Debt Portfolio Capstone

Author: Mark Yingling

[1 Problem Statement 3](#_Toc527451191)

[2 Gather Data 4](#_Toc527451192)

[3 Exploratory Data Analysis (EDA) 5](#_Toc527451193)

[3.1 Data Dictionary 5](#_Toc527451194)

[3.2 Data interpretation and the workflow 6](#_Toc527451195)

[3.3 Feature Engineering 6](#_Toc527451196)

[4 Model Pre-processing 7](#_Toc527451197)

[4.1 Scaling Data 7](#_Toc527451198)

[4.2 Leave-One-Out Cross Validation 7](#_Toc527451199)

[4.3 SMOTE (Synthetic Minority Over-sampling Technique) 7](#_Toc527451200)

[4.4 GridSearch 7](#_Toc527451201)

[5 Model the data 8](#_Toc527451202)

[5.1 Logistic Regression 8](#_Toc527451203)

[**5.1.1** Confusion Matrix Evaluation Metrics 8](#_Toc527451204)

[**5.1.2** ROC Curve 8](#_Toc527451205)

[**5.1.3** Cross Validation Score 8](#_Toc527451206)

[**5.1.4** IBM Watson Studio Model Evaluation 8](#_Toc527451207)

[5.2 DecisionTreeClassifier 8](#_Toc527451208)

[**5.2.1** Confusion Matrix Evaluation Metrics 9](#_Toc527451209)

[**5.2.2** ROC Curve 9](#_Toc527451210)

[**5.2.3** Cross Validation Score 9](#_Toc527451211)

[5.3 RandomForestClassifier 9](#_Toc527451212)

[**5.3.1** Confusion Matrix Evaluation Metrics 9](#_Toc527451213)

[**5.3.2** ROC Curve 9](#_Toc527451214)

[**5.3.3** Cross Validation Score 9](#_Toc527451215)

[**5.3.4** IBM Watson Studio Model Evaluation 10](#_Toc527451216)

[5.4 GradientBoostingClassifier 10](#_Toc527451217)

[**5.4.1** Confusion Matrix Evaluation Metrics 10](#_Toc527451218)

[**5.4.2** ROC Curve 10](#_Toc527451219)

[**5.4.3** Cross Validation Score 10](#_Toc527451220)

[**5.4.4** IBM Watson Studio Model Evaluation 10](#_Toc527451221)

[5.5 KNeighbors 10](#_Toc527451222)

[**5.5.1** Confusion Matrix Evaluation Metrics 11](#_Toc527451223)

[**5.5.2** ROC Curve 11](#_Toc527451224)

[**5.5.3** Cross Validation Score 11](#_Toc527451225)

[6 Evaluate the Model 12](#_Toc527451226)

[7 Risks and Limitations 13](#_Toc527451227)

[8 Answer the Problem 14](#_Toc527451228)

# Problem Statement

Debt is a large problem in the United States. Many people have fallen into large amounts of debt much of which have a staggering interest rates attached to it. A significant number of these people have chosen not to repay this debt. This can be attributed to any number of reasons (loss of employment, other costs taking precedence, with the interest being so high they don't see any reduction in the principle amount, etc…) After a period of time trying to collect on the debt creditors will write this debt off and sell it in hopes of recouping some of what is owed. The process by which debt is sold comes in the form of debt portfolios i.e., a number of different debtors are bound together and sold as a percentage of the total amount owed. This could me pennies on the dollar, but the amount paid varies on the amount of information given to the buyer about the debtors included in the portfolio. For instance, all statements and correspondence each debtor has had with the selling creditor. The more data provided the more the portfolio will cost, however the more data you have the better you will be able to predict if someone will eventually repay what is owed.

Getting people who have a demonstrated a propensity of ignoring their debts to pay is the key to success when purchasing portfolio s of debt. How then do we do this? It is our belief that people in this position would like a way out. Either to just have the load of the debt off their shoulders, or to get their credit back so they my buy a house, car, etc… This was implemented successfully with this portfolio by forgiving the interest and fees currently associated with the loan and reducing the interest rate to prime plus 3 for debtors meeting the following criteria:

* Pass a bankruptcy check
* Lives in the Denver metro local
* Remaining balance < 15k
* Didn’t look at interest rates of 8%
* Married and own their home

Can we accurately identify if the portfolio is good one to purchase by determining the Likelihood of someone repaying their debt?

* What were your findings?
* What risks/limitations/assumptions affect these findings?

# Gather Data

We have a portfolio of debt that was purchased on October 24, 2008. This portfolio is comprised of 42 debtors with a total debt amount of $182,222.36. The cost of the portfolio was $6,000.

The data came in the form of two paper spreadsheets. I used Abbyy FineReader, which is an optical character recognition (OCR) piece of software that allowed me to read the paper documents into a .csv file. From there I needed to separate several columns that the software merged together because of several reasons; some of the cells were highlighted, there were several hand written notes, and the pages were taped together so that the rows could be seen with all the columns. I also needed to fix several of the date, name, and address fields for the same reasons as above.

