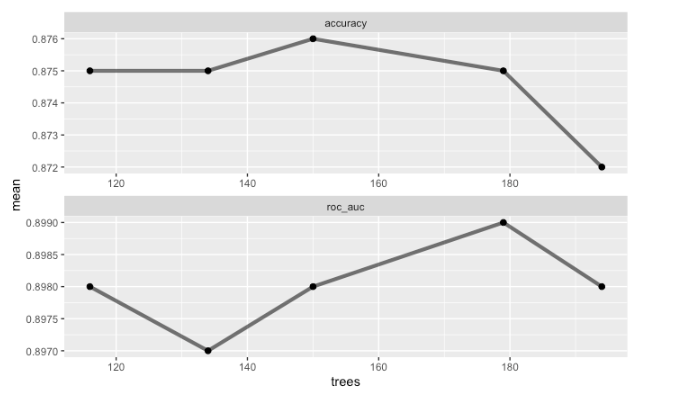
Random Forest Model

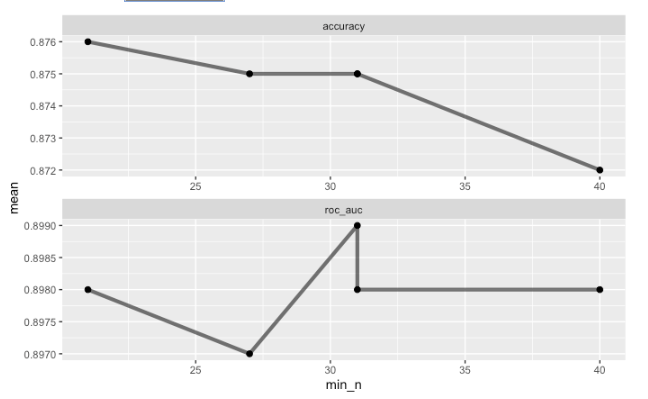
The next step in this modeling process will be to develop a more complex machine learning model, and in this case a random forest will be next. A random forest is a more complex model than a logistic regression and therefore a more accurate model might be created. To start the model, the same recipe was used as the logistic regression recipe. But because the model overfit, some variables were removed, and these were: “collections\_12\_mths\_ex\_med”, “chargeoff\_within\_12\_mths, installment”, “policy\_code”, “fico\_range\_high”, “open\_acc”, “total\_acc”, “dti”, “grade”, “home\_ownership”. The list of variables included and not included is shown below. The next step in the random forest modeling process is to identify and implement hyper parameters.

| Random Forest Recipe | | | |
| --- | --- | --- | --- |
| Variable | Status | Variable | Status |
| id | Excluded | fico\_range\_low | Included |
| member\_id | Excluded | fico\_range\_high | Included |
| loan\_amnt | Included | inq\_last\_6mths | Included |
| funded\_amnt | Excluded | mths\_since\_last\_delinq | Excluded |
| funded\_amnt\_inv | Included | mths\_since\_last\_record | Excluded |
| term | Included | open\_acc | Included |
| int\_rate | Included | pub\_rec | Excluded |
| installment | Included | revol\_bal | Excluded |
| grade | Included | revol\_util | Included |
| sub\_grade | Included | total\_acc | Included |
| emp\_title | Excluded | out\_prncp | Included |
| emp\_length | Included | out\_prncp\_inv | Included |
| home\_ownership | Included | total\_rec\_late\_fee | Included |
| annual\_inc | Included | last\_pymnt\_d | Included |
| verification\_status | Included | last\_pymnt\_amnt | Included |
| issue\_d | Excluded | next\_pymnt\_d | Excluded |
| loan\_status | Target | last\_credit\_pull\_d | Included |
| pymnt\_plan | Excluded | collections\_12\_mths\_ex\_med | Included |
| url | Excluded | policy\_code | Included |
| desc | Excluded | application\_type | Excluded |
| purpose | Included | acc\_now\_delinq | Excluded |
| title | Excluded | chargeoff\_within\_12\_mths | Included |
| zip\_code | Excluded | delinq\_amnt | Excluded |
| addr\_state | Excluded | pub\_rec\_bankruptcies | Excluded |
| dti | Included | tax\_lines | Included |
| delinq\_2yrs | Included | int\_rate | Included |
| earliest\_cr\_line | Excluded | revol\_util | Included |

