The Graduate Center

Re-classification of Violations

Understanding Miscellaneous Violations

Eva Silva

5-22-2020

Table of Contents

Background 2

Research Question 3

Methods 3

Figure 1: Dataset Features 3

Figure 2: Visualizing I_CODE & New I_CODE 4

Results 5

Figure 3: K Nearest Neighbors 5

Figure 4: Precision – Recall – F1 Score for K Nearest Neighbors 6

Figure 5: Precision – Recall – F1 Score for Decision Tree 7

Figure 6: Decision Tree – Feature Importance 8

Figure 7: PCA – Unscaled (left), Scaled (right) and X-Scaled Variance Ratio (bottom) 9

Figure 8: Heat-Plot 9

Figure 9: K Means Clustering (using feature 2 and feature 4) 10

Figure 10: Agglomerate 11

Figure 11: DBSCAN 12

Figure 12: ARI (top) & Silhouette Coefficient (bottom) 12

Conclusion 13

Background

The data set I chose to use for this study was the Stop Work Order – ECB Violation dataset from the Department of Buildings. This dataset is composed of two datasets:

1. **Stop Work Order:** an order issued by an inspector which prohibits all work on a construction site/property due to the unsafe practices being conducted on site.

This dataset is completely publicly available. A portion of it is available on Open Data, under the DOB Complaints data set (all violations that are equivalent to an A3 or L1 disposition).

2. **ECB Violations:** a violation issued by an inspector which contains an order to correct the conditions cited, and these can be challenge in court.

This dataset as mentioned before, is completely available on Open Data, under ECB Violations. The data contains various features from penalty imposed to the type of violation that was issued. This data was merged with the Stop Work Orders, as every Stop Work Order has an ECB Violation associated with. However, please note, not every ECB violation does not have Stop Work Order (SWO) associated with it.

I chose this data as its data I am familiar with and became interested with the manual process of SWO and Violation reclassification. There are certain SWO and their associated violations considered and labeled to be “miscellaneous violations”. This category of violation is one of the agency’s number one type of violation issued. However, there are some complications when reporting data with miscellaneous violations, which is that it is very hard to explain the results for a group of violations that can range from missing railings to expired insurance.

The usage of this category is so that in the creation of amendments to the construction code, it is a long process to have a violation category to be developed. Therefore, the miscellaneous violation was created to be used by the inspector when enforcing new construction laws (i.e. the OSHA license in 2019). Now, the program has its own violation code (infraction code) 1K6, but most of the violations issued initially under the new law were classified as miscellaneous, 106.

When generating reports, miscellaneous violations are reclassified, but the process can be tedious and long. Upon reading the course material, I grew interested in the classification supervised/unsupervised learning methods and sought to apply these methods onto the dataset. Hence, I understood that by applying these methods I would get the same results as the manually classified Stop Work Orders and their associated violations data available. Still, I was willing to see how these methods could facilitate the process.

Research Question

With the features chosen, how well does reclassification machine learning methods work on the data set?

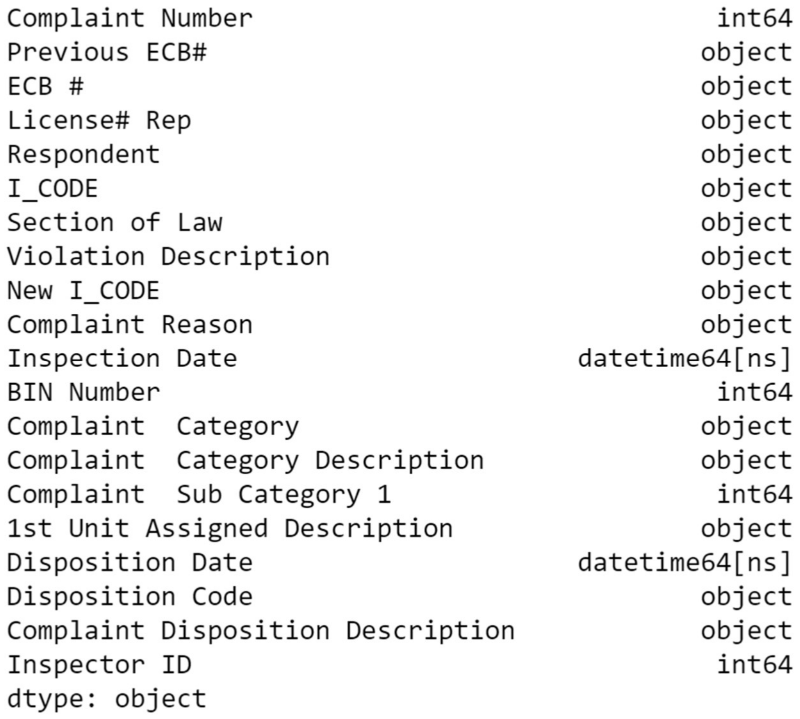
Methods

Data for this study was collected by the Department of Buildings: Administrative Enforcement Unit (AEU). Variables chosen from the dataset as shown in Figure 1, are as follows:

- **I_CODE:** Infraction Code, the type of violation issued.
- **Section of Law:** the section of law cited on every violation issued.
- **Inspector ID:** the inspector who issued the violation
- **New I_CODE:** the manual reclassification of the violation.