# Exploratory Data Analysis (EDA)

Upon the initial look at the two spreadsheets it became clear that they were the same group of accounts. When the portfolio was purchased two people were doing the analysis. I’m guessing that the two spreadsheets were each individual’s work product. The first data set had less missing data so we chose to work with it.

## Data Dictionary

There are 42 rows of data with 28 columns

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| Account Number | int64 | Issuers account number |
| PortID | object | Issuer id |
| Remaining Balance | float64 | The remaining balance owed |
| Interests Fees | float64 | Interest and fees associated with the account |
| Date Opened | datetime64[ns] | Opening date for the card |
| Charge Off Date | datetime64[ns] | When account was written off by the issuer |
| Last Pay Date | datetime64[ns] | Last date a payment was made on the loan |
| Last Pay Amount | float64 | The amount last paid |
| Last Activity Date | datetime64[ns] | The date of the last payment |
| Interest Rate % | float64 | Interest rate of the loan |
| Issuer | object | The issuer of the credit card i.e., Chase, or some other bank |
| Merchant | object | This is the company that offers the card to its customers i.e., United Airlines Mileage Plus or the bank itself |
| FCFRA Date | datetime64[ns] | Fair Credit Reporting Act – This is the last time the issuer reported to the credit bureau. |
| OOS Date | datetime64[ns] | Out of Statute – Six years after the last payment made by the consumer |
| Account Type | object | Type of debt portfolio |
| Last Name | object | Debtors last name |
| First Name | object | Debtors first name |
| Middle Name | object | Debtors middle name |
| Address | object | Debtors address |
| City | object | Debtors city of residence |
| State | object | Debtors home state |
| Zip | object | Debtors zip code |
| County | object | Debtors home county |
| SSN | float64 | Debtors social security number |
| Home Phone | float64 | Debtors home phone number |
| Employer Phone | float64 | Debtors employers phone number |
| Cycle | object | How “fresh the account is i.e., how many times did the issuer send it out to a collection agency prior to sale |
| Paid | int64 | Was the debt paid |

## Data interpretation and the workflow

In doing the manual analysis my SME took the following into account:

* Bankruptcy check
* Denver metro local
* Home owner weighted higher then renters
* Remaining balance < 15k
* Didn’t look at interest rates of 8%
* Look to see if they were married
* Looked at the cycle the loan was in
* Calls were also made to each of the debtors and questions were asked to determine their current financial situation.

## Feature Engineering

* Because the Out of Statute date is an important date I subtracted the date the portfolio was purchased from it to determine how many days remained until the loan was out of statute, and then added a column to the dataframe.
* The Cycle is also an important column in determining if someone will pay back the debt. I created dummy values for this column which generated three new columns; 'Cycle\_QUATERNARY', 'Cycle\_SECONDARY', 'Cycle\_TERTIARY'

# Model Pre-processing

Prior to fitting any models some pre-processing of the data will help build better models.

## Scaling Data

Feature scaling can vary your results a lot while using certain algorithms and have a minimal or no effect in others. Because many of the models I will be building are based on the distance between two data points. Not having scaled data will cause problems.

## Leave-One-Out Cross Validation

Because my data set is very small I incorporated the leave-on-out cross validation. Leave-one-out cross validation is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the function approximator is trained on all the data except for one point and a prediction is made for that point.

## SMOTE (Synthetic Minority Over-sampling Technique)

Because there is not a lot of data we need to over-sample the data to get more samples of the minority class. SMOTE is an oversampling method which creates “synthetic” example rather than oversampling by replacements. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen.

## GridSearch

Prior to fitting any models, I ran a grid search looking to fine the best hyper-parameters.

# Model the data

## Logistic Regression

The first model I tried was a logistic regression.

Based on all the scores I think that this might be a good model, even though the test set was 100%. The observations from the test set might come from the fact that there is only one observation in that set.

### Confusion Matrix Evaluation Metrics

I evaluated the model using model evaluation metrics such as accuracy, precision, and recall.

* Accuracy: Classification rate of for the training set was 80.82%. The test set scored 100%.
* Precision: How accurate is your model. In other words, you can say, when a model makes a prediction, how often it is correct. In this case, the training set scored 80.56% and the test was correct 100% of the time.
* Recall: Sensitivity number was 80.56% for the training and the test set scored 100%.

### ROC Curve

Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

* AUC score for the case was 0.936. AUC score 1 represents perfect classifier

### Cross Validation Score

The cross-validation score was 78.08%

### IBM Watson Studio Model Evaluation

I wanted to see how Watson Studio’s automated model builder would do with this data. It’s version of a Logistic Regression model gave a AUC score of .7, however the overall evaluation was “poor”.

