Project Part III

Noah Keogh (njk6) S01264950

1 Data Preparation & Wrangling

Before I could train and test different models to achieve the business objective, I needed to prepare the data using different data wrangling techniques. The techniques outlined in the data proposal II will be carried out, so that the data is properly cleaned and ready for the training of each type of model. The dataset consists of $150,000$ entries, with each entry containing 11 features. The data does have a label for achieving the business objective which is called "SeriousDlin2yrs." This label is an indication of whether a given person / datapoint has experienced serious delinquency. This feature will be used as the label for any supervised machine learning model since the business objective is to identify whether accurate predictions can be made for determining if someone will experience financial hardship/serious delinquency. The results of this classification can then be used for assessing someone's credit worthiness. The dataset will also be looked at in an unlabeled manner for use in an unsupervised machine learning model to assess whether there appears to be any similarities or clustering in the financial tendencies of the people recorded in the dataset. The identification of shared financial traits in people will, therefore, be an additional business objective.

Before any data wrangling or transformation of the dataset was performed, the data was split into two sets of data: training and testing. 20% percent of the total dataset was reserved and set aside for testing the final model made, and the remaining 80% of the dataset will be used for training the models. A stratified shuffle split was used to split the data so that the distribution of each of the features would be captured in both the training and testing data. This type of split was used to help ensure that there would be less chance for bias since the training and testing datasets would be less likely to be skewed when compared to the original distribution of the dataset.

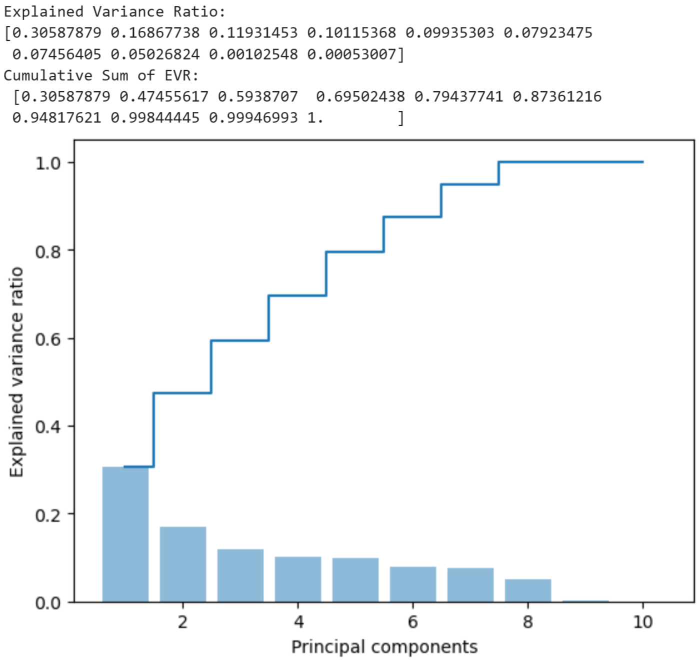
After splitting the data, the first step in preparing it was to fill the missing values that were present for some entries in the dataset. Two features in the dataset contained missing values. The monthly income feature was missing from $29,731$ entries and the number of dependents features was missing from $3,924$ entries. In order to fill these missing values, an imputer was made that would fill the missing entries of each feature with the median value of that feature in the training dataset. The imputer – that was made with the median value of each feature of the training dataset – will be applied to both the training dataset and the testing dataset.

After filling the missing values, it was determined that the number of samples for each label (1 = serious delinquency experienced, 0 = serious delinquency not experienced) was not balanced. The delinquency (1) label had $8,021$ samples in the training dataset and the non-delinquency (0) label had $111,979$ samples in the training dataset. It was determined that a large imbalance in the labels could lead to problems with model training. Therefore, it was determined that a random undersampling of the majority label (non-delinquency label) would be performed. Thus, the number of samples of each label were made equal such that there were $8,021$ delinquent and $8,021$ non-delinquent samples.

