LendingClub Risk Assessment

Courtney Hrdy and Shannon Gambuti

IST 707: Applied Machine Learning Final Project

December 16th, 2021

**Introduction**

LendingClub:

LendingClub was founded in 2006 and initially launched as one of Facebook’s first applications. After receiving $10.26 million in 2007, LendingClub developed into a full scale peer-to-peer lending company. In 2014, the company went public with an IPO evaluation of over $10 billion. Investors began to notice duplicate loans taken out by the same individuals with vastly different interest rates. In one particular case, one $15,000 loan had a 15% interest rate while the same individual took out a loan with double the principal and only a 9% interest rate. This meant that investors who funded only the second loan were missing out on a large sum of money. Allen Grimm, a data scientist, developed an algorithm to classify duplicate borrowers and began data mining LendingClub’s database, only to notice how bad things have gotten. In 2016, CEO Renaud Laplanche, was forced to resign due to ethical breaches that involved misdated loans and conflicts of interest. LendingClub’s stock price dropped to just $8. Grimm uncovered that they had falsified $22 million worth of loans sold to Jeffries at Goldman Sachs. In 2018, LendingClub experienced a huge surge in defaulted and charged off loans which prompted them to turn to crowdsourcing and posted their applicant data in efforts to develop a new plan to assess applicants.

The Problem:

LendingClub facilitates three to five year loans between $1,000 and $40,000 to accepted applicants based on a risk score calculation. The assigned risk score to all loan applicants determines whether a loan is accepted and the interest rate the approved loans will receive. This scoring system has been utilized for the past five years, however it requires evaluation and improvement, given the increase in defaulted loans from accepted applicants.

The Data:

For our analysis, we used two datasets provided by LendingClub: the accepted loans dataset and the rejected loans dataset. The accepted loans dataset contains 235,629 records with 107 attributes. The rejected loans dataset contains 16,384 records with 9 attributes. Between the two datasets, there are 7 common attributes: amount requested, month, purpose, debt-to-income ratio, state, years of employment, and class (reject/approve). In order to accomplish our analysis we merged these two datasets. Each loan applicant is assigned a grade A-G based on LendingClubs risk calculation.

Our Goal:

Our project aims to identify target variables indicative of risk and determine how LendingClub is currently grading their applicants. Next, LendingClub needs a new grading system for accepted loan applicants due to an increase in the number of accepted applicants defaulting/charging off their loans. We used the accepted loans dataset for data exploration and training data to find attributes indicating risk. Finally, we are attempting to understand how different attributes, including applicant grade, can predict whether or not a loan will be charged off.

