RSA Project Report

Our Web Application

Team Members - Rohan Mathur – rma135, Siddharth Goradia – srg6, Anmol Malhotra – ama302

Problem Definition

Our project aims at providing crime analytics for assisting the Vancouver Police Department in better crime patrolling across neighborhoods in Vancouver, along with providing predictions of types of crimes for the safety of the citizens. Our data is being sourced from the official Vancouver Police Department Website. The data ranges from 2003 to 2022.

Currently, the Vancouver Police Department relies mostly on cold calls through 911 and historically proven neighborhoods which are internally divided into four districts. Moreover, patrolling vehicles are distributed throughout these districts as per routine instructions. Our analysis aims to further provide analytics based on numerous factors that provide targeted enforcement of Patrolling Vehicles. These analyses will be particularly helpful for the Crime Analysis Unit in the VPD, who are tasked with investigating crimes and can allocate patrolling resources according to the need.

Methodology

1) Data Collection

Data Acquisition was done through the Vancouver Police Department's website. Our collection did not require scraping as the data was downloaded in a structural manner (CSV File). We had to manually enter the range of data we needed (2003-2022). We had to also select manually the neighborhoods that we require, in this case we selected all as they were all within Vancouver.

2) Data Cleaning

Our downloaded dataset initially had approximately 843,000 rows. For data cleaning, we used Pyspark to work on the dataset. Data cleaning involved steps like removing null values, removing masked data and keeping unmasked data, conversion of coordinates to Latitude Longitude type and more. Our cleaned data was then stored on a MySQL hosted database.

3) ETL Pipeline

We are using AWS RDS to host our MYSQL database. The AWS RDS database acts like a central repository where all the tables reside with all the data, from cleaned data to tables for each visualization. The code pushes the data frame to the AWS RDS database making it accessible to the Dash app to fetch and use it on the visualizations. The basic pipeline being followed is that a CSV file is ingested by the cleaning pyspark code and cleaned data is generated and pushed to the AWS RDS database. From there we create specialized pyspark code to create data tables that can be directly ingested into the Dash app. To avoid importing all the data and slowing up the dash app we are creating specialized tables for each visualization.

4) Visualizations

The dashboard is made using Dash and fetches data tables to create visualizations to create specific visualizations. We also are using the ML pickle files to provide real-time predictions and use an Iframe tag to provide map visualization. Our web application is deployed through Heroku.
Visualizations are done by directly accessing the cleaned data. Some Are -

1) Total Number of Crime by Type which can be sorted by Year by the user.
2) Total Number of Crime by Neighborhood which can be sorted by Year by the user.
3) Count of the types of crime occurring in Each Neighborhood
4) Total crime per month for all years
5) Total crime per hour for all years
6) Sum of Crimes for Each District in Last three years
7) Total crimes in top three hundred blocks in a district with lowest crimes
8) Total crimes in top three hundred blocks for all four districts
9) Total Crimes happening on the top three holidays on which crime occurs the most for all the years of data

5) Machine Learning

Our project incorporates Machine Learning to predict the crime type ('theft from vehicle', 'mischief', 'other theft') as the target based on year, month, hour, neighborhood and hundred block as the input features. During data cleaning, we realized that multiple crime types like homicide, residential break in etc. have very few data points (in range of a few hundred/(s)) to accurately assign them a prediction. We decided to filter them from our target variable to model the features appropriately. The Machine Learning pipeline is designed using Spark ML concepts such as Vector Assembler, IndextoString, String Indexer, Random Forest Classifier and Naïve-Bayes Classifier (for modelling and training), Pipeline, Multiclassification Evaluator. We also implemented Machine Learning using concepts of Pandas and scikit-learn (KNN for modelling and training). We incorporated the use of Confusion Matrix (Accuracy, Precision, Recall, F1Score) as a metric to judge the predictions produced by our model.

Problems

1) Our initial thought for proceeding with Dash as our frontend had Pyspark on it. However, Pyspark on Dash turned out to be very heavy and was repeatedly crashing on the laptop. Eventually, we had to turn to Pandas for reading the data from our database.
2) With this issue, a direct code for creating visualizations could not be added to the application. Pyspark code was a necessity in this project, hence we decided on operating on it away from the app itself and use Pyspark to read and write on our database by operating locally. This resulted in the creation of Tables through Pyspark for our advanced visualizations which were pushed separately as tables to our hosted database. Our Dash App would then call these tables instead of our cleaned data table to make visualizations. These were done for only four visualizations.
3) Machine Learning could also not be incorporated into our Dash App as it was coded in Pyspark. We used Naïve Bayes and Random Forest as our classifier algorithms in Pyspark. However, when we found out PySpark model would not run, we decided on creating an ML model using Pandas, which was then pickled along with the target labels.

