**CMPSC 463 PROJECT 2 REPORT:**

**Police Response Time Predictors**

By: Sam Axler

*(Notes: Inside the GitHub is an MP4 showcasing the project in case the project does not work on your computer.)*

**OVERVIEW:**

This project is intended to put into perspective how long it might take for the police to solve/sort out a case during a situation. From the moment someone calls 911 to the moment the situation is completely solved, this program intends to calculate and predict the exact time it takes for things to be sorted out. The overall goals of the project are as follows:

**GOALS:**

* Calculate and predict the time it would take for police to solve an isue based on the level of urgency and the circumstances leading up to the event.
* Allow for as many specifics as possible to make an accurate prediction.
* Allow for user input to create a more seamless experience.

**SIGNIFICANCE:**

This is important for a multitude of reasons. The first is that it puts into perspective how long it takes for help to arrive during a situation, but it also raises awareness of how long it takes for the police to solve a situation. Many people think most situations are done and dusted in a day or two, or maybe even a few hours. However, rarely is a 911 situation ever settled in less than 2 days, even in minor situations. This is because there are several other contributing factors than simply the police arriving and solving the situation, there’s also a whole load of paperwork and information to fill out based on each response.

**REQUIREMENTS/INSTRUCTIONS:**

**INSTALLATION:**

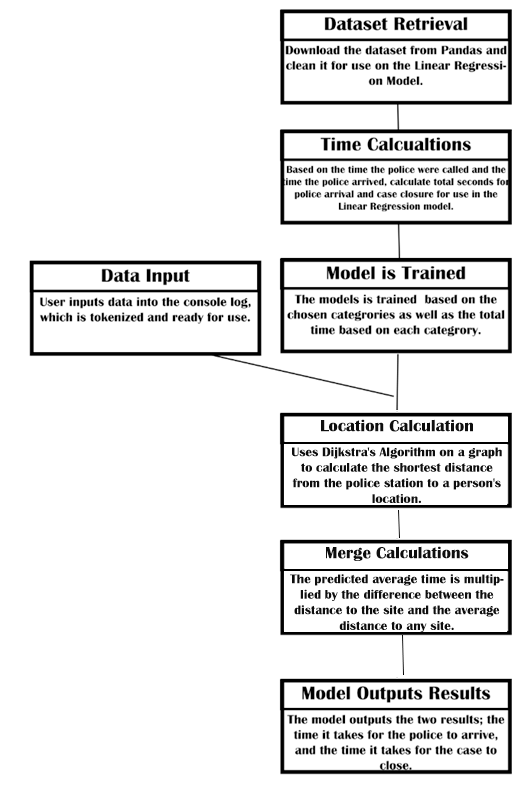
Download the project, and run the following commands in your Terminal:

**USAGE:**

Once these dependencies are installed, run the first two Jupyter Lab nodes, both the imports node and the linear regression model node. Once complete, “model has been trained” should appear in the console.

Finally, run the final node at the bottom of the Jupyter notebook. Enter the text into the output log to get your results.

**CODE FLOW:**



**CODE FLOW CONTINUED:**

The code uses a linear regression model to predict the time it will take for police to arrive based on four criteria:

* Contact Type: (Phone or Email)
* Impact of Call (Low, Medium, or High.)
* Urgency of Call (Low, Medium, or High.)
* Time of Call (XX:XX format)
* Location of call (In pre-generated sites A,B,C,D,E,F for testing purposes.)

When the location is inputted, Dijkstra’s Algorithm finds the shortest path from the police station (Node 0) to the chosen location. The average distance between the police station to every node is also calculated in advance, and is our baseline.

The Linear Regression model uses those categories to estimate the time it takes for the police to arrive on the scene.

Once the time is estimated, it is multiplied by the difference between the distance that Dijkstra’s Algorithm calculated and the average distance. The formula looks like this:

The estimated time that the case is closed is NOT effected by the distance to the location, but rather, just the urgency and impact of the call.

**RESULTS:**

Below are a few results and examples of the program when it runs.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The average response times between the Linear Regression model and real life statistics of Philadelphia are as follows:

**Urgent:**

Model: 16 minutes, 11 seconds.

Actual, PA: 9 Minutes

**Non-Urgent:**

Model: 38 minutes, 11 seconds.

Actual, PA: 15 Minutes

The discrepancy between the two values is likely due to the variance in data. The dataset I used is likely on a much larger scale than simply the city. Several factors, including distance to the police and traffic at the time, determine the true median values of these statistics. The distances between locations in the dataset may have been different than the ones in the graph.

**CONCLUSION:**

**LIMITATIONS:**

The project had quite a few limitations. The first, of course, being time. If I had more time and to dedicate specifically to this project, I’d have investigated more accurate means to calculate response times. However, one glaring issue was the issue of police codes and how inconsistent they are.

Police codes and incident codes are formatted like “Scenario 12 Subsection 205” or something like that. These scenarios outline specific incidents, such as a house fire, or a robbery, or something requiring the bomb squad, ETC. The issue with these incident codes is that they’re different between each police department, meaning it’s impossible for any consistency to be derived from these codes. This was the biggest limitation of the project, and by the time I realized this, it was too late to change course.

**ENDING STATEMENT:**

In conclusion, this project offered a unique challenge to me. Crime and Violence is mostly uncharted waters for computer science, since analyzing the worst of human activity is still a highly volatile subject. However, I managed to get a decent program working to calculate certain response times. I learned a lot about computer science techniques, as well as police department procedures and codes.