**Methods**

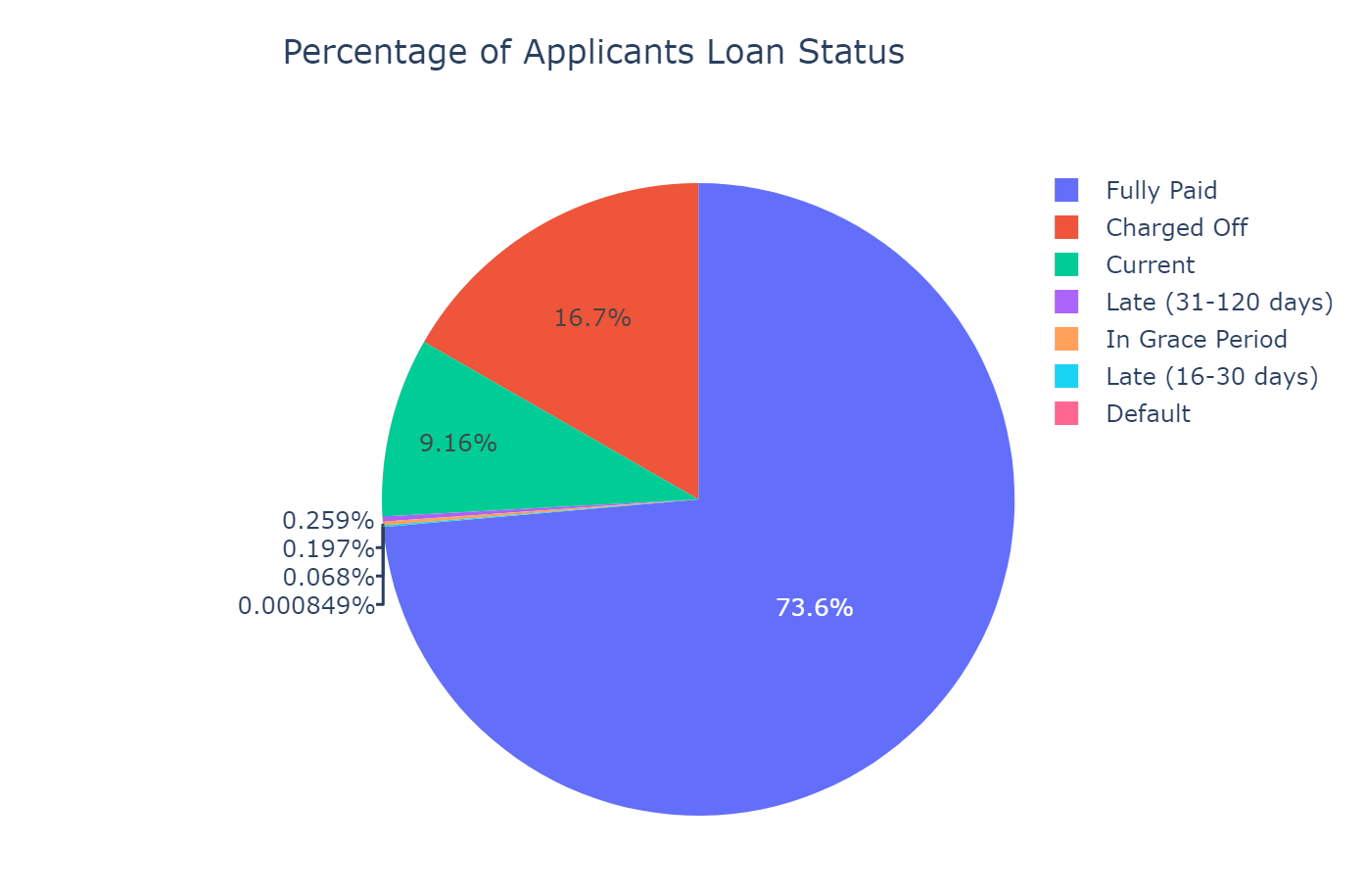
To accomplish our analysis, we used Python packages Sci-Kit Learn, Pandas, NumPy, Matplotlib, Seaborn, and Plotly written in Jupyter Notebook. We completed the following models: SVM, KNN, Gradient Boosting, Random Forest, and Logistic Regression,

Preprocessing:

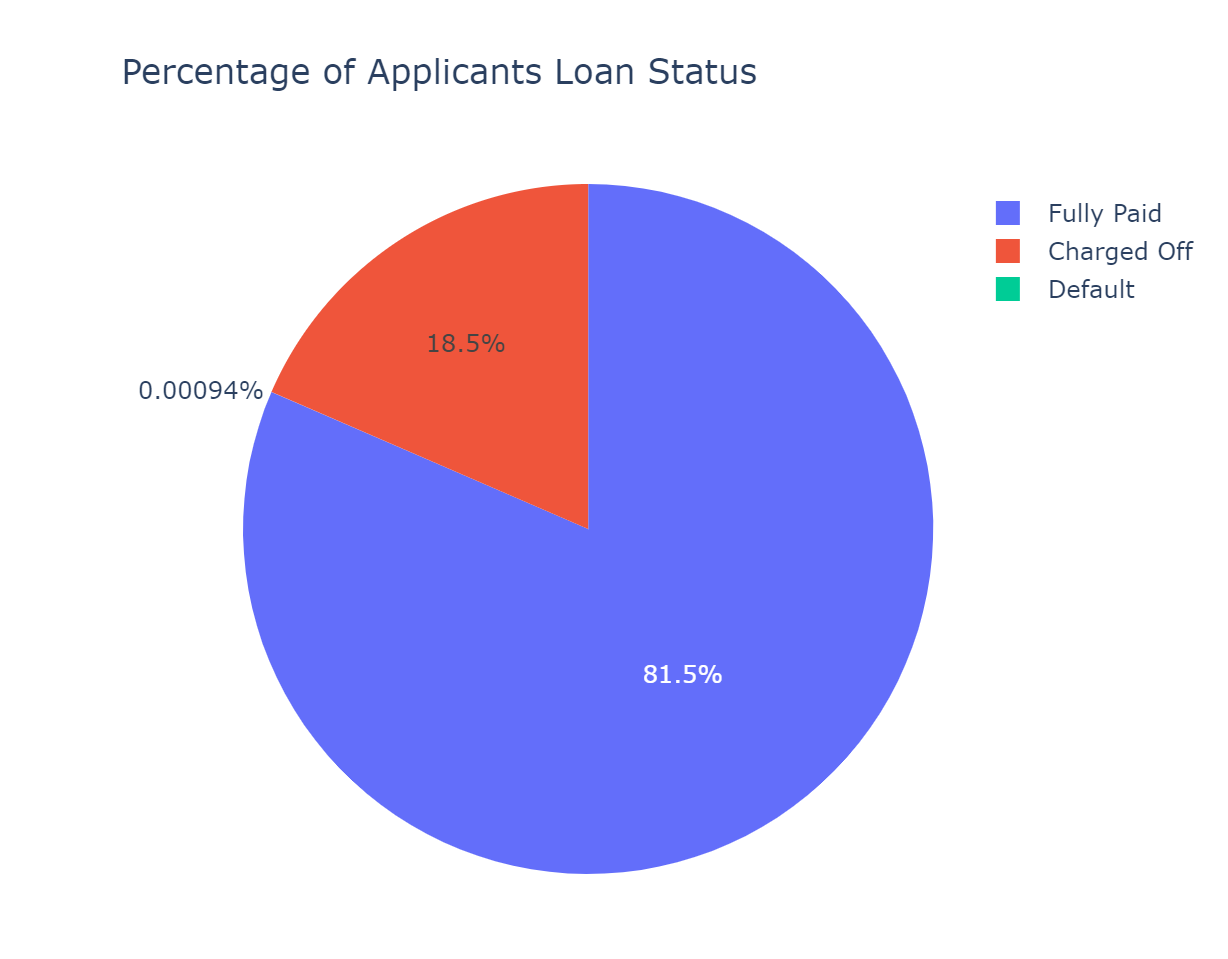
To begin, we needed to complete the following preprocessing steps. First, we removed all rows with missing values, this accounted for 39% of our original data. We then removed 23 columns that had a large number of null values and weren’t useful. We then converted interest rate from percentages to decimals, removed “<”, “+”, “year”, “years” from the emp\_length column, converted the issue date column from m-yy format ro m-yyyy, and finally merged the rejected and accepted datasets together.

Exploratory Analysis:

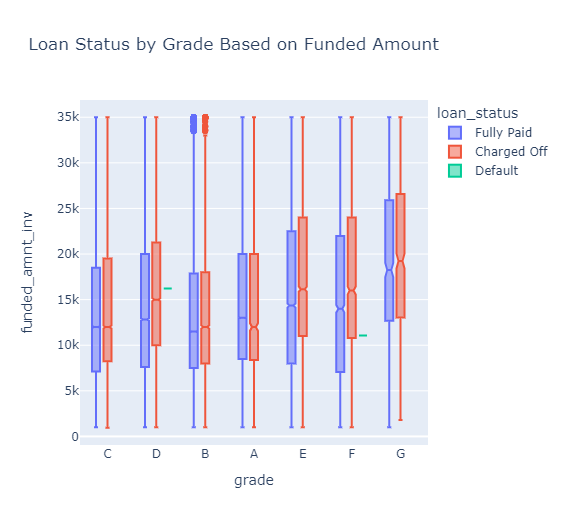
We first wanted to understand the percentage of approved applicants for each category in column ‘loan\_status’. The categories include: Fully Paid, Charged Off, Current, Late (31-120 days), Late (16-30 days), In Grace Period, and Default. The pie chart below depicts where LendingClub approved applicants stand:

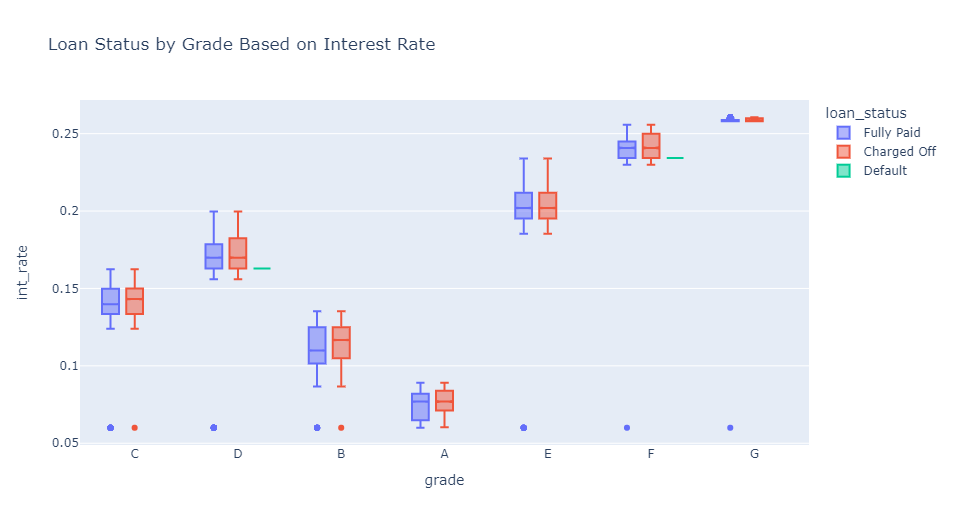
****

As you can see, the majority of applicants have fully paid off their loan. However, there is a large percentage of applicants that have charged off their loan. When a loan is charged off, it means that LendingClub has given up on being repaid according to the original terms of the loan. This does not mean that the applicant is no longer responsible for the amount owed, but rather LendingClub has removed that particular loan from their balance sheet. We then removed the loans that have been classified as in grace period, late, and current to better visualize the percentage of loans that were fully paid and charged off. Our dataset contains just 2 records of applicants who have defaulted, so we are using ‘charged off’ as our target variable throughout our analysis. The following pie chart depicts the percentage of fully paid and charged off loans:

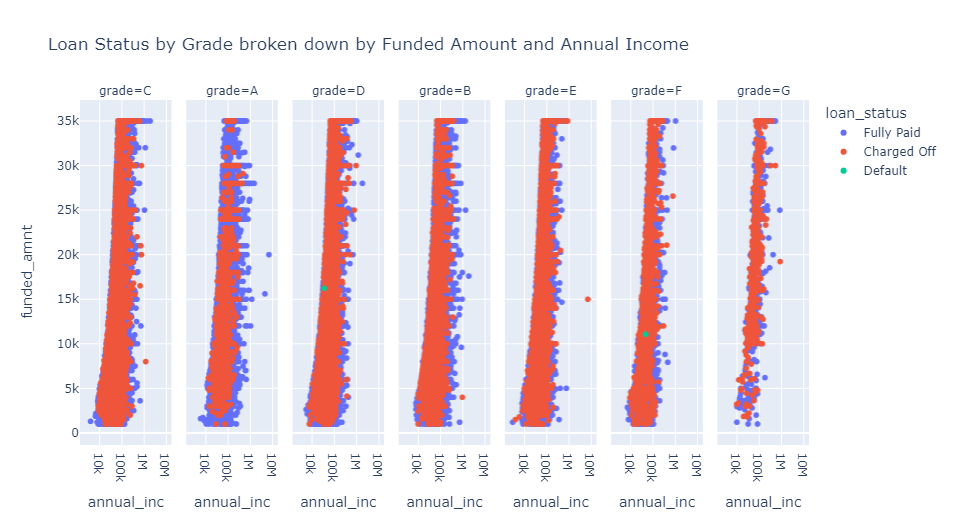


Next, we wanted to look at the proportion of fully paid and charged off loans broken down by the applicant’s grade and loan amount. We found that the lowest grade has the highest loan amounts, interest rate and funded amount for charged off loans. Better grades have lower interest rates and have less funding. This is indicative of LendingClub algorithm optimizing net annualized return for investors without accounting for risk properly. We can see that they are prioritizing loans with high interest rates to receive higher loan amounts. The following boxplots depict these findings:



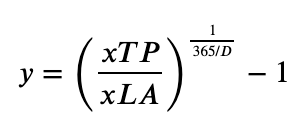


Next, we wanted to take a look at how applicants’ annual income, the funded amount, and grade correlate to the loan status. From the below scatterplot, we can see that



