**Final Project**

**Jasmine Dumas (1523905)**

**CSC 478: Programming Machine Learning Applications - Autumn 2016**

**Due: Tuesday, November 22, 2016**

**Executive Summary**

A K nearest neighbor classifer was developed for the lending club loan data to classify loan status from issued loans from Lending Club. The data was sampled, cleaned, visualized, pre-processed, modeled and scored on unseen data.

**Details on the data set used**

The original data set was downloaded from Kaggle, as an aggregate of issued loans from Lending Club through 2007-2015. Lending Club is a US peer-to-peer lending company. The original data set contains 887383 rows and 75 columns.

**Design decisions**

Due to computing power on my Macbook Pro, I choose to reduce (sample) the data to perform the data analysis to 5% of the original. I also choose to perform some pre-processing by removing categorical variables with high cardinality. I also chose to impute NaN values to zero, as the other option of removing slect rows with NaN would results in eliminating the entire data set.

**Description of how the KDD process was applied**

The steps I took during this project were:

1. Data Acquistion: I loaded the necessary libraries and download the Zip package containing the CSV file from Kaggle. After viewing the data and its shape I took a random 5% of the data to perform the analysis on.
2. Exploratory data analysis: During this step I perform some descriptive analysis and determined the target variable. I also explored how many classes were in the target and a selection of other possibly problamatic (high cardinality) variables. I also visualized the target variable in a histogram which is a good technique for understanding the distribution of the data to assist in parameter tuning.
3. Data Cleaning: I dropped those high cardinaility variables during this step as a pre-cursor to the pre-processing step.
4. Pre-processing & Transformation: I removed the target variable from the entire data set and transformed the categorical variable into a model matrix with one-hot encoding. This is sometimes the requirements for certain algorithms to process the data in a sparse matrix format. Other statistical software such as R, automates this step when generating models. I imputing the missing values in the data to 0. I scaled the contiuous variables using min-max normalization which transforms values from 0 to 1 to prevent variables on different scales heavily impacting the coefficients.
5. Data Partition: I partitioned the pre-processed data into a training and test data set.
6. Modeling: I built a k-NN classifier model, using 10 neighboor classes and the euclidean distance.
7. Evaluation: I scored the classifier on unseen test data and calculated the R squared values for both the training and test data. A confusion matrix and classification report were conducted.

**Tools used to perform specific data analysis tasks**

I used python 3 in a jupyter notebook to perform this data analysis. The libraries used were: numpy, pandas, matplotlib, and sklearn. From the sklearn library I also used: feature\_extraction, preprocessing, neighbors, metrics, cross\_validation.

**Results & Conclusion**

The classifier performed really well with a train R-squared value of 83% and the test R-squared value of 80% explained variance of the target. The classification report is below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
|  |  |  |  |  |
| avg / total | 0.77 | 0.8 | 0.77 | 13311 |