Math 748: Course Project Final Report

Who Did We Stop This Time?

Gabrielle Salamanca
San Francisco State University

December 20, 2024

Contents

I. Executive Summary

There can be multiple reasons why a police officer would stop someone on the road. Perhaps the license plate has expired registration tags, someone was speeding, traffic violations, or someone was driving under the influence to name a few. And any of these can be issued a citation, meaning the culprit is being legally charged with violating the traffic law.

For this project, I wanted to find out if there's a pattern to receiving a citation, and if certain characteristics or circumstances are more likely in receiving one. The dataset being used in this project was published under Vincent Arel-Bundock's Github projects under available R datasets, but the original was from Open Data Minneapolis. It is called Police Stop Data, and it's still being updated to this day. However, under Arel-Bundock's Rdatasets, it only pulled all the stops during the year 2017.

The dataset had 51,920 rows and 15 variables; and it included information about the type of stop, about the area it happened in, about any processes that happened, and about the person that was stopped. Because I wanted to use both supervised and unsupervised learning for this project, I had to split the data into two, one for each. For unsupervised learning, I had to remove any categoricals I couldn't make binary, and the response variable citationIssued. For supervised learning, I kept nearly everything, besides any variables I believed to be unnecessary.

Now, I did have to down sample the variable citationIssued due to an imbalance of no's and yes's after removing all the missing values, just so the results won't be skewed. So, my results were wildly different from when I didn't do this extra step.

Now, I'll admit, I unfortunately wasn't able to do more after the presentation due to the other deadlines I had. I would have loved to try out the other methods, but for the time being, we had: PCA, K-Means Clusters, Decision Tree, and LASSO.

II. Dataset Cleaning and Editing

As I've mentioned previously, the dataset was originally 51,920 rows and 15 columns. However, I trimmed out 5 variables, because I believed they were irrelevant to the project. They were: rownames, idNum, MDC, latitude, longitude. Rownames and idNum were mainly identifiers for the row, but I believed the built-in rows in R were sufficient. MDC was a variable that was concerned about how the data was collected, but that didn't seem important for the project. Then, I had to convert the date column into an actual date and split it into three: month, day, year. I removed the year, since it was known to be only within 2017 for all of them. However, I could have converted it into days of the week once I turned it into an actual date, but as mentioned, I wasn't able to.

Then, when I looked through the original dataset, citationIssued's values were: 15,899 no's, 3,211 yes's, and 32,810 empty cells. So, I had to remove the missing values, then down sample the variable.

Then, because I needed to create a separate dataset for unsupervised learning, I turned any categoricals into binary values if possible. If I couldn't, then those were removed as well as citationIssued when I made the new dataset. The variable that I turned into binary values was problem: 1 was traffic, and 0 was suspicious. The categorical variables I had to remove were unable to be converted to binary values, because there were too many possibilities, and I didn't feel comfortable weighing one possibility heavier than the others.

III. Feature Selection: Unsupervised Learning

Unsupervised learning was my main objective for this project. As stated previously, I wanted to specifically see if there was some sort of pattern with the Minneapolis police stopping cars, and so I have decided to use PCA and K-Means Clustering.

A. Principal Component Analysis

PCA is a dimensionality reduction technique in statistics and machine learning. It transforms the original data into a lower-dimensional space, which means transforming it with possibly correlated features into a smaller set of uncorrelated features called principal components. As this happens, it retains most of the original variance in the data.

B. K-Means Clustering

K-Means Clustering is a machine learning algorithm used to partition data into a specified number clusters (k) based on similarity. It groups the unlabeled dataset into different clusters, where the data points are in the same cluster or more similar to each than to those in other clusters.

IV. Feature Selection: Supervised Learning

I decided to include supervised learning for this, because I can predict if a citation was issued or not, considering we were provided the variable. So, I wanted to try a method that I could have done on my Math 448 project, and use one of the familiar ones.

A. Decision Tree

A Decision Tree is used for both classification and regression tasks, which we will be using for the former. It splits the data into subsets based on the value of input features. It then builds a model in the form of a tree. Each node in the tree represents a decision on an attribute, each branch represents an outcome of the test, and each leaf represents the final prediction.

B. LASSO

LASSO is a shrinkage method, shrinking the estimates' coefficients towards 0. This shrinkage reduces the variance and can perform variable selection. LASSO has a l_1 penalty,

which has the effect of forcing some of the coefficient estimates to be exactly 0 when the λ is sufficiently large.

