CAPSTONE PROJECT 1
Lending Club Loan Data

Paul Duehlmeyer
January 5, 2018

1. Introduction

a. About

Will this investment work out for me? Is the first question everyone asks when they are about spend money, even when the costs are relatively low. This project will use Lending Club data about loan applicants to predict which loans will be quality investments

b. Problems/Goals

i. Problem: When picking loans, how do I know which loans will go into default or which will be paid. What features from loan application data predict the quality of a loan? E.g. income, home ownership, etc.
ii. Goal: Predict which loan applicants would have the lowest risk of defaulting.

c. Client

i. Any person that invests through lending club's platform for peer to peer lending.
ii. It may possibly be used to indicate loan quality in other peer-to-peer or loan situations

2. Data Cleaning

a. Dataset

The dataset includes a sqlite database with 857383 rows with 75 columns of features.

b. Data Wrangling

i. Data types

There were several columns that pandas had considered text, but were numerical values, which I cast as floats. There were additional columns that contained percent sign symbols. I removed the symbols and cast the values as floats. There was a column for the length of employment of the person applying for the loan. The values were text indicating less than a year, 1-9 years, or more than 10 years. I converted these to float numbers using '.map'. A column for "home_ownership" contained text data with several categories, and I used '.map' to create two groups for people who owned a home or people who rented.

The most important thing I did to clean the data was to examine the "loan_status" and determine which values would be considered a Good loan and which values were considered bad. This is important because it is what will determine the quality of the loan. The were 11 values, which I had to map to a "good" or "bad" status. Here is how they

were mapped: ' Current': True, 'Fully Paid':True, 'Charged Off': False, 'Default':False, 'Late (31-120 days)':False, 'In Grace Period':False, 'Late (16-30 days)':False, ‘Does not meet the credit policy. Status:Fully Paid':True, 'Does not meet the credit policy. Status:Charged Off':False, None:False, 'Issued':True.

ii. Missing Data

Since the database contains over 887,383 rows and 75 columns of data and virtually all of the data is present for the majority of features, I dropped the features that had less than 80% of the values present. There were features that did not intuitively provide any value, such as “joint debt to income” ratio, and many applicants could probably not populate this field. After dropping these columns, there were 51 columns of data left. I used .dropna to drop rows with any missing data. The table size is now 816725 rows.

iii. Create new columns

There were several columns where the features had a categorical value. I used “.create_dummies”, which created new features for each category with a binary value. As a result of the new features created using “.create_dummies”, the dataset now has 104 columns.