After making the labels balanced, a standard scaling procedure was performed on the training dataset. The reason the data was standardized is that if the different features have different scales this can have a negative effect on different models possibly giving more weight to one feature over another based solely on its scale. Not performing scaling could also have an impact on principal component analysis (PCA), since PCA is sensitive to the scale of the features.

After standardizing the features of the dataset, PCA was performed on the dataset to reduce the number of dimensions in the dataset. A large number of dimensions can lead to what is known as "the curse of dimensionality" which means that a larger number of dimensions may result in poorer performing models, and require exponentially more training samples than those with a fewer number of dimensions. It was determined that the threshold for an acceptable level of explained variance would be 90%. Thus, after performing PCA analysis, it was determined that 7 principal components achieved this level of variance. This is shown in Figure 1. Thus, it was determined that a PCA with 7 components would be used to transform and perform dimensionality reduction on the dataset.

After performing the operations above, it was determined that the data was now properly processed to move on to the data modeling phase. At this point, the missing values in the dataset had been filled, the labels had been balanced, the features scaled, and the number of dimensions reduced from 10 to 7.



```
Explained Variance Ratio:
[0.30587879 0.16867738 0.11931453 0.10115368 0.09935303 0.07923475
 0.07456405 0.05026824 0.00102548 0.00053007]
Cumulative Sum of EVR:
 [0.30587879 0.47455617 0.5938707  0.69502438 0.79437741 0.87361216
 0.94817621 0.99844445 0.99946993 1.        ]
```

Figure 1: Depicts the explained variance by each principal component.

2 Data Modeling

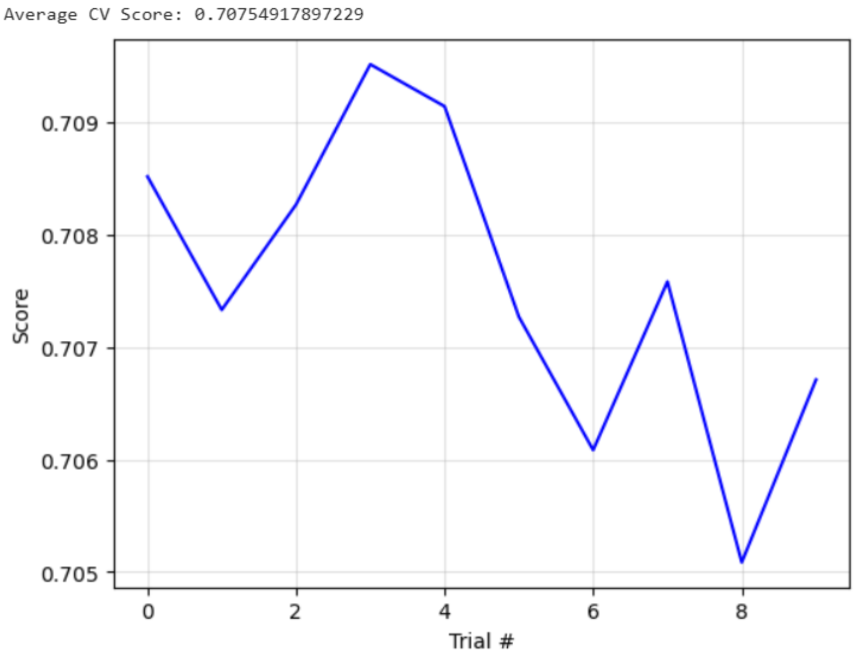
Several supervised and unsupervised models will be explored in order to determine which type of model may best achieve the business objective. In order to assess which model may be the best to further optimize, several rounds of training and testing with various parameters for each model will be tested against a validation dataset. More specifically, in order to properly assess the performance and accuracy of each model for achieving the business objective, nested cross validation will be used. Nested cross validation will allow me to test multiple hyperparameters for any given model and then score the model on a validation portion of the data. This will allow me to easily and compare the scores of each model and determine which model should be pursued and further optimized to best achieve the business objective. Models with low interpretability (such as neural networks) were not considered since the final model used to achieve the business objective must have high interpretability. The reason that it must have high interpretability is because since this model will be used in determining a person's credit score, the model must have high transparency in its classification decisions. The main reason we need high interpretability is to ensure the model is not biased in a particular manner in which it should not be.

