Predicting Traffic Issues for the Purpose of Delivery Route Planning

Cintas is a company that delivers clean uniforms to customers around the city. While GPS can help navigate routes in real-time, route planning is done in advance based on sequencing stops based on geographical considerations. The geography of a series of stops is not always the greatest predictor of what will complete the series of stops in the least amount of time. Can we predict traffic issues before they occur so that we may avoid them?

Criteria for Success: Reduce the time of truck routes by improving route planning
Scope of Solution Space: Saving time will reduce fuel, vehicle maintenance, and labor costs.
Constraints with solution space: Recommendations may conflict with other route considerations.
Stakeholders to provide key insight: Cintas route planners'

Data

The data was collected by Austin's Open Data Portal. This portal provides easy access to open data and information about your city government. The public data that the City of Austin has published is intended to spark innovation, promote public collaboration, increase government transparency, and inform decision-making. The specific dataset Real-Time Traffic Incident Reports contains traffic incident information from the Austin-Travis County traffic reports RSS feed, available at https://www.austintexas.gov/qact/default.cfm.

Data Wrangling

Original shape: 249650, 9

Column Title	Original Data Type
Traffic Report ID	object
Published Date	object
Issue Reported	object
Location	object
Latitude	object
Longitude	float64
Address	object
Status	object
Status Date	object

This dataset came straight from Austin's Open Data Portal and was in its raw form. Much of the data cleaning was necessary in order to allow functionality and analysis.

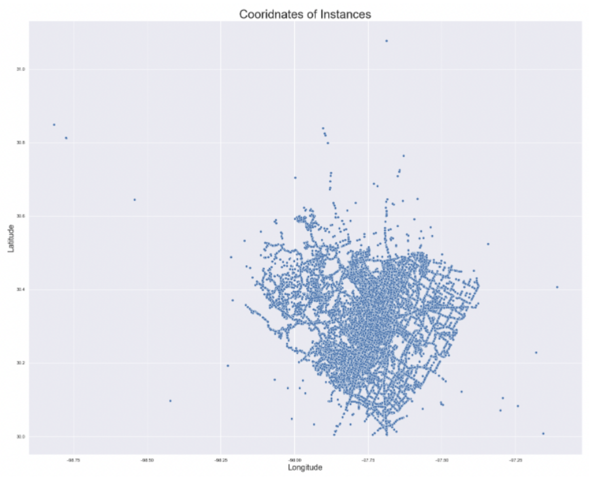
Problem 1: Column names that had more than one word in it had spaces in them
Solution 1: Rename applicable columns

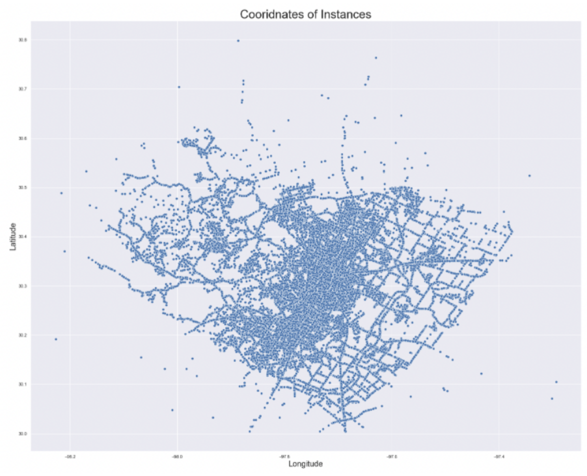
Problem 2: Latitude and the two date columns were listed as objects
Solution 2: Convert Latitude to float data type, import data using parse_dates

Problem 3: Null and 0.0 values in the Logitude and Latitude columns
Soultion 3: This only eliminated 589 rows

Problem 4: Several columns were deemed unnecessary after further analysis
Solution 4: Drop Traffic Report ID, Location, Address and Status columns

Problem 5: A few coordinates when plotted where well outside the city limits
Solution 5: Drop the 14 rows outside the following range:
(Latitude >30) & (df.Latitude < 30.8), (Longitude < -97.25) &(Longitude > -98.25)





Problem 6: Status Date column is useful only in that it helps provide a resolution timespan
Solution 6: Replace Status Date with issue_timespan which subracts the Published Day column from the Status Day column and convert the datetime64[ns, UTC] data type to a float represented in minutes. Some of these timespans represented negative time which is impossible so those rows were removed.

EDA

Question: How much time should an incident take to clear off the road?

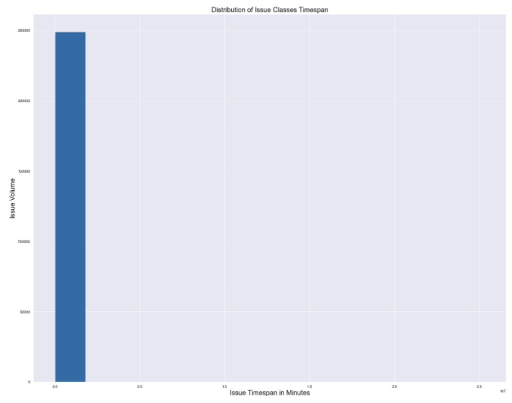
The mean time to resolve an incident is 3 days 04:57:48 according to the summary statistics. The max value is a whopping 233 days 08:49:00.

According to this legal website, the average wait time in neighboring Houston in 2011 was 33.5 minutes.

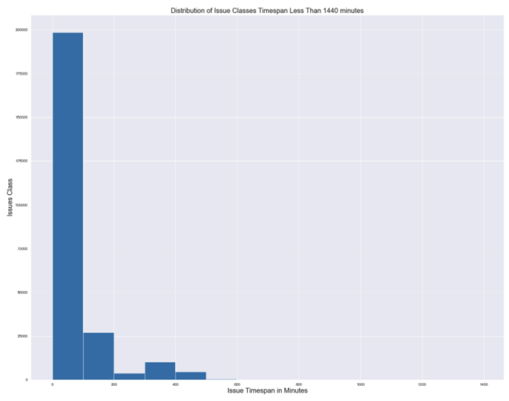
This article out of Washington state claims that the average wait time in 2018 was 13 minutes. The article goes on to say there was also an uptick in accidents that took over 90 minutes to clear. They also reference "extraordinary incidents" that take an average of 12 hours to clear.

