BANA 212 - DATA PROGRAMMING & ANALYTICS
FINAL PROJECT REPORT

LendingClub Default Risk Prediction

Team Members:
Karl Hickel, Candice Biying Han, Kathy Yu-Hsin Lee, Salem Arthur.

Table of Contents

Introduction & Executive Summary

LendingClub is the world's largest peer-to-peer lending platform. Founded in 2006 by Renaud Laplanche, the company has risen to become one of the largest online platforms for personal loans by offering those loans on a secondary market. Headquartered in San Francisco, it takes a customer-centric approach to providing loans of up to $40,000 for interested borrowers by pairing them with investors who would vet the borrower, the purpose of the loan and it's overall viability before providing them with the funds for the loan. By creating such a marketplace, the company makes money from fees collected from both investors and borrowers on the platform.

In 2016, the company was marred with controversy and the board fired its founder and CEO after allegations concerning "data integrity and contract approval monitoring and review processes" which sent the company's stock tumbling about 26% that day. There were multiple reports of dubious loans with doctored requirements to fit an investor's criteria in order to sell large sum loans, inaccurate statements and inferences being made within the company's registration portal and some customers even receiving loans that violated state usury limits. This all snowballed and led to a grand jury subpoena from the United States Department of Justice and investigation by the Securities Exchange Commission, resultantly charging the company with fraud in the fall of 2018.

Interestingly, the data that we have available was collected over a two-year period during which the company was virtually embattled, from January 2016 to December 2017. The effects of the revelations during this time period were felt all over the lending marketplace and LendingClub as expected, was dealt numerous blows. The company executed several layoffs, its stock price was cut in half by the end of 2016 and the company's management and data integrity practices underwent massive change. The biggest blow dealt was the incredible loss of borrowers and investors alike, and the general loss of trust within the public.

The company has since made huge strides; in 2019, it returned to adjusted net income profitability and in 2020 announced the acquisition of Radius Bank, an incredibly bold move that could help bring back its glory days.

Project Outline

Disclaimer: The results in this report will differ from those we presented. Unfortunately, we made the mistake of dropping too many columns in our data cleaning for the analysis because there was a huge amount of NAs and we only focused on attributes that had completed data. We then went back and later discovered after a reanalysis that we could have included some other attributes to improve on our machine learning models.

As mentioned in our introduction, our dataset is collected from January 2016 to December 2017 and contains data for loans issued within that time period. As a company reeling from the effects of such a scandal there are various business implications that can be learned and applied based on the sets of data we have available.

We want to try to assess which loans will be defaulted based on the loan status and information about the borrowers and possibly draw inferences on how LendingClub can maximize their profit and reduce defaults. We also want to be able to predict both customer loan types and their risk of not fully paying back the loan provided their information. Such as income, house type, and zip code just to name a few.

By drawing from the various data exploration techniques that have been taught to us in class in Python using the Pandas & MatPlotLib packages, we also hope to be able to better understand the customer base and the types of loans they apply for. We can look at how the borrowers' status such as employment title and length, homeownership, annual income, and previous lending history can affect the grade of the loan. We'll be able to look at how interest rate is affected by the borrowers' status (employment length, homeownership, annual income, and so on). And we'll also be able to answer the question of whether borrowers with lower grades tend to have higher default rates.

We'll be using various packages such as MatPlotLib, Seaborn and GraphViz to create easy-to-read and tangible graphs that depict some of the learnings that we are able to garner from our project and will employ the SkLearn and PyDotPlus packages in the creation of our decision tree and random forest algorithms.

Exploratory Data Analysis

```
data
```

	id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	home_ownership	...	total_rec_prncp	total_rec_int	total_rec_late_fee	re
0	112435993	2300	2300	2300.0	36 months	12.62	77.08	C	C1	OWN	...	270.09	112.90	0.0	
1	112290210	16000	16000	16000.0	60 months	12.62	360.95	C	C1	MORTGAGE	...	1186.93	873.16	0.0	
2	112436985	6025	6025	6025.0	36 months	15.05	209.01	C	C4	MORTGAGE	...	684.18	348.28	0.0	
3	112439006	20400	20400	20400.0	36 months	9.44	652.91	B	B1	RENT	...	2501.19	736.61	0.0	
4	112438929	13000	13000	13000.0	36 months	11.99	431.73	B	B5	MORTGAGE	...	1539.34	597.66	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
759333	65854936	6000	6000	6000.0	36 months	7.89	187.72	A	A5	OWN	...	3668.59	646.34	0.0	
759334	66055600	6000	6000	6000.0	36 months	9.17	191.28	B	B2	RENT	...	3641.68	775.37	0.0	
759335	66141895	14400	14400	14400.0	60 months	13.18	328.98	C	C3	RENT	...	4219.84	3007.18	0.0	
759336	65673209	34050	34050	34050.0	36 months	15.41	1187.21	D	D1	MORTGAGE	...	19919.22	7561.51	0.0	
759337	65744272	5000	5000	5000.0	36 months	11.22	164.22	B	B5	MORTGAGE	...	2998.76	775.18	0.0	