## DecisionTreeClassifier

The next model I tried was a single decision tree classifier. I found that there were four important features for this model; “Interests Fees”, “Cycle\_Quanternary”, “OOS Days Left”, and “Last Pay Amount”.

Based on all the scores I'm suspicious of this model not being over fitted.

### Confusion Matrix Evaluation Metrics

I evaluated the model using model evaluation metrics such as accuracy, precision, and recall.

* Accuracy: Classification rate of 100% for both the training and test sets.
* Precision: How accurate is your model. In other words, you can say, when a model makes a prediction, how often it is correct. In this case, the training and the test sets were correct 100% of the time.
* Recall: Sensitivity number was 100% for the training and the test set.

### ROC Curve

Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

* AUC score for the case is 1. AUC score 1 represents perfect classifier.

### Cross Validation Score

The cross-validation score was 93.15%

## RandomForestClassifier

Our next model was a random forest. I found that there were eight important features for this model; “Interest Fees”, “Last Pay Amount”, “OOS Days Left”, “Remaining Balance”, “Cycle\_Quanternary”, “Cycle\_Tertiary”, “Interest Rate”, and “Cycle\_Secondary”.

Based on all the scores I'm suspicious of this model not being over fitted.

### Confusion Matrix Evaluation Metrics

* Accuracy: Classification rate of 100% for both the training and test sets.
* Precision: How accurate is your model. In other words, you can say, when a model makes a prediction, how often it is correct. In this case, the training and the test sets were correct 100% of the time.
* Recall: Sensitivity number was 100% for the training and the test set.

### ROC Curve

Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

* AUC score for the case is 1. AUC score 1 represents perfect classifier.

### Cross Validation Score

The cross-validation score was 87.67%

### IBM Watson Studio Model Evaluation

I wanted to see how Watson Studio’s automated model builder would do with this data. It’s version of a Random Forest model gave a AUC score of .5, however the overall evaluation was “poor”

## GradientBoostingClassifier

The next model was a gradient boost. I found that there were eight important features for this model; “Cycle\_Quanternary”, “Interest Rate”, “Last Pay Amount”, “OOS Days Left” ,“Remaining Balance”.

Based on all the scores I'm suspicious of this model not being over fitted

### Confusion Matrix Evaluation Metrics

* Accuracy: Classification rate of 97.26% for the training and 100% for the test. Because I only have one hold out for the test, so I might have gotten lucky
* Precision: How accurate is your model. In other words, you can say, when a model makes a prediction, how often it is correct. In this case, the training set was correct 94.74% of the time and the test set was correct 100% of the time.
* Recall: Sensitivity number was 100% for the training and the test set.

### ROC Curve

Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

* AUC score for the case is 0.996. AUC score 1 represents perfect classifier.

### Cross Validation Score

The cross-validation score was 87.67%

### IBM Watson Studio Model Evaluation

I wanted to see how Watson Studio’s automated model builder would do with this data. It’s version of a Random Forest model gave a AUC score of .833, also the overall evaluation was “good”

## KNeighbors

The final model used was K Nearest Neighbors. I used the following as features; “Cycle\_Quanternary”, “Cycle\_Tertiary”, and “Cycle\_Secondary.

Based on the ROC curve and the cross validation score this model did not perform very well, and would not be deployed.

### Confusion Matrix Evaluation Metrics

* Accuracy: Classification rate of 58.9% for the training and 100% for the test. Because I only have one hold out for the test, so I got lucky
* Precision: How accurate is your model. In other words, you can say, when a model makes a prediction, how often it is correct. In this case, the training set was correct 100% of the time so was the test set.
* Recall: Sensitivity number is 16.67% for the training set but was 100% for the test set.

### ROC Curve

Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate against the false positive rate. It shows the tradeoff between sensitivity and specificity.

* AUC score for the case is 0.595. AUC score 1 represents perfect classifier, and 0.595 represents a worthless classifier.

### Cross Validation Score

The cross-validation score was 41.1%

# Evaluate the Model

Because the data set is small evaluation is a bit tricky, however based on the cross-validation scores and the ROC curves I believe that all the models except for the K-Nearest Neighbors would do a good job at answering the problem statement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | | Recall | | Cross-Validation |
| Training | Test | Training | Test |
| Logistic Regression | 80.82% | 80.56% | 100% | 100% | 78.08% |
| Decision Tree | 100% | 100% | 100% | 100% | 93.15% |
| Random Forest | 100% | 100% | 100% | 100% | 87.67% |
| GradientBoosting | 97.26% | 100% | 100% | 100% | 87.67% |
| K Nearest Neighbors | 58.9% | 16.67% | 100% | 100% | 41.1% |

# Risks and Limitations

Would like to create a confidence column to weight the debtor based on questioning by the SME.

# Answer the Problem

Going back to the original problem statement; Can we accurately identify if the portfolio is good one to purchase? I believe that the model can give a very good indication as to whether a portfolio of debt will have enough debtors that will repay the debt to produce a good return on the investment.

End of document