2.1 Logistic Regression Model - Supervised Model

The first model that was tested was a logistic regression model. Since the business objective is to identify or classify people or data entries as being likely or not likely to experience a serious delinquency, the logistic regression model will be a good model to test. The logistic regression model can be used as a classifier to classify people as being at risk or not as risk. The model can also retrieve the raw prediction probabilities and can be used to identify the probability of someone experiencing a serious delinquency. Thus, not only will the model be able to perform the classification task, but it may also provide further insight into the probability that each person may experience delinquency.

To test the model, the "X_train_bal_pca" dataset was used. This dataset is a training dataset which had gone through the process of filling missing values, label balancing (through undersampling the majority class), standard scaling, and a PCA dimensionality reduction using 7 principle components, as described in the data preparation and wrangling section. The reason that this dataset was used is because logistic regression is not a model which is robust to imbalanced labels. Thus, balancing the labels was required in order to make sure the model was properly trained. In addition, to avoid the curse of dimensionality it was determined that training the logistic regression model on a dimensionality reduced dataset would be helpful in training an accurate model. The logistic regression model is not a very complex model, therefore, it is essential that the data it receives is not too complex. Thus, to allow the model a better chance at learning the dataset, it was determined that using PCA dimensionality reduction may allow the model to better learn, and since the dimensionality reduction still captures over 90% of the training dataset variance, the model may still have good predictive capabilities for the test dataset.

In order to determine the performance of the model trained on the training dataset described above, nested cross validation was used. For each trial of nested cross validation run, several different hyperparameters were tested in the inner cross validation loop and the optimal model for that trial was then selected and tested against a validation dataset in the outer cross validation loop. The hyperparameters that were tested were C values of 1, 10, or 100, and the solvers tested were lbfgs and liblinear. Ten trials were conducted with the results shown in Figure 2 below.



```
Average CV Score: 0.70754917897229
```

Figure 2: Depicts the nested cross validation of the logistic regression model.

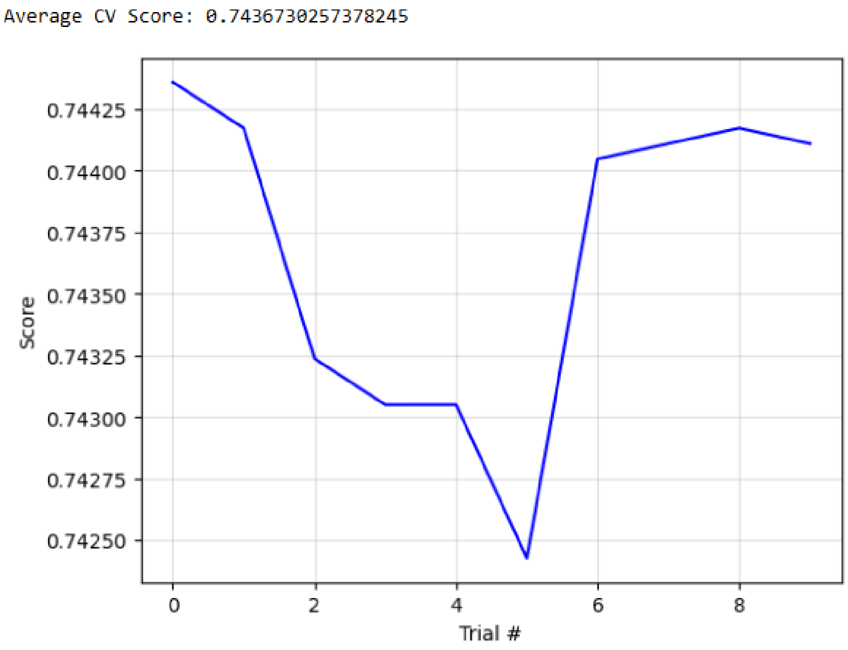
As can be seen from the graph above, the logistic regression model appears to be performing relatively well with an average accuracy of 70.75% over 10 independent trials. There is some small fluctuation between the accuracy of the independent trials, but this fluctuation is only around 1%.