All other features available in the dataset were removed. Most of the data available wasn't pertinent to the study and one feature was complex in nature to clean and include: Complaint Reason. Complaint reason contains the explanation by the inspector as to what he saw pertaining to the violation. As inspectors can write in abbreviation and each inspector has different abbreviations for different items, the process to have each individual inspector form of writing to be understood exceeded the time given to approach the subject.

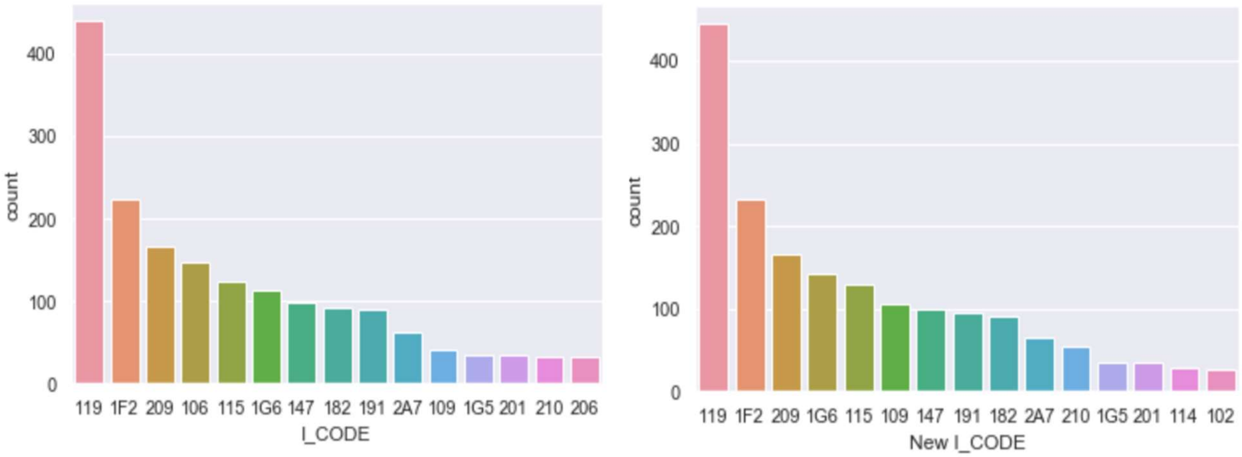
Figure 1: Dataset Features



```
Complaint Number                                int64
Previous ECB#                                  object
ECB #                                          object
License# Rep                                   object
Respondent                                     object
I_CODE                                         object
Section of Law                                 object
Violation Description                          object
New I_CODE                                     object
Complaint Reason                               object
Inspection Date                        datetime64[ns]
BIN Number                                      int64
Complaint  Category                            object
Complaint  Category Description                object
Complaint  Sub Category 1                       int64
1st Unit Assigned Description                  object
Disposition Date                       datetime64[ns]
Disposition Code                               object
Complaint Disposition Description              object
Inspector ID                                    int64
dtype: object
```

Exploring the data, as seen in Figure 2, led to a few challenges along the way. There were many classes in both I_CODE and New I_CODE and this became a problem with the Stratified Shuffle Split, as these two features contained classes with less than 5 in the group, specifically in the New I_CODE. All the classes with less than 10 in group were dropped from the study. All the of categorical variables in the dataset were label encoded. However, as I look back at the project, I realize that the label encoding should have been the same labels for the same infraction in I_CODE and New I_CODE; i.e. taking infraction code 101 = 0, in I_CODE, if infraction code 101 were present New I_CODE, it should have continued to be labeled as 0. In this specific case, it was. Yet, it was not in all cases—the miscellaneous infraction code was no longer an existing class in New I_CODE.

