Data Mining Final Project

1. Where is your dataset from and what type of data does it contain?

   I found my dataset from Kaggle, the link to the dataset is:

   https://www.kaggle.com/felix4guti/traffic-violations-in-usa/data

   What it contains is traffic violations in the USA over the time period of a few years and lots of information given from each incident, such as the location, car manufacturer etc.

2. Preprocessing steps you have taken to ensure the correctness and usability of your data.
   - Specifically, what preprocessing did you perform, why?
     - Since traffic violations not only include vehicles operated by people, they also include violations of people on foot, but I only want operated vehicles. So, within the “Vehicle type” column I have removed rows that are people violating traffic laws on foot.
     - Looking at the counts for each attribute, some information is missing on things like “Geolocation”, "Driver City", "Driver State", "DL State", "Arrest Type", "Geolocation", “Color”, “Article”. This could be seen since the max count is 5906 and the count for those attributes is less than that.

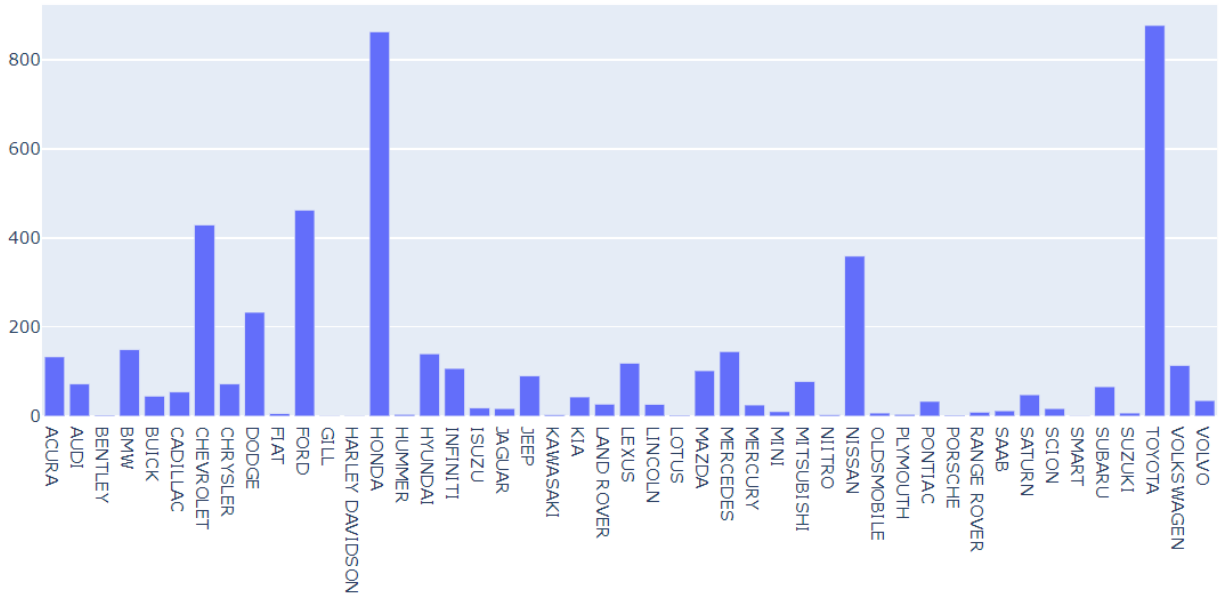
```
VehicleType                5906
Year                       5906
Make                       5906
Model                      5906
Color                      5885
Violation Type             5906
Charge                     5906
Article                    5809
Contributed To Accident    5906
Race                       5906
Gender                     5906
Driver City                5902
Driver State               5906
DL State                   5889
Arrest Type                5906
Geolocation                5309
```

