Rushabh Shah (1893142)
CSC 478: Final Project Report
Due: Tuesday March 19, 2019

Lending Club Loan Analysis: Final Report

Executive Summary

For the Lending Club loan data, a K-Nearest Neighbor (KNN) classifier model was developed to classify the statuses of the loans in the Lending Club dataset. The data was manually explored, cleaned, reduced/scaled and visualized. All of this was done before pre-processing, creating a KNN-classifier model and scoring that model for how well it fit.

The Lending Club Dataset

The Lending Club loan dataset contains data on all loans issued from 2007-2015. The dataset contains all relevant data regarding each issued loan. Each loan contains information regarding the status of the loan, with the status being any one of seven statuses. The seven loan statuses include *Charged Off, Current, Default, Fully Paid, In Grace Period, Late (16-30 days), Late (31-120 days).*

The original dataset contains 887,383 row and 75 columns (or attributes), but I reduced the original dataset to 5% of the original size. I reduced the size of the dataset to make the analysis step more manageable because at the original size, the application crashed at every attempt at analysis. I furthered reduced the size of the dataset by removing categorical variables with high cardinality. In addition to these two pre-processing steps, I replaced all NaN values with zero's because I did not want to remove rows with NaN values.

How the KDD Process was Applied

For my analysis, I applied the KDD process using seven steps. The first step was the loading step, where I loaded all the necessary libraries and uploaded the loan data CSV files for each of the years from 2007-2015. The second step in the KDD process was the exploratory analysis step. In this step I attempted to run some descriptive statistics before the application crashed, so I reduced the dataset by 50% and attempted again.

Since the application crashed again, I reduced the dataset to 5% and ran some descriptive analytics to determine my target variable. I also checked how many attributes were in the dataset and determined which variables had the highest cardinality values. As the last part of the exploratory analysis step, I visualized the target variable's distribution with a histogram, which showed a bell curve distribution.

The third step in applying the KDD process was the data cleaning step, in which I removed the categorical variables with the highest cardinality values. The fourth step in took in applying the KDD process was the pre-processing and transformation step. In this step, I removed the target variable, and transformed the categorical variables into a matrix model using one-hot encoding. As part of this step, I also replaced all NaN values with zeros since I did not want to remove rows that contained NaN values, and I scaled continuous variables using min-max. Min-max was replaced all continuous variables with values scaled to values ranging from 0 to 1, so that the coefficients are less impacted.

The fifth step in the KDD process that I applied was the partitioning step, where I partitioned the dataset into a training and a test set so that I can develop my model on one set and test it's fit on the second set. The sixth step was the modeling step, where I built and developed a K-Nearest Neighbor model for the Lending Club dataset. I built my model using Euclidian distance with 10 neighbor classes.

The final step in the KDD process that I applied was the scoring step, where I scored the classifier model by testing it on both the training set that it was developed on and on the test set which the model was not developed on. As part of this step, I also created both a classification report and confusion matrix, as well as calculated the r-square value's for both the training and test set scoring.

Tools Used for Specific Data Analysis Steps

To perform my analysis of the Lending Club loan data, I used Python 3 in a Jupyter notebook. For my analysis, I used the numpy, pandas, sklearn, and matplotlib libraries. From the sklearn library, I used the preprocessing, metrics, feature_extraction, neighbors and cross_validation libraries.

Results and Conclusion

The k-Nearest Neighbor classifier model that was developed, performed well on both the training and test sets. The classifier model had a r-square value of 83% on the training set and a r-square value of 80% on the test set. This means that the model explains 80% of the variability for the target variable in the test set and 83% of the variability in the training set. The classification table is seen in figure 1. As we see in the classification table, we had an overall precision of 77%, meaning 77% of our total positive predictions were predicted correctly.