Figure 2: Visualizing I_CODE & New I_CODE



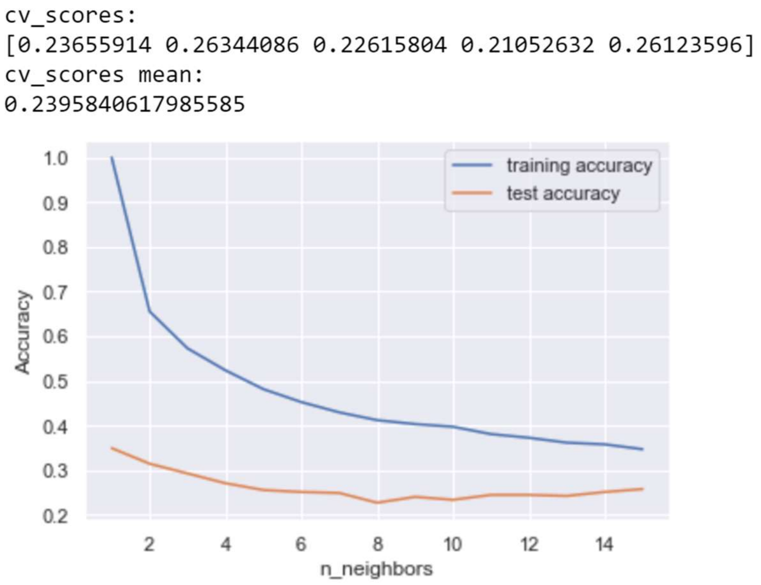
By visualizing the data using bar graphs to see the complexities of I_CODE and New I_CODE, I was able to determine which classes to drop from the study. I was able to firmly consider the machine learning methods I would apply to my data to answer my research question.

Results

After data cleaning and exploration, the following methods of supervised learning were used on the data set: K Nearest Neighbors and Decision Trees. These classification methods were used to analyze relationship between the features and the target and see whether the clustering method would replicate the manual classification (see Figure 3 & 4). The decision tree application was to get a better determine which features were important in the classification of violations (see Figure 5 & 6).

When using k-NN on the training dataset— as the data was previously split using the Stratified Shuffle Split, I hoped to have the pattern recognition algorithm, through clustering analysis classify the violations. Below in Figure 3, we can see the accuracy of the algorithm over the change in number of neighbors.

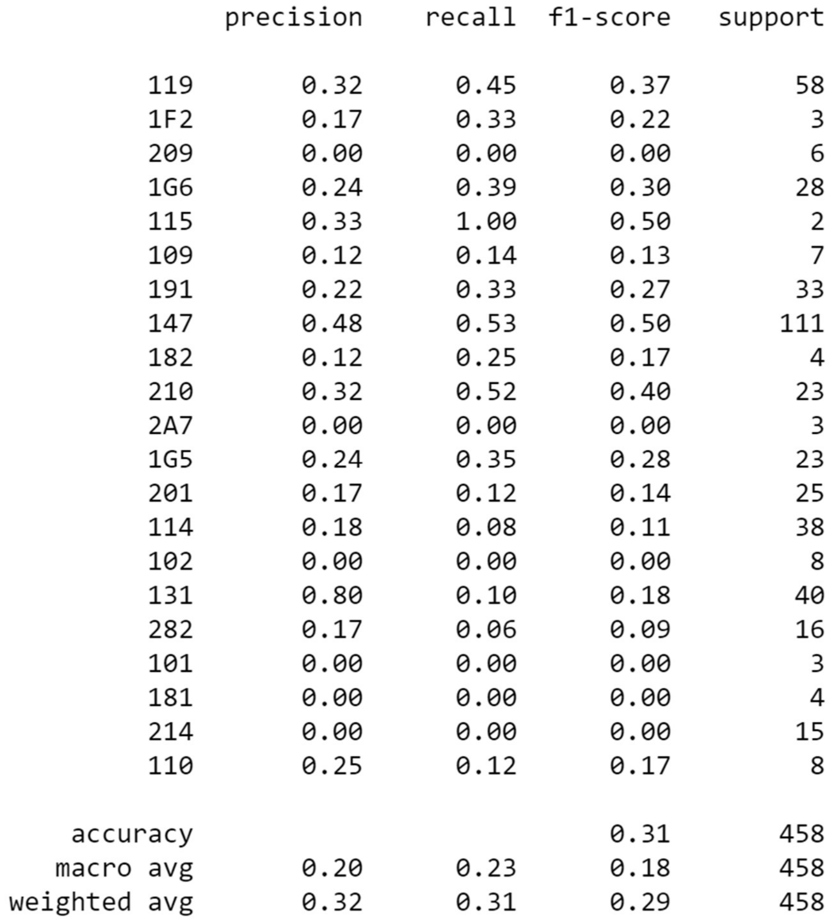
Figure 3: K Nearest Neighbors



```
cv_scores:
[0.23655914 0.26344086 0.22615804 0.21052632 0.26123596]
cv_scores mean:
0.2395840617985585
```

From the graph, we see that the accuracy decreases drastically as we increase the number of neighbors from one to fifteen. It was observed, that the best parameters for number of neighbors is one. However, not knowing how well the model would perform on the new data, we tested using cross validation (k-fold cross validation). Five-fold cross validation was applied to new the data and the average accuracy of the model was determined to be 0.23959, which is very low. Thus, these parameters, the "best parameters" n_neighbors = 1, were used to determine the precision-recall-f1 score table.