- Did you have to fill-in missing values? Why? How? Was there a "fill-in" technique that gave better results than some other technique?
    - There isn't a good way to fill in the attributes that are missing because they are all nominal (strings) and would need further research. But since this dataset is so big, dropping the rows with the missing values will not have a large effect on it.
    - Many of the car manufacturer names have been misspelt, so I had to find all the misspellings and locate them then change them to the correct spellings, this reduced the unique count by about 100! Since there were so many and the only way to do them is by hand, I had to make a for loop for each car manufacturer and put a list of all the misspellings and set them to the correct ones. As for the names that are empty or ones that I cannot understand, I ended up dropping them as that was the best choice for that instance.
- Normalization or Standardization?
    - None has been needed
- Discretize continuous values?
    - The "Time of Stop" has been discretized, because for the information I'm trying to get out of it, having something like Afternoon and Midnight etc. is better than the time down to the seconds. Morning is from 6:00 to 12:00. Afternoon is from 12:01 to around 17:00. Evening is from 17:01 to 20:00. Night is from 20:01 PM until 5:59.

- I have not found the need to discretize any other continuous values like the dates because I have no use for them during this project.

3. Using a broad range of descriptive statistics, describe the nature of your data using graphs, plots, measures of central-tendency, descriptions of the distributions, histograms for categorical data, frequent item sets, extensive correlation analysis, similarity metrics, etc. Be sure to include relevant or important visualizations in your report. This may include, but is not limited to, distribution plots, histograms, bar-plots, scatter plots, similarity heat-maps.

    - Looking at the statistics of the traffic violations, it is clear that some makes of cars will have a higher count of violations in the US. As can be seen in the chart below:

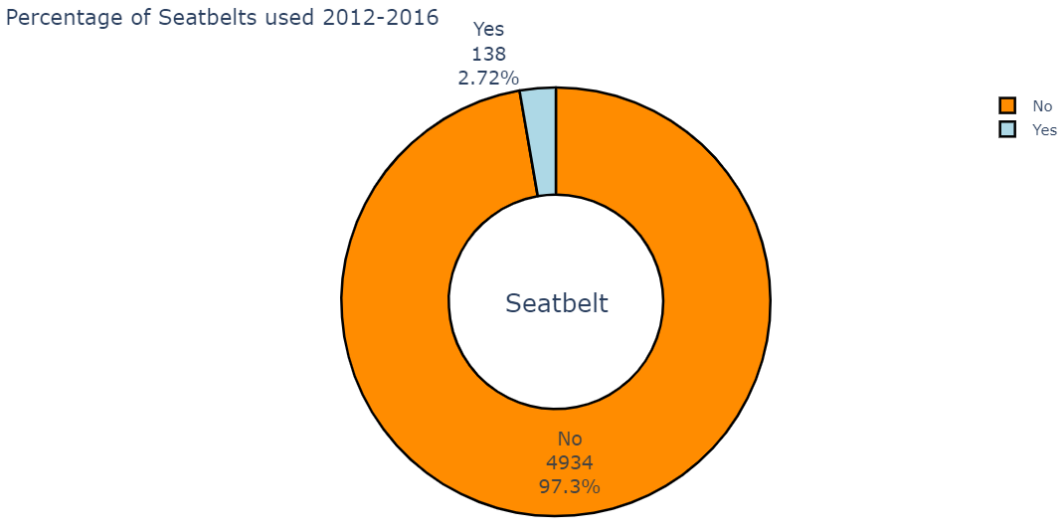


From the looks of this chart, it is shown that the top 5 car makes that cause traffic violations (in order) are: Toyota (877), Honda (863), Ford (462), Chevrolet (429), and Nissan (359). Now this dataset has data recorded from the years 2012 to 2016, and during those years according to an article from Daily Kanban, the car brands that sold the most cars in order are Toyota (13.7% of cars sold), Chevrolet (11.1% of cars sold), Honda (10% of cars