Let's visualize a timespan with no limit, a 24-hour limit, and an 8-hour limit

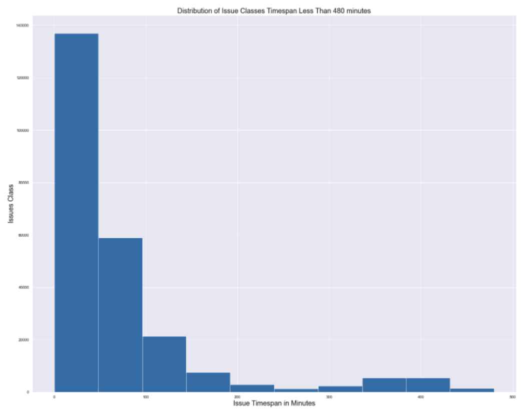
Histogram of all Timspan Values



Histogram of Timespan Values < 1440



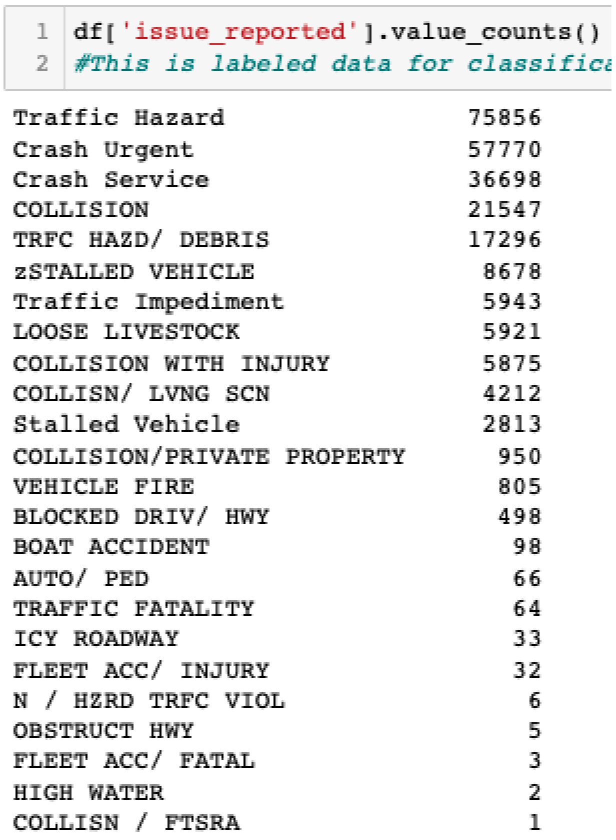
Histogram of Timespan Values < 480



We can at least see all the bins involved at 480 minutes. The vast majority of instances have a time span of fewer than 200 minutes. Cutting that many rows would degrade the dataset significantly. It's best to bring the total minutes to the equivalent of 8 hours.

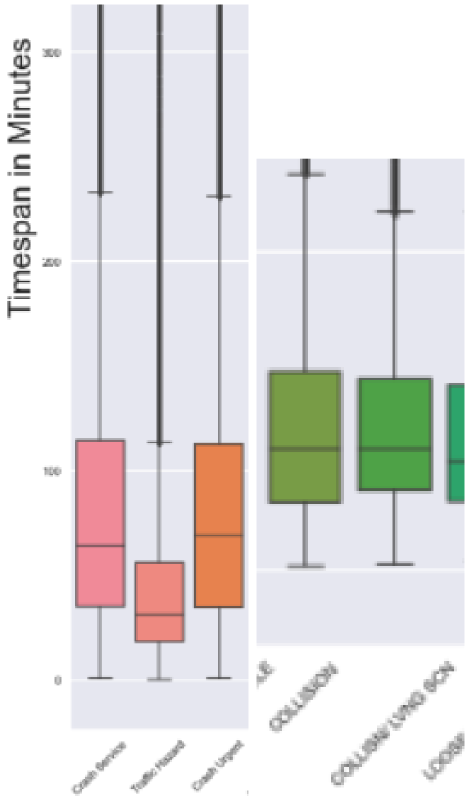
Question: Are some of the categories for issue_reported redundant or unnecessary?

Originally there were 23 different categories under the issue_reported columns.

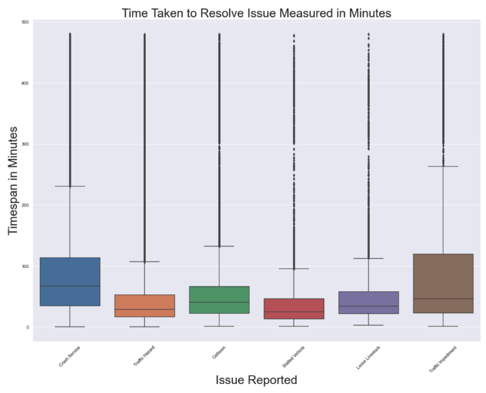


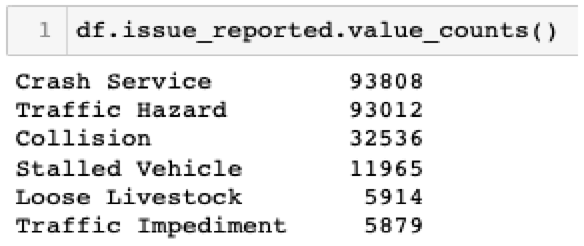
```
df['issue_reported'].value_counts()
#This is labeled data for classifica
```

```
Traffic Hazard                  75856
Crash Urgent                    57770
Crash Service                   36698
COLLISION                       21547
TRFC HAZD/ DEBRIS               17296
zSTALLED VEHICLE                 8678
Traffic Impediment               5943
LOOSE LIVESTOCK                  5921
COLLISION WITH INJURY            5875
COLLISN/ LVNG SCN                4212
Stalled Vehicle                  2813
COLLISION/PRIVATE PROPERTY        950
VEHICLE FIRE                      805
BLOCKED DRIV/ HWY                 498
BOAT ACCIDENT                      98
AUTO/ PED                          66
TRAFFIC FATALITY                   64
ICY ROADWAY                        33
FLEET ACC/ INJURY                  32
N / HZRD TRFC VIOL                  6
OBSTRUCT HWY                        5
FLEET ACC/ FATAL                    3
HIGH WATER                          2
COLLISN / FTSRA                     1
```



Looking closely at the categories we can see that some have very similar labels as well as similar timespan resolution. In a piece of the boxplot for all categories it is easy to see that Crash Service and Crash Urgent are close in name and in how long it take to resolve their issues. To help reduce the number of labels and make prediction easier it is necessary to combine these two labels and others like them. All issues with less than 1000 incidents will be dropped totalling around 1000 rows.





```
df.issue_reported.value_counts()
```

```
Crash Service         93808
Traffic Hazard        93012
Collision             32536
Stalled Vehicle       11965
Loose Livestock        5914
Traffic Impediment     5879
```

This plot is much easier to read and gives us a much better idea of what issues take a long time to resolve.