Figure 4: Precision – Recall – F1 Score for K Nearest Neighbors



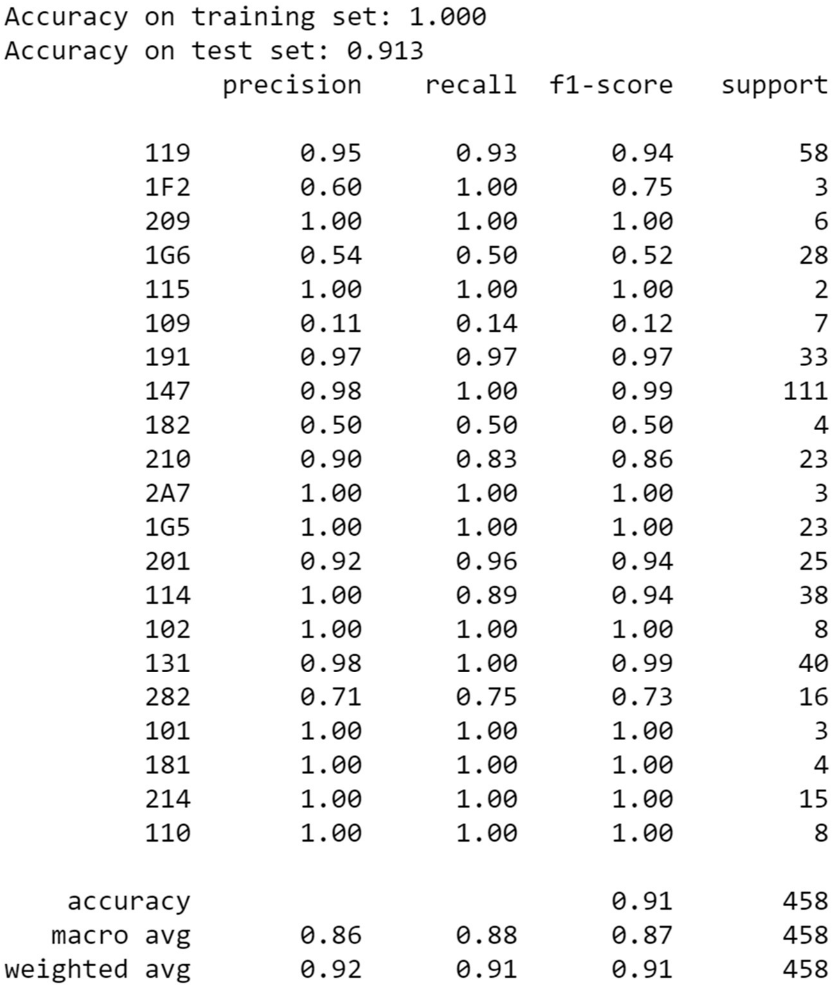
```
              precision    recall  f1-score   support

         119       0.32      0.45      0.37        58
         1F2       0.17      0.33      0.22         3
         209       0.00      0.00      0.00         6
         1G6       0.24      0.39      0.30        28
         115       0.33      1.00      0.50         2
         109       0.12      0.14      0.13         7
         191       0.22      0.33      0.27        33
         147       0.48      0.53      0.50       111
         182       0.12      0.25      0.17         4
         210       0.32      0.52      0.40        23
         2A7       0.00      0.00      0.00         3
         1G5       0.24      0.35      0.28        23
         201       0.17      0.12      0.14        25
         114       0.18      0.08      0.11        38
         102       0.00      0.00      0.00         8
         131       0.80      0.10      0.18        40
         282       0.17      0.06      0.09        16
         101       0.00      0.00      0.00         3
         181       0.00      0.00      0.00         4
         214       0.00      0.00      0.00        15
         110       0.25      0.12      0.17         8

    accuracy                           0.31       458
   macro avg       0.20      0.23      0.18       458
weighted avg       0.32      0.31      0.29       458
```

As seen in Figure 4, the precision, or the ratio of correctly predicted positives over the total predicted positive observations, ranged in its results. Some classes like 181 had a higher precision rate, while others like 2A7 had a low precision rate. For 181, this meant that it had a low false positive rate. Recall, the ratio of correctly predicted positive over all observations, not many classes rated above the acceptable 0.5. Apart from 115 and 210, all the other classes had a low recall rate. Now in evaluating the F1 Scores, the F1 score weighs the average of the precision and recall and useful when having an uneven class distribution. The score fore the classes were low, as to be expected, seeing that their relative precision and recall scores were low. The low accuracy of k-NN results could be due to the use of irrelevant features and/or scaling not being consistent (signs of underfitting of the model). These two subjects are discussed in this paper, BIN Number mentioned as an irrelevant feature and the improper scaling of the infraction code features.

