DATA 606: Capstone Term Paper

Introduction

The Baltimore Police Agency launched a massive overhaul to its new Records Management Systems in May 2020. This improvement enabled the department to migrate from a paper-based system to a completely digital reporting environment. As a consequence of this major shift, we had significant difficulties in appropriately transferring data from the new records system to the previous Open Data Baltimore system. The "Arrests" dataset is one of many open datasets made publicly accessible by Baltimore's police department on the city's Open Data website (https://www.baltimorepolice.org/crime-stats/open-data). This information is provided to us in an effort to foster greater openness and data exchange between the local administration and the residents of the city. This dataset contains arrest records for offenses such as assault, theft, and property damage in the City of Baltimore. [1]

Our business question: What methodology that we've learned over the course of our graduate student journey will provide the most accurate model in the least amount of time? To answer this question, we sought to use their database to create a model that can predict which district a crime occurs in based on various details related to the arrest of the perpetrator, such as what he or she was charged with, their gender, etc. Our hope with this model is that the department can then use this model to efficiently spread their resources, tackling the more likely arrests that would be made in a certain district. In addition, we hope that the department can modify their policing efforts to decrease bias in said policing efforts in certain districts (if any are found).

Accessing the Data

We begin where all great classification problems begin, acquiring the data. Due to different strengths of our computers, we used a mix of anaconda, jupyter lab and google colab to access the data. We used a simple .read_csv method and checked the general info of the dataframe so we're aware of what we're working with.

Features

The following are a complete list of the raw features at our disposal:

RangeIndex: 175923 entries, 0 to 175922

Data columns (total 20 columns):

#	Column	Non-Null Count	Dtype
0	X	90128	float64
1	Y	90128	float64
2	RowID	175923	int32
3	ArrestNumber	167697	float64
4	Age	175836	float64
5	Gender	175884	object
6	Race	175884	object
7	ArrestDateTime	175923	object
8	ArrestLocation	91488	object
9	IncidentOffence	175923	object
10	IncidentLocation	91488	object
11	Charge	156541	object
12	ChargeDescription	175923	object
13	District	89891	object
14	Post	89891	object

15	Neighborhood	89870	object
16	Latitude	90128	float64
17	Longitude	90128	float64
18	GeoLocation	175923	object
19	shape	0	object

---

dtypes: float64(6), int32(1), object(13)

memory usage: 26.2+ MB

Data Cleaning and Exploratory Data Analysis (EDA)

Exploratory Data Analysis refers to “the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. Exploratory Data Analysis, or EDA is one of the most important steps in Data Analysis. During this step, the data analyst thoroughly examines the dataset to ensure it is in the form they need for modeling.” [2]

After the initial investigations into the data, the first thing we noticed is that there are some features that will not contribute to our classification problem:

- 'X', 'Y', 'RowID', and 'ArrestNumber' are all similar indexing variables that provided nothing to the classification
- 'ArrestLocation', 'IncidentLocation', 'Neighborhood': This didn’t fit the classification problem, as our model shouldn't need location information to determine which district the crime happened in

- 'Latitude', 'Longitude', and 'GeoLocation' all provide the same information. This didn't fit our classification problem, as our model didn't need location information to determine the district the crime happened in
- 'Shape': Shape is a feature column that provides no information to us, and is mostly filled with NA values
- 'Charge', 'post': Charge and Charge Description hold nearly identical information. Post is a 3 digit code that relates to one's charge. Thus, when we use OHE on them, they will provide redundancy, which we don't want.

Therefore, we dropped those columns before moving on with the EDA.

For our analysis, it is better to have the Date and time of the arrests in separate columns instead of having a timestamp. We split the ArrestDateTime column into two columns and dropped the column. Now we have our final dataframe, which we used for our analysis.