759338 rows × 38 columns

We can see here from the above image that we have 759,338 different observations in our dataset, or in other words 759,338 unique loan IDs or applications. Originally, we had 72 different attributes to choose from and dropped the columns that we viewed were not necessary for our analysis in order to allow for more accurate and easy to generate interpretations. We were left with 38 variables after dropping those we did not need and will briefly describe some of them below:

- funded_amnt - the total amount of money a borrower was approved for and awarded
- term - the length of time a borrower will have to make monthly payments on a loan
- int_rate - the interest rate on a loan
- installment - the agreed upon sum of money to be paid on a monthly basis of a loan's term
- grade - a quality score assigned to each loan based on the borrower's credit history, collateral and likelihood of repayment
- annual_inc - a borrower's stated annual income
- loan_status - the approval status or progress made on disbursement/payment for an individual loan application
- pymnt_plan - a loan payment plan
- dti - debt to income ratio

- total_pymnt - the total payments made by a borrower on their loan
- last_pymnt_d - the last payment date on a loan

We used WEKA to determine what the most important attributes were for our machine learning algorithms that did not overfit and were not duplicates of other attributes (for example: total_pymnt and total_pymnt_inv had the same values). We decided to use the information gain evaluator and then ranked our top 10 attributes, with recoveries as the most important.

```
Ranked attributes:
 0.23543   28 recoveries
 0.23521   29 collection_recovery_fee
 0.10436    5 int_rate
 0.05238   25 total_rec_prncp
 0.02772   26 total_rec_int
 0.02079   24 total_pymnt
 0.01661   11 pymnt_plan
 0.01315   30 last_pymnt_amnt
 0.01071   27 total_rec_late_fee
 0.00543    7 grade
```

After our data cleaning is done, and we are satisfied with the results and our new data we decide to create a new dictionary to help us understand the default risk among our customers based on our understanding of the status of their loan. We are able to classify all current and fully paid loans as no risk, charged off loans and those marked as default were classified as default, loans late by 31-120 days were categorized as high risk and those late by 16-30 days and also those in grace period were classified as medium risk.

```
Current               565523
Fully Paid            130718
Charged Off            37197
Late (31-120 days)     15354
In Grace Period         6634
Late (16-30 days)       3876
Default                   36
Name: loan_status, dtype: int64
No Risk        696241
Default         37233
High Risk       15354
Medium Risk     10510
Name: default_risk, dtype: int64
```

```
No Risk        91.69
Default         4.90
High Risk       2.02
Medium Risk     1.38
```

We can see here that a majority of the loan applications, 565523 to be exact, are classified as no risk whilst we find that our medium risk classification, containing those late by 16-30 days and those in grace period which is the time during which a borrower isn't expected to make any payments is listed as having the least number of loans. Here, we are thus able to calculate a default rate of 4.90% using this information which basically tells us that close to 5% of the total number of loans issued between 2016-2017 had either not been paid off yet or most likely would not be paid off in full.

```
return on investment for defaulted loans:  -70.61041780037785
return on investment for fully paid loans:  8.489282767312927
```

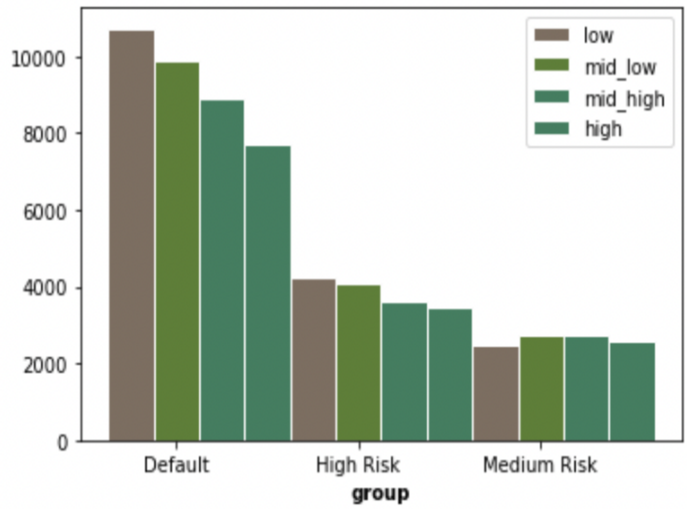
This begs the question of what our return on investment is for those loans which have defaulted, and we can calculate this by subtracting the funded amount for defaulted loans from their total payment amounts and then dividing that by the funded amount. Repeating the same logic for our fully paid loans, we can see that we are a net negative of 70.6% for the defaulted loans and a small profit of 8.489% for the fully paid loans.
In an attempt to better understand the customers that are applying for these loans, we decide to consider a couple of didn't attribute to determine the various income types and associating default risks as well as looking at loan purposes and their associated default risks trying to identify any opportunity gaps for minimizing loan defaults.

```
          risk   average_inc
0      No Risk  80462.130154
1      Default  73482.530487
2    High Risk  74316.058167
3  Medium Risk  78450.516216
```