V. Results

A. Principal Component Analysis

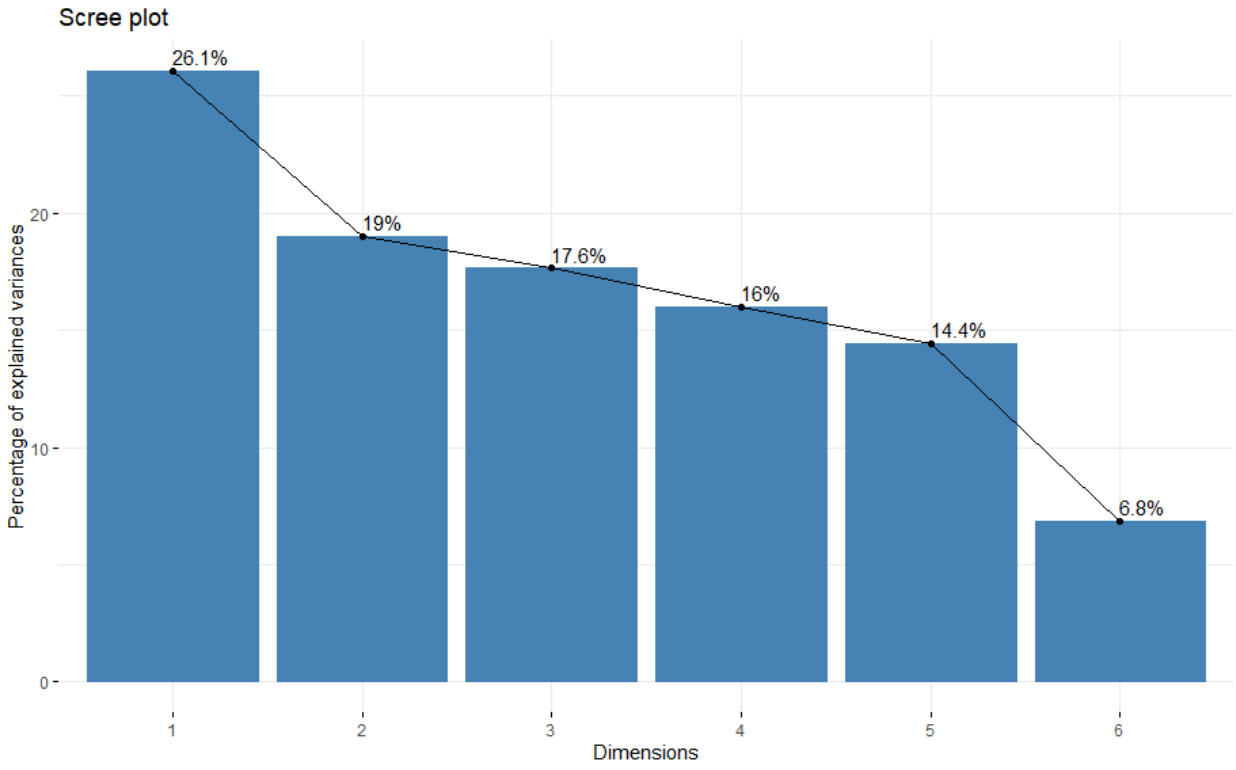

Figure 1: Scree Plot

Importance of Components:

	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6
Standard Deviation	1.2509	1.0684	1.0288	0.9795	0.9300	0.6409
Prop of Variance	0.2608	0.1902	0.1764	0.1599	0.1442	0.0685
Cum Prop	0.2608	0.4510	0.6275	0.7874	0.9315	1

The goal of PCA is to retain enough components to explain most of the variance while also reducing dimensionality, so we want a cumulative proportion threshold to be between 80-95%. By Component 3, 62.75% of the variance can be explained, meaning they capture the

majority of the patterns in the data. But we're not quite at the 80-95% threshold, so we would go all the way up to Component 5 and drop the last one.

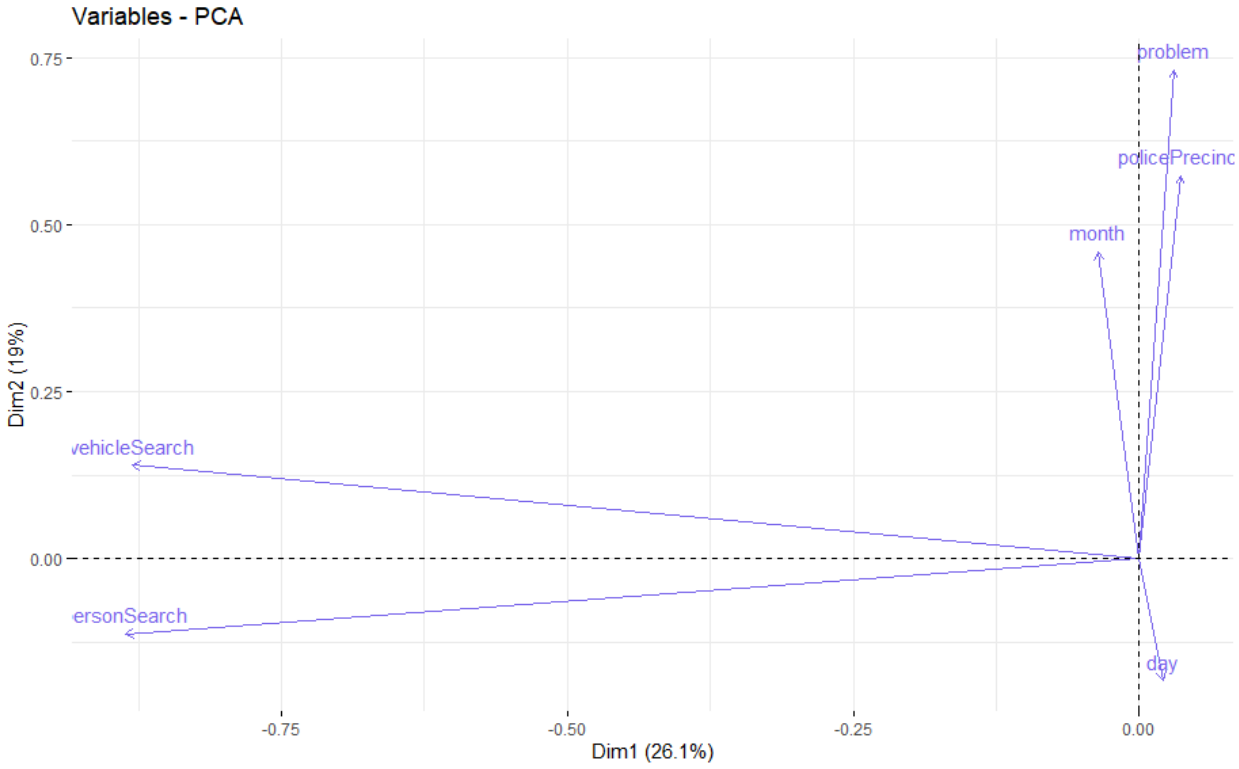

Figure 2: PCA Variable Plot

Loadings:

	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6
problem		0.685	0.251	0.300	0.586	0.186
personSearch	-0.709	-0.105				0.695
vehicleSearch	-0.704	0.130				-0.692
policePrecinct		0.537	0.337	-0.600	-0.484	
day		-0.171	0.708	0.562	-0.392	
month		0.430	-0.563	0.482	-0.514	

Loadings represent the contribution of each variable to a principal component, indicating how strongly each variable correlates with the corresponding component. In a loading, if the value is either a large positive or negative, it means that the variable heavily influences the component. If it's close to zero, the variable doesn't contribute as much to the component.

In Component 1, both personSeach and vehicleSearch strongly contribute to it, but they also suggest being strongly correlated. As the first principal component increases, both variables tend to decrease. In Component 2, problem, policePrecinct, and month have the strongest contributions and will tend to increase. personSearch, day, and vehicleSearch have the weakest contributions, and the first two are negative. Overall, perhaps the type of problem varied across police precincts and month. In Component 3, the strongest contributors were month and day, but in two different directions; and there's moderate influence from policePrecinct and problem. Daily patterns are strongly positive, but there's a trade-off with monthly patterns. The moderate influences may play a role with how those temporal patterns manifest.

Component 4's strongest contributors are policePrecinct and day, but in opposite directions. The variable month could be debated as another strong influence, but problem would be categorized as moderate influence. There's a relationship between the spatial and temporal factors, perhaps there's a trade-off between precinct-related effects and time-based trends. Component 5 has a positive influence from problem, while the rest are negative. Depending on the policePrecinct and date, there could be issue-specific trends. Finally, Component 6 has weak contributions from problem, while both searches have strong contributions, but in opposite directions. There's perhaps more of a focus on what type of searches, mostly person ones it seems.