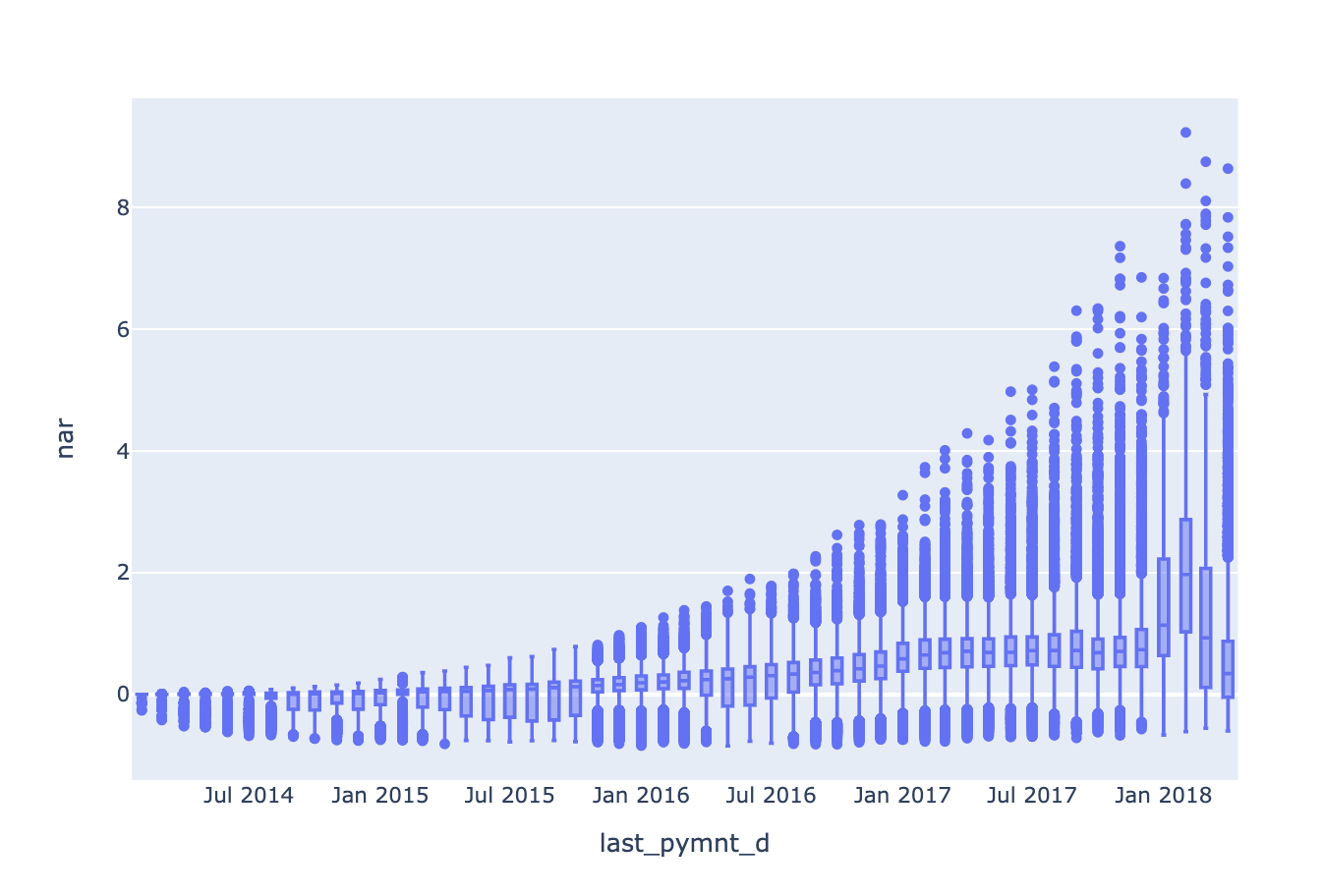
Net Annualized Return Calculation

Net Annualized Return (NAR), according to LendingClub, is “an annualized measure of the rate of return on the principal invested over the life of an investment. NAR is based on the actual Borrower payments received each month, net of service fees, actual charge off amounts and recoveries.” We added a column to for NAR to our dataframe using the following calculation:

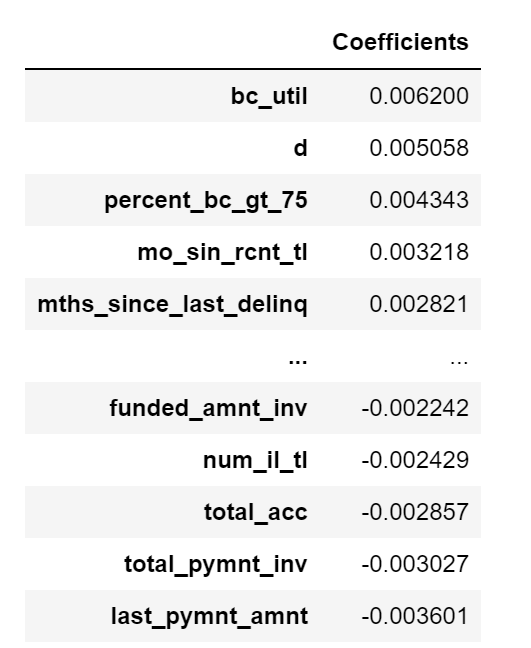


**(Total Payment / Funded Amount) \*\* (1 / (365/ days)) - 1**

We then plotted NAR dependant on the last payment date. From the plot below, you can see that NAR began to decrease in April 2018



Model: NAR - Linear Regression



Linear Regression is used to calculate the relationship between the numerical financial information and the dummy variables created for loan status, loan purpose, address state, and application type and the calculated NAR column. This model assumes that there is a direct correlation between the x and y variables and can be represented with a straight line. To measure accuracy of this model we determined the mean squared error and r squared values. The coefficients describe the mathematical relationship between each independent variable and the dependent variable. Below are the MSE and R² values and on the left, a data frame with the coefficients for each independent variable.

| MSE | R² |
| --- | --- |
| 4.515035530271146e-27 | 1.0 |

We found that it is difficult to accurately predict the NAR value through a regression model. Though the R² value is 1, meaning that all variance in our data is explained through the independent variables, we understand that there is more that goes into calculating net annualized return.

Model: PCA

The Principal Component Analysis is used to map standardized features into a low dimensional space where they are separated by maximum variance. Dimensionality reduction keeps relationships between variables, but changes the dimensionality by producing a series of linear combinations of the data where each linear combination is uncorrelated with the others. This allows us to look at the data in a space where they are orthogonal to each other. PCA is used extensively in business and finance for risk analysis. By creating a principal component representation of the portfolio of borrowers, an investor can analyze the weights of different borrowers to better understand the percentage return on a statistical risk factor. From the PCA constructed using all borrowers in the dataset, the negative values in the first principal component correspond to borrowers with lower interest rates and lower debt to income ratios, indicating lower risk. This makes sense as the grades with negative values range from A-C. The positive grades have higher interest rates, higher debt to income ratios indicating higher risk. This also makes sense, as the positive grades correspond to grades F and G. Applying PCA allows the data to be clustered by matching systematic risk and return characteristics without any prior knowledge of their fundamentals.