2.2 Support Vector Machine (SVM) Model - Supervised Model

The second model tested was a support vector machine (SVM). Since the goal is to identify people who are likely to experience serious delinquency, the SVM model should work relatively well, since it can be used as a classification model. Since SVM tries to identify the optimal decision boundary between the two classes, my hypothesis was that it may be able to outperform the accuracy obtained by the logistic regression model.

To test the model, the ”X_train_bal_pca” dataset was used. This dataset is the same dataset that was used to train and test the logistic regression model. This dataset has gone through filling missing values, label balancing (through undersampling), standard scaling, and a PCA dimensionality reduction using 7 principal components. The main reason that this dataset was selected for training and testing a SVM model is because SVMs are not robust to imbalanced labels. This means that if a dataset was used that did not have balanced labels, the model trained may have a bias towards one label and, therefore, be more likely to derive incorrect classifications on the test dataset. The reason that it was determined that dimensionality reduction was required is because of the nature of how SVM models work. Similar to logistic regression models, SVM models draw a decision boundary to differentiate the two classes. Thus, the higher dimensional data that is present, the more considerations and complexity is required for the model to draw the boundary. This means that we can run into issues described as the ”curse of dimensionality.” The main issue is that as the number of dimensions increase the number of training samples required grows exponentially. Thus, since there is limited data, it is essential that the number of dimensions be reduced. This is the reason why the dataset used to train the model had PCA dimensionality reduction applied to it. In addition, SVM may be affected by features of different scales, therefore, it was essential that the data also be standardized.

In order to determine the performance of the model trained on the training dataset described above, the nested cross validation procedure was used. For each trial conducted, several different hyperparameters were tested in the inner cross validation loop. After performing a grid search CV comparing the different models obtained from training models on each of the hyperparameters, the model with the hyperparameters obtaining the highest score was used for the outer cross validation loop. The hyperparameters tested were a C value of 1 or 10, and a gamma value of 0.01 or 0.1, using rbf for the kernel.The C hyperparameter represents a trade off between misclassification and model complexity. Higher C values favor more complex models while lower values of C favor less complex models. Gamma is a parameter which defines how much influence a single training example has on the model. Gamma, therefore, controls the shape of the decision boundary with smaller gammas resulting in smoother curves whereas higher gammas leads to a less smooth and more complex decision boundaries. These two hyperparameters were selected for cross validation testing since they are essential to controlling the behavior of the SVM model. In the outer cross validation loop, the optimal model obtained was then tested against the validation split of the data. Since, the optimal model did not have access to this data during the training procedure, the accuracy result can give a good sense of how the model may perform on the testing data. The results of the nested cross validation can be seen in Figure 3.



```
Average CV Score: 0.7436730257378245
```

Figure 3: Depicts the nested cross validation of the SVM model.

As can be seen from the figure above, the SVM model appears to be performing quite well on the training data producing an average accuracy of 74.37% over 10 independent trials of nested cross validation. It appears that there is less fluctuation as compared to the logistic regression model. The SVM model only produced a fluctuation of 0.1% to 0.2% in accuracy.

2.3 K-means Clustering - Unsupervised Model

The third model tested was K-means clustering. This model (unlike the previous two models tested) is an unsupervised learning model. This means that the training data used for this model will not contain labels. Since, this model will not be trained using labels, there is a different business objective for the unsupervised model. Instead of trying to predict whether a given data point or person will experience serious delinquency, the business objective will be to cluster the data and determine whether there are any identifying characteristics of those that experience delinquency and those that do not. To perform this, the data will be clustered using the K-means clustering algorithm, without the use of labels, and after clustering is performed, each sample will be identified using its label to determine if any relationships can be drawn between the clusters and whether or not someone experiences serious delinquency. Thus, the goal is not to predict delinquency, but, rather, determine if there are any identifying characteristics of people who do experience delinquency.