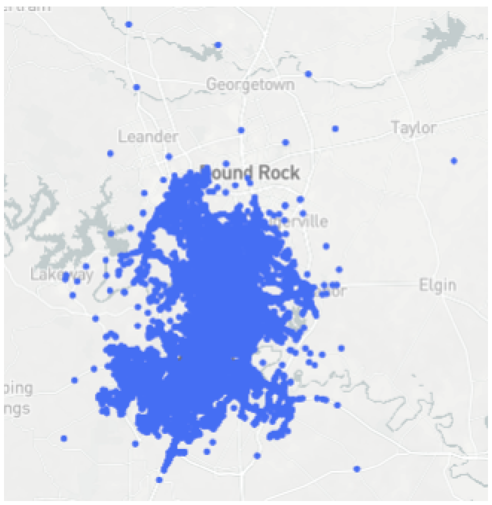
Takeaways:

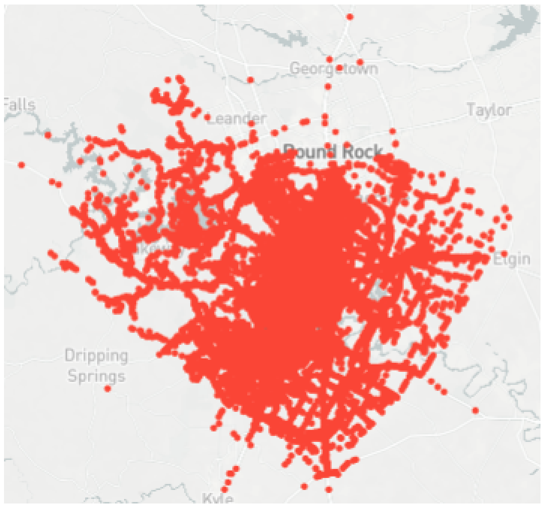
1. Crash Service and Traffic Impediment take the longest to resolve.
2. Stalled Vehicle and Traffic Hazard take the least amount of time to resolve.
3. Collision and Loose Livestock take a moderate amount of time to resolve.

What is the Geographic Distribution of the Different Issues?

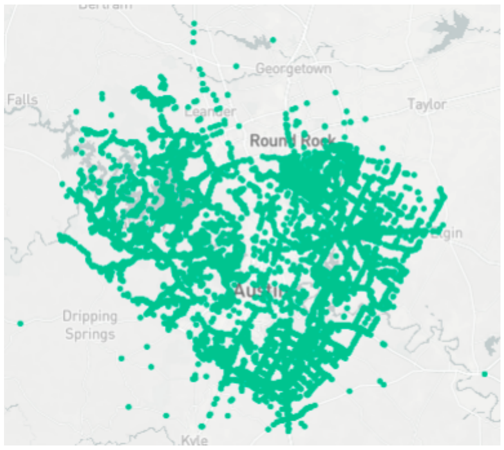
Perhaps examining the geographic distribution of the different types of traffic incidents will provide insights into how we can narrow our focus

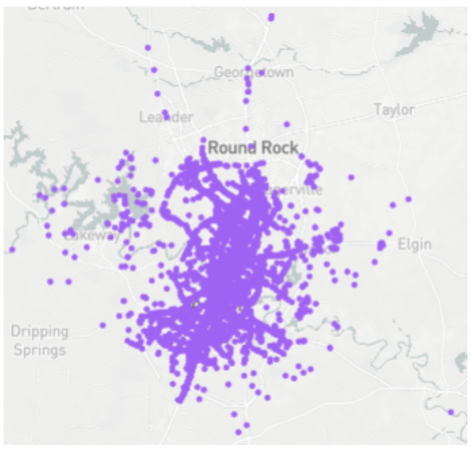
Map of Crash Service Incidents **Map of Traffic Hazard Incidents**



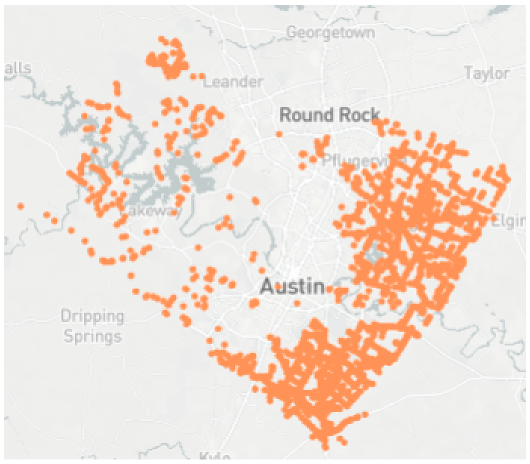


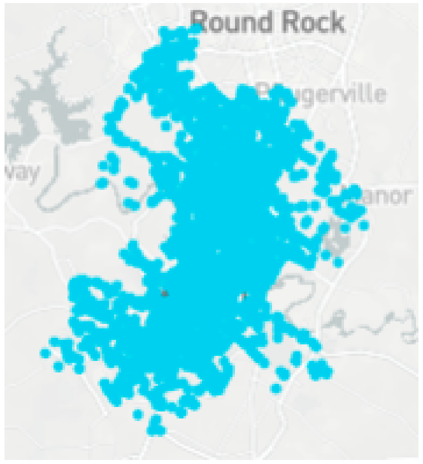
Map of Colission Incidents **Map of Stalled Vehicle Incidents**





<u>Map of Loose Livestock Incidents</u> **<u>Map of Traffic Impediment Incidents</u>**





Traffic hazards and collisions seems evenly distributed throughout the city.

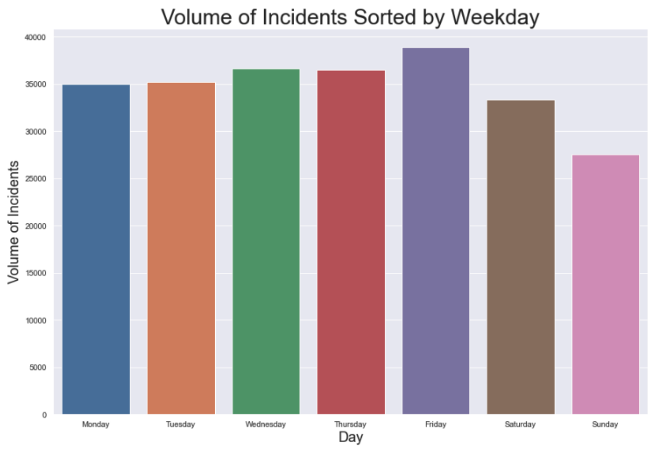
Loose livestock appears to be clustered around the eastern part of the city. This makes sense as there are not many suburbs on the east side and the landscape turn from urban to rural quite quickly.