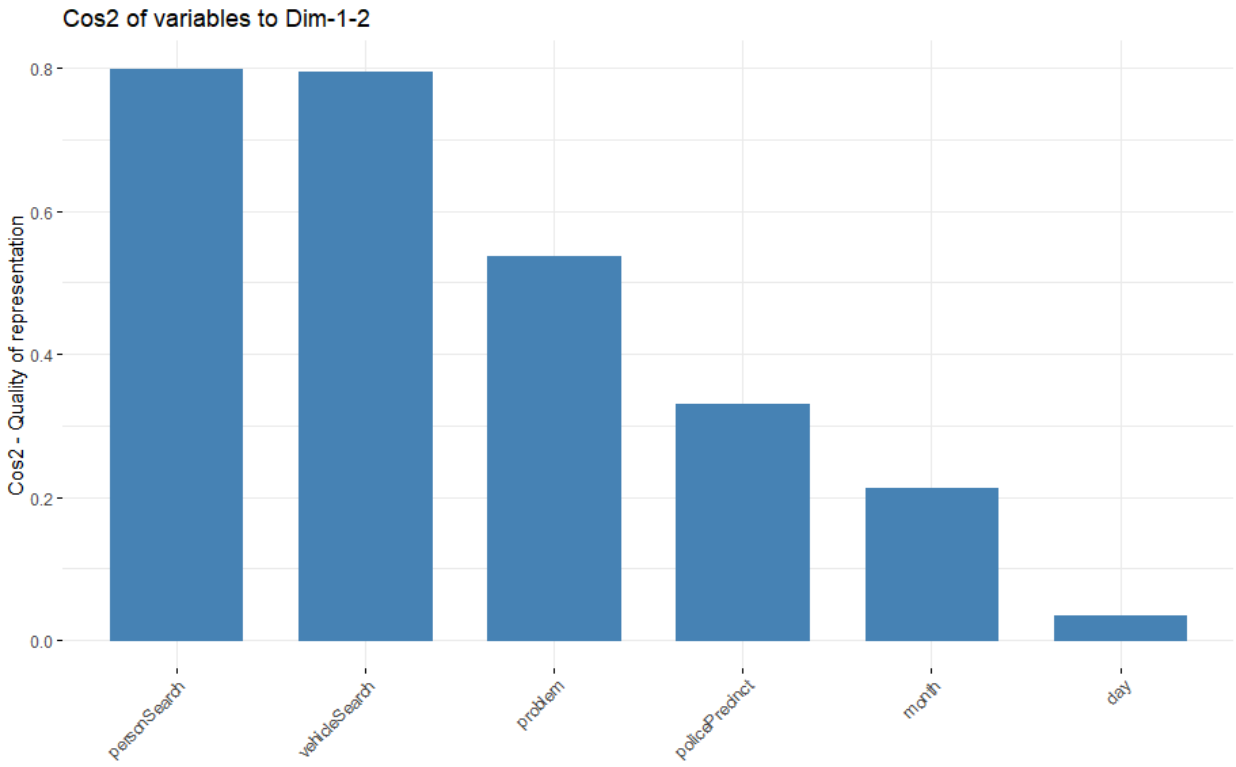

Figure 3: cos^2 of Variables to Dim-1-2

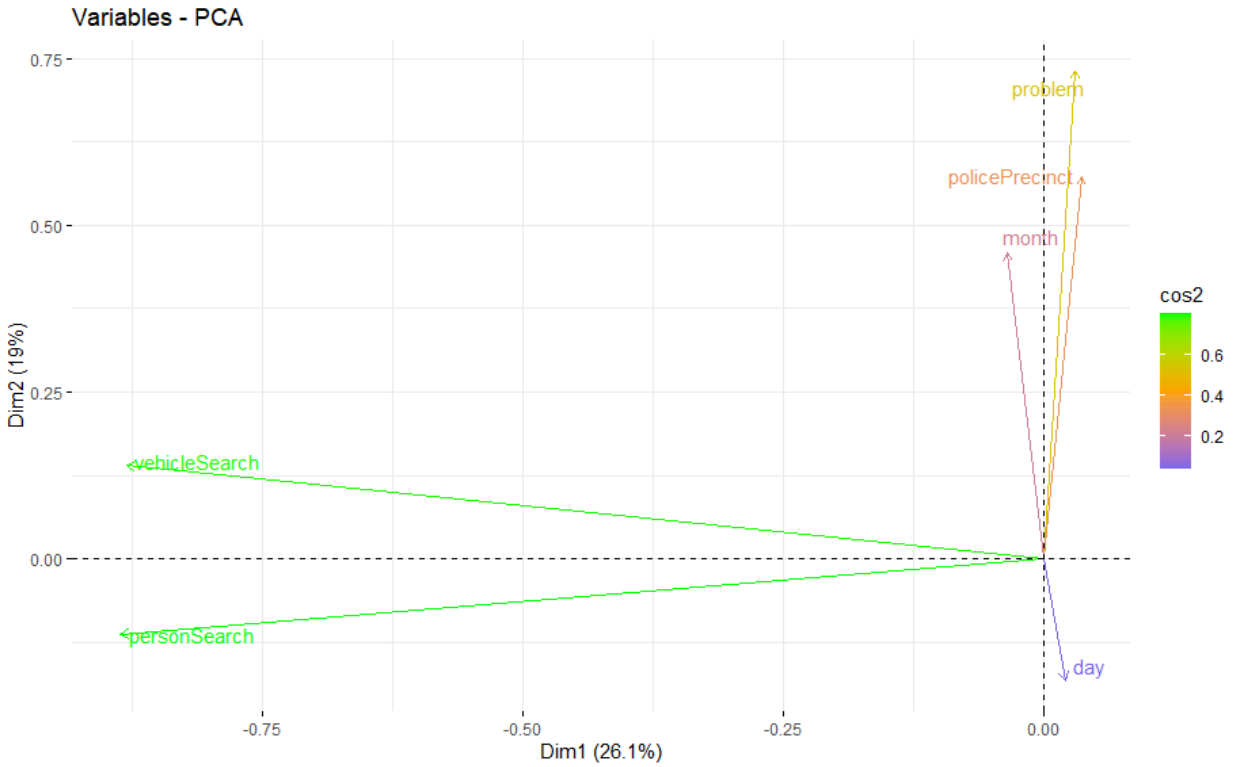

Figure 4: PCA Variable cos^2 Plot

In these two figures, they represent the proportion of the variance of each variable that's captured by the selected dimensions. It's a measure of how well the variable is represented in the reduced-dimensional space defined by those dimensions. The variables represented well in the Dim-1-2 plane are both search variables, where most of their variances are captured by these dimensions. I would say the ones partially represented are problem and policePrecinct, where some of their variances are captured by other dimensions. I could also see an argument where policePrecinct should be under poorly represented like month and day, where most of their variances are in dimensions outside of Dim-1 and Dim-2. So, we could explore additional dimensions if we wanted to explain the last three variables.