Figure 5: Precision – Recall – F1 Score for Decision Tree



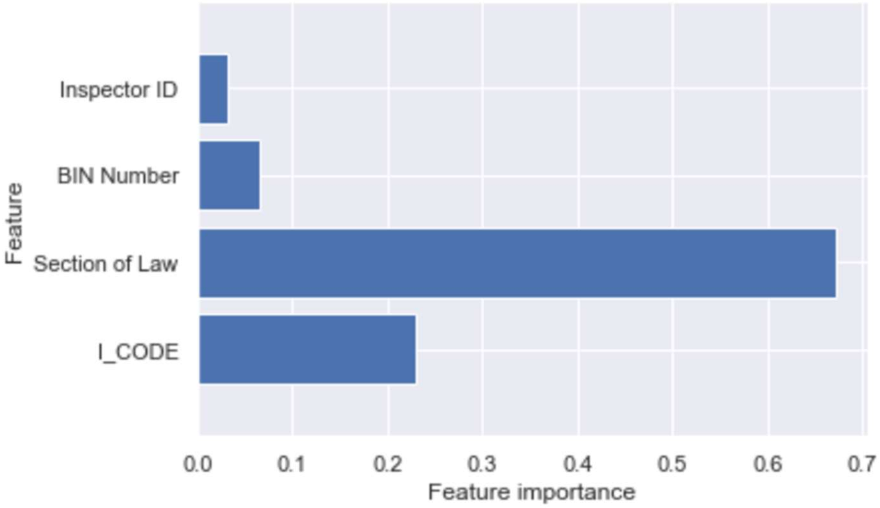
```
Accuracy on training set: 1.000
Accuracy on test set: 0.913
              precision    recall  f1-score   support

         119       0.95      0.93      0.94        58
         1F2       0.60      1.00      0.75         3
         209       1.00      1.00      1.00         6
         1G6       0.54      0.50      0.52        28
         115       1.00      1.00      1.00         2
         109       0.11      0.14      0.12         7
         191       0.97      0.97      0.97        33
         147       0.98      1.00      0.99       111
         182       0.50      0.50      0.50         4
         210       0.90      0.83      0.86        23
         2A7       1.00      1.00      1.00         3
         1G5       1.00      1.00      1.00        23
         201       0.92      0.96      0.94        25
         114       1.00      0.89      0.94        38
         102       1.00      1.00      1.00         8
         131       0.98      1.00      0.99        40
         282       0.71      0.75      0.73        16
         101       1.00      1.00      1.00         3
         181       1.00      1.00      1.00         4
         214       1.00      1.00      1.00        15
         110       1.00      1.00      1.00         8

    accuracy                           0.91       458
   macro avg       0.86      0.88      0.87       458
weighted avg       0.92      0.91      0.91       458
```

In Figure 5, we see the results of using a decision tree model. There was no max_depth applied to it. From the results, we can see that there are signs of overfitting from the accuracy of the test. Thus, this model cannot be trusted in the classification. When setting new parameters for max_depth = 6, the accuracy for the training and test set were 0.602 and 0.600. Low results. Fiddling with max-depth and setting it to 10, the accuracy for the training and the test were 0.935 and 0.921. Although, still signs of overfitting, the results were good. Potentially, the aim is to be able to find the max_depth or do some pruning to obtain better results.

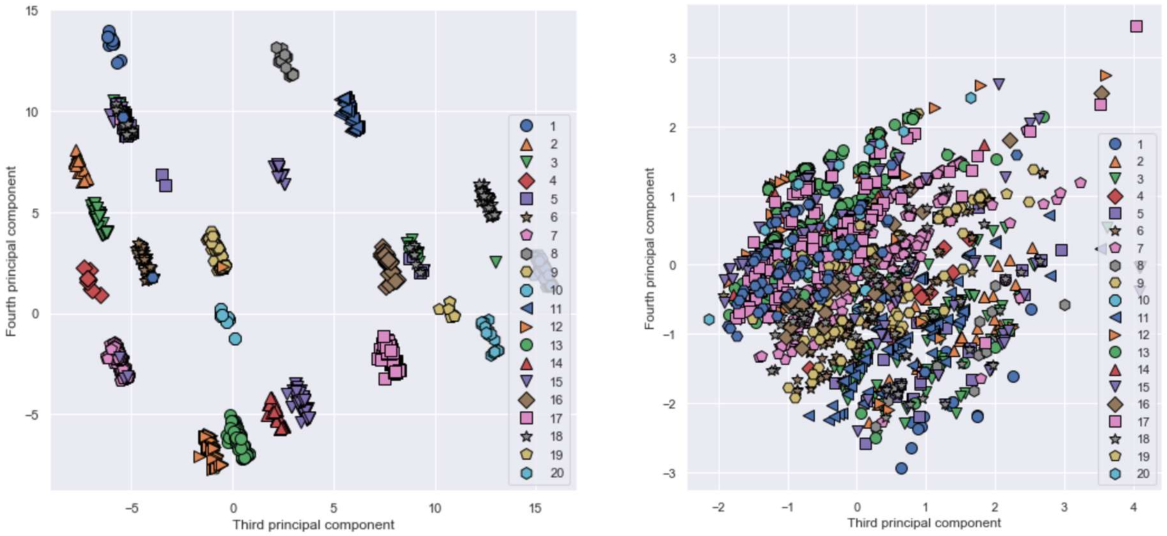
Figure 6: Decision Tree – Feature Importance

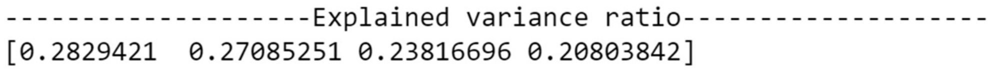


Using the decision tree algorithm, I was able to obtain feature importance. As observed in Figure 6, the results obtain from the decision tree with no max_depth parameters set, we obtained a bar-graph showing the features used in the model and their level of importance. The most important features in the model for each all parameters used for the decision tree were I_CODE (infraction code) and Section of Law. This was a key factor that interpret the importance of the section of law in classifying the data sets.