In order to test the model, the ”X_bal_scaled” dataset was used. This is the dataset that was created after the missing values had been filled, the labels balanced by undersampling, and the features standardized. This is because a different type of dimensionality reduction would be performed on this dataset as compared to the dataset used in the previous two models. Instead of performing a dimensionality reduction with 7 principal components, PCA dimensionality reduction was performed using 2 principal components. The reason 2 principal components was used instead of 7 is because 2 principal components allow for easy visualization of the results of the clustering algorithm. In addition, 2 principal components still explains around 47.46% of the variance as can be seen from Figure 1.

After performing dimensionality reduction with two principal components, an elbow plot was generated in order to determine the optimal number of clusters for the K-means clustering algorithm. This elbow plot can seen below in Figure 4.

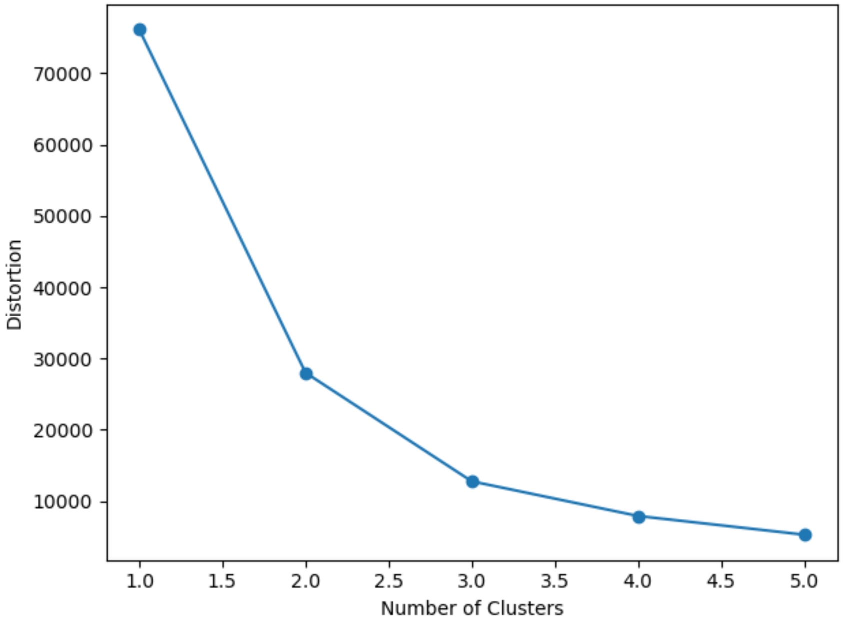


Figure 4: Depicts the elbow plot of the K-means clustering algorithm.

Based on the elbow plot above, it appears that the optimal number of clusters to use for K-means clustering is two clusters. This can be seen in the figure where two clusters produce the characteristic protruding ”elbow,” thereby, signaling that it is the optimal number of clusters to use. This is further shown from the plot which shows that there is a significant decrease in distortion from one to two clusters and then increasingly smaller decreases in distortion after the number of clusters increases past two. It could also be argued that three clusters could be used since there is still a significant decrease in distortion from two to three clusters. However, based on visual inspection, it appears that two clusters is more likely for the dataset.

Based on the elbow plot, the K-means clustering algorithm was run using two clusters (K=2). The model initialized the points using the k-means++ algorithm and performed 10 iterations using different initial centroids before selecting the optimal model. In addition, a limitation of 300 iterations and tolerance of $1e-4$ was used. After running the model with these parameters, the results obtained can be visualized in Figure 5.