B. K-Means Clustering

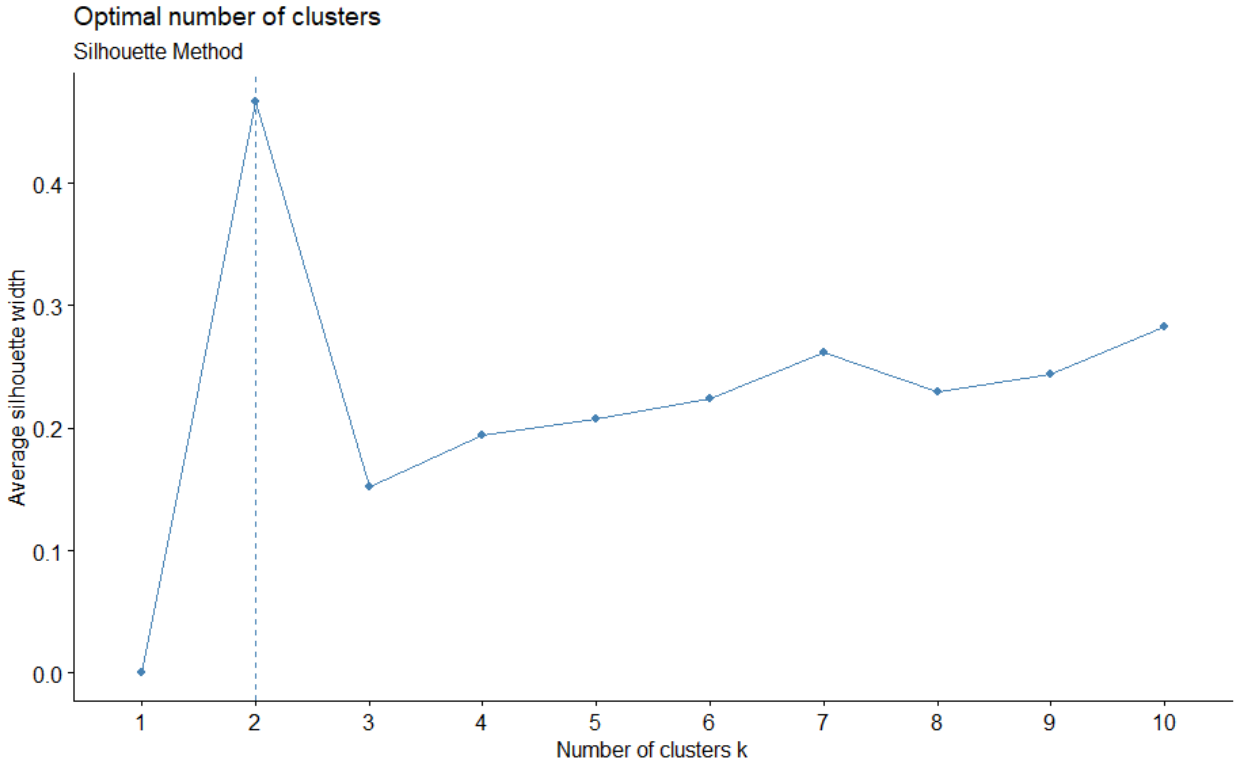

Figure 5: Optimal Cluster Number Plot

To find out the optimal number of clusters for K-Means Clustering, I ran the fviz_nbclust function to visualize it by using the silhouette method. The silhouette method assesses how similar each point is to its own cluster and how different it is from other clusters. The silhouette score will range from -1 and +1: +1 indicates the point is well-matched to its own cluster. 0 indicates that it's on or very near the boundary between 2 clusters, and -1 indicates that it might be misclassified and is closer to another cluster. All the values didn't fall under 0, but there was one value that was far above the rest: k = 2. So, I ran that value through the kmeans function, printed its centers, and visualized its clusters.

Centers:

	problem	personSearch	vehicleSearch	policePrecinct	day	month
1	-0.2079	2.3403	1.7239	-0.1092	-0.0171	0.0024
2	0.0334	-0.3756	-0.2767	0.0175	0.0027	-0.0004

Every center has a negative and positive value in each variable. A negative value suggests that on average, the values of that variable in the data points within that cluster are below the overall mean of that variable if it's standardized. If it's a positive value, then they're

above the overall mean on average. And since all centroids have a negative and positive value, this suggests that there's a contrast in how the clusters differ with respect to that variable.

In Cluster 1, the dominant variables are both searches, having strong positive associations. The variable problem is weakly negatively associated, perhaps having minimal relevance. However, policePrecinct, day, and month have negligible influence. So, this cluster groups observations strongly related to these search types. Cluster 2 meanwhile is the same, but with some key differences. The dominant variables are negative contributors, and problem was weakly positively associated. This cluster perhaps is capturing cases where other factors are slightly more relevant.

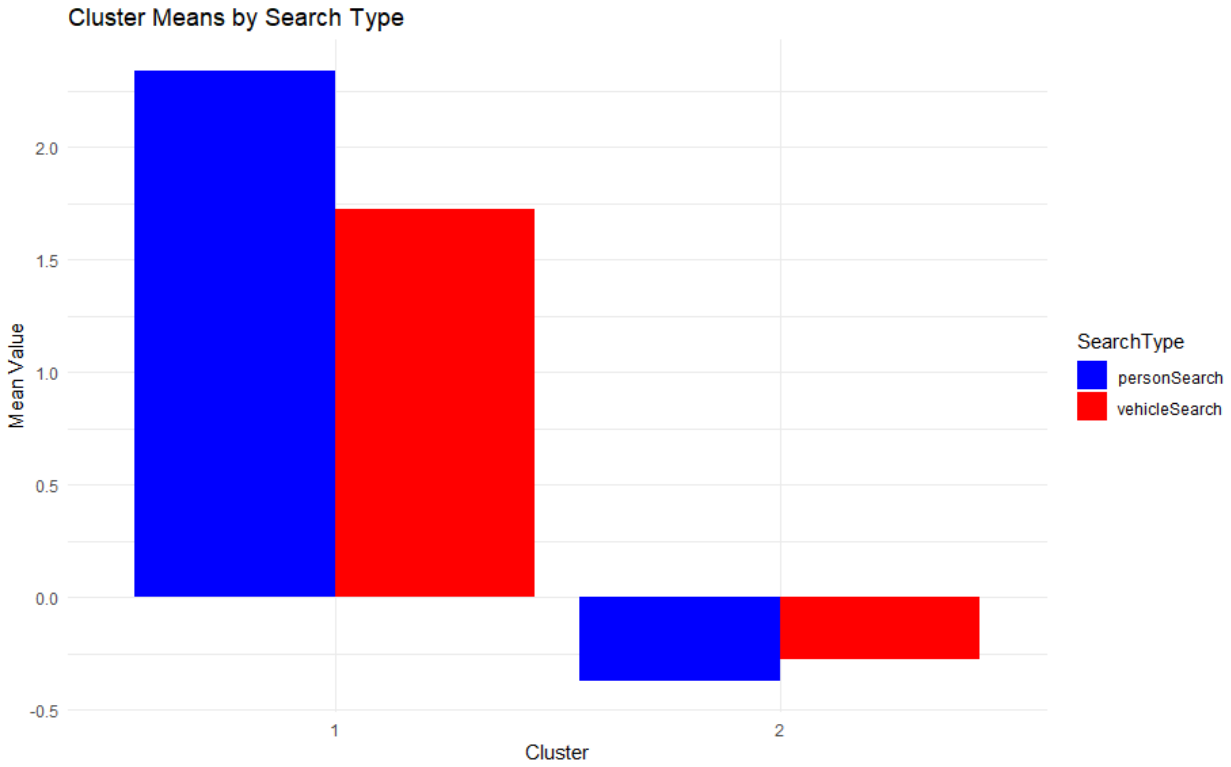

Figure 6: Cluster Means by Search Type

Both clusters seem to favor personSearch, but Cluster 1 has it as a key factor, and Cluster 2 isn't strongly characterized by it.