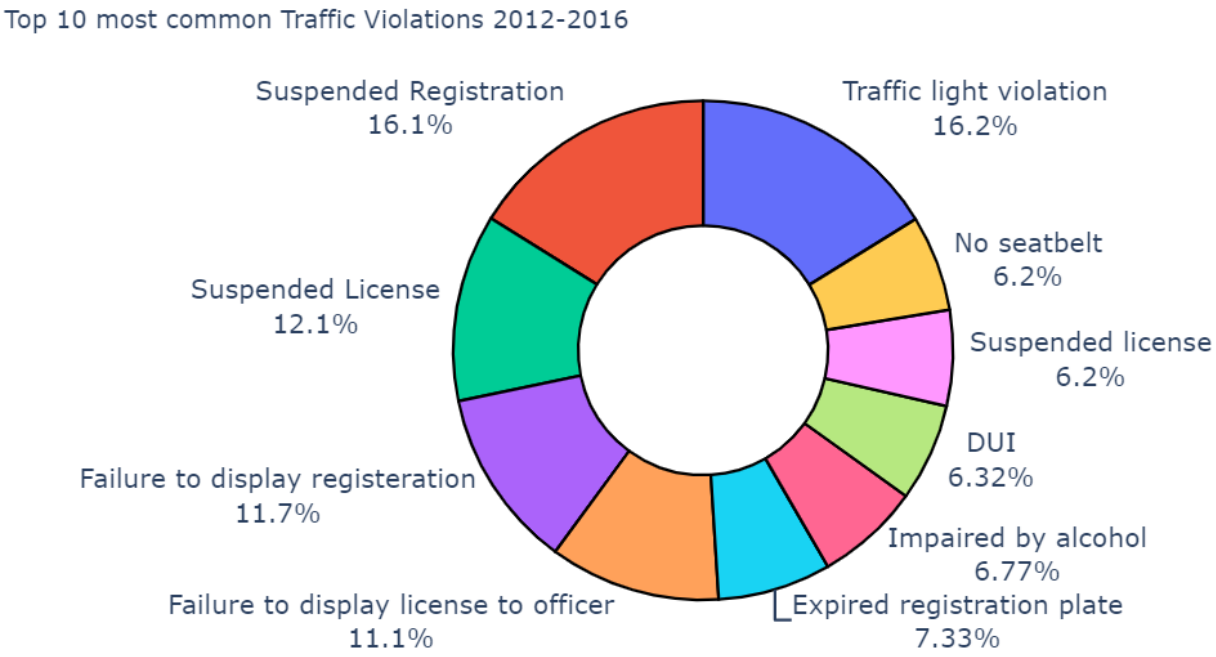
sold), Nissan (10% of cars sold) , and Ford (9.6% of cars sold). While of course older cars (not sold in those years) could be causing traffic violations, it is clear there is at least some time of connection since the 5 most sold car brands are the same 5 brands that cause the most traffic violations; even though they are not in the same order. However, an interesting thing to notice is that Toyota leading in car sales during those years and leading in the most traffic violations occurred.

What I had to do to get this the chart above is I took the count of the occurrences of each unique make name, and a list of all of the makes from the dataset. Then I had to set both sets of information into the same order so when they were put into the bar chart they matched, since before doing so I ran into an issue of them not matching.