Hyper Parameters/K-Fold Cross Validation

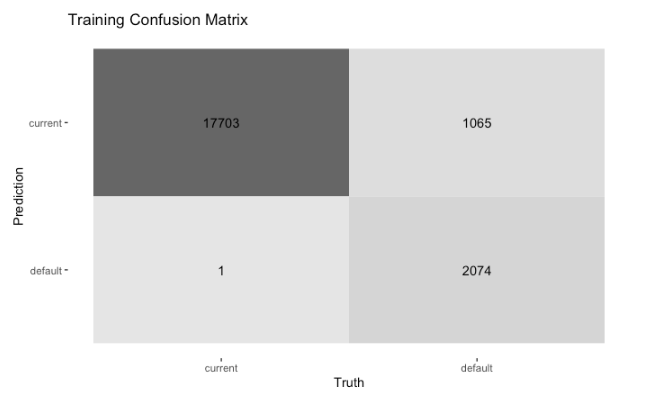
In a random forest model it is important to set hyper parameters that maximize the metrics of the model as well as use k-fold cross validation. In this case, the data was split into 5 folds. For the hyper parameters, the two for a random forest are the number of trees and min\_n is the number of observations. After performing the tuning on the recipe, the best parameters are outlined below on the roc\_auc, with the best number of trees equalling 150 and the min\_n equalling 21.

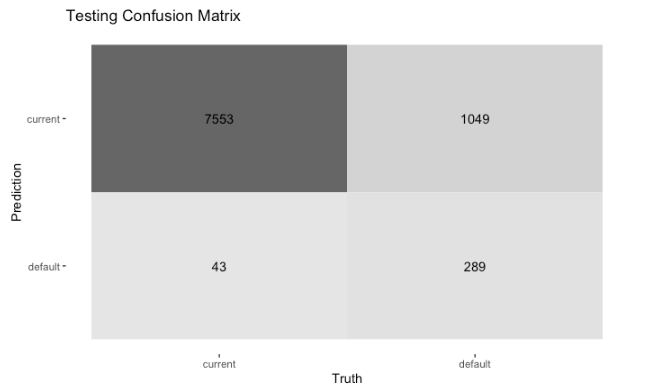


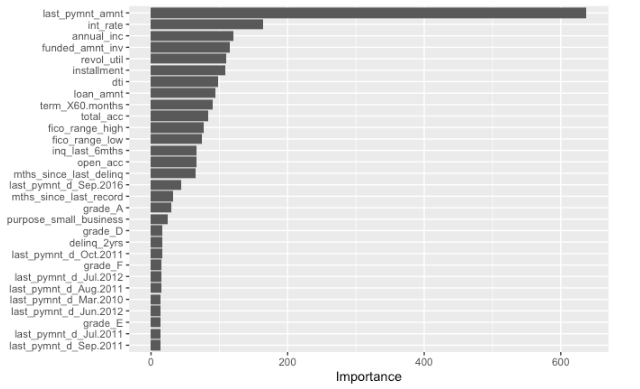


After applying the hyperparameters back to the workflow and running the model, these are the metrics, the most important variables that affect the model, and the confusion matrices for the training and testing data shown below. It is evident this model performs well on the training data, but not as well on the test data. This can be a sign of additional overfitting, as mentioned before. The discrepancy between the training and testing for every metric is a sign that the model might not be a great predictor of defaulted loans.

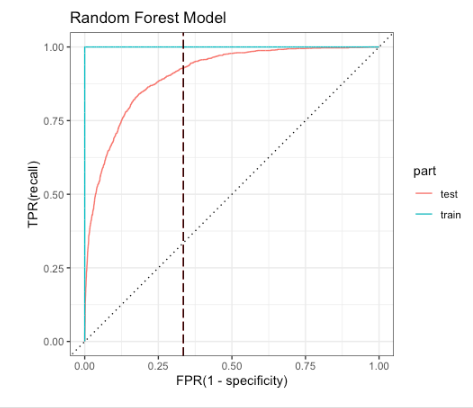
| Random Forest Model Metrics | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Partition | Accuracy | ROC AUC | LogLoss | Precision | Recall | F1 |
| Training | 0.9488557 | 0.9987762 | 0.1693901 | 0.9995181 | 0.66072 | 0.7955504 |
| Testing | 0.8777703 | 0.9047777 | 0.2817482 | 0.8704819 | 0.215994 | 0.3461078 |







As seen in this ROC curve with the same 5% operating threshold, the training data is prone to overfitting because of the shape of the curve.