Stalled vehicles, crash services, and traffic impediments seem to be a much larger issue in the central part of the city. This could be due to the lack of lanes in the city center, providing little room to go around stalled vehicles and other large objects.

<u>Feature Engineering: Months, Days, Hours</u>

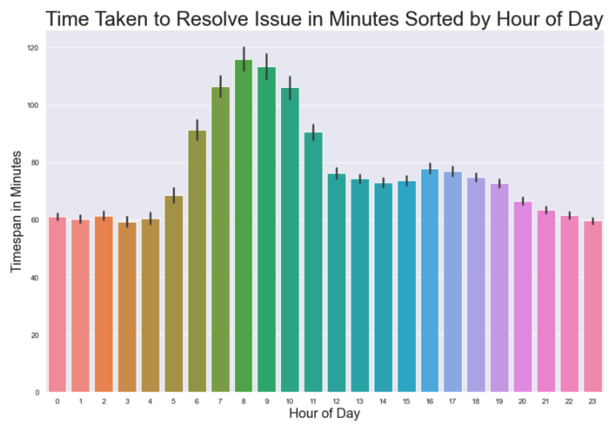
Using the column published_date I was able to create three features that isolated the month, day, and hour of an incident. These features provided some interesting insights and the visualizations allow us to infer certain conclusions.

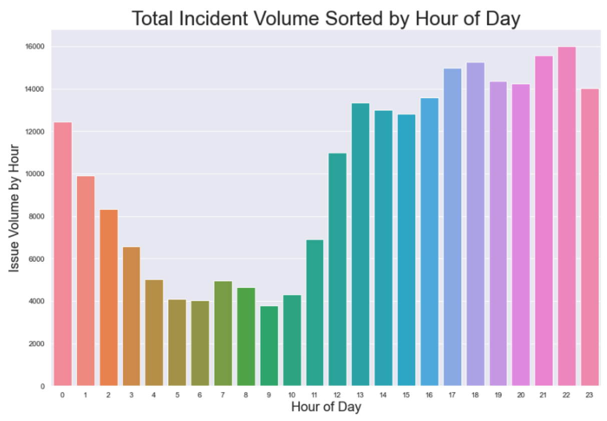
<u>Day</u>



The incident a fairly evenly distributed except for Friday (higher than normal) and Sunday (lower than normal). This could be due to the likelihood of people driving from work to a bar and getting into an accident on the way home. Sunday has fewer drivers on the road in general. Monday and Tuesday hold steady at around 35,0000 Wednesday and Thursday there is a slight increase to around 37,000 Saturday has fewer instances than weekdays at about 33,000

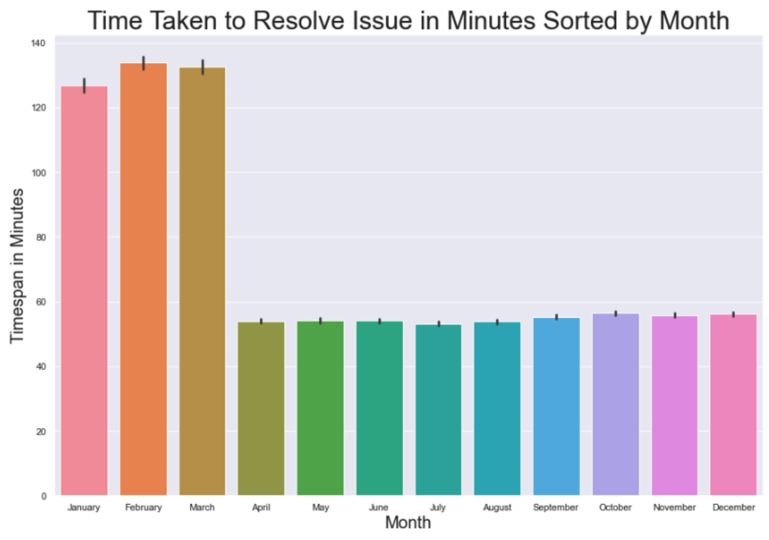
Time





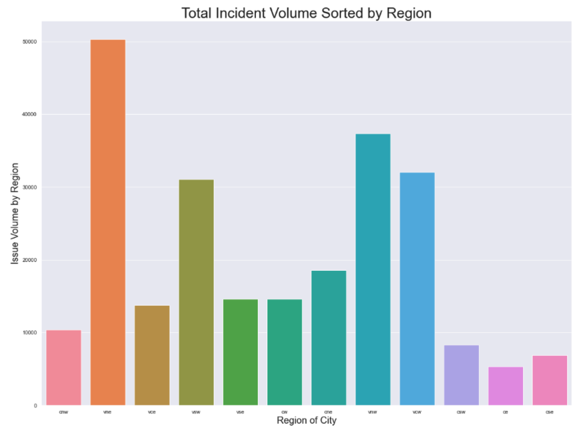
Looking at these two bar charts helps us draw two conclusions. The first is that traffic incidents that occur between 7:00 am and 11:00 am take longer to resolve. The second is that there is a noticeable increase in incident volume starting after 12:00 pm. These two conclusions seem counterintuitive. If fewer incidents happen in the morning, why would it take longer to resolve those issues?

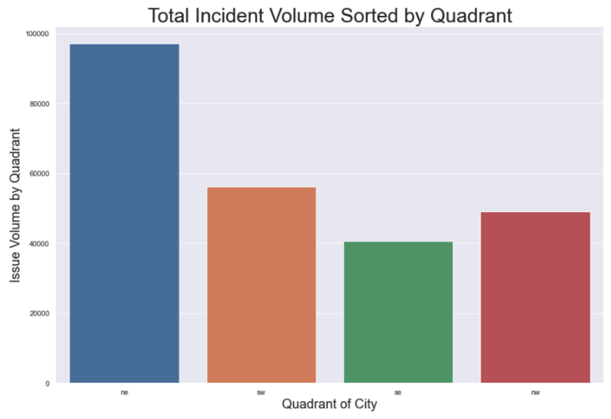
Month



Not much was revealed with the month feature except for this barplot which shows that traffic incidents took longer to resolve in the first quarter of the year. This could be due to winter weather. Winter's are mild in Austin, but that does mean the populace is often unprepared for when bad weather does arrive.