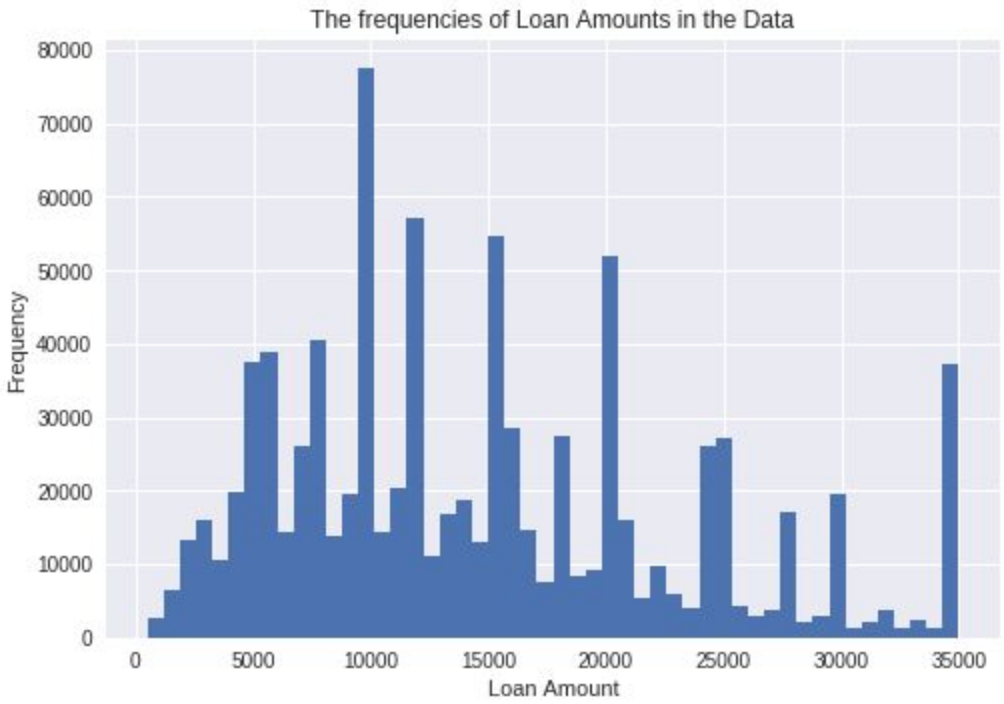
3. Exploratory Data Analysis

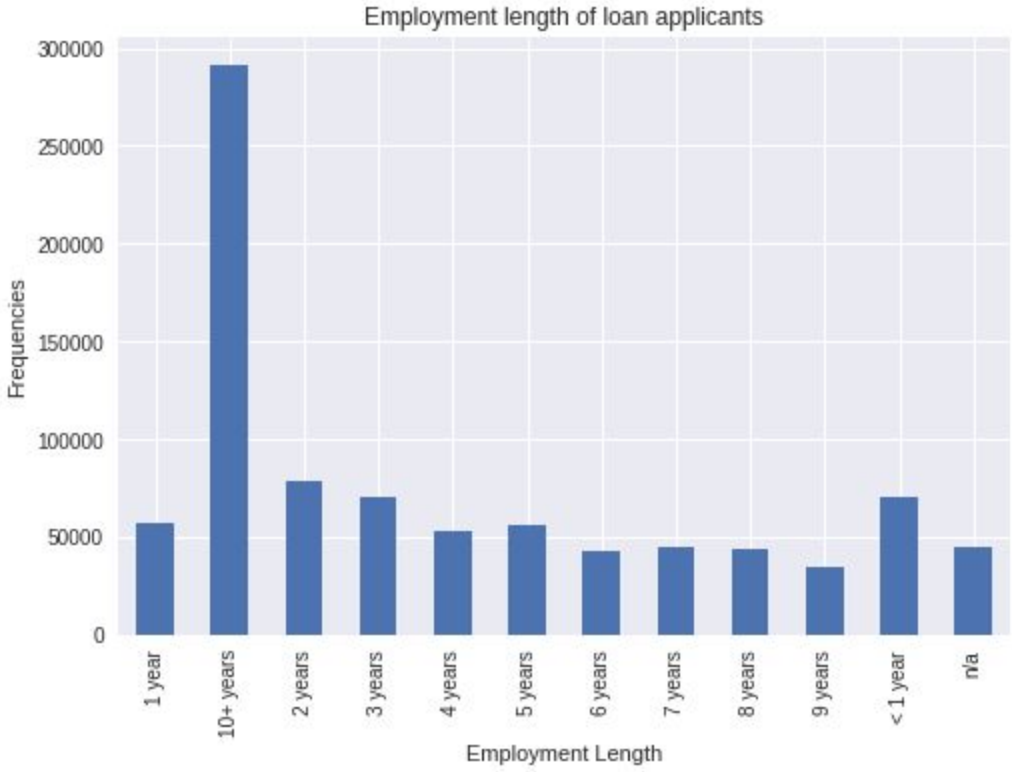
a. Features

Fifty-one of the features contain data that the applicant entered, such as accounts now delinquent, annual income, FICO range high, FICO range low, and so on. Fifty-three of the features contain binary values.