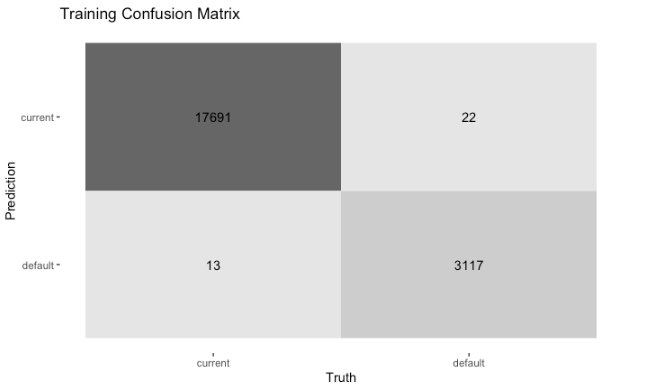
Gradient Boosting Model

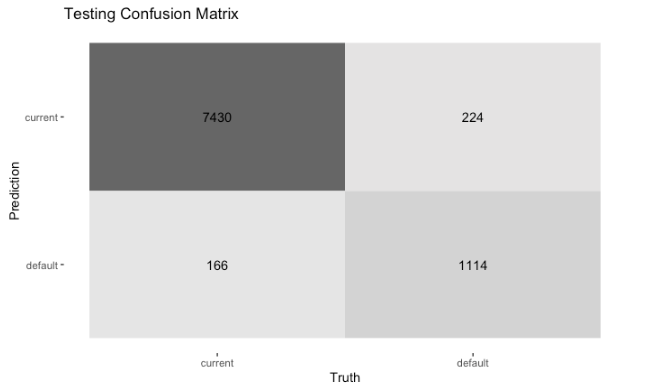
Another model that is more complex than a logistic regression model is a gradient boosting model, otherwise known as “XGBoost”. The parameters for the XGBoost are the number of trees in this case, and the number of trees was set to 350 for optimal performance. The recipe for this model is the same as the logistic regression recipe because all the variables render as significant predictors to the model. The list of variables and whether or not they are included are below:

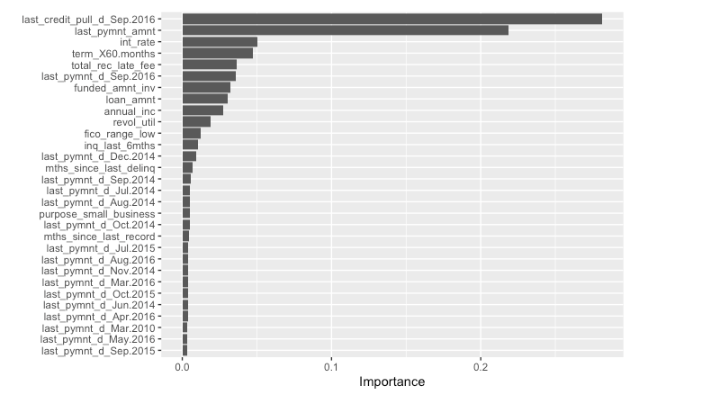
| XGBoost Recipe | | | |
| --- | --- | --- | --- |
| Variable | Status | Variable | Status |
| id | Excluded | fico\_range\_low | Included |
| member\_id | Excluded | fico\_range\_high | Excluded |
| loan\_amnt | Included | inq\_last\_6mths | Included |
| funded\_amnt | Excluded | mths\_since\_last\_delinq | Excluded |
| funded\_amnt\_inv | Included | mths\_since\_last\_record | Excluded |
| term | Included | open\_acc | Excluded |
| int\_rate | Included | pub\_rec | Excluded |
| installment | Excluded | revol\_bal | Excluded |
| grade | Excluded | revol\_util | Included |
| sub\_grade | Included | total\_acc | Excluded |
| emp\_title | Excluded | out\_prncp | Included |
| emp\_length | Included | out\_prncp\_inv | Included |
| home\_ownership | Excluded | total\_rec\_late\_fee | Included |
| annual\_inc | Included | last\_pymnt\_d | Included |
| verification\_status | Included | last\_pymnt\_amnt | Included |
| issue\_d | Excluded | next\_pymnt\_d | Excluded |
| loan\_status | Target | last\_credit\_pull\_d | Included |
| pymnt\_plan | Excluded | collections\_12\_mths\_ex\_med | Excluded |
| url | Excluded | policy\_code | Excluded |
| desc | Excluded | application\_type | Excluded |
| purpose | Included | acc\_now\_delinq | Excluded |
| title | Excluded | chargeoff\_within\_12\_mths | Excluded |
| zip\_code | Excluded | delinq\_amnt | Excluded |
| addr\_state | Excluded | pub\_rec\_bankruptcies | Excluded |
| dti | Excluded | tax\_lines | Included |
| delinq\_2yrs | Included | int\_rate | Included |
| earliest\_cr\_line | Excluded | revol\_util | Included |