Feature Engineering: Region, Quadrant, Distance From Center



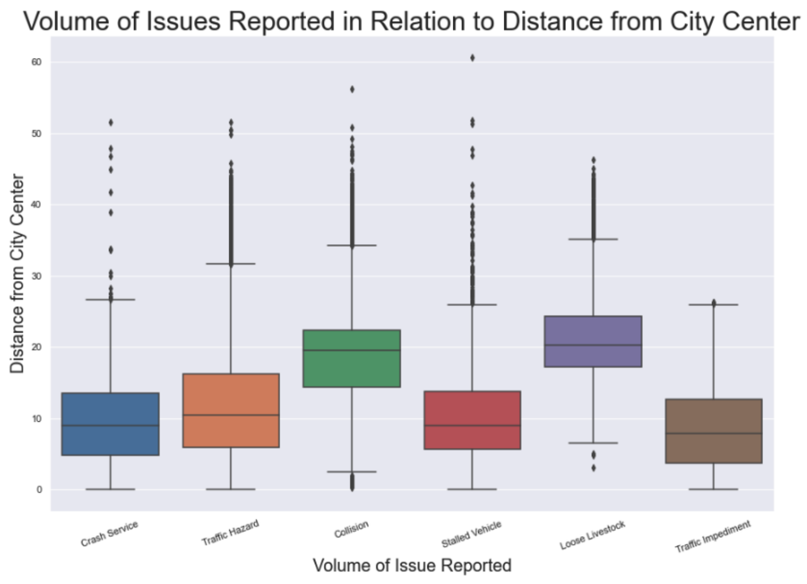


Region & Quadrant

The two charts represent different methods for dividing up the city geographically. The first, Region, uses major city highways to subdivide the city into twelve sections. A key to those intersections and their corresponding coordinates can be found in the notebook feature_engineerning4, inside the feature engineering folder of my project. The second, Quadrant, simply takes the central coordinates (30.2672, -97.7431) of the city and creates city quadrants. These two methods are not precise. I used two different methods in hopes to glean something, and I did. Most accidents occur in the northeast part of the city.

Distance From Center

This function takes coordinates and measures the distance an incident that occurred from the city center (30.2672, -97.7431) in kilometers. The graph below examines which issues happen farthest from the city center.



If we compare this boxplot to our map plots on pages 5 and 6 we can see some confirmations and some clarifications

Similarities/Confirmations:

Loose livestock appeared mostly in the east, far from the city center, and confirmed on the boxplot.

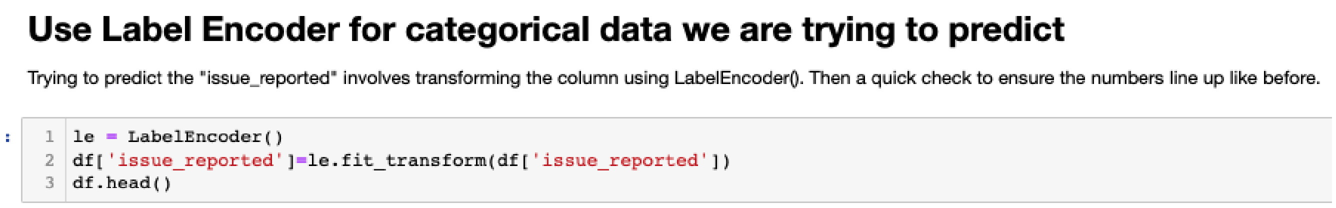
The stalled vehicle, crash service, and traffic impediment boxes seem to add to the evidence that they are much larger issues in the central part of the city.

Differences/Clarifications

On the maps it appeared that traffic hazards and collisions were evenly distributed. Now we see that this is not quite the case and that traffic hazards occur in more central locations and for collisions the opposite is true.

Label Encoding and One Hot Encoding

Step 1: Label Encode target data

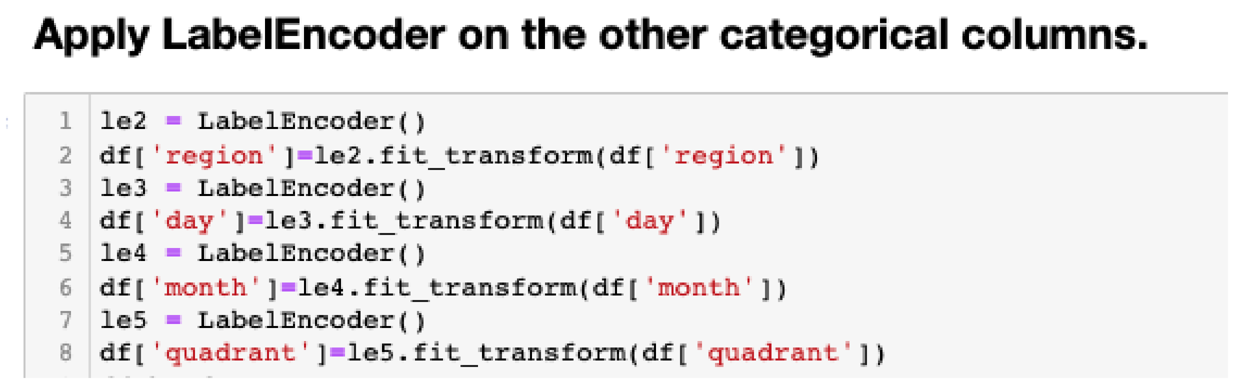


Use Label Encoder for categorical data we are trying to predict

Trying to predict the "issue_reported" involves transforming the column using LabelEncoder(). Then a quick check to ensure the numbers line up like before.

```
le = LabelEncoder()
df['issue_reported']=le.fit_transform(df['issue_reported'])
df.head()
```

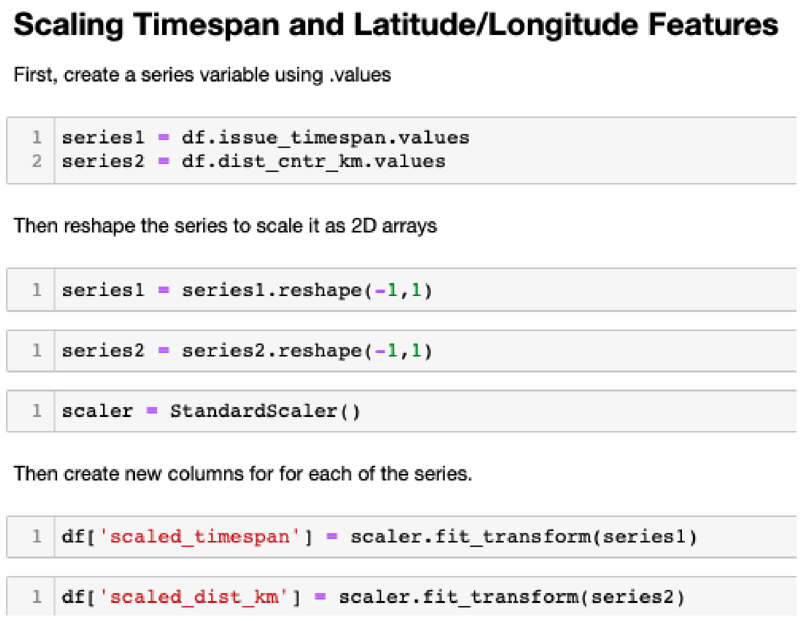
Step 2: Label Encode other categorical features