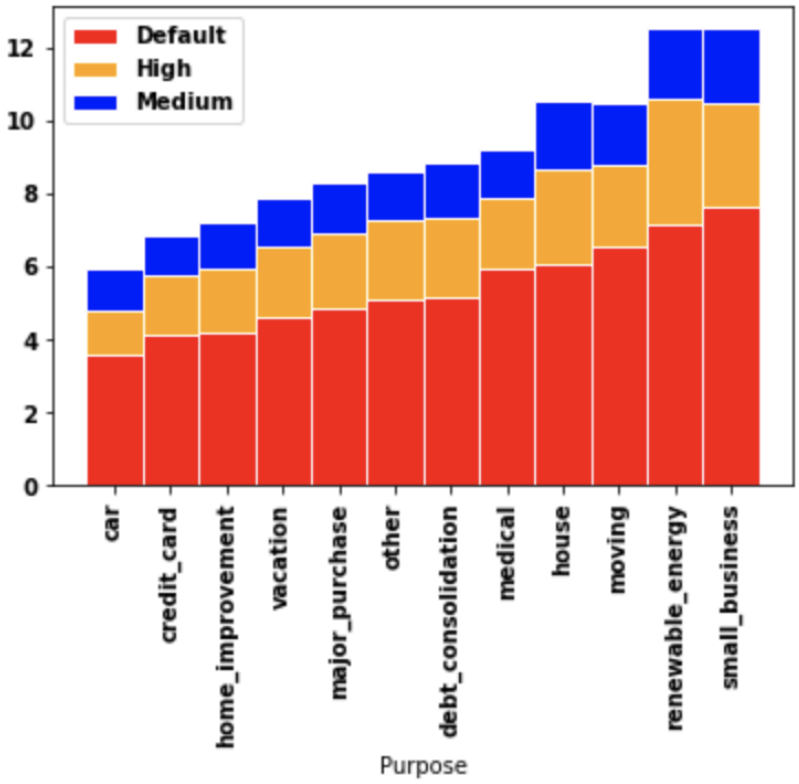
By calculating the mean income amounts for customers based on the risk classifications we created we identified that the default risk class on average had the least annual

income whereas the no risk had the highest average as expected. The maybe worrisome observation here though was how close these different annual income averages were.

income_level	default_risk	
high	Default	7708
	High Risk	3441
	Medium Risk	2579
	No Risk	182162
low	Default	10723
	High Risk	4254
	Medium Risk	2486
	No Risk	170599
mid_high	Default	8918
	High Risk	3590
	Medium Risk	2734
	No Risk	169322
mid_low	Default	9884
	High Risk	4069
	Medium Risk	2711
	No Risk	174158



By reclassifying customers based on their average incomes we arrive at 4 different classes; high for those earning above $95000, mid_high for customers earning between $67000 and $95000, mid_low for those earning between $48000 and $67000 and low for customers earning below $48000 per year. This allows us to take a closer look at the amounts of customers per default risk.



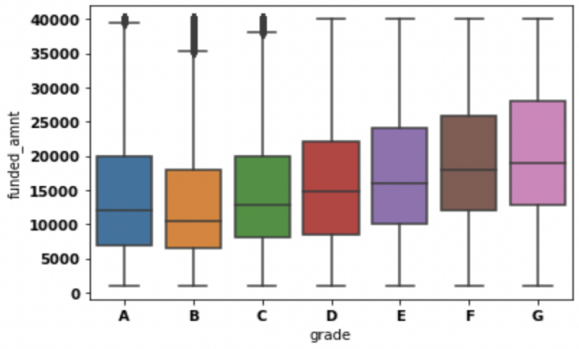
After getting a better idea of the various default risks associated with each income level, we decide to try to decipher the relationship between the purpose of a loan application and the various default risk classes that we have created by finding the respective number of loan IDs within each unique purpose. For instance, if we take a close look at customers who took out loans for debt consolidation purposes, we can assess that a relatively large amount of that group defaulted on their loans as opposed to those that were high and medium risk and the same can be said for those who took out loans to start or support small businesses as well as those who used the loan for renewable energy purposes.

```
Average interest rate for A is:  6.926255049504337
Average interest rate for B is:  10.383187338324486
Average interest rate for C is:  14.012157466488318
Average interest rate for D is:  18.43361093529974
Average interest rate for E is:  23.277900567212093
Average interest rate for F is:  27.224676925219033
Average interest rate for G is:  29.948677984467878
```

With various loan grades, term durations, interest rates and total loan amounts we can further categorize or classify our customers in the dataset using these attributes and