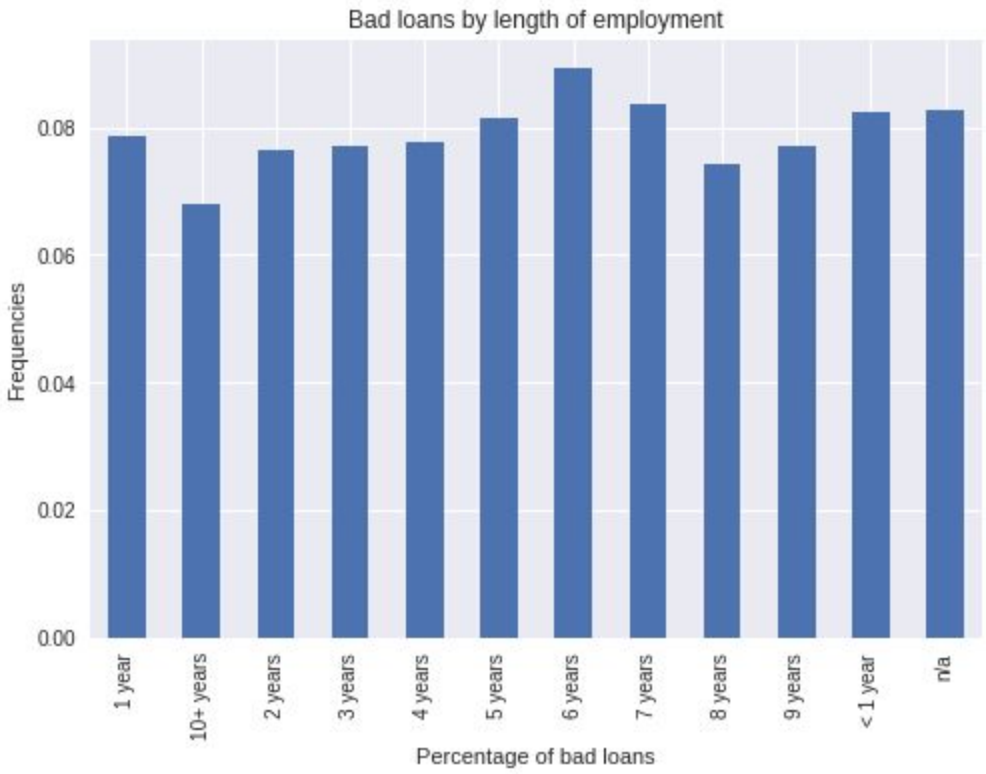
b. Preliminary Hypotheses

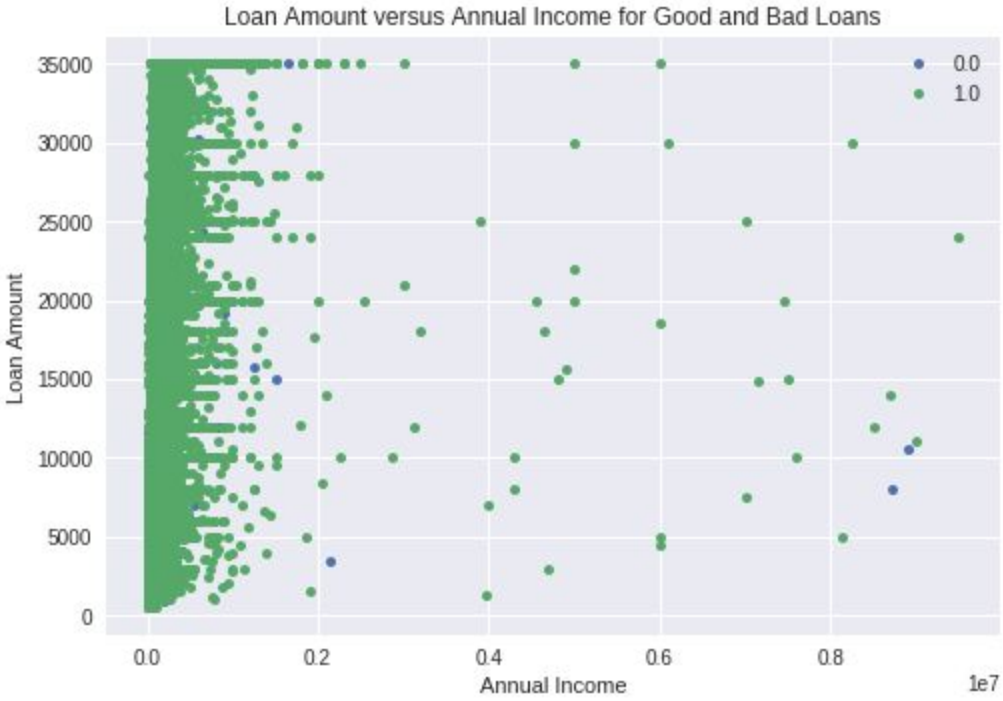
I had hypotheses that several features could predict whether a loan would default, such as a high correlation between default and applicant’s zip code. I believed by calculating and plotting the bad loan rates over income groups or loan amounts or another feature, I could find a feature that would predict a bad loan. After spending significant time attempting to visualize the data, I was not able to find any significant correlations or groupings. Below is a plot of loan amounts in the data set.





Just looking at the ratio of good loans and bad loans for each segment of the employment length, there is no trend that indicates the length of employment would predict a loan's quality.


Bad loans by length of employment
Frequencies
0.00
0.02
0.04
0.06
0.08
1 year
10+ years
2 years
3 years
4 years
5 years
6 years
7 years
8 years
9 years
< 1 year
n/a
Percentage of bad loans