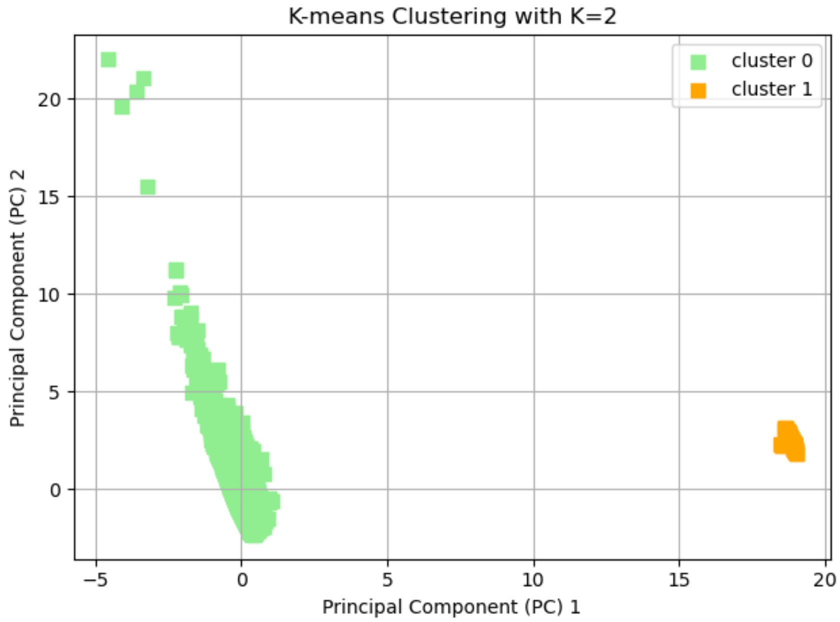
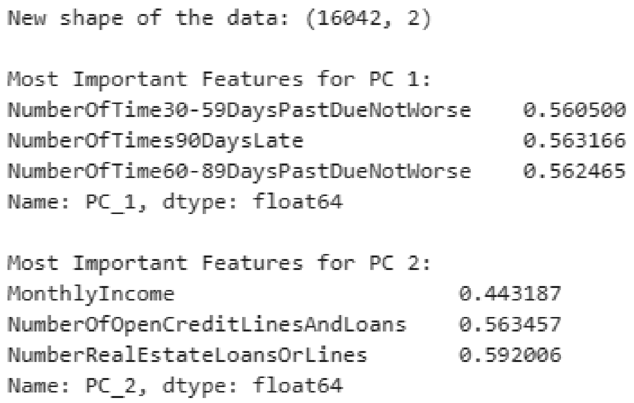


Figure 5: Depicts the result of K-means clustering with K=2.

After clustering the data, it is clear that there are two clusters present in the dataset. In order to get a sense of what differentiates these two clusters let's look at the contribution of the features towards each of the principal components. This analysis was performed and is shown below in Figure 6.



```
New shape of the data: (16042, 2)

Most Important Features for PC 1:
NumberOfTime30-59DaysPastDueNotWorse    0.560500
NumberOfTimes90DaysLate                 0.563166
NumberOfTime60-89DaysPastDueNotWorse    0.562465
Name: PC_1, dtype: float64

Most Important Features for PC 2:
MonthlyIncome                       0.443187
NumberOfOpenCreditLinesAndLoans     0.563457
NumberRealEstateLoansOrLines        0.592006
Name: PC_2, dtype: float64
```

Figure 6: Shows the contribution of each feature to the two principal components.

As we can see from the analysis above, principal component one appears to be heavily influenced by the number of times that someone has been late on payments. Principal component two appears to be heavily influenced by the income and number of open lines of credits or real estate loans a person has open. Thus, with this knowledge, looking at the two clusters in Figure 5 we can see that cluster 1 appears to have a very large value for principal component one but a relatively small value for principal component two. This means that these people have have been late pretty frequently (as shown from high value for PC 1), and have an income that is close to the mean (as shown from low value for PC 2). Looking at cluster 0 it appears that the samples generally have a low principal component one value but a diverse value for principal component two. Thus, this means that people in this group generally have been late a number of times that is closer to the mean, but they differ greatly in their income and number of credit lines or real estate loans they have open. Thus, it appears that there are two distinct types of people in the dataset: those that are late frequently and have a close to mean income, and those that are not late frequently but have a diverse income and number of credit lines open. To identify if anything about the labels can be learned from this clustering, the labels were added and can be visualized in Figure 7.