	Age	Gender	Race	IncidentOffence	ChargeDescription	District	date	time
0	27	M	B	96BINVESTIGATIVE STOP	HAND GUN VIOLATION	Southwest	2020-12-31	23:50:00
1	45	F	B	Unknown Offense	AUTO THEFT	Western	2020-12-31	23:45:00
2	42	F	W	Unknown Offense	2ND DEGREE ASSAULT	Southwest	2020-12-31	23:40:00
3	26	M	B	Unknown Offense	PWID	Southern	2020-12-31	21:45:00
4	19	M	B	Unknown Offense	PWID	Southern	2020-12-31	21:45:00

Dealing with missing values (if any):

Now that we have all the relevant columns for our classification, we need to ensure that the values in said column are not going to cause any modeling pipeline issues. We began by checking how many NA values there are in each column.

```
print(dfConverted.isna().sum())
```

```
Age                   87
Gender                39
Race                  39
IncidentOffence        0
ChargeDescription      0
District           86032
date                   0
time                   0
dtype: int64
```

We had NA values in Age, Gender, Race and District. Due to the sheer size of the data, we filled those with a value to denote that they were "unknown". However, if they already had a value denoting "unknown", that would muddy the purity of the data. So, before making that modification, we checked for unique values for gender and race.

- If **age** has no "unknown" value already then substituting 0 for all

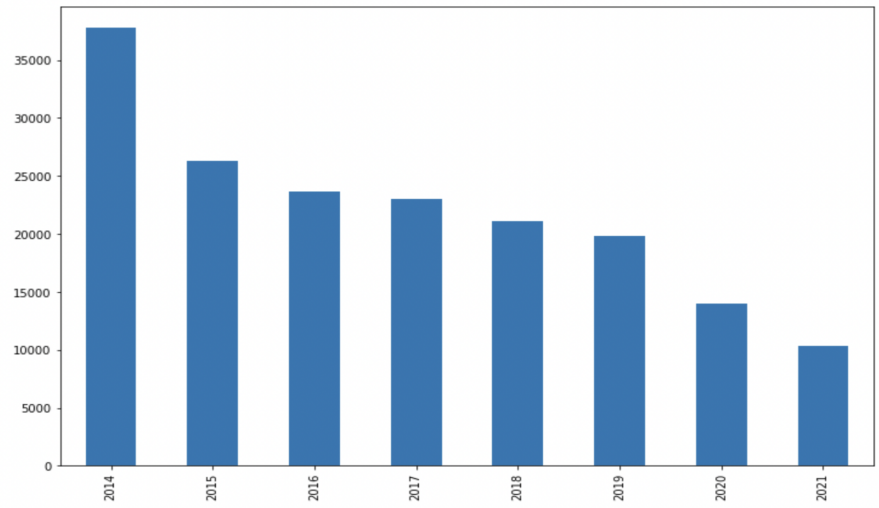
We used non-positive age.

- We checked this by making sure all ages are positive numbers
- Gender should have 2 unique values. A correctional officer told us that any unknown values are most likely due to gender being unknown at the time of the arrest, so any different values were considered unknown.
- Similarly, we checked the unique values of each of the values in race. We found values that all relate to a particular race and/or contain a letter denoting "other race". We determined our "unknown" hypothesis to be true, and modified the NAs accordingly.

After using appropriate means to handle the NA values in all our columns, we have our final dataframe which was used for our analysis.

Year:

Now, let us see the number of arrests made each yeart:

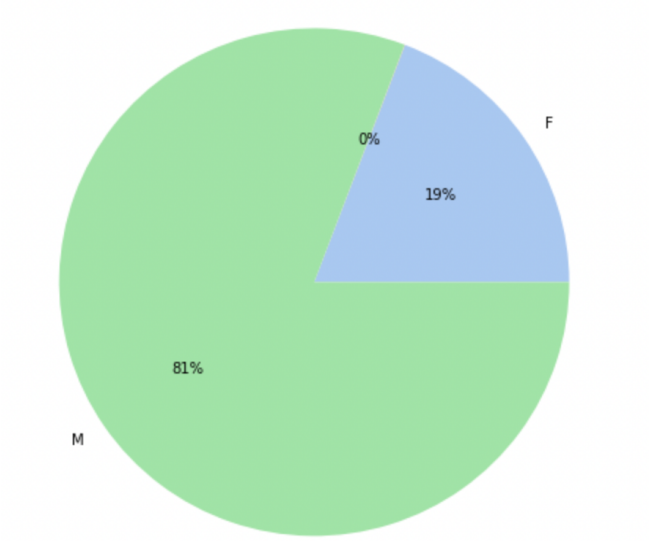


From the plot we can see,

- After cleaning the data, the number of arrests made each year are decreasing over the years. T
- The largest number of arrests was made in 2014.