Loan Amount versus Annual Income for Good and Bad Loans
Loan Amount
0
5000
10000
15000
20000
25000
30000
35000
0.0
0.2
0.4
0.6
0.8
1e7
Annual Income
0.0
1.0

c. Distributions:

This data contains 817,725 rows of data. Of these rows 760,344 are loans in good standing and 56,381 rows are loans in bad standing.

4. Machine Learning

a. Model the Data - Imbalanced Data

Since the data is imbalanced, RandomOverSampler, ADASYN (Adaptive Synthetic Sampling Approach for Imbalanced Learning), and SMOTE (synthetic minority oversampling technique) are used to balance over sample the minority class. I will test out which sampling method is best for modeling.

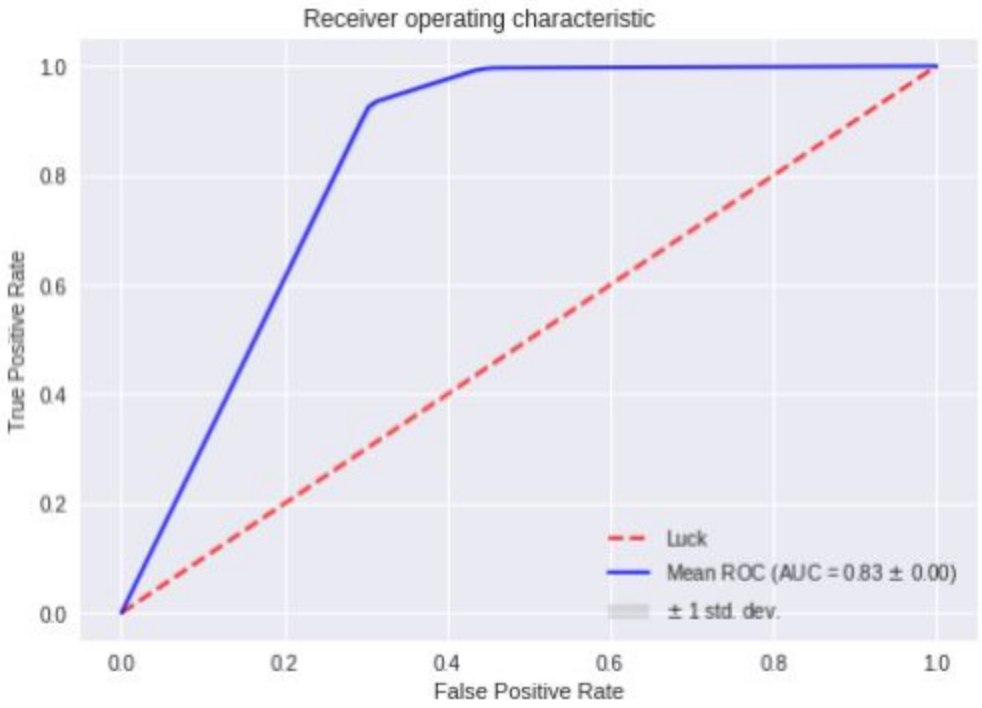
The area under the receiver operating characteristic curve (ROC AUC) is used to score the accuracy of each sampling method. As shown below, the resampled data yields the highest scores across all three classifiers. For this model, I will use SMOTE resampled data and tune the logistic regression model for the best parameters.