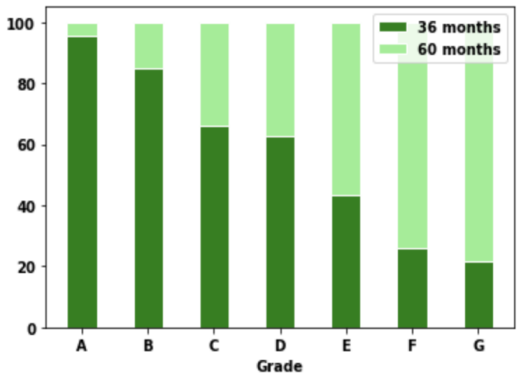
thus calculate the average rates and amounts for customers in these various sections. For example, we first notice as expected that loans with the highest grade, A, typically have the lowest average interest rate at about 6.9%. On the flip side, customers with a loan grade of G had typically observed the highest interest rates with an average of 29. 94%.

```
Average funded amount for A is:  14020.677096300104
Average funded amount for B is:  13375.611116647242
Average funded amount for C is:  14952.73843113396
Average funded amount for D is:  15959.760815859858
Average funded amount for E is:  17573.30501284724
Average funded amount for F is:  19028.284709650467
Average funded amount for G is:  20143.981397868884
```

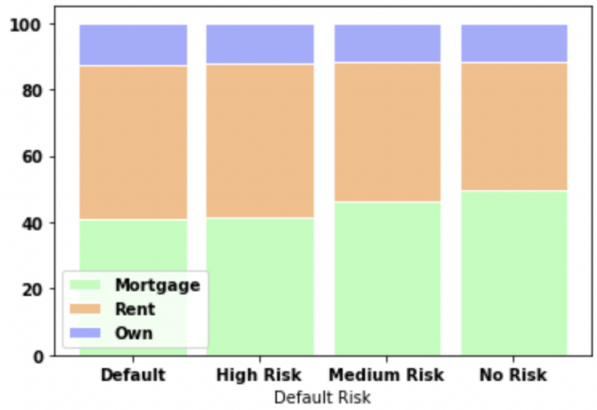


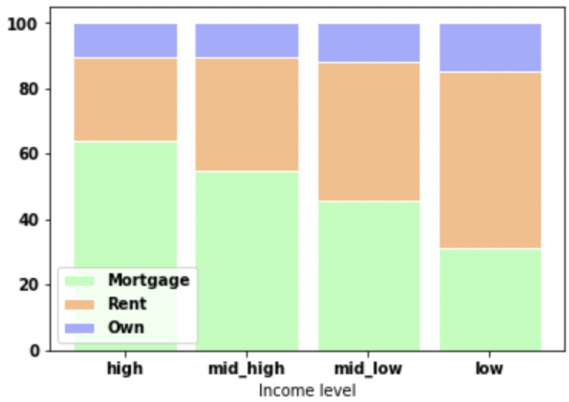
Calculating the averages for total funded amounts for the various customers based on their loan grade produces some interesting results. Surprisingly, the section of customers with the lowest loan grade of G tend to be funded the most amount of money with an average of $20,143.98 whilst customers with a loan grade of B are funded the least amount with an average of $13,375.61. With our analysis, based on customer's loan terms which is the amount of time a customer will have to make monthly payments back to an investor for a loan, we take a look at the count of customers for each loan grade per term duration. This reveals that a massive number of customers with loan grades of A, opted into 36-month loans instead of 60-month loans, with 119,053 borrowers choosing the shorter term compared to just 5,464. There's a more even distribution for borrowers loan grades of E with 17,816 choosing the 36-month term as opposed to 23,438 that chose the longer term. And just like we have observed with various other variables, borrowers with the loan grade of G typically behave in a juxtaposed manner with far more borrowers (4,333) choosing the longer 60-month term compared to those who chose the 36--month loan term (1,204).

	36 months	60 months
A	119053	5464
B	196392	34415
C	161410	83493
D	61612	36346
E	17816	23438
F	3750	10612
G	1204	4333