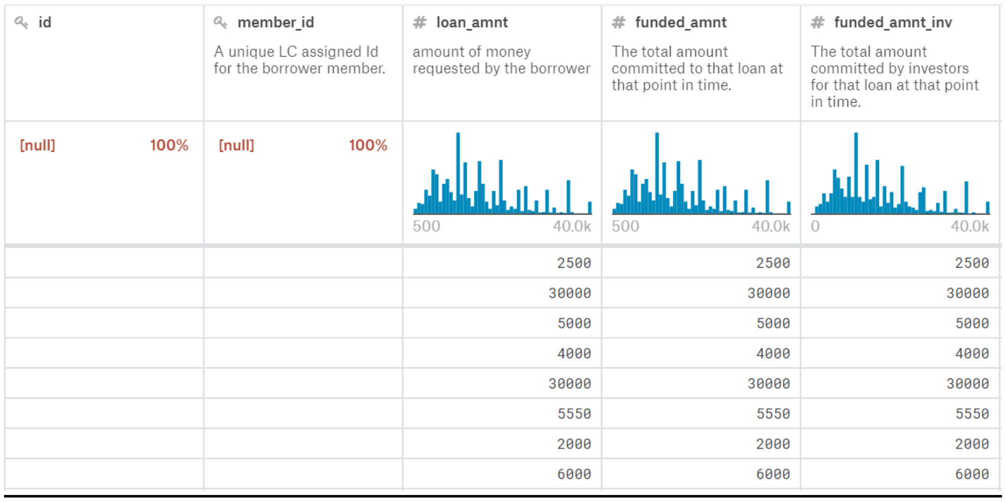
A precision of 77% shows that our classifier model is good at predicting. The average recall for our model was 80%, and the closer our recall value is to 100%, the more sensitive our model is at predicting positive predictions correctly in comparison to all correct positive predictions + all false negatives, meaning the ratio of all correct positive predictions to all observations in the dataset. The F1-score represents the weighted average of precision and recall, so an f1-score of 0.77 would be considered good.

Class	Precision	Recall	F1-Score	Support
Charged Off	0.69	0.12	0.2	663
Current	0.8	0.96	0.88	9101
Default	0	0	0	20
Does not meet the credit policy. Status:Charged Off	0	0	0	7
Does not meet the credit policy. Status:Fully Paid	0	0	0	24
Fully Paid	0.81	0.6	0.69	3069
In Grace Period	0	0	0	95
Issued	0	0	0	147
Late (16-30 days)	0	0	0	39
Late (31-120 days)	0	0	0	146
(average/total)	0.77	0.8	0.77	13311

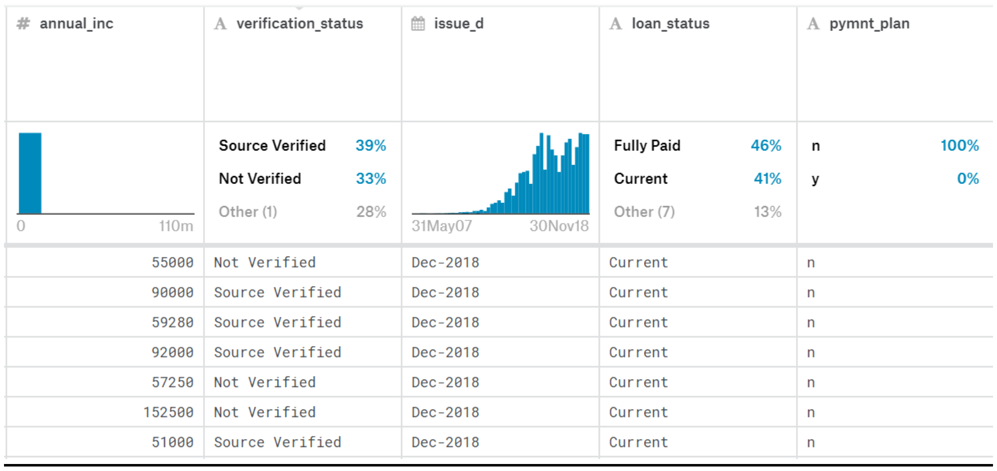
Figure 1: Lending Club Loan Data Classification Table

Appendix

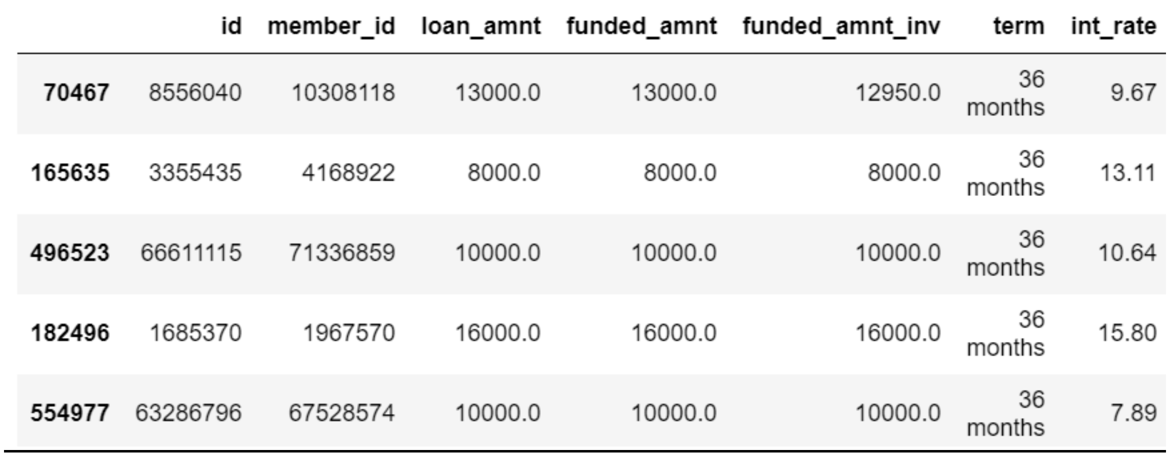
Data Samples:



id	member_id	loan_amnt	funded_amnt	funded_amnt_inv
	A unique LC assigned Id for the borrower member.	amount of money requested by the borrower	The total amount committed to that loan at that point in time.	The total amount committed by investors for that loan at that point in time.
[null] 100%	[null] 100%	500 – 40.0k	500 – 40.0k	0 – 40.0k
		2500	2500	2500
		30000	30000	30000
		5000	5000	5000
		4000	4000	4000
		30000	30000	30000
		5550	5550	5550
		2000	2000	2000
		6000	6000	6000

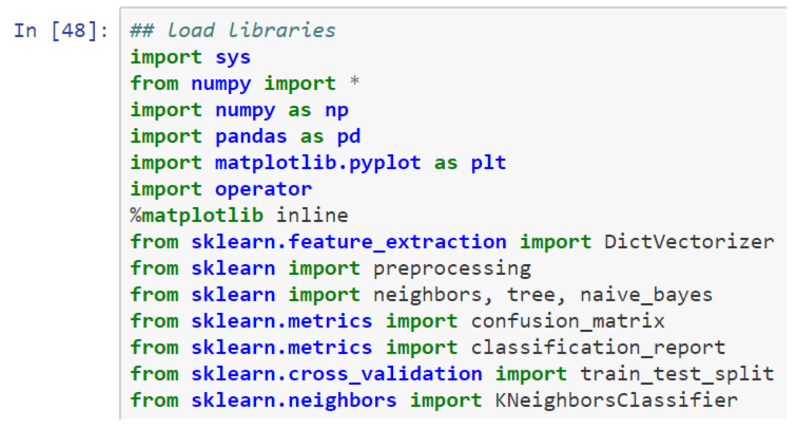


annual_inc	verification_status	issue_d	loan_status	pymnt_plan
0 – 110m	Source Verified 39% Not Verified 33% Other (1) 28%	31May07 – 30Nov18	Fully Paid 46% Current 41% Other (7) 13%	n 100% y 0%
55000	Not Verified	Dec-2018	Current	n
90000	Source Verified	Dec-2018	Current	n
59280	Source Verified	Dec-2018	Current	n
92000	Source Verified	Dec-2018	Current	n
57250	Not Verified	Dec-2018	Current	n
152500	Not Verified	Dec-2018	Current	n
51000	Source Verified	Dec-2018	Current	n

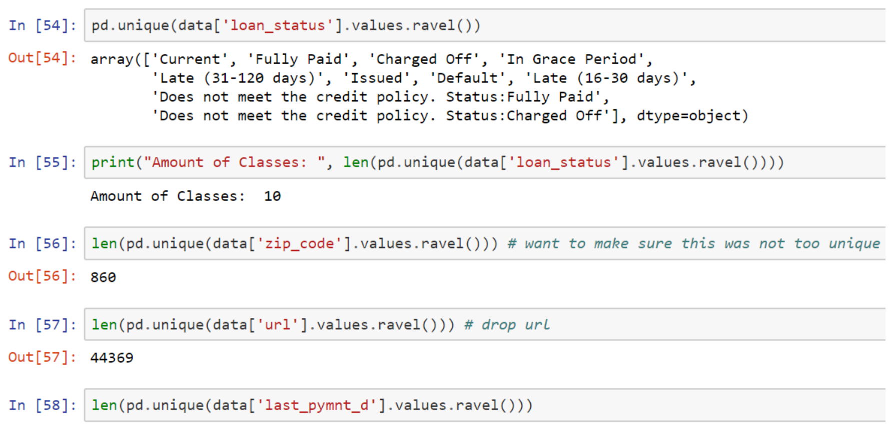


	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate
70467	8556040	10308118	13000.0	13000.0	12950.0	36 months	9.67
165635	3355435	4168922	8000.0	8000.0	8000.0	36 months	13.11
496523	66611115	71336859	10000.0	10000.0	10000.0	36 months	10.64
182496	1685370	1967570	16000.0	16000.0	16000.0	36 months	15.80
554977	63286796	67528574	10000.0	10000.0	10000.0	36 months	7.89