Gender:

Now, let us see the percentage of male and female arrests made:

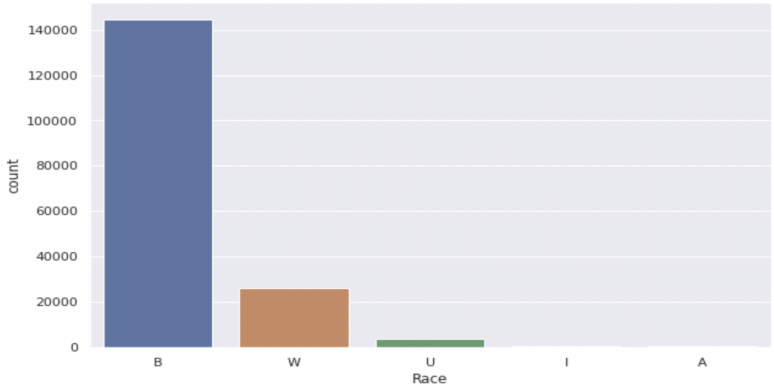


From the plot we can see,

- 81% of arrests have male arrestees.
- 19% of arrests have female arrestees.

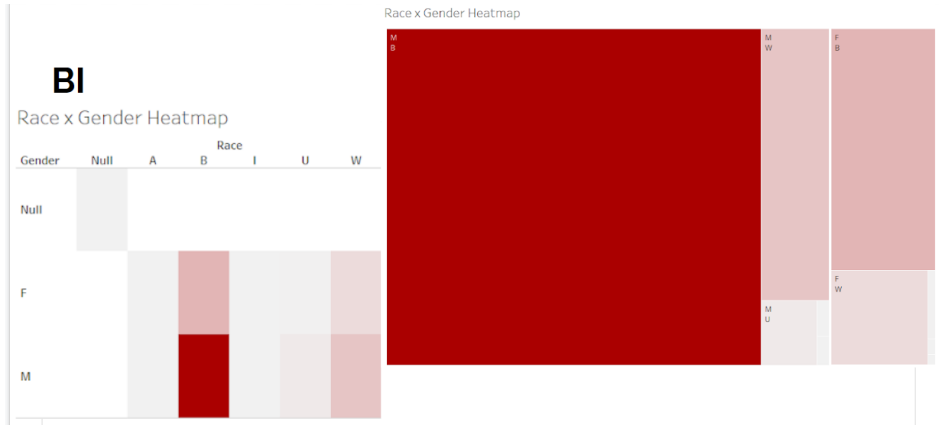
Race:

Now, let us the see races of the people who are arrested made:



From the plot we can see,

- A disproportionately large majority of people arrested are black, followed by white, with all other values trailing behind.
- Here, U represents race unknown at the time of the arrest.
- Arrests made of Indian and Asian people are very low.

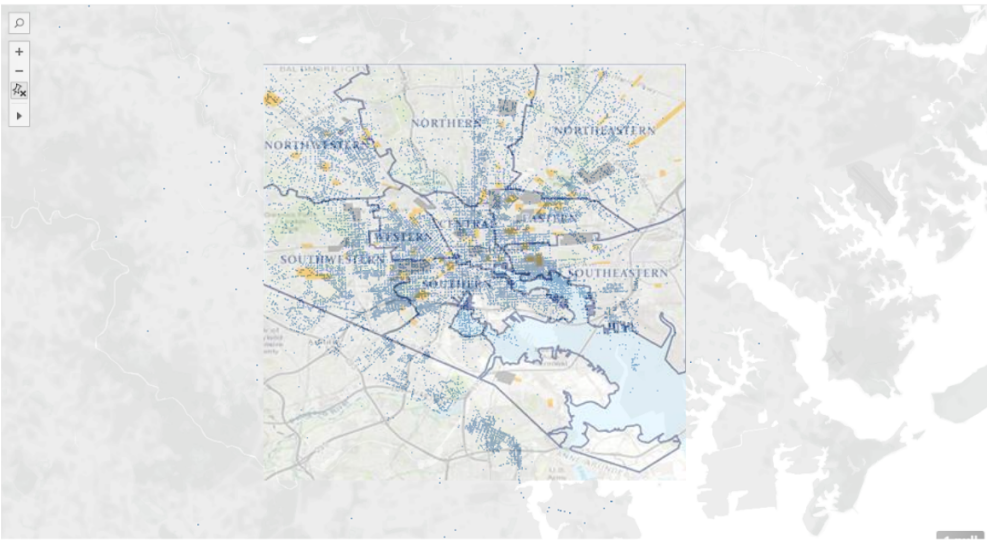
- 

As we can see from the above Race x Gender Heatmap, the majority of arrests made are of Black males followed by White males and Black females.