When using PCA, a method used to reduce the dimension of the feature space, it wasn't as effective with the data present, as the data only contain 4 features. Given the nature of PCA, the data was scaled as the fitting of the algorithm is dependent on the scaling of the features. The use of standard scaler was use on the training set and the distinction is made in Figure 7.

Figure 7: PCA – Unscaled (left), Scaled (right) and X-Scaled Variance Ratio (bottom)

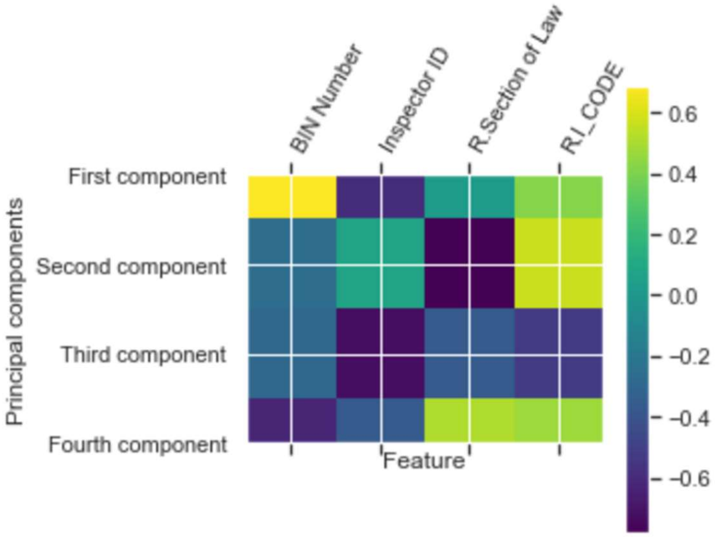




```
--------------------Explained variance ratio--------------------
[0.2829421  0.27085251 0.23816696 0.20803842]
```

When using PCA, the number of components used was four. Since PCA is used for dimension reduction and they were only 4 features available, I chose to use PCA to understand the variance of the data. From the results of variance ration of the principal components (using scaled data), all 4 components helped to explain 95 percent of the variance. A heat plot was also generated to understand the correlation of the original features and the number of components, as seen in Figure 8. Based on these 4 components we visualized the data and saw some separation of the data. However, heavy overlap of the data points.

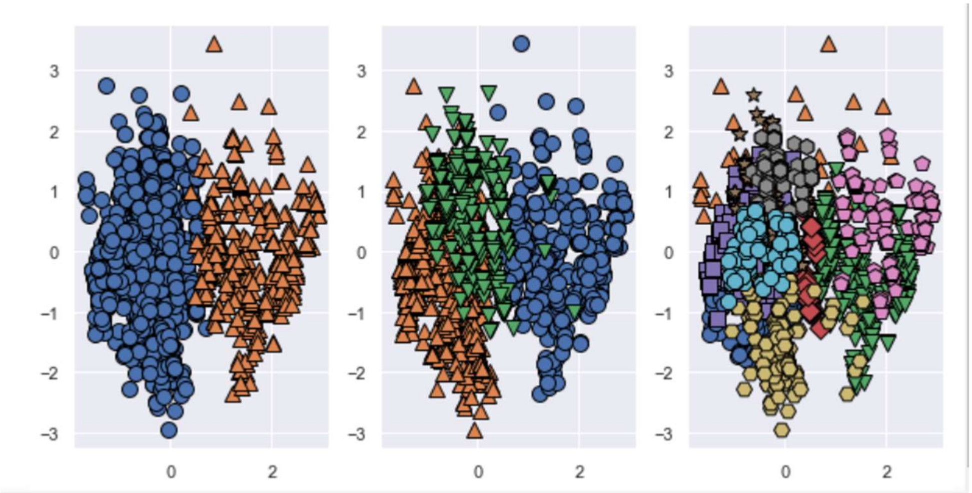
Figure 8: Heat-Plot



Using PCA as a pre-processing step for clustering, did improve the accuracy on the training set and test set from 0.24 to 0.61. This is not the best results, but the result was an improvement than that compared to the k nearest neighbors' accuracy. Still, this processes training data was then used for the following algorithms: K-Means, Agglomerate and DBSCAN.