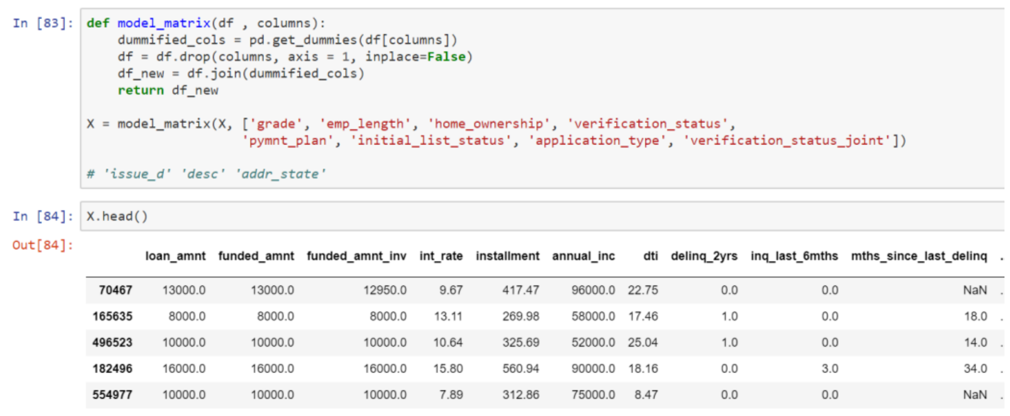
Fully Documented Code Segments:



```
In [48]: ## load libraries
         import sys
         from numpy import *
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import operator
         %matplotlib inline
         from sklearn.feature_extraction import DictVectorizer
         from sklearn import preprocessing
         from sklearn import neighbors, tree, naive_bayes
         from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report
         from sklearn.cross_validation import train_test_split
         from sklearn.neighbors import KNeighborsClassifier
```

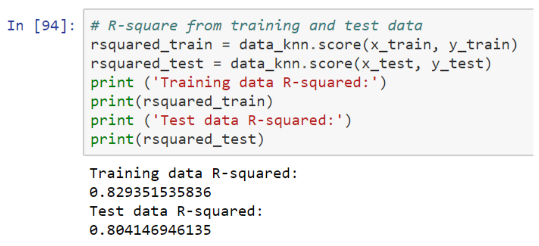


```
In [54]: pd.unique(data['loan_status'].values.ravel())

Out[54]: array(['Current', 'Fully Paid', 'Charged Off', 'In Grace Period',
                'Late (31-120 days)', 'Issued', 'Default', 'Late (16-30 days)',
                'Does not meet the credit policy. Status:Fully Paid',
                'Does not meet the credit policy. Status:Charged Off'], dtype=object)

In [55]: print("Amount of Classes: ", len(pd.unique(data['loan_status'].values.ravel())))

         Amount of Classes:  10

In [56]: len(pd.unique(data['zip_code'].values.ravel())) # want to make sure this was not too unique

Out[56]: 860

In [57]: len(pd.unique(data['url'].values.ravel())) # drop url

Out[57]: 44369

In [58]: len(pd.unique(data['last_pymnt_d'].values.ravel()))
```



```
In [83]: def model_matrix(df , columns):
             dummified_cols = pd.get_dummies(df[columns])
             df = df.drop(columns, axis = 1, inplace=False)
             df_new = df.join(dummified_cols)
             return df_new

         X = model_matrix(X, ['grade', 'emp_length', 'home_ownership', 'verification_status',
                              'pymnt_plan', 'initial_list_status', 'application_type', 'verification_status_joint'])

         # 'issue_d' 'desc' 'addr_state'

In [84]: X.head()

Out[84]:
```

	loan_amnt	funded_amnt	funded_amnt_inv	int_rate	installment	annual_inc	dti	delinq_2yrs	inq_last_6mths	mths_since_last_delinq	.
70467	13000.0	13000.0	12950.0	9.67	417.47	96000.0	22.75	0.0	0.0	NaN	.
165635	8000.0	8000.0	8000.0	13.11	269.98	58000.0	17.46	1.0	0.0	18.0	.
496523	10000.0	10000.0	10000.0	10.64	325.69	52000.0	25.04	1.0	0.0	14.0	.
182496	16000.0	16000.0	16000.0	15.80	560.94	90000.0	18.16	0.0	3.0	34.0	.
554977	10000.0	10000.0	10000.0	7.89	312.86	75000.0	8.47	0.0	0.0	NaN	.



```
In [94]: # R-square from training and test data
         rsquared_train = data_knn.score(x_train, y_train)
         rsquared_test = data_knn.score(x_test, y_test)
         print ('Training data R-squared:')
         print(rsquared_train)
         print ('Test data R-squared:')
         print(rsquared_test)

         Training data R-squared:
         0.829351535836
         Test data R-squared:
         0.804146946135
```

Programs Used

- *Python 3 (https://www.python.org/download/releases/3.0/)*
- *Jupyter Notebooks (https://jupyter.org/)*