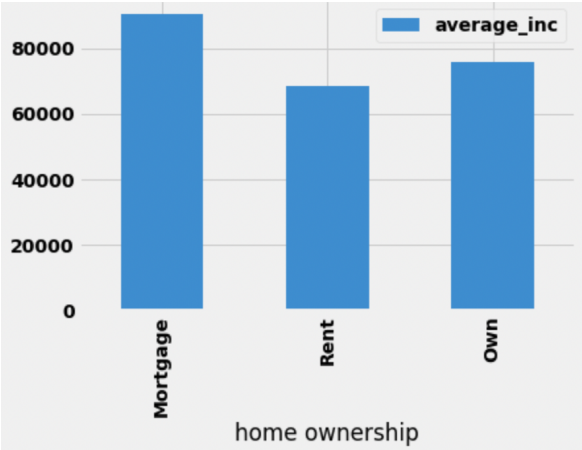


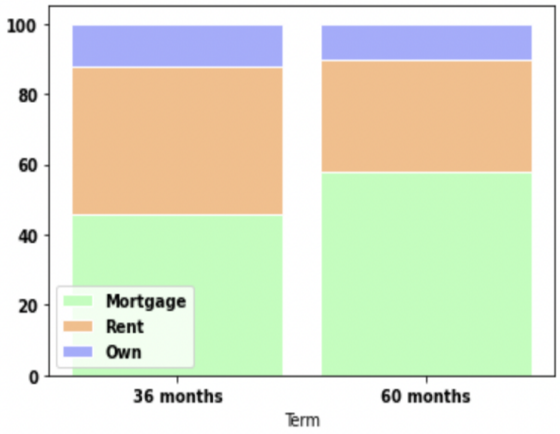
One last variable our team deemed very important to look at was of homeownership as we believe that this could help provide even more insights into the makeup of our borrowers and possibly even help us make inferences on default risk.





We do so by calculating the various counts of borrowers for each homeownership type and then comparing those per loan grade, loan term, income level and finally their default risk. The very first observation that we make is regardless of whether we compare borrowers per income level or loan term, most borrowers tend to have mortgages within those classes. The average income for borrowers with mortgages is also quite higher than those for borrowers who rent or own.





Out of the 37,233 borrows that are at risk of defaulting on their loans, we are able to observe that 15,260 of those borrowers have mortgages on their homes which would make up almost half of that subgroup as opposed to 17,276 borrowers renting their properties. A look at our descriptive statistics simply call for the creation of models to enable us to be able to best identify and predict patterns that would help us to identify applicants who are most likely to default on their loans.

Algorithm Analysis

After looking at the descriptive statistics and summaries that we have been able to generate; we are fairly confident that this is a classification problem and thus decided to use the following machine learning algorithms in our analysis:

1. Decision Tree
2. Random Forest
3. K-Nearest Neighbor

Using these models, we will be able to make predictions on whether a borrower or loan applicant will default on their loan with LendingClub. To do so, we'll first have to divide our dataset into two different feature sets to help us generate the most optimal classifiers as using different attributes with our models should certainly produce different results. We'll then observe the overall accuracy of our analysis considering the different feature sets we are able to form and then will summarize the results of our findings.

```
Ranked attributes:
 0.23543   28 recoveries
 0.23521   29 collection_recovery_fee
 0.10436    5 int_rate
 0.05238   25 total_rec_prncp
 0.02772   26 total_rec_int
 0.02079   24 total_pymnt
 0.01661   11 pymnt_plan
 0.01315   30 last_pymnt_amnt
 0.01071   27 total_rec_late_fee
 0.00543    7 grade
```

Our image above here depicts the list of attributes that we selected as part of our first feature set for algorithm analysis and includes the total loan amount applied for, the funded amount, interest rate on the loan, the grade of the loan and finally a borrower's annual income.

Decision Tree

Decision trees are a non-parametric supervised learning model that are typically used for classification and regression analysis. Here, our goal is to simply keep splitting the data based on different tests or conditions until we arrive at our desired output and can quantify the accuracy of that prediction.

```
Predicted values:  [3 3 3 ... 1 3 3]
[[  451   229  2466]
 [  432   605  3649]
 [  255   429 10414]]
Model accuracy :  60.59165346011621
              precision    recall  f1-score   support

           1       0.40      0.14      0.21      3146
           2       0.48      0.13      0.20      4686
           3       0.63      0.94      0.75     11098

    accuracy                           0.61     18930
   macro avg       0.50      0.40      0.39     18930
weighted avg       0.55      0.61      0.53     18930
```