Geographic plot:

Next, we have plotted the each arrest made on the map using the location of the arrest made:

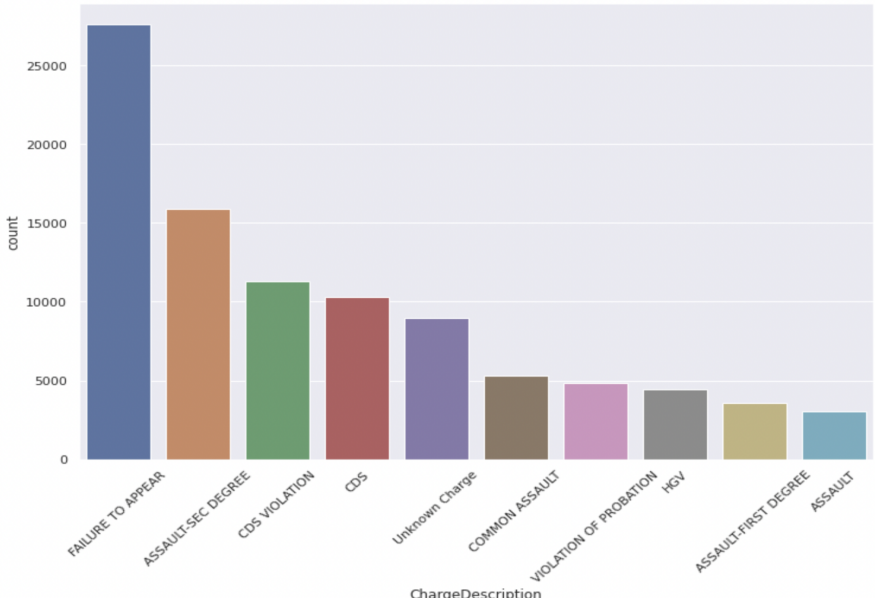
Visual representation of where arrests occur



From the plot, we can see each arrest represented by a blue dot on the map, across the different districts. This is to get a visual representation of the different districts in Baltimore and the spread of arrests in each district.

Charge:

Now, we will see on what are the different charges most of the arrests are made:



From the plot we can see,

- These are the top 10 charges on which most of the arrests are made.

- Most of the arrests are made because of the failure of the person to appear at the court, people are missing their court dates, which is leading to their arrests.
- Assault-Sec Degree and CDS violation also had a very high number of arrests.

Dataset basics:

We began by gathering basic info on our new dataframe by checking the 5 number summary for each of the features

```
print(dfConverted.dtypes)
print(dfConverted.describe(include='all').T)
```

```
Age                    Int64
Gender                string
Race                  string
IncidentOffence       string
ChargeDescription     string
District              string
date                  object
time                  object
dtype: object
                       count unique                top  ...   50%   75%    max
Age                 175923.0    NaN                NaN  ...  30.0  40.0  100.0
Gender                175923      3                  M  ...   NaN   NaN    NaN
Race                  175923      6                  B  ...   NaN   NaN    NaN
IncidentOffence       175923    172    Unknown Offense  ...   NaN   NaN    NaN
ChargeDescription     175923   7028  FAILURE TO APPEAR  ...   NaN   NaN    NaN
District               89891      9           Southern  ...   NaN   NaN    NaN
date                  175923   2866         2014-01-09  ...   NaN   NaN    NaN
time                  175923   1440           11:00:00  ...   NaN   NaN    NaN

[8 rows x 11 columns]
```

```
print("Arrest Data is logged from: ", dfConverted.time.min(), "to ", dfConverted.time.max())
```

```
Arrest Data is logged from:  00:00:00 to  23:59:00
```

```
print("Arrest Data Is collected All day from: ")
print(dfConverted.date.min())
print("To")
print(dfConverted.date.max())
```