The first step in model building was to identify variables with the strongest correlation to the target variable “charge off. ” Logistic regression and random forest models used these features as independent variables so we could better understand the feature importances and identify the coefficient weights that indicate higher probability of being charged off. From these baseline models, it was understood that higher values of debt-to-income, interest rate, and annual income had higher probabilities of being charged off. The second step was to fit all of the financial information to Support Vector Machines, Gradient Boosting, K-Nearest Neighbors, and Logistic Regression.

Model:SVM

Support Vector Machines are a powerful supervised learning classification algorithm. The SVM finds a hyper-plane creating a boundary between the types of data and maximizes that boundary. We ran three SVMs using the Linear Kernel, Gaussian Kernel, and Polynomial Kernel. We first ran the SVMs using all of the numeric columns in our dataset with the target variable being ‘charged\_off’. We used the train\_test\_split method from sklearn to split our data with a test size of .33. The Linear Kernel and Gaussian Kernel performed the same with the following results:

| Accuracy | F1 | Precision | Recall |
| --- | --- | --- | --- |
| .87 | .53 | 1 | .36 |

The Polynomial Kernel performed slightly worse:

| Accuracy | F1 | Precision | Recall |
| --- | --- | --- | --- |
| .84 | .31 | .99 | .18 |

Model: Gradient Boosting

Gradient Boosting combines weak ‘learners’ into a single strong learner iteratively. Each iteration is weighted based on whether misclassification increases in the current iteration, thus higher misclassification has higher weight. The weight indicates the likelihood each case is selected again in the iterative sample. The focus on the misclassification count is what ultimately optimizes model performance.

| Accuracy | F1 | Precision | Recall |
| --- | --- | --- | --- |
| .99 | .96 | 1.00 | .93 |

Model: KNN

K-Nearest Neighbor (k-NN) reads in all training examples, and immediately makes predictions based on the similarity between the test example and all training examples with the majority-voted category label in the k nearest training examples. The decision boundary has no predefined shape and there are no assumptions made about independence. Given that kNN is sensitive to noisy training data and works best when all attributes are relevant to prediction, the algorithm likely was unable to filter some of the inconsistencies to make predictions.

| Accuracy | F1 | Precision | Recall |
| --- | --- | --- | --- |
| 0.83 | .45 | .61 | .35 |

Model: Logistic Regression

Logistic regression performed the best of all of our models, indicating that probability algorithms work best with our data. Ranking features by highest coefficient weights descends from funded\_amnt (total amount funded for the loan), total\_rec\_int (the interest received to date), revol\_bal (the amount of credit the borrower is using relative to all available revolving credit), total\_il\_high\_credit\_limit (the total installment/high credit limit), and the installment (​​the monthly payment owed by the borrower if the loan originates). Logistic regression has a discrimination threshold (the cutoff imposed on the predicted probabilities for assigning observations to each class) that would allow Lending Club decision-makers to adjust cut-off values for installments based on this model. The logistic regression supports the initial analysis we made, in that higher loan amounts are given to lower grades without accounting for risk factors.

| Accuracy | F1 | Precision | Recall |
| --- | --- | --- | --- |
| 1.00 | .96 | 1.00 | .93 |

**Conclusion**

LendingClub has discontinued issuing notes and ceased all new loan accounts on their website as part of their restructuring plan. As of December 31, 2020, LendingClub is no longer operating as a peer-to-peer lending operation. From our analysis, we have uncovered that LendingClub was scoring their applicants based on Net Annualized Return, rather than using an accurate risk assessment per applicant. Our models all proved to be accurate in predicting if an applicant will charge off their loan based on the independent variables mentioned in this report. Our highest performing model was Logistic Regression after applying a step-wise approach and adding dummy variables for address state, employment title, and grade.

Reference:

https://www.bloomberg.com/news/features/2016-08-18/how-lending-club-s-biggest-fanboy-uncovered-shady-loans