C. Decision Tree

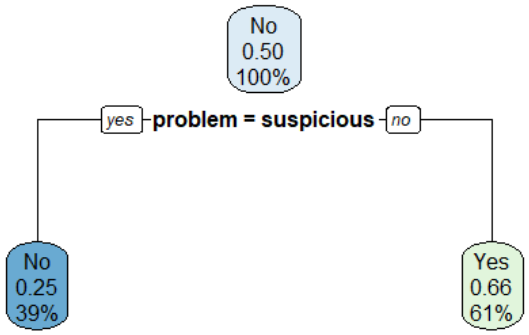

Figure 7: Decision Tree

After downsampling, it's a 50% draw between being issued a citation or not. The only split is the variable problem, particularly if the input is suspicious. If it's a suspicious stop, then the branch leads to either a 25% chance of being issued a citation or 39% chance of not being issued one. If it's a traffic stop, then it's a 66% chance of being issued a citation and a 61% chance of not.

Confusion Matrix:

	No	Yes
No	356	135
Yes	286	507

It has an 67.2% accuracy, meaning it can correctly predict approximately 67% of all cases. It has an 78.9% precision, meaning the model predicts "Yes" correctly approximately 80% of the time. 63.9% recall means it correctly identified 64% of all actual "yes" cases. 72.6% specificity means it correctly identifies the actual "No" cases approximately 73% of the time. An F1-score of 70.5% means the model strikes a reasonable balance between identifying "Yes" cases and avoiding false positives.

Overall, not great accuracy, so there could be improvements.

D. LASSO

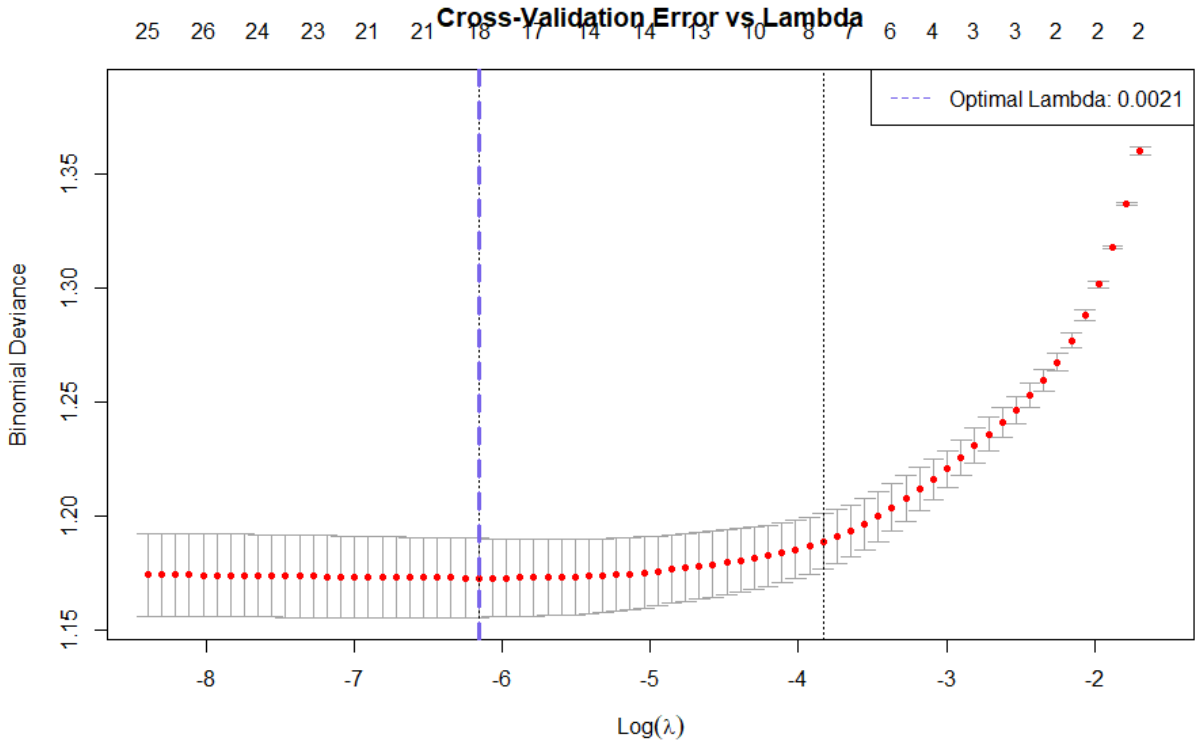

Figure 8: Cross-Validation Error vs λ

Originally, LASSO was chosen, because when I was running the data, it told me there was high multicollinearity within. With this method, the best lambda was 0.00211598, which is extremely small. It retains more features, and it indicates that the regularization is almost negligible. So, the LASSO model is behaving more like an OLS regression model. However, there's a risk that the model may be overfitting the data.