Apply LabelEncoder on the other categorical columns.

```
le2 = LabelEncoder()
df['region']=le2.fit_transform(df['region'])
le3 = LabelEncoder()
df['day']=le3.fit_transform(df['day'])
le4 = LabelEncoder()
df['month']=le4.fit_transform(df['month'])
le5 = LabelEncoder()
df['quadrant']=le5.fit_transform(df['quadrant'])
```

Step 3: Scale the ordinal data



Scaling Timespan and Latitude/Longitude Features

First, create a series variable using .values

```python
series1 = df.issue_timespan.values
series2 = df.dist_cntr_km.values
```

Then reshape the series to scale it as 2D arrays

```python
series1 = series1.reshape(-1,1)
```

```python
series2 = series2.reshape(-1,1)
```

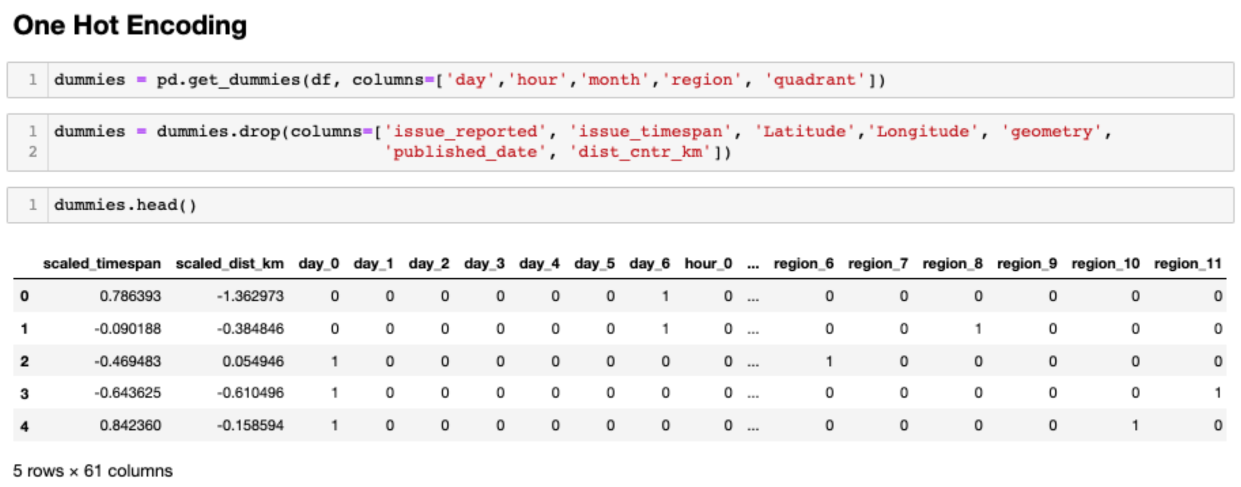
```python
scaler = StandardScaler()
```

Then create new columns for for each of the series.

```python
df['scaled_timespan'] = scaler.fit_transform(series1)
```

```python
df['scaled_dist_km'] = scaler.fit_transform(series2)
```

Step 4: One Hot Encoding



One Hot Encoding

```python
dummies = pd.get_dummies(df, columns=['day','hour','month','region', 'quadrant'])
```

```python
dummies = dummies.drop(columns=['issue_reported', 'issue_timespan', 'Latitude','Longitude', 'geometry',
                                'published_date', 'dist_cntr_km'])
```

```python
dummies.head()
```

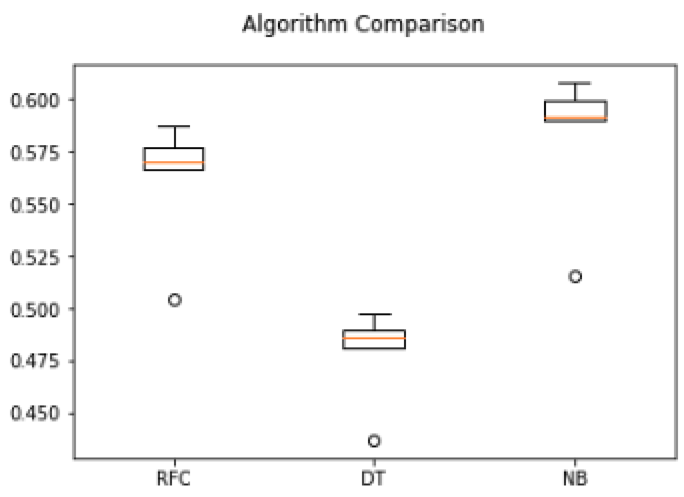
	scaled_timespan	scaled_dist_km	day_0	day_1	day_2	day_3	day_4	day_5	day_6	hour_0	...	region_6	region_7	region_8	region_9	region_10	region_11
0	0.786393	-1.362973	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0
1	-0.090188	-0.384846	0	0	0	0	0	0	1	0	...	0	0	1	0	0	0
2	-0.469483	0.054946	1	0	0	0	0	0	0	0	...	1	0	0	0	0	0
3	-0.643625	-0.610496	1	0	0	0	0	0	0	0	...	0	0	0	0	0	1
4	0.842360	-0.158594	1	0	0	0	0	0	0	0	...	0	0	0	0	1	0

5 rows × 61 columns

We now have 61 columns as we proceed to find the best model to use moving forward

Model Selection

Comparing three classifier model algorithms will provide some insights as to which model to pursue in terms of hyperparameter tuning. The three models I choose were Random Forest, Decision Tree, and Naive Bayes classifiers. The result is represented in the boxplot below.