4) This is what is currently being used in our Dash App, the final model that is being used is made using K- Nearest Neighbours which was resulting in a 0.60 Accuracy (approx). We have kept both our Machine Learning scripts, Pandas and Pyspark in our repository that can be run.

Results

1) Analytics

Advanced visualizations require accessing other tables in MySQL which were done in the ETL pipeline. These were done based on the following questions posed -

1) How were the crimes ranging for each district made by the VPD, for the last three years?

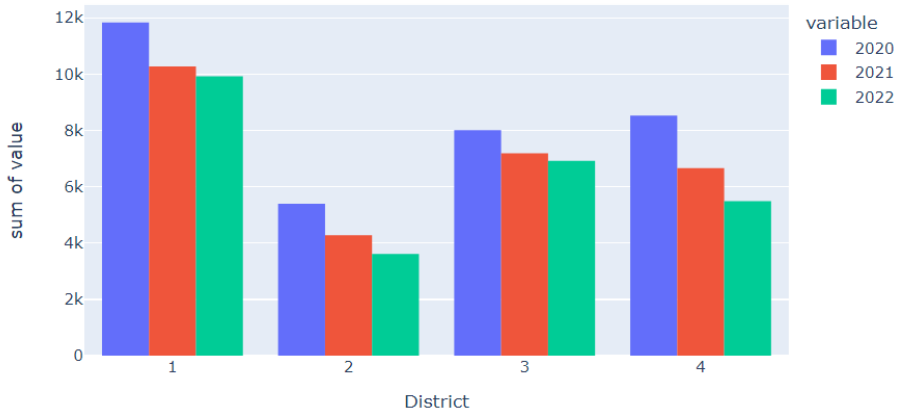

Fig 1. Top Crimes Per District for Last Three Years (2020-2022)

As observed, the trend has been that crimes are decreasing as the years pass. It is observed that District 1 has the greatest number of crimes among all districts, and District 2 has the least. To answer our problem statement, it is deduced that VPD can assign a larger number of patrolling vehicles in that district.

2) From the above, it was observed that District 2 had the lowest crimes for the past three years. Can we locate the hundred blocks within this district which are having the highest crimes only in this district?

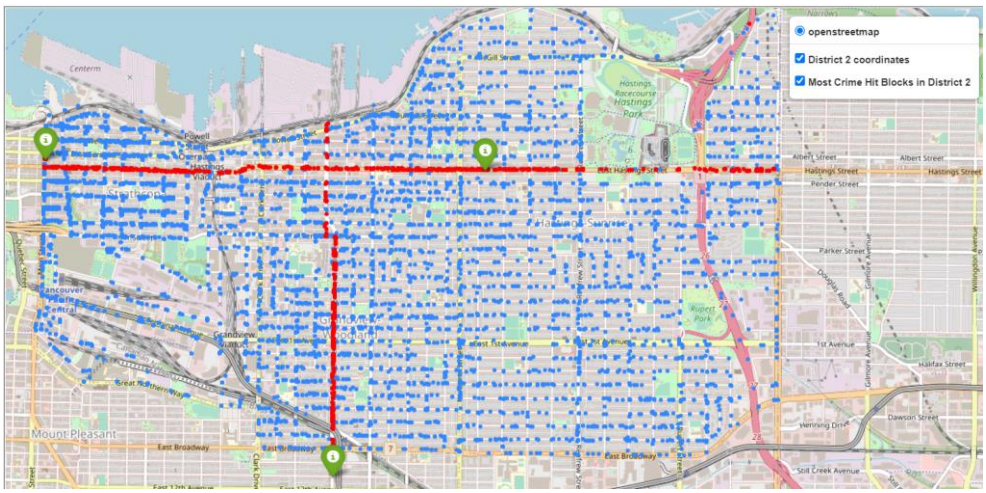

Fig 2. The 3 Blocks in District 2 with Most Crimes for Past Three Years (2020-2022)

We observed three blocks within this district (among all different neighborhoods) which were having the highest crimes. These were – Nk_Loc_Street, E Hastings Street, Commercial Drive.