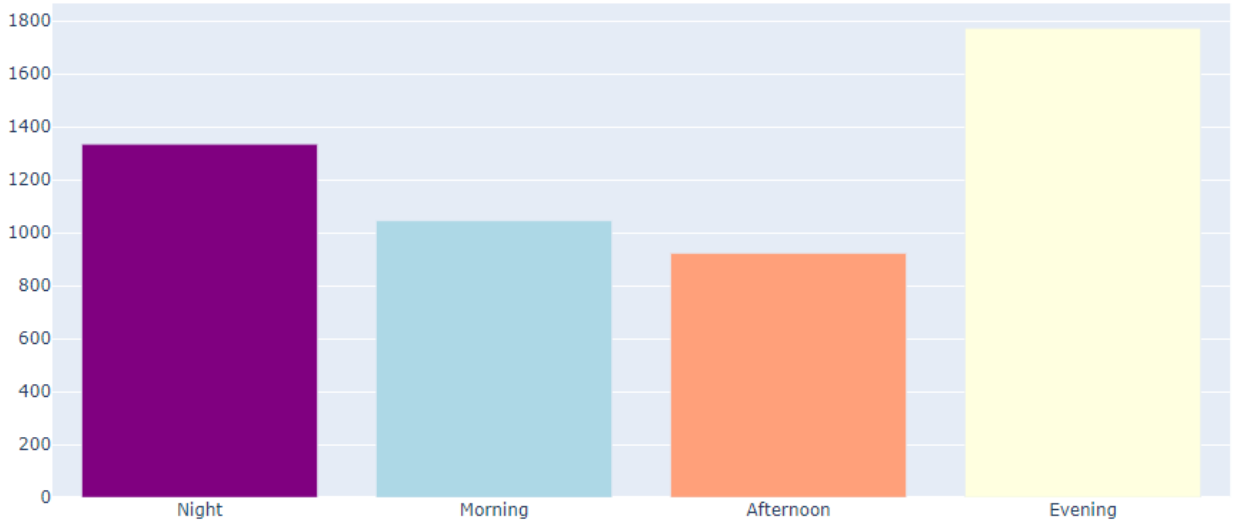
- Since this dataset includes if the person violating the traffic rules is wearing a seatbelt or not, I though it would be interesting to see what the percentage. And as it turns out, 97.3 % percent of traffic violators don't put on their seatbelt.



- While the percentage may seem staggering at first, it does make sense though considering that the dataset used is from traffic violations, and assumingly people that violate traffic laws will most likely not be wearing a seatbelt, and this information confirms that assumption.
- Part of this data set is a description given by the police officer of the violation, however the sentences are quite long to be easily displayed in a visualization, so I sliced the violation descriptions by occurrences, taking the top 10 most common violations. I then shortened the wording for each violation and visualized them into a pie chart so the violations can be easily seen and compared



- After considering the time of each violation recorded, and distributing them into bins, I thought it would be very interesting to see them visualized.



As can be seen, the most common time for a traffic violation to incur is during the evening, while the least to happen is during the afternoon. This fact makes sense since the evening is when rush hour happens, meaning more people on the streets driving which also means more chances for traffic violations.

4. Show results (training and test-sets) from at least one algorithm which you use for categorical prediction on multi-dimensional data (i.e. not single independent variable and single dependent variable). What was your motivation for using the algorithm? What assumptions did you make? What were the results? Did you try to improve the results? How was this done? Ideas about why it is performing well, or why it isn't.

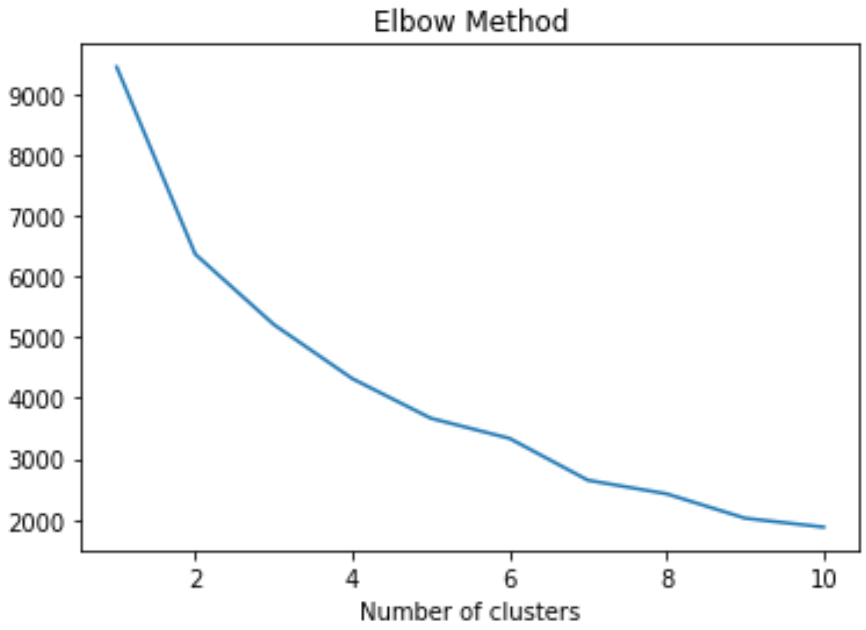
    - Since almost all of the data in this dataset is categorical, the only way to implement predicative algorithms was to encode the data somehow, after many long and excruciating tries to do so, I finally ended up encoding the data within two methods. Method one: was to equal the yes or no values to binary (1 and 0) for all of the data that