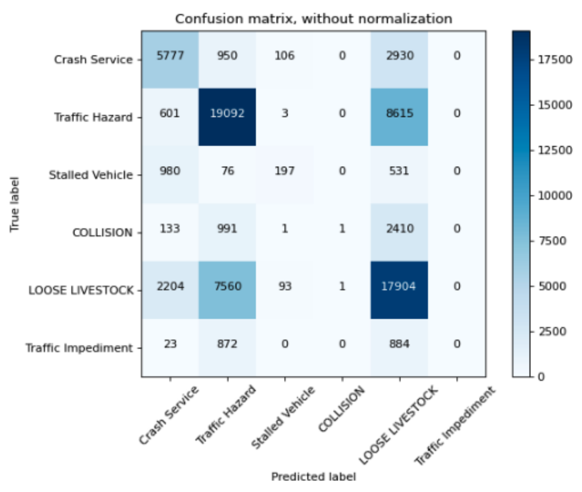


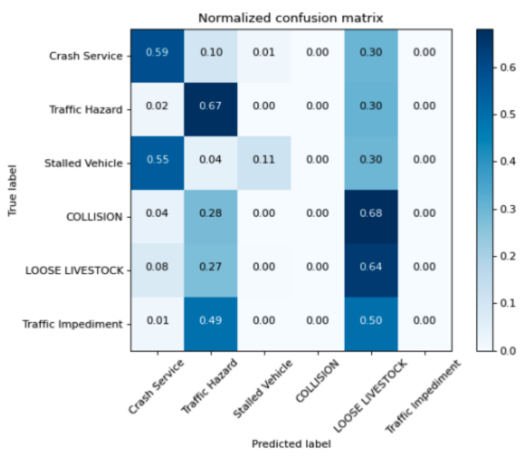
Both Random Forest and Naive Bayes scored well above the Decision Tree model.

Hyperparameter Tuning for Random Forest Classifier using GridSearchCV
The best parameters were determined as the following:

```
Best n_estimators: 500
Best max_depth: 25
Best min_samples_split: 15
```

Running the improved model the following confusion matrices

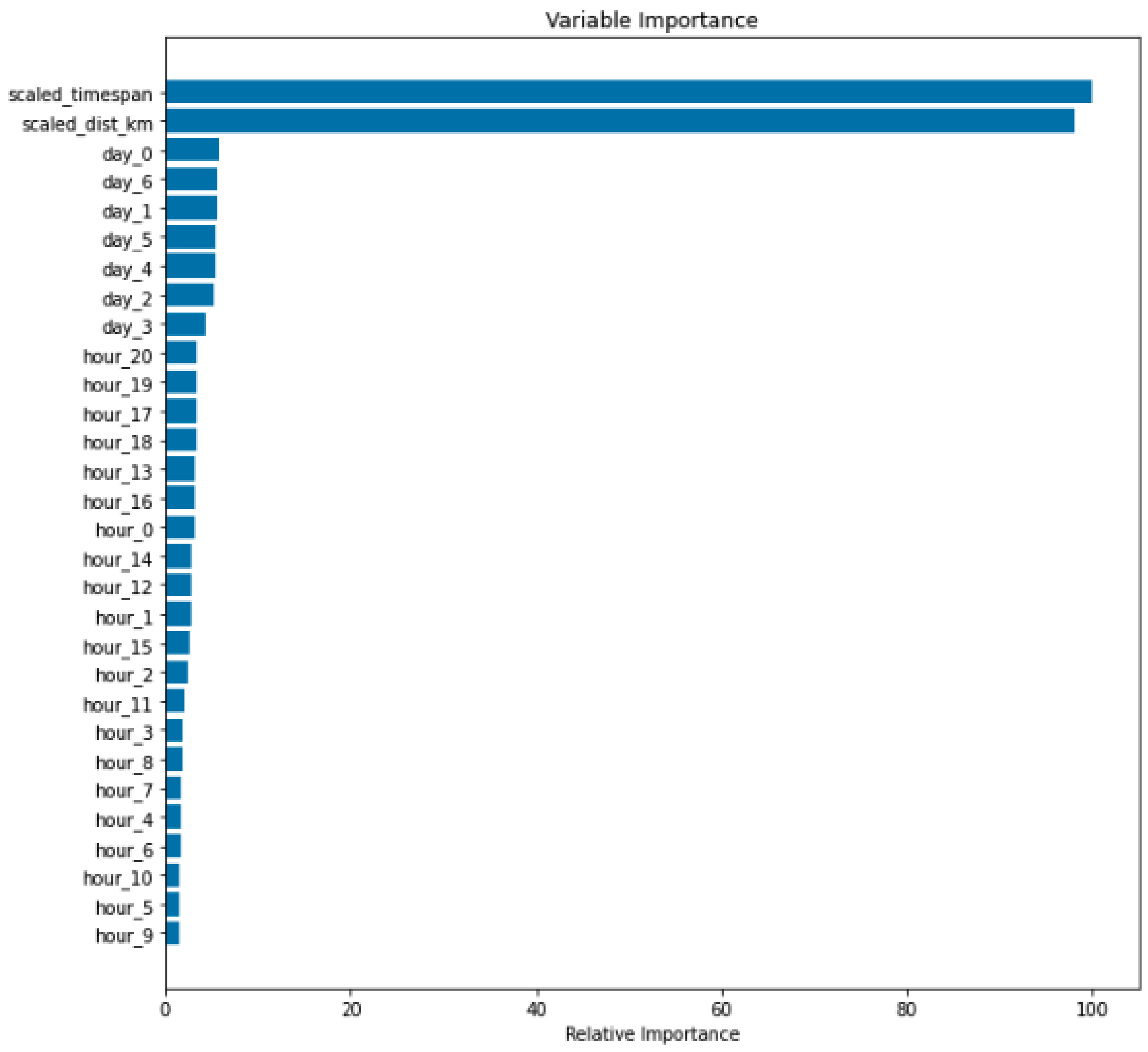




We can see here that we are doing fairly well, predicting half of our categories and predicting terribly the other half.

The 3 classes that stand out: **Crash Service, Traffic Hazard, Loose Livestock**

What were the most important variables?

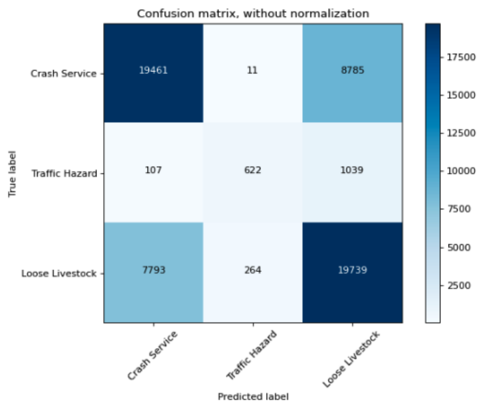


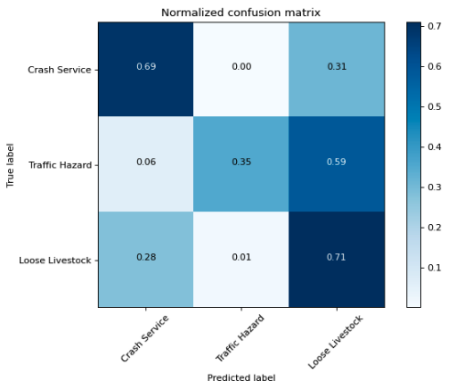
It’s not really that close, the issue_timespan and dist_from_cntr columns are by far the most important features, with day and hour making an appearance.