Using K-Means clustering and applying different number of clusters (3-5), the algorithm takes the training data set (processed) and groups similar data points together by using the underlying patterns. A set number of clusters were stated, and three cluster plots were created. Upon, observing the three plots, it was evident that there was a lot of overlap between the points when more clusters are declared (see Figure 9).

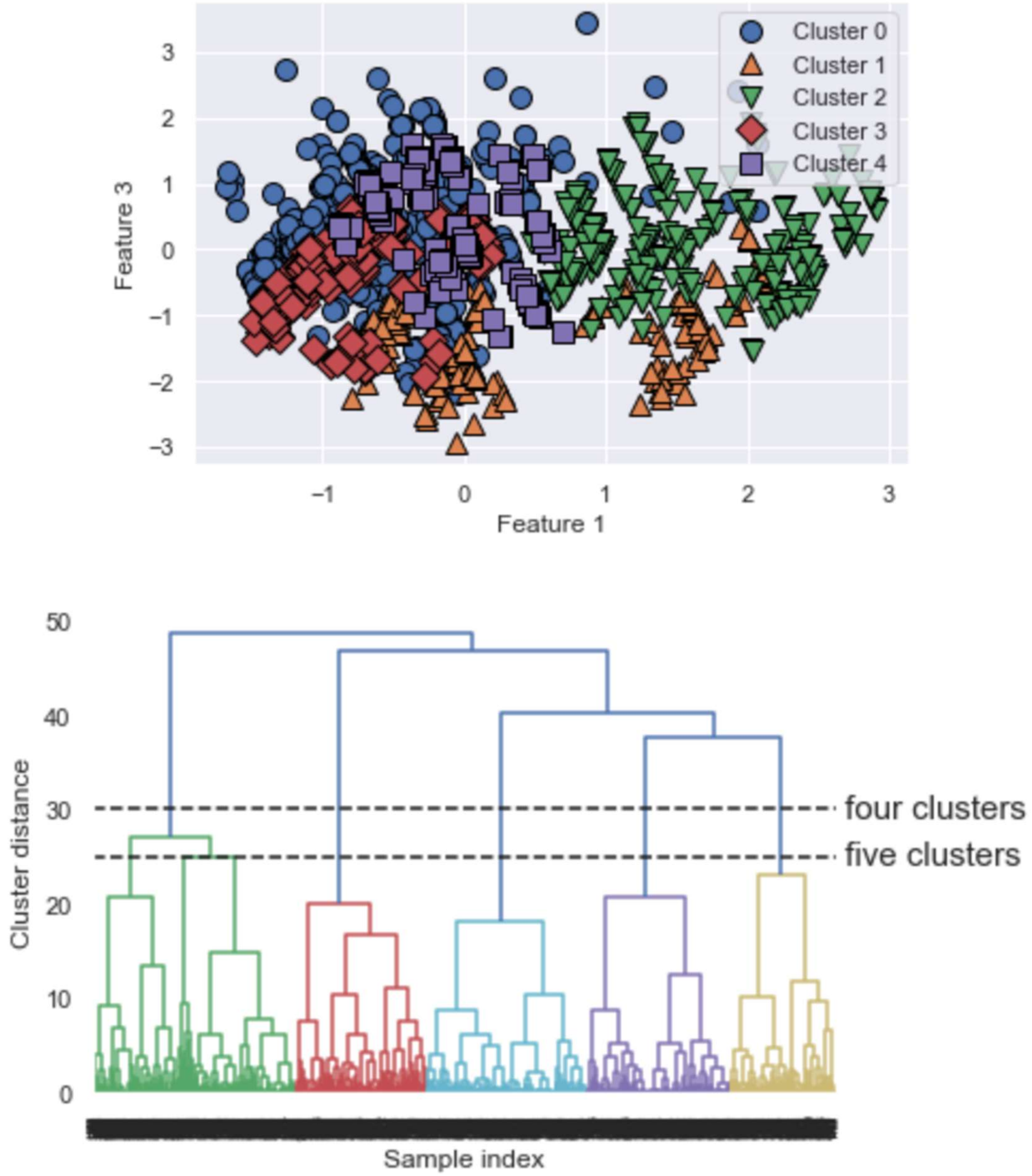
Figure 9: K Means Clustering (using feature 2 and feature 4)



It was later determined by the using the elbow method (plots) the determine the optimal value of k. In the case of this dataset it as 5. From there the data was the clustered once more using k=5.