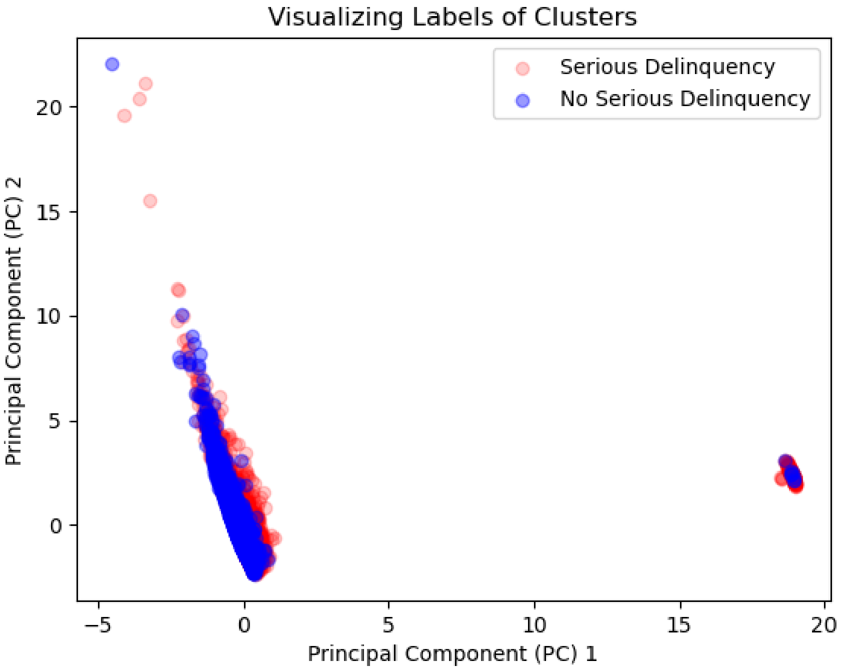


Figure 7: Shows the label makeup of the two clusters.

Based on the figure above, it is clear that not much can be learned about the labels from the clustering. It appears that in both clusters there is large amount of people who experience serious delinquency and those that do not. One surprising fact is that those that have a history of late payments are not necessarily at risk of serious delinquency. Thus, I would say that it is interesting that there are two diverse sets of people, however, these two diverse clusters are not very useful in identifying the characteristics of someone who may experience serious delinquency.

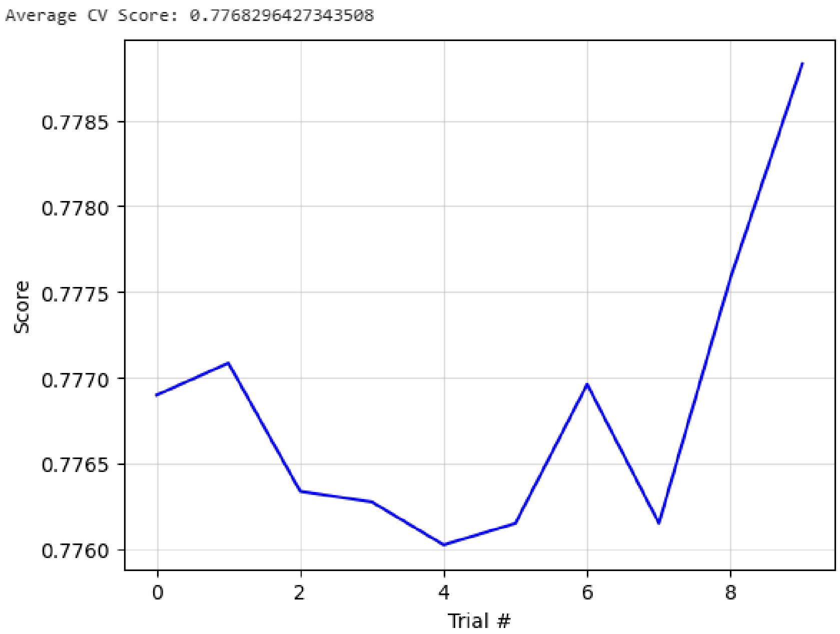
2.4 Random Forest Classifier - Supervised Model

The fourth model that was tested was a random forest classifier. Since the business objective is to identify people who are likely to experience serious delinquency, the random forest classifier will be a good model to test. A random forest classifier was decided upon over a regular decision tree, since decision trees are relatively unstable models, since small changes in the inputs of a decision tree can have drastic outcomes on the final decision tree derived. Thus, a random forest model was decided upon since this type of model makes many different decision trees based on random sub-samplings of the data points and the features within the dataset. This will lead to a much more stable model as compared to a singular decision tree.