It was deduced that District 2 has the lowest crime count. We can say that patrolling can be deployed here on a lower intensity. However, within this District there should be more concentration on the above green marked blocks.

3) We observed a great concentration of crimes in a few hundred blocks in District 2. Can we also locate the top hundred blocks in all districts for the past three years?

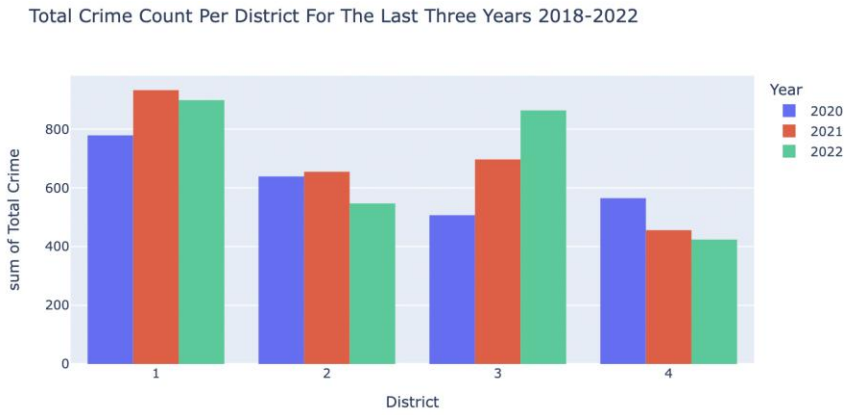

Fig 3. Top 3 Blocks for each District Having the Most Crimes for Last Three Years (2020-2022)

As seen through our graph, concentrations for each district show up along with their count. It was observed that 2020 was a year that had spikes (perhaps during the pandemic) for almost all districts, but District 3. District 3 shows a surge in number of crimes in the top hundred blocks in recent years. Hence, patrolling should be allocated at a larger scale for these mentioned blocks.

4) Holidays are meant to be for creativity, or a festival to celebrate. However, crimes are bound to happen these days on a larger scale. Can we quantify how the crimes range on special occasions that happen in Vancouver?

Fig 4. Top 3 holidays with Most Crimes and The Range of The Total Crimes For Past Twenty Years

We gather data from 2003 to 2022 on the special days that exist in Vancouver, which are termed as public holidays. After matching these with our data, we took the total count observed for the top three public holidays. These were – New Years's Day, Labour Day, and Canada Day.

Overall, all three have a comparable graph, however, there has been a spike in recent times for the crimes that happen on Canada Day. Hence, patrolling should be on high alerts during these holidays, especially Canada Day.

2) Machine Learning

The following table highlights the results obtained from our multiple Machine-Learning Models

SPARK-ML MODELS:

Classifier (Spark-ML)	Metric Name	Metric Value
Random Forest Classifier	Precision	0.527591
	Recall	0.733574
	Accuracy	0.547105
	F1 Score	0.490002
Naïve-Bayes Classifier	Precision	0.556199
	Recall	0.385946
	Accuracy	0.485317
	F1 Score	0.435867

Table 1. Model Accuracies for Pyspark Model

SCIKIT-LEARN MODEL (PANDAS):

K Nearest Neighbors (KNN) Classifier

Crime Type	Precision	Recall	F1 Score	Support
mischief	0.36	0.11	0.17	16271
other theft	0.59	0.64	0.61	29879
theft from vehicle	0.58	0.72	0.64	37442
Accuracy (Total Support)			0.57	83592

Table 2. Model Accuracies for Pandas Model

Project Summary

Getting The Data: Gathered data directly from the VPD Website. (1)

ETL: Used Pypspark for cleaning the dataset. Used AWS RDS for hosting MySQL database. (2)

Problem: Brainstormed to arrive on Patrolling as a solution to work towards after discussing numerous problems solvable using Crime Analysis. (3)

Algorithmic Work: Worked on creating ML Models using Pyspark and Pandas. Used Pyspark ML's Naïve Bayes and Random Forest. Used KNN and Random Forest through Scikit learn for Pandas. (2)

Bigness/ Parallelization: Can be scalable to larger data, more importantly live feed data. However computational cost is limited for our project as of now. Also using MySQL in the long run would not work. (2)

UI: Used Dash and Plotly for making the dashboard, hosted using Heroku. Used Folium to generate maps. (2)

Visualization: Used Pyspark to create datasets which were further implemented using Dash. (5)

Technologies: Pyspark, Python, Dash, Plotly, Heroku, Spark ML, AWS, MySQL, Pandas, Folium (3)