Narrowing the Categories from 6 to 3

Running the Tuned Random Forest Model using only Crash Service, Traffic Hazard, Loose Livestock produced the

```
Random Forest: Accuracy=0.689
Random Forest: f1-score=0.687
Random Forest: Balanced Accuracy=0.584
```

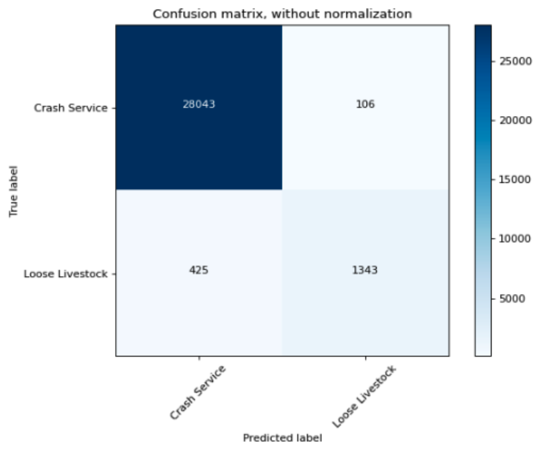


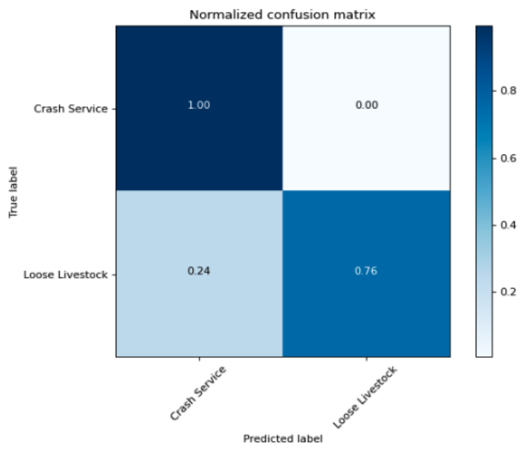


Narrowing the Categories from 3 to 2

Running the Tuned Random Forest Model using only Crash Service, Loose Livestock.

```
Random Forest: Accuracy=0.982
Random Forest: f1-score=0.981
Random Forest: Balanced Accuracy=0.878
```

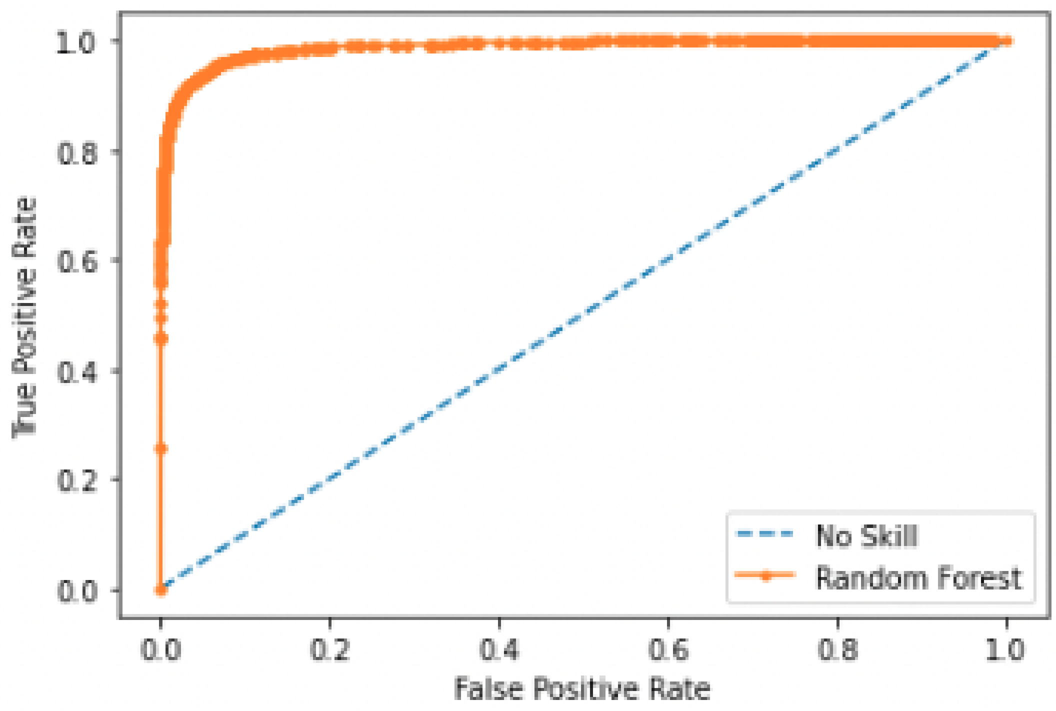




The model was able to predict Crash Service and Loose livestock with 98% efficiency. If we convert the last confusion matrix into a binary classifier we get the following:

True Positive - 1343/0.76
True Negative - 28043/1.00
False Positive - 106/0.0
False Negative - 425/0.24

```
No Skill: ROC AUC=0.500
Random Forest: ROC AUC=0.988
```



The Naive Bayes model scored the exact same and as the Random Forest model even without tuning the hyperparameters.

Why were crash service and loose livestock so much easier to predict than the others?

Thinking back to the variable importance graph on page 12, we saw that the two most important variables were timespan and dist_km. This is likely because these two issues are unique in those two categories. Crash service takes a lot longer to resolve than other issues so it is unique in that regard. Loose Livestock has a strong correlation to its distance from the city center. This should not come as too much of a surprise.

Recommendations

- All routes should avoid traveling long distances on the east side of town if not necessary for a specific delivery. All travel routes should start and end away from high-volume areas like the northeast.
- Workweeks should consist of four 10-hour days rather than five 8-hour days. This way Friday can be avoided entirely as we know this is a high incident volume day.
- Routes should depart as early as possible to avoid rush hour and the lengthy time spans associated with incidents at that time.

Future Improvements:

- In the future, I would want to go back to my original question of route planning and see if I can isolate when and where high volume incidents occur.
- Run a dummy classifier on the model with the crash service and loose livestock classes to compare with the accuracy score of 98%
- Why is it that fewer incidents happen in the morning, but it takes longer to resolve those issues? If the volume is down, why are timespans increased?
- Why are average timespans for issues so much higher in January, February, and March compared to the rest of the year?
- Why do so many incidents occur in the northeast of the city?