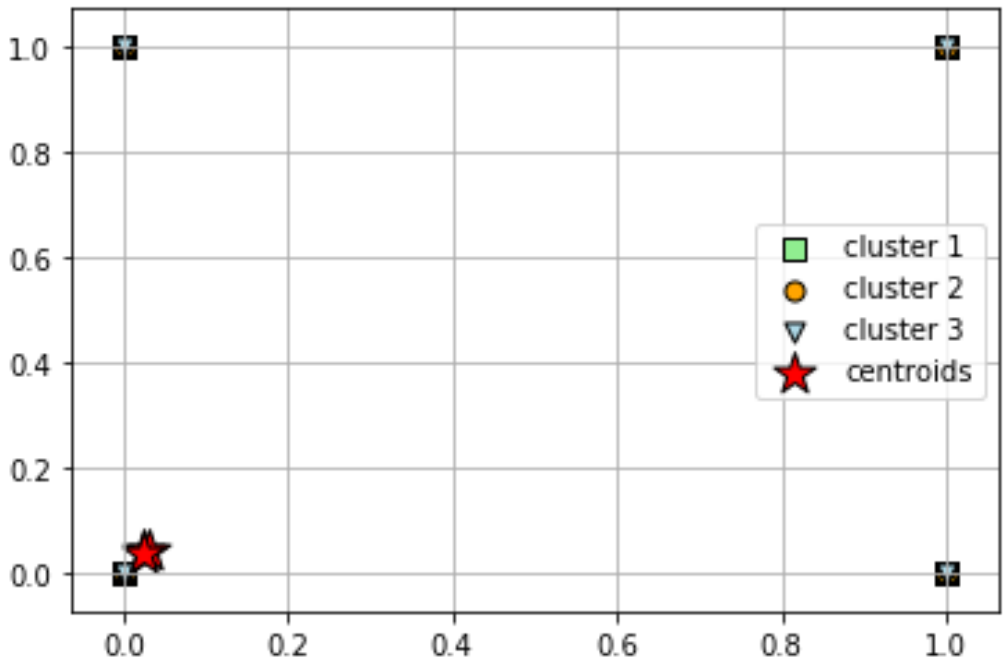
has yes and no for their value. Method two was to make panda dummies of the other data that was categorical and doing so caused other problems because it was creating unwanted linear relationships between variables that are supposed to be independent. So, after removing half of those, since they were essentially multiplied by two. I then chose a few variables that were all independent of each other, and one variable that was dependent for the program to predict. After that, I wanted to apply linear regression to the data using the sklearn linear model. First, I had to train test and split the data into X and Y, using 20% for the test size, after that I fitted the X train and Y train into the linear model in order to predict the dependent variable that I had chosen earlier. After running the code, I was met with the predictions and ran a test on it, specifically the R^2 score test. The score value given was a 0.02523877306955369 which is not good, because the closer the R^2 score is to 1, the better. So that meant the algorithm was not preforming well for my dataset and I can only assume the reason has to do with the encoding. I fully expected to get a higher score since the given variables were not dependent on each other and all of them are dependent by predicted variables.

5. Show results (training and test-sets) from at least one clustering algorithm. How can you interpret the results and what might they be useful for? (e.g. training curves, scatter plot of resulting clusters, etc.