	ADASYN	RandomOverSampler	SMOTE
Gradient Boosting Machine	.621	.795	.695
Random Forest	.820	.735	.803
Logistic Regression	.738	.815	.850

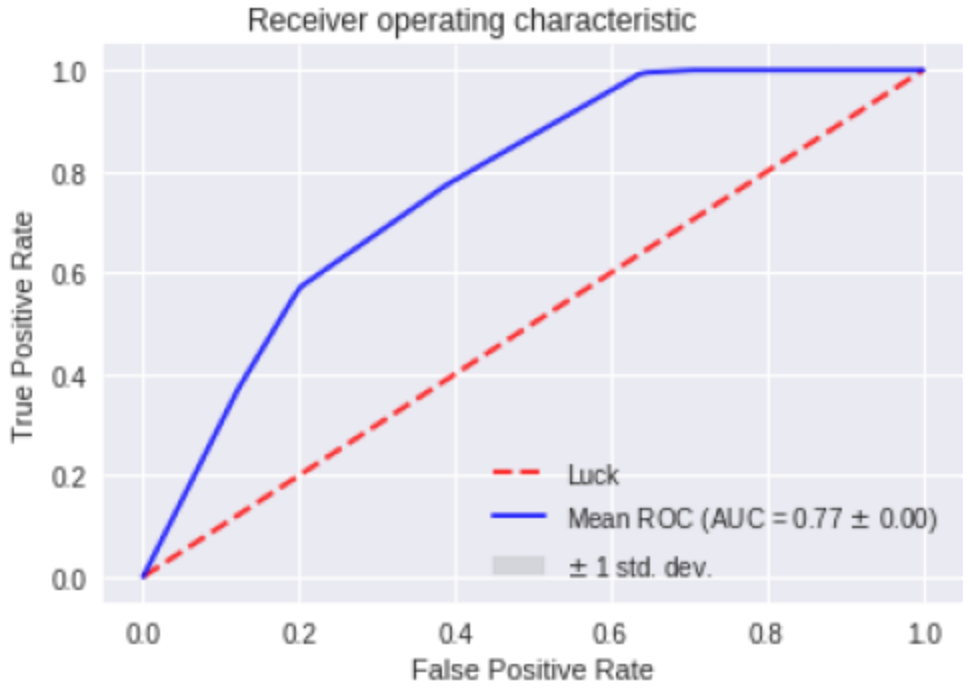
b. Model the Data - Resampled Data

Overall, the three models performed reasonably well, with precision scores of above 0.7 in most cases. The Logistic Regression algorithm with the SMOTE resampling method implemented performed the best compared to the Gradient Boosting Machine and Random Forest.