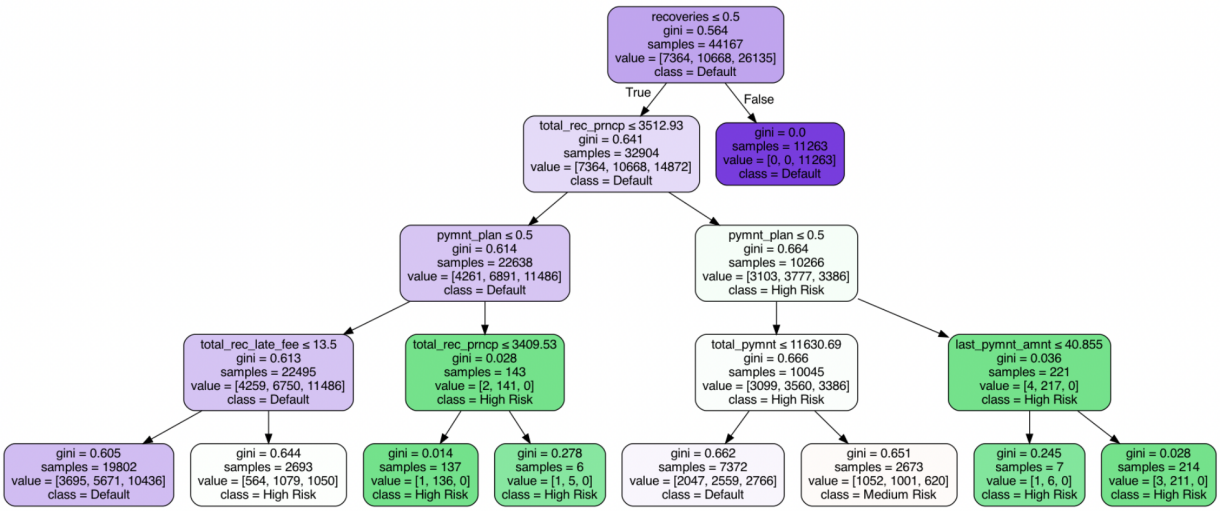
Although decision tree classifiers conceptually work just fine with categorical and numeric data, we'll have to convert our categorical attributes into numeric values and will map the binary values that we derive to its corresponding categorical attribute. With this feature set, we do so first with the default risk variable that contains different categories for default risks which we calculated earlier as this variable is the category we are trying to predict. So for instance, the "No Risk" default group is represented by

the number 0 and “Medium Risk” by the number 1. We’ll do the same thing for the loan grade variable as well for easy understanding and processing. So here, “A” is mapped to 0, “B” is mapped to 1, “C” mapped to 2 and so on.

After that, we will use the SciKitLearn package to split our data for our first feature set into two halves, a training and testing set using a 70/30 split. Separating our data into these two sets in an extremely important part of algorithm analysis that will help us identify any discrepancies with our model and help us optimize its accuracy. This time around, we choose not to standardize our data in the first feature set as this will allow for a more easy to interpret tree.

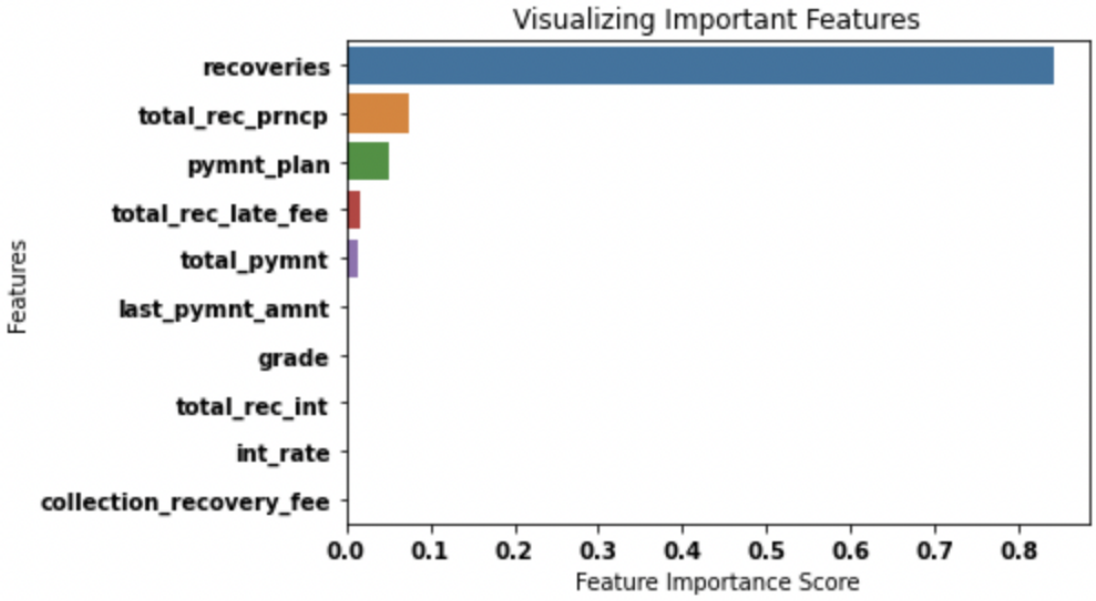
Next, we’ll try to understand the uncertainty of our first feature set using the Gini Index, which calculates the probability of a specific feature or variable that is classified incorrectly when selected at random or in other words, the quality of a split. The Gini Index ranges from 0 to 1 and whilst we create our tree, the features with the lowest Gini index score will be selected as the most preferred since they carry the least uncertainty or randomness.

After setting the maximum depth of the tree to 4 and then specifying 5 as the minimum number of samples required to be at a leaf node in order to help smooth the model, we run the decision tree classifier on our training sets and after the creation of the model run that again using our test set to generate some predictions. Once those predicted values are generated we can look at the confusion matrix, the model’s accuracy and a classification report to get a better understanding of our results. Once our results are generated, we can visualize them using the SciKitLearn and PyDotPlus packages.