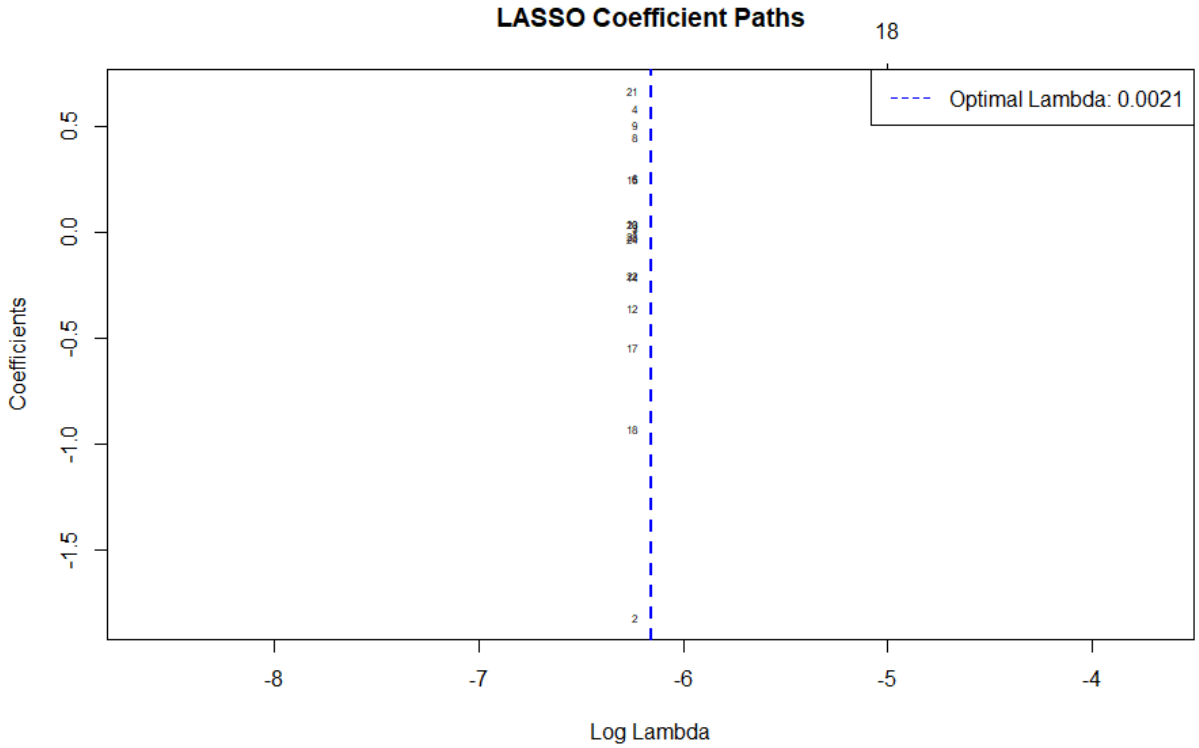

Figure 9: LASSO Coefficient Paths

It did shrink categories within these variables: date, preRace, race, gender, and month. And we ended with this confusion matrix:

Confusion Matrix:

	No	Yes
No	384	139
Yes	258	503

It has a 69.2% accuracy of correctly classified instances, nearly 70%. The precision is 78.3%, recall is 66.1%, specificity is 59.8%, and F1 score is 71.6%. LASSO performs better than the decision tree, but not by much. There could be improvements for specificity, the model's ability to correctly predict the "No" class; and recall, which suggests that the model may be missing some positive cases.