The results from this model are much better than the previous two models once the parameters were changed, and the best metrics are as follows along with the most important variables to the model:

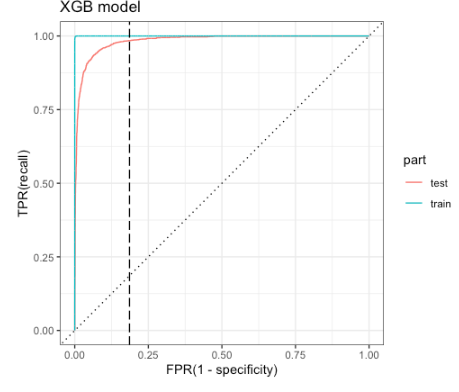
| XGBoost Model Metrics | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Partition | Accuracy | ROC AUC | LogLoss | Precision | Recall | F1 |
| Training | 0.99832 | 0.99998 | 0.02523 | 0.99585 | 0.99299 | 0.99442 |
| Testing | 0.95635 | 0.98291 | 0.11701 | 0.87031 | 0.83259 | 0.85103 |







As seen with this ROC curve for the XGBoost model, the test data performs the best with the same 5% threshold and has the largest AUC out of any of the models previously.



Model Comparison

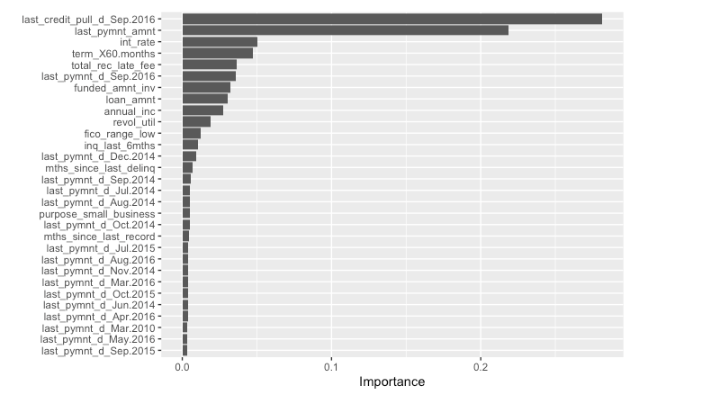
| Overall Models Comparison on Test Dataset | | | |
| --- | --- | --- | --- |
| Model | ROC AUC | Accuracy | Precision |
| Logistic | 0.9702392 | 0.9449295 | 0.8363493 |
| Random Forest | 0.9047777 | 0.8777703 | 0.8704819 |
| XGBoost | 0.9829147 | 0.95635 | 0.87031 |

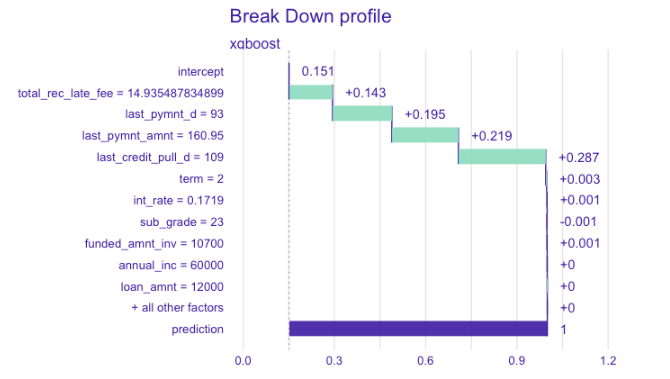
Explanation of Best Model

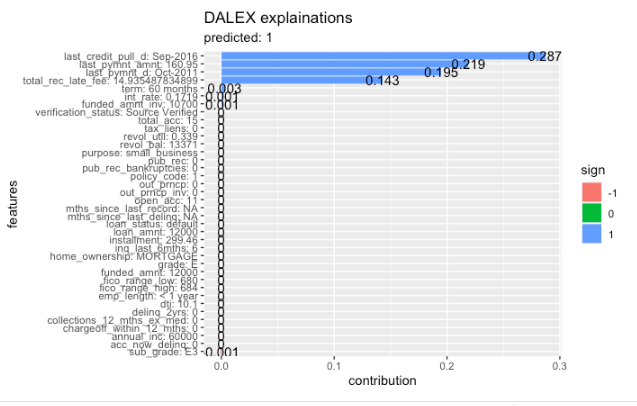
After comparing the logistic, random forest, and XGBoost models, we can arrive at a decision regarding which model is the best predictor of “loan\_status”. After looking over the metrics and other key factors, it is clear that the XGBoost model is superior to the other two models. The model has the highest ROC AUC out of the three, the highest accuracy out of the three, and the second highest precision but practically the same as the random forest. In this model, there are 22 variables contained in this model, which is a severe reduction from the original 51 predictors in the full dataset. It is quite simple to choose this model, as the random forest displays signs of overfitting due to the test data not performing nearly as well as the training, and the XGBoost simply performs better than the logistic model with the same variables contained in the recipe.