Our decision tree model generates an accuracy score of 61% and especially adept at making predictions for Class 3 with a precision score of 0.61. We are also able to

calculate the most important features in our model to best understand which variables help us predict a borrower's default risk more effectively and from our first feature set we realize that recoveries are extremely important variables to consider.



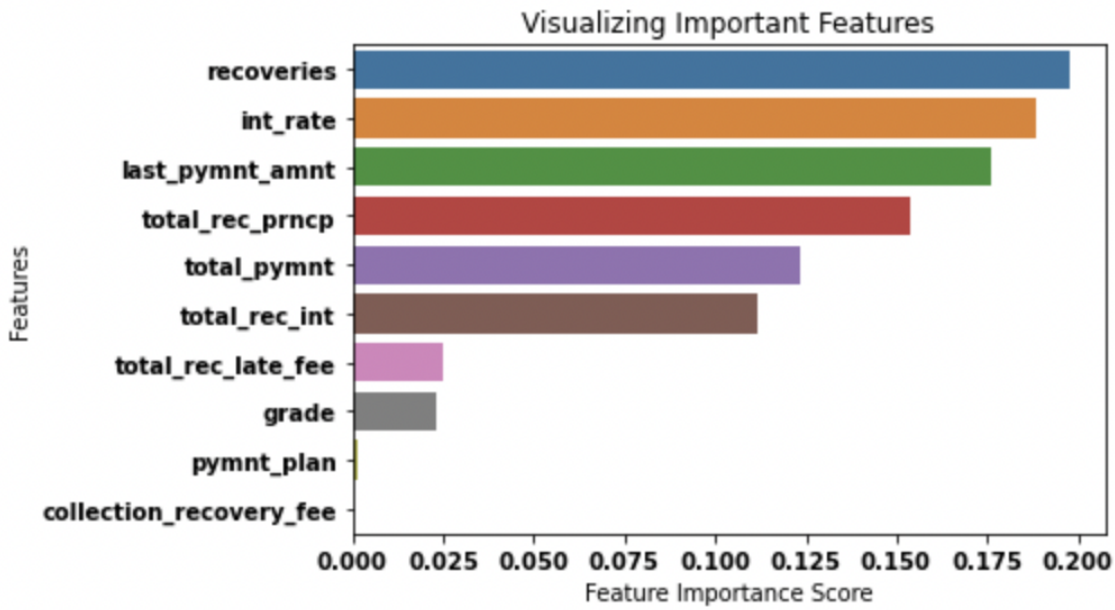
Random Forest

Random Forest is a classification algorithm that relies on the idea of two or more heads being better than just one and thus is simply a collection of various decision tree classifiers that make up a forest whose collective outputs and accuracy should help us arrive at the most optimal prediction.

Similarly to the decision tree we run prior to this we run the random forest classifier on our training sets first and in our criteria, specify that we want 100 different trees to be generated using the "n_estimators" parameter.

```
Predicted values:  [2.1        2.61       2.02       ... 1.57       1.83       2.04666667]
Mean Absolute Error: 0.36882854129857995
Mean Squared Error: 0.30775378618511745
Root Mean Squared Error: 0.5547556094219485
Model accuracy: 77.304152761702
```

After running our model we achieve an accuracy of 77.3% and just like with our decision tree, we are also able to identify the most important features here with our random forest classifier. Somewhat surprisingly the results we see here greatly differ from those from our decision tree. Here, recoveries is ranked as the most important feature with the collection recovery fee being the least impactful.