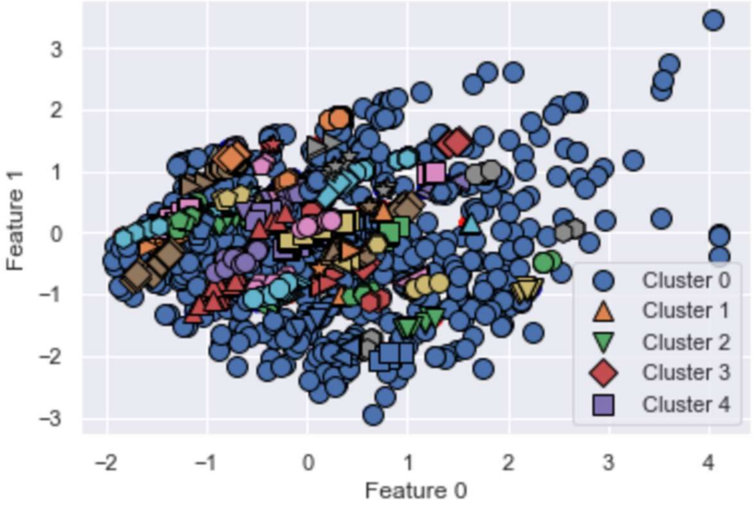
With agglomerate/hierarchical clustering analysis the goal is to build a tree diagram violation more similar are placed on the branches closes together. Originally, the number of clusters used was determined by the previous clustering analysis. As previously stated, the output wasn't easy to read as the clusters overlapped tremendously. Thus, the application of a dendrogram which create a tree-like diagram that records the split sequence of the data.

Figure 10: Agglomerate



The dendrogram records the distance between the clusters in Figure 10 (top). A threshold distance is set, 30 mark and horizontal number and the number of vertical lines it intersects is the optimal number of clusters to use 4. This is how the number of clusters is decided using hierarchical clustering analysis (Figure 10 – bottom).

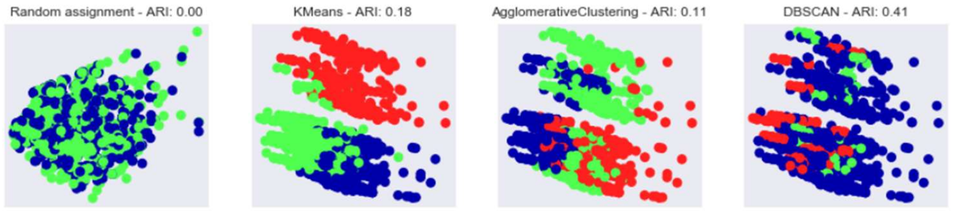
Figure 11: DBSCAN

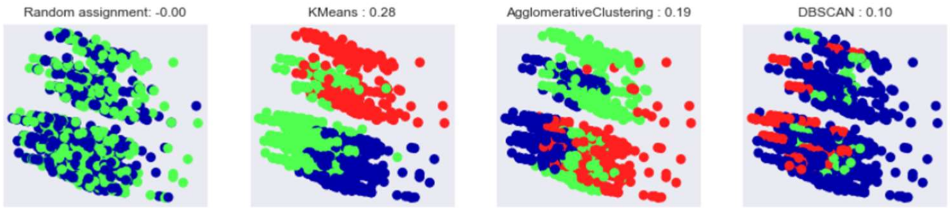


Density-based clustering analysis was also used on the training set. The use of density-based clustering is unlike k-means as it works best with arbitrary shaped clusters and detecting outliers. In this analysis, the number of clusters does not have to be specified. As seen in figure 11, outliers are marked in the blue circles. For this plot, the set number of eps 0.2 and the minimum number of data point was of 5. The use of this type of clustering was not useful as most of the data was noise/outliers and it's because DBSCAN is not fit to define clusters and the constrains of defining the number of eps and minimum points, especially when clusters have different densities. The violations data as seen in the bar graph populated in the beginning shows a range of densities for the classes of the features used.

Using the clustering analysis did help understand and visualize the training set. However, as seen in Figure 12, finding both the adjusted rand score and silhouette coefficient showed that the consistency of the cluster data was low. The values are on a scale of -1 to 1 and the as the figure shows, the scores did not surpass .41. High values indicate that the clustering configurations are appropriate and thus, low scores as the ones obtained from this study show the opposite.

Figure 12: ARI (top) & Silhouette Coefficient (bottom)




Random assignment: -0.00

KMeans : 0.28

AgglomerativeClustering : 0.19

DBSCAN : 0.10

Conclusion

I believe that now with the results gathered from the study, I understand now that I should try incorporating new features that can provide a better understanding of all the variables that lead to the manual reclassification of the data. These new features will aid in more developed classification using the supervised and unsupervised learning methods. From the different model applications, I gathered that section of law and infraction code are good features to use for classification, but it is not enough to work a proper model. It is possible that working with less classes would help understand the dataset more in depth. Although, the algorithms used on the dataset did not provide great results (higher accuracy levels), this study has allowed me to understand the techniques I can use to apply to my data and facilitate the manual classification of these miscellaneous violations. Understanding when to apply certain models/algorithms versus others, i.e. the distinction between the three-types of clustering analysis used in this study, classification methods used during the supervise learning methods part of this project and learning the use of PCA as a pre-processor. I feel like I gain a lot from this study even though I was not able to answer my research question.