    - For this clustering problem, I had no idea what to algorithm to use with this all categorical dataset, K means clustering is the go-to algorithm when it comes to this stuff but that only works with non-categorical data. I looked online and it was recommended to use the one hot encoding method then using a “Kmeans” clustering algorithm on top of it. The result is expected since the data has been one hot encoded.

- Using the elbow method, and some testing with a different number of clusters, 3 came out to be the best choice.





6. Show results (training and test-sets) from at least one algorithm for continuous value prediction used on multi-dimensional data (i.e. not single independent variable and single dependent variable). Why did you choose the algorithm? What assumptions did you make? Was it appropriate for the data you are using? Were the results good or bad? Why? Did you try to improve the results? How? Include relevant graphs (e.g. training curves) If your dataset is entirely categorical, you may skip item 6 for continuous value prediction. Instead, you must implement a second algorithm for multi-dimensional categorical prediction. Explain how these results differ from the first. Why do you think this is the case? Include relevant plots and explanations.

    - Since the last attempt at using a predictive didn't go so well, I had to change the algorithm for something that should work for an all categorical dataset, so I came to the conclusion that I could use Logistical Regression with the help of some encoding. In order to set a good comparison from the last attempt at using a predictive algorithm, I used the exact same dependent and independent variables. So I took those variables and their information and put them into a train test split again, using the same test size as last time (20%). After fitting, I called the logistic regression algorithm and predicted the new values. Then using the SKLean metrics, accuracy score I was given a 0.7172413793103448 which is way better than last time. I have also noticed that changing the dependent variables makes this score higher but not for the other test from earlier.

- Logistic regression models are specifically designed for analyzing binary data and categorical data. Which is why the accuracy score was so much higher when using logistical regression in comparison to linear regression, since all of the data in this dataset is categorical with some binary.

Takeaways from this project:

- While cleaning data is not supposed to easy, for this project specifically it was time consuming. Cleaning the names of car manufacturers was time consuming because there will be many misspelt variations of one name, for example the name Toyota had around eleven different spelling variations in this dataset.
- Using one hot encoding can give out the wrong data if used incorrectly, for example if there is a binary column that is one hot encoded, when analyzed, it could give out misinformation because the algorithm will make two columns, one for 'yes' and one for 'no' when only one is needed.
- When wanting to predict data from a dataset, having a categorical only dataset makes it a lot harder than for example an all numerical dataset, or even a categorical and numerical dataset.
- Linear regression is not a good choice as an algorithm to use when wanting to predict categorical data, the best choice for predicting categorical data is logistical regression.

Takeaways from the data

- Toyota cars have the most traffic violations in this dataset.
- The most common time for a traffic violation is during the evening.
- A person violating traffic laws will most likely not be wearing a seatbelt.
- The most common violation type is a traffic light violation (ex. Running a red light)
- The attributes "Contributing to an accident" and "causing property damage" have the highest correlation in this dataset.

Sources:

Analysing Categorical Data Using Logistic Regression Models
Sarah Littler - https://select-statistics.co.uk/blog/analysing-categorical-data-using-logistic-regression-models/

K-means Clustering Python Example
Cory Maklin - https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203

Examples Of Numerical and Categorical Variables
365 Team - https://365datascience.com/numerical-categorical-data/

Train/test Split and Cross Validation in Python
Adi Bronshtein - https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

Simple and Multiple Linear Regression in Python
Adi Bronshtein - https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9

Visualizing Data with Pairs Plots in Python
Will Koehrsen - https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166

Dealing with Categorical Data Fast - an Example
Samir Gadkari - https://towardsdatascience.com/dealing-with-categorical-data-fast-an-example-d4329b44253d

Clustering Based Unsupervised Learning
Syed Nazrul - https://towardsdatascience.com/clustering-based-unsupervised-learning-8d705298ae51

An Introduction To Clustering Algorithms in Python
Jake Huneycutt - https://towardsdatascience.com/an-introduction-to-clustering-algorithms-in-python-123438574097

Similarity Measures
Saif - https://towardsdatascience.com/similarity-measures-e3dbd4e58660