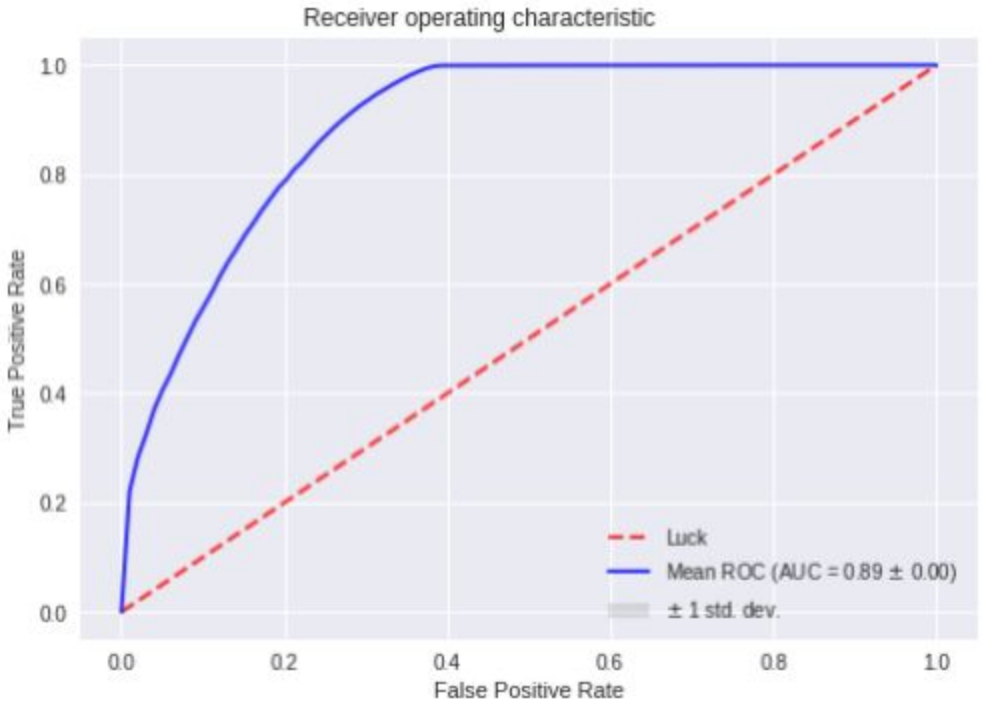
i. Random Forest



ii. Gradient Boosting Machine



iii. Logistic Regression



c. Hyperparameter Tuning - Logistic Regression

Using Python's GridSearchCV method, I checked for the best parameters of C to tune the model for the optimal complexity and fit.

Table comparing the scores between default parameters and tuned parameters for the model.

	ROC AUC	Test Accuracy
Default	.89	.850
Optimized Parameters	.89	.849
Improvement	0	-0.001

Default Parameters - Precision / Recall Table:

	Precision	Recall
Bad Loan	.69	.65
Good Loan	.97	.98
Average	.95	.96

Optimal Parameters - Precision / Recall Table:

	Precision	Recall
Bad Loan	.63	.65
Good Loan	.97	.97
Average	.95	.95

The optimal parameters from GridSearchCV actually have marginally the same scores in Precision and Recall as the default parameters. The default parameters for a model provide adequate performance and changing the 'C' value does not significantly change the performance of the model.

5. Conclusion

a. Summary

The goal of this project was to find a way to predict whether an applicant would default on a loan that was issued through lendingclub.com. Although, I was not able to find clear relationships through visualizing the data, I was able to develop a model that can very accurately predict applicants that will not default on their loans.

After trying a Gradient Boosting Machine, Random Forest, and Logistic Regression algorithm, I was able to use a logistic regression algorithm to very accurately predict which applicants would be good loans. The model does not predict bad loans as precisely and the recall score is quite low; however, a model that can guarantee a good applicant is acceptable for customer who wants to feel more secure about their investment in a loan.