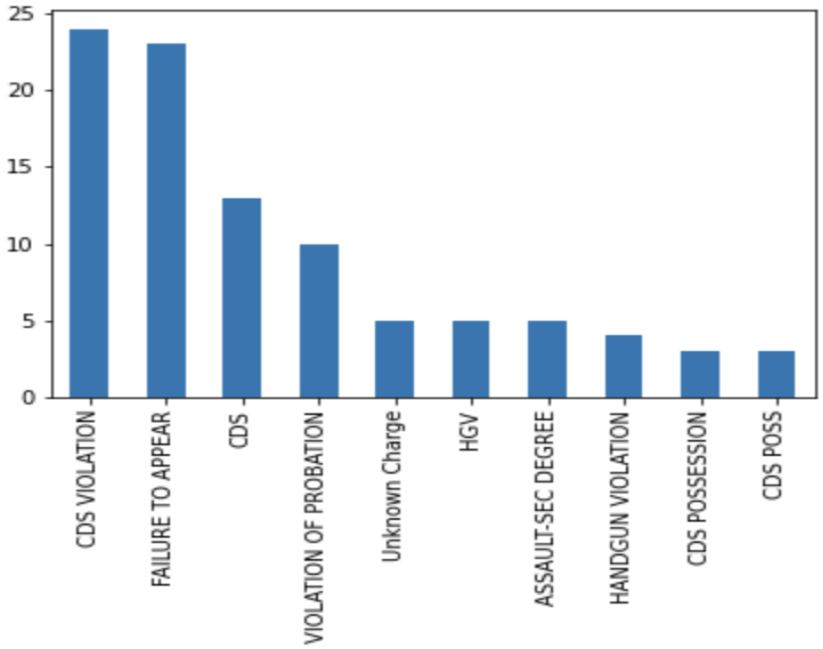
```
Arrest Data Is collected All day from:
2014-01-01
To
2021-11-05
```

5 Number Summary Analysis:

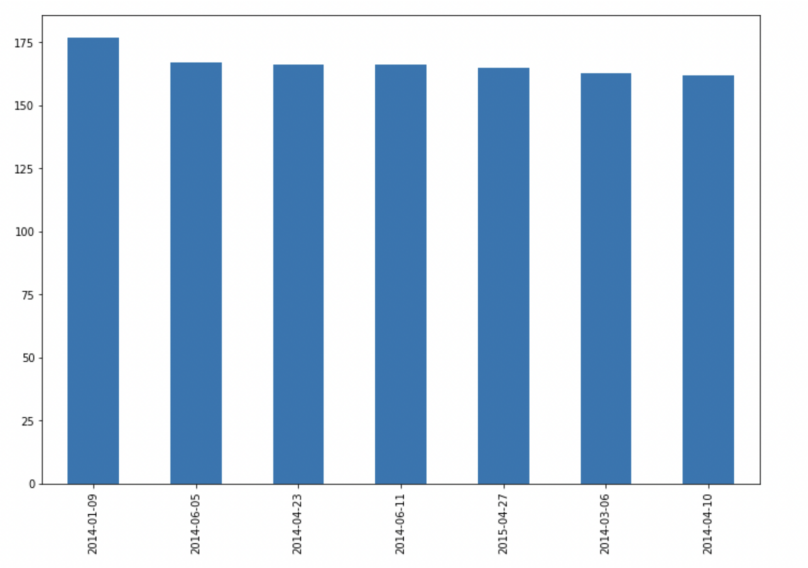
From the 5 number summary alone, we acquired a wealth of information:

- The most frequent reason for being arrested is not showing up to your appointed court date. Whether it's a major or minor offense, one should always do their best to make their court date.
- The most common time to be arrested is around 11am.
- The highest number of arrests made on a single day was on January 9th, 2014
- In the 7 years this data spans, black males are the most likely to be arrested. Published articles give credence to policing bias contributing to this trend[4]

What happened on January 9th 2014? Is the question we get when we see that the most number of arrests have been made on that day. Now we will visualize to gain insights from the arrests made on that day:

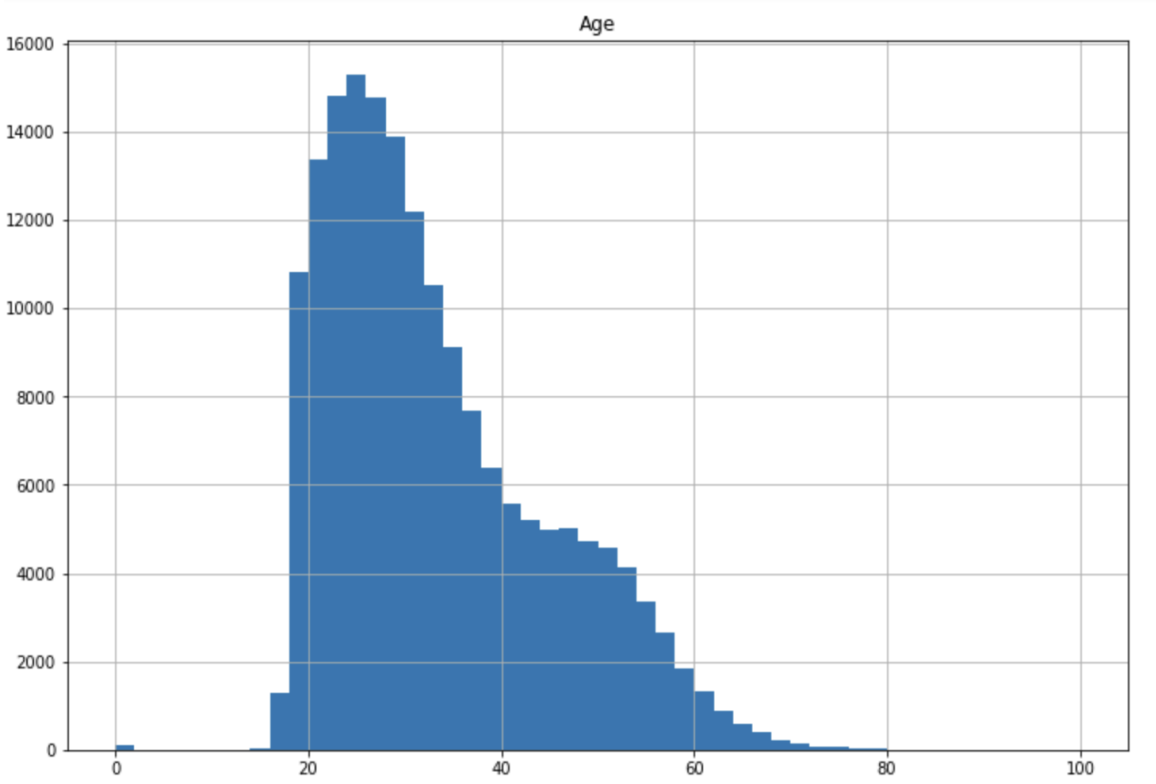


Disregarding "Failure To Appear", it seems most of these arrests were CDS, aka Controlled Dangerous Substance charges. This makes sense, as people are coming off of the holidays at this point. Perhaps they are trying out their new "presents".



This is further backed up by the chart above. Arrests seem to occur:

- a couple days following holidays like christmas or easter
- a couple days after pseudo holidays like 4/20 (weed day)
- and in warmer months like June when people are more out and about.



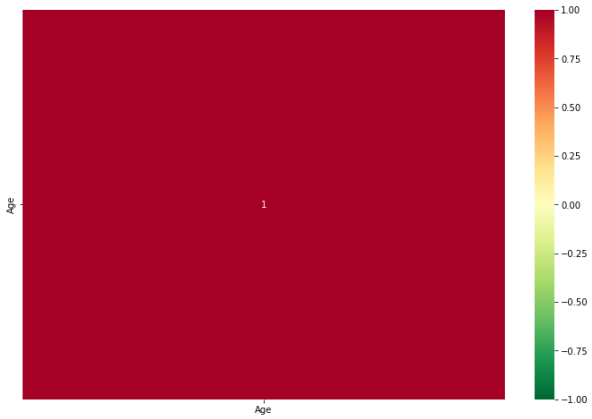
Most arrests are being made on those in their mid twenties, with a less noticeable hump occuring in their late 40s.

There are many factors that can play into why people in their mid 20s are being arrested as a professor of criminal justice told the New York Times[4]

Multicollinearity check:

Decision trees were one of the methods we wanted to apply, so we needed to make sure that the columns were not highly correlated to each other.