In order to test the model the ”X_train_bal” dataset was used. This dataset is different to the ones tested previously. The dataset has gone through several pre-processing steps including filling missing values with the median value of a given feature and balancing the labels of the dataset through undersampling the majority class. This dataset has not been scaled and no dimensionality reduction has been performed to it. The reason for this is that scaling of the data is not necessary for a random forest classifier, since it does not affect the way in which the algorithm splits the data. Since a random forest model performs a random sub-sampling of the features, not all the features may be used in the final model, therefore, it was determined that dimensionality reduction was not required. One major benefit of not scaling or performing dimensionality reduction on the model is that it increases the interpretability of the model. This is a big benefit. Since this model will be used to asses people’s credit scores, it is essential to ensure that there are no unwanted biases present in the final model. The random forest model, however, derives its result based on a ”forest” of decision trees, therefore, the interpretability of the overall model may be less than logistic regression, but may can still be made interpretable. For example, the interpretability of the model can be increased through looking at particular decision trees within the model and identifying feature importance.

In order to determine the performance of the model on the training dataset described above, nested cross validation was used. For each trial of the nested cross validation, several different hyperparameters were tested in the inner cross validation loop to derive the optimal model for the outer cross validation loop. The random forest model tested had 10 estimators or 10 trees in the forest and split features based on gini index values. One hyperparameter tested in the inner cross validation included max depths of 4, 6, and 8. Another hyperparameter tested in the inner cross validation step was the minimum number of samples in a leaf which included values of 30 or

100. Ten trials were conducted with the results shown in Figure 8 below.



```
Average CV Score: 0.7768296427343508
```

Figure 8: Depicts the nested cross validation of the random forest model.

Based on the figure above, it appears that the random forest model performed relatively well with an average accuracy of 77.68% across all ten trials. The scores across all ten trials had only a small variation of around 0.2%. Thus, we can see that the random forest model appeared to show promising results as it had relatively high accuracy compared to the other models tested.

3 Choosing the Optimal Model to Optimize

Based on the data above, I would say that all of the supervised models appeared to have relatively high accuracy. The unsupervised K-means clustering model, however, appeared to not be very insightful. This was shown through the analysis conducted above which identified two distinct clusters within the data; however, these clusters did not provide insightful knowledge into the characteristics that may identify someone who is likely to experience delinquency. Instead it showed that there are people who are frequently late on payments with average income and others who are not late frequently but have a range of incomes and number of credit or real estate loans open. These behaviors, however, did give any indication as to whether a given person in either cluster would be likely to have a serious delinquency. For these reasons, this model will not be investigated further.

The supervised models appeared to have relatively high accuracy. Looking at the logistic regression model, it appeared to have the worst performance with an accuracy of around 71% during nested cross validation. The support vector machine model appeared to have a slightly better accuracy during nested cross validation with an average accuracy of 74%. The random forest model appeared to have the highest accuracy of the model tested averaging around 78% accuracy during nested cross validation. The SVM model only had slightly better classification results as compared to the logistic regression model. Since the SVM model had only slightly better accuracy than the logistic regression model, the decrease in interpretability does not justify choosing it over the logistic regression model. Comparing the logistic regression model to the random forest model, it is clear the random forest model performed much better than the logistic regression model. Though the logistic regression model may have higher interpretability a random forest model with a smaller number of forests will still have relatively high interpretability. Thus, based on these results, the random forest model will be used for further optimization. With a smaller number of forests the random forest model will still have relatively high interpretability and an indication to the confidence of each classification decision may be ascertained through the number of trees that voted in favor of a sample's final classification.