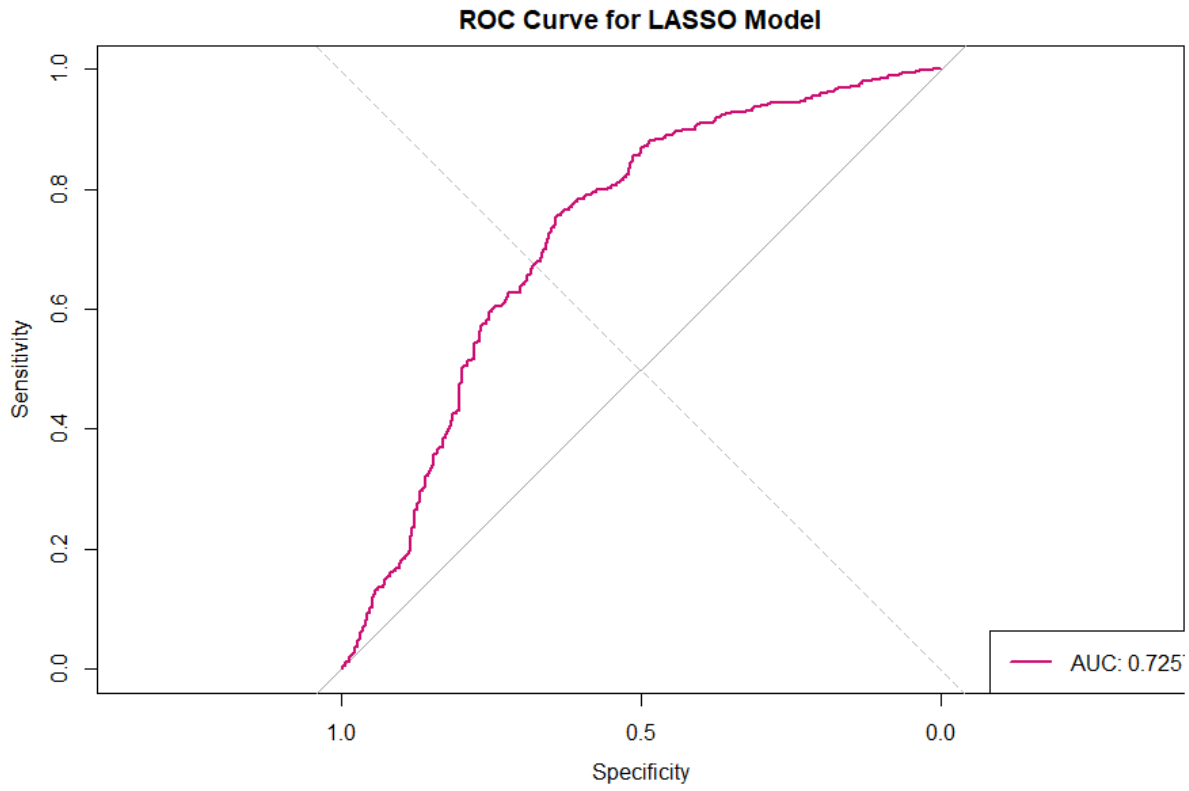

Figure 10: LASSO ROC Curve

Despite the relatively low accuracy, the model has a moderate discriminatory power, meaning it can reasonably distinguish between the classes. In practice, this AUC value is considered acceptable, but I would prefer it to be higher. Also, the curve isn't exactly smooth. This could suggest one or more underlying problems: model instability, high threshold sensitivity, class imbalance, model calibration issues, too few positive/negative instances, or extreme thresholds. However, I did handle the class imbalance, so let's look over the predicted probabilities.

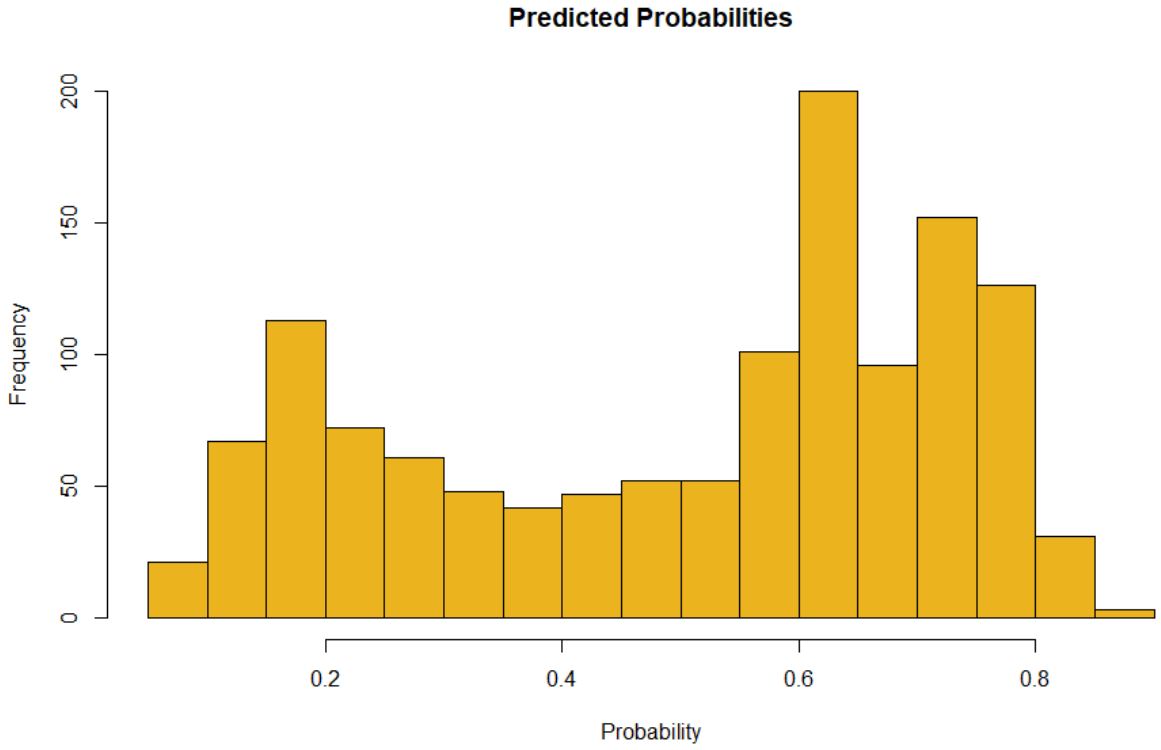

Figure 11: Predicted Probabilities

They are skewed towards the right, so the model is heavily inclined to classify most instances as belonging to the positive class, so we may need to adjust the lambda or even adjusting the predicted probabilities.

VI. Conclusion

Overall, there could have been more work done on this dataset. In PCA, the total variance can be explained by five principal components, specifically 93.15%. In K-Means Clustering, it's noted that both searches are significant, albeit in opposite directions. Both the decision tree and LASSO method did not do too well in accuracy, needing major improvements.

I definitely should have made time to try bagging, boosting, and SVM, considering it is known that decision trees aren't known to perform well.

References

Arel-Bundock, V. (2020, June 9). *Minneapolis Police Department 2017 stop data*. R. https://vincentarelbundock.github.io/Rdatasets/doc/carData/MplsStops.html

Salamanca, G. (2024, September 25). *Asphodelian/2017-Minneapolis-police-stops*. GitHub. https://github.com/asphodelian/2017-Minneapolis-Police-Stops