The first thing we checked for is multicollinearity in the numerical data:



We only have 1 numerical variable, so it would make sense that it is fully correlated with itself. Next, we checked for multicollinearity with our other variables. According to Deepanshu Bhalla of Listen Data [3], "For categorical variables, multicollinearity can be detected with Spearman rank correlation coefficient for ordinal variables and chi-square test for nominal variables."

A nominal variable has no inherent ordering. Race, for example, is a categorical variable because there are 6 categories (W, B, A, I, U, and most recently O) and they can't be placed in an order naturally.

A variable with an ordinal value has a distinct ordering. The date and time columns would both be ordinal, as the objects have a clear timeline. i.e. 2014-01-09 comes before 2014-01-10 and 03:40:00 is earlier than 11:00:00

chi-square test: The easiest way to perform a Chi-Square Test is to create a contingency table from the frequency of values[5]. We began by checking the multicollinearity between Gender and Race.

After performing the chi-square test, we got a p-value of 0, which means that we do not reject the null hypothesis (Gender and Race are independent) at 95% level of confidence[5]. Therefore, we now know that Gender and race are independent from each other.

Next, we compared the other nominal variables (incident offense and charge description) to determine their p statistics.

```
Pstat btwn Gender and Incident Offecnce is:  0.0
Pstat btwn Gender and Charge Description is:  9.044134031092239e-170
Pstat btwn Race and Incident Offecnce is:  0.0
Pstat btwn Race and Charge Description is:  1.4298290944869867e-265
Pstat btwn Charge Description and Incident Offecnce is:  0.0
```

All of our nominal variables had little to no multicollinearity with each other. As such, we decided that we could move ahead with our modeling.

Processing and Modeling

Before modeling, we built a pipeline to separate the District column, apply the Standard Scaler, one-hot encode the non-numeric features, and impute missing values. We then used train_test_split with a test size of .25, splitting the data into 67418 training and 22473 tests.

- To begin, we did logistic regression with grid search, then succeeded in further refining it using the best correlation strength.
- We attempted to use ADABoost, but it resulted in lower accuracy.
- Next, we tried a decision tree classifier, further refining it using max and minimum ranges.
- Our last unique model was SVM, which was refined using a grid search. SVM took an extremely long amount of time to run, but returned with the highest accuracy.
- However, we used pyspark to expand our options in parameters for logistic regression, which resulted in what was by far the highest accuracy model, while not taking too much longer to run than SVM.

Results and Conclusion

- The Logistic Regression model produced the second highest accuracy (23.6 percent) in around a minute and a half. It appears that attempting to change the Logistic regression resulted in a decrease in accuracy.
- The SVM model produced the highest accuracy (25 percent) in 1483.9 seconds, or a little under 25 minutes. This was after a preliminary grid search in an attempt to determine best parameters (taking nearly 24 hours). As this was already beginning to be prohibitively slow, even on a powerful home PC, further refinement was not considered.
- Since Logistic Regression had such high accuracy without having an extreme runtime, we decided to try it with different parameters using pyspark. This resulted in an accuracy of 50.02 percent in 3146.6 seconds (52 minutes 26 Seconds), more than doubling our previous model accuracy.

References

[1] "Baltimore Police Department." Crime Stats | Baltimore Police Department, https://www.baltimorepolice.org/crime-stats.

[2] Ijraset. "IJRASET Journal for Research in Applied Science and Engineering Technology." *An Enhanced Approach of Finding Probability of Road Accidents By Using ANOVA*, https://www.ijraset.com/research-paper/enhanced-approach-of-finding-probability-of-road-accidents.

[3] "Detecting Multicollinearity In Categorical Variables, https://www.listendata.com/2015/04/detecting-multicollinearity-in-categorical-variables.html

[4] Goode, E. (2011, December 19). *Many in U.S. are arrested by age 23, study finds*. The New York Times. Retrieved November 20, 2022, from https://www.nytimes.com/2011/12/19/us/nearly-a-third-of-americans-are-arrested-by-23-study-says.html

[5] Pipis, G. (2020, September 6). *How to run chi-square test in Python*. Predictive Hacks. Retrieved November 20, 2022, from https://predictivehacks.com/how-to-run-chi-square-test-in-python/