Global Explanations on XGBoost

Now that the best model has been identified, the next steps are to look at the global explanations for the best model in XGBoost starting first with the variable importance using VIP again and DALEX. The vip plot shown once again and the variable importance using DALEX are shown below. Using these on the model, the most important variables can be identified and then partial dependence plots can be created off those variables:



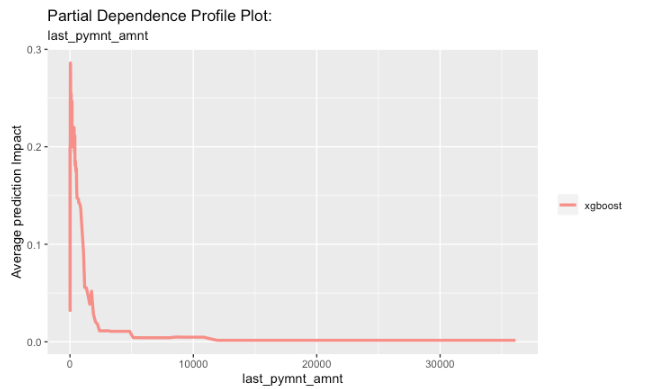




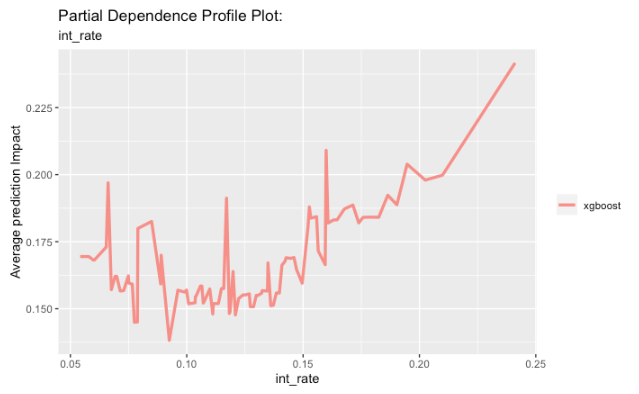
Using these plots we can identify some of the top variables in the model. The variable last\_credit\_pull\_d is one of the most important variables in the model, but can be looked at as an outlier in this case. So some of the other important variables include:

* last\_pymnt\_amnt (numeric)
* int\_rate (numeric)
* total\_rec\_late\_fee (numeric)
* funded\_amnt\_inv (numeric)
* term (character)

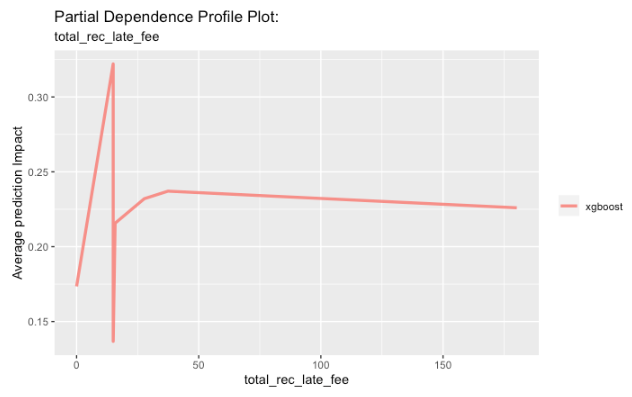
Using these variables, partial dependence plots can be constructed to determine the marginal effect they have on the predicted outcome of the target variable of “loan\_status”. The plots are shown below:



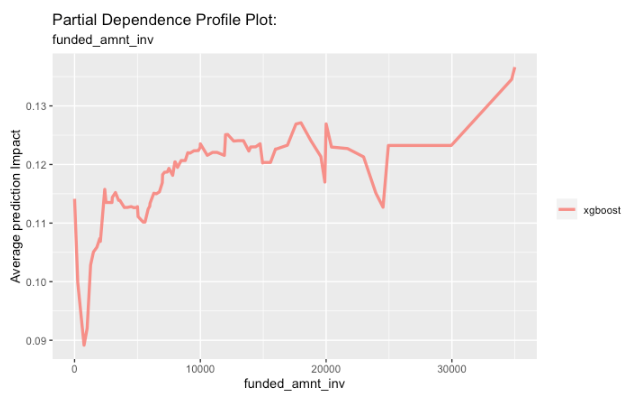
As seen above, last\_pymnt\_amnt shows a high increase in impact around the $1000-$2000 range, so it can be concluded that this variable has a significant impact as a predictor when the value is within that range.



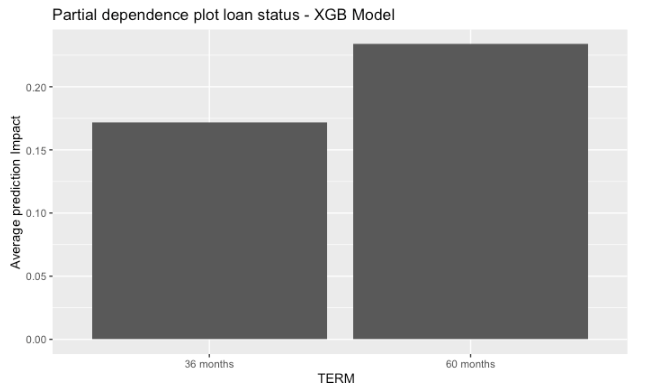
As seen in this case, there is a mostly positive correlation between an increasing value of int\_rate and increasing impact on the model.



Similar to last\_pymnt\_amnt, there is a significant outlier aspect to the total\_rec\_late\_fee variable. As seen above, in the range between about 0 and 20, there is a sharp increase and decrease in the impact, signifying that within that range the variable is important.



Similar to int\_rate, funded\_amnt\_inv displays a positive correlation between increasing and impact on the prediction.



Finally the categorical variable in term displays a discrepancy with the 60 months term showing a higher impact on prediction than the 36 months term. This discrepancy renders this variable significant and important to the model.

Local Explanations on XGBoost

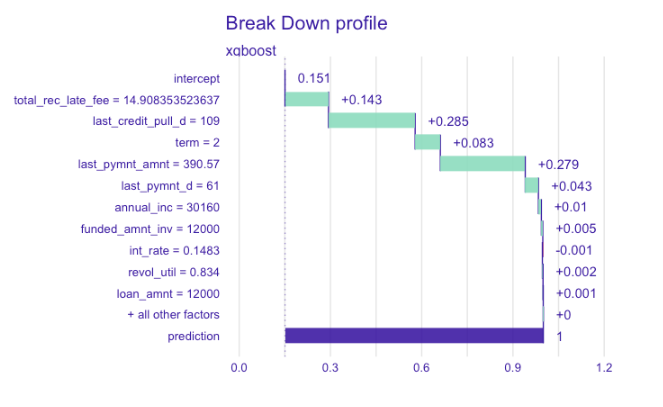
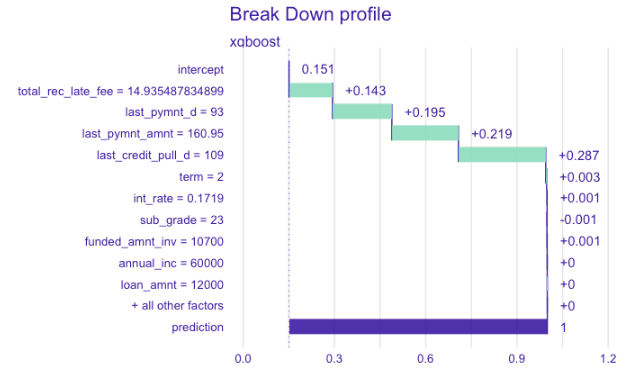
For the local explanations, the SHAP method as well is used to predict the top 10 true positives, top 10 false positives, and top 10 true negatives for the model. These output tables are displayed below:

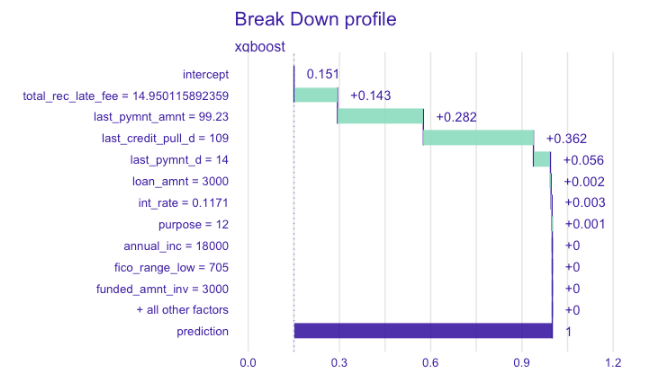
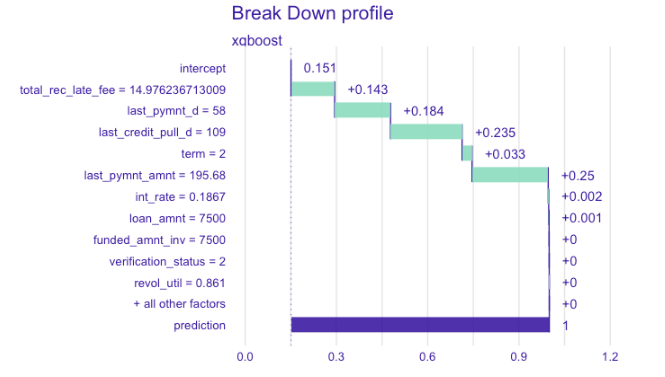
| Top 10 True Positives | | |
| --- | --- | --- |
| .pred\_current | .pred\_default | .pred\_class |
| 4.893500E-05 | 9.999511E-01 | default |
| 6.600064E-05 | 9.999511E-01 | default |
| 7.642327E-05 | 9.999236E-01 | default |
| 7.899893E-05 | 9.999210E-01 | default |
| 8.755324E-05 | 9.999124E-01 | default |
| 9.385834E-05 | 9.999061E-01 | default |
| 9.444676E-05 | 9.999056E-01 | default |
| 1.129375E-04 | 9.998871E-01 | default |
| 1.262785E-04 | 9.998737E-01 | default |
| 1.395080E-04 | 9.998605E-01 | default |

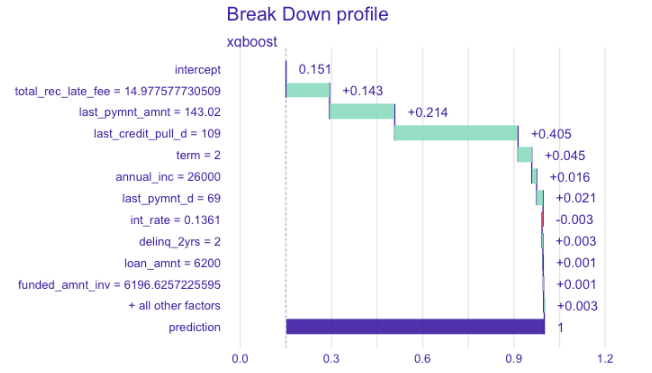
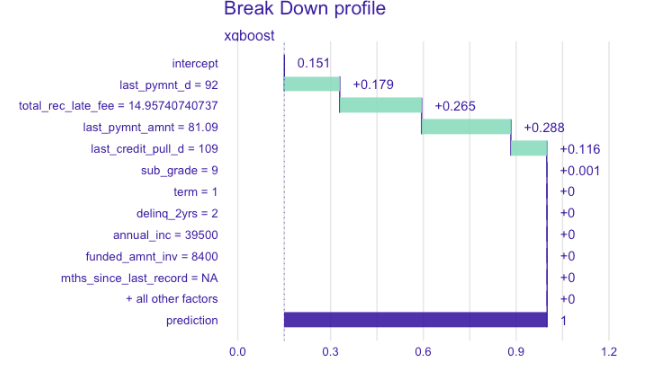
| Top 10 False Positives | | |
| --- | --- | --- |
| .pred\_current | .pred\_default | .pred\_class |
| 9.999999E-01 | 1.192093E-07 | current |
| 9.999999E-01 | 1.192093E-07 | current |
| 9.999999E-01 | 1.192093E-07 | current |
| 9.999998E-01 | 2.384186E-07 | current |
| 9.999998E-01 | 2.384186E-07 | current |
| 9.999998E-01 | 2.384186E-07 | current |
| 9.999998E-01 | 2.384186E-07 | current |
| 9.999998E-01 | 2.384186E-07 | current |
| 9.999996E-01 | 3.576279E-07 | current |
| 9.999996E-01 | 3.576279E-07 | current |