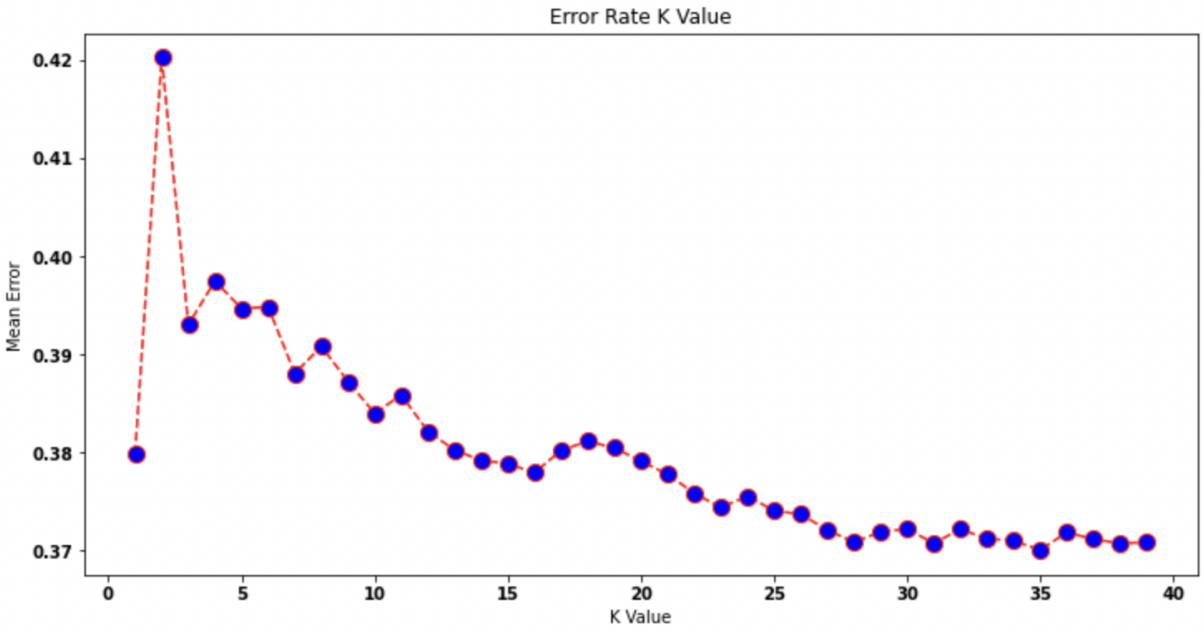


KNN

K-Nearest Neighbors quite like decision trees are fairly simple non-parametric supervised learning models great for classification and regression purposes. KNN as an algorithm assumes that similar variables or attributes should exist in proximity and thus can be visualized in easily identifiable classes or groups to make accurate predictions. This idea of proximity is measured using the distance between two variables which we will do using the Euclidean approach.

Here, we'll first split our data into training and testing sets using the 70/30 split like we did previously for our analysis. We'll then scale the various features in order to have values that are more easily comparable and analyzed. Next, we calculate the error rate to help us identify the most optimal number of the nearest neighbors to use in our calculation by selecting the k-value with the smallest error rate. This helps us identify 30 as the best number of neighbors to use for our model.

```
Text(0, 0.5, 'Mean Error')
```




```
Predicted values:  [2 3 3 ... 1 2 2]
[[ 610 1081 1455]
 [ 609 1737 2340]
 [ 352 1183 9563]]
              precision    recall  f1-score   support

           1       0.39      0.19      0.26      3146
           2       0.43      0.37      0.40      4686
           3       0.72      0.86      0.78     11098

    accuracy                           0.63     18930
   macro avg       0.51      0.48      0.48     18930
weighted avg       0.59      0.63      0.60     18930
```

Above we can see the error rate in the k value, we selected a K value of 28 as the marginal diminishing values for mean error is no longer decreasing significantly. Our KNN model resulted in a 63% accuracy at a K value of 28.

For future business applications, what we have is the models to understand factors that affect one's ability to pay back a loan presuming they are at risk. For example, we can create new business practices that consider interest rates for specific individuals based on their loan grade. By considering the factors of risk for individuals we as an organization can make better decisions when issuing loans to individuals who are considered at risk for default.

Works Cited

Data Manipulation

Lynn, Shane. “Pandas Groupby: Summarizing, Aggregating, and Grouping Data in Python.” *Shane Lynn*, 30 June 2020, www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-python-pandas/.

“Matplotlib - Bar Plot.” *Tutorialspoint*, www.tutorialspoint.com/matplotlib/matplotlib_bar_plot.htm.

The Pandas Development Team. “Pandas.DataFrame.drop.” *Pandas.DataFrame.drop - Pandas 1.1.4 Documentation*, 2020, pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html.

Jeon, Tom. “Importing Data with Pandas' read_csv().” *DataCamp Community*, 9 May 2018, www.datacamp.com/community/tutorials/pandas-read-csv.

The Pandas Development Team. “How Do I Select a Subset of a DataFrame?” *How Do I Select a Subset of a DataFrame? - Pandas 1.1.4 Documentation*, 2020, pandas.pydata.org/pandas-docs/stable/getting_started/intro_tutorials/03_subset_data.html.

The Pandas Development Team. “Pandas.DataFrame.sample.” *Pandas.DataFrame.sample - Pandas 1.1.4 Documentation*, 2020, pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sample.html.

Decision Tree

Navlani, Avinash. “Decision Tree Classification in Python.” *DataCamp Community*, 28 Dec. 2018, www.datacamp.com/community/tutorials/decision-tree-classification-python.

W3Schools. “Machine Learning - Decision Tree.” *Python Machine Learning Decision Tree*, 2020, www.w3schools.com/python/python_ml_decision_tree.asp.

Hiros (h1ros). “Introduction to Graphviz in Jupyter Notebook.” Step-by-Step Data Science, 27 Feb. 2019, h1ros.github.io/posts/introduction-to-graphviz-in-jupyter-notebook.

Random Forest

Navlani, Avinash. “Random Forests Classifiers in Python.” *DataCamp Community*, 16 May 2018, www.datacamp.com/community/tutorials/random-forests-classifier-python.