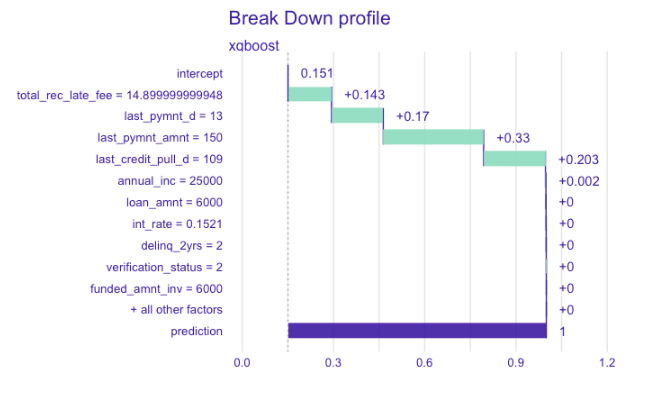
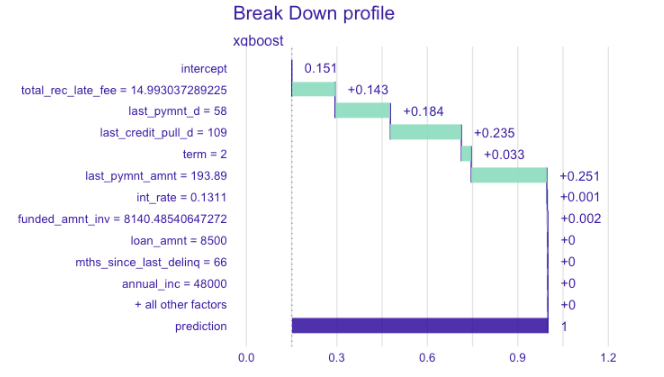
| Top 10 False Negatives | | |
| --- | --- | --- |
| .pred\_current | .pred\_default | .pred\_class |
| 4.893500E-05 | 9.999511E-01 | default |
| 6.600064E-05 | 9.999511E-01 | default |
| 7.642327E-05 | 9.999236E-01 | default |
| 7.899893E-05 | 9.999210E-01 | default |
| 8.755324E-05 | 9.999124E-01 | default |
| 9.385834E-05 | 9.999061E-01 | default |
| 9.444676E-05 | 9.999056E-01 | default |
| 1.129375E-04 | 9.998871E-01 | default |
| 1.262785E-04 | 9.998737E-01 | default |
| 1.395080E-04 | 9.998605E-01 | default |

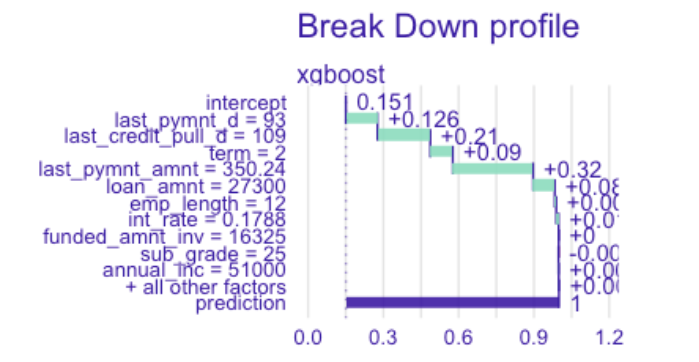
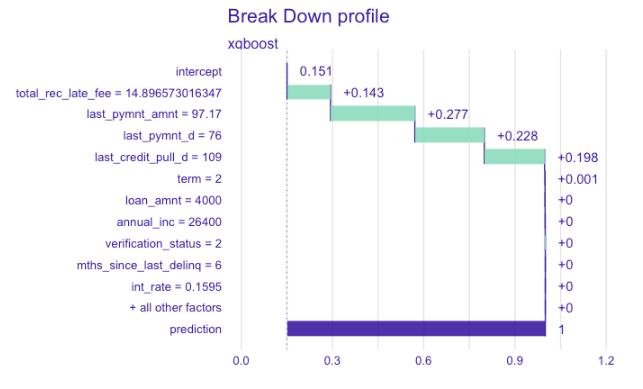
These predictions give insight into the predictability of the model, and what are the most accurate instances for both positives and negatives regarding defaulted loans. Additionally, the SHAP method can help us understand the most important variables within each of these instances for the top 10 of each category. These outputs are seen below:











Predictions

The final step in this report is to use the best model which as mentioned before is the XGBoost, and predict the loan\_status on the variables within the holdout dataset. The csv file with the predictions is attached to this report.