Geospatial Analysis on Toronto Traffic, Pedestrians, and Air Quality Levels

Amr Shalaby

Abstract

In this report, I explored the statistical relationships among traffic volumes, pedestrian counts, air quality, and their respective temporality in Toronto. The focus was on the Traffic and Pedestrian Volumes and validating the anecdotal evidence of the visualized hotspots. I utilized Python, SQL, and Load Testing in streamlined automation. The pipeline integrated Autonomous Machine Learning Methods to embed the predictive layers in the web maps. There are six data sources ranging from Environment and Climate Change Canada, the City of Toronto's Open Data Portal, to ArcGIS Online Dataset that are curated and ingested autonomously. The approach utilized the H_2O AI Platform in a high-level implementation to produce three different map types: Turf, Mapbox, and Folium. The pipeline requires little to no user input and no ad-hoc code used. The three produced maps are stress-loaded under different browsers and finally, the map that is presented to the user scored the lowest loading time in its respective optimal browser.

§0. Git Repository

- Online Maps:
  - https://amr-y-shalaby.github.io/ggr_472_project/
- Python Code:
  - https://github.com/amr-y-shalaby/ggr_472_project
- Guide to Setting up Config.ini:
  - https://github.com/amr-y-shalaby/ggr_472_project?tab=readme-ov-file#01-configini
- Pipeline Executor:
  - https://github.com/amr-y-shalaby/ggr_472_project/blob/main/Pipeline/main.py

§1. Introduction

Urban air quality, a critical determinant of public health and environmental sustainability, has garnered increasing attention in recent years, particularly in the context of rapidly urbanizing cities. Among various pollutants, vehicular emissions stand as a primary contributor to urban air pollution [1]. The Auto Machine Learning Layer thoroughly separated temporality from spatial, physical locations of the regions that descriptively manifested hotspots of air quality, traffic, and pedestrian volumes. The maps' anecdotal evidence needed refutation, or affirmation, by the application of statistical rigour since adopting the wrong beliefs can be deeply seated for the non-technical audience.

Similar data portals online went to great lengths in displaying traffic congestion zones without presenting any statistical validation of the perceived truth presented by descriptive maps [2]. Accordingly, I decided to embed descriptive plots into the markers capturing multiple sources: air quality, vehicular volume,

pedestrian counts, and air quality indices into one map such that a collective, cohesive perception can be secured by combining many data sources into a single map without sacrificing rendering efficiency, responsiveness, or optimality of the selected browser to view the map. The Embedded Machine Learning layers have undergone extensive batteries of model selection and validation before the predictive layers are integrated into the maps utilizing the famous Super Learner Approach published in 2010 [3] whose statistical robustness was secured six years later due to the limitations of the computational power.

The findings are valuable recommendations for urban planners, policymakers, public health officials, and general commuters guiding understanding, and statistical validation, of problematic regions of traffic congestion, air quality, and their joint impact on pedestrian flow.

§2. Data Sources

The study utilizes five primary data sources: the Air Quality Health Index (AQHI) data from Environment and Climate Change Canada, Traffic Volume data collected by the City of Toronto's Transportation Services Division, and Pedestrian and Vehicle Counts collected by ArcGIS Open Data Portal.

§3. Primary Variables of Measure

§3.1 Air Quality Health Index

The AQHI is measured on a scale ranging from 1 to 10+. The AQHI index values are grouped into health risk categories as shown below (Figure 1). These categories help you to identify the level of risk easily and quickly [4]. The locations of Toronto Air Quality Monitoring Stations are indicated in Figure 2 [5]. Due to the precision of measure, the Air Quality Index secured a healthy Signal-to-Noise Ratio (SNR) of 3.26 as its mean of 2.73 surpassed its noise, or standard deviation, at 0.84 which is reflected in the speed of generating a predictive model due to less variability.

Figure 1: Air Quality Index Interpretation [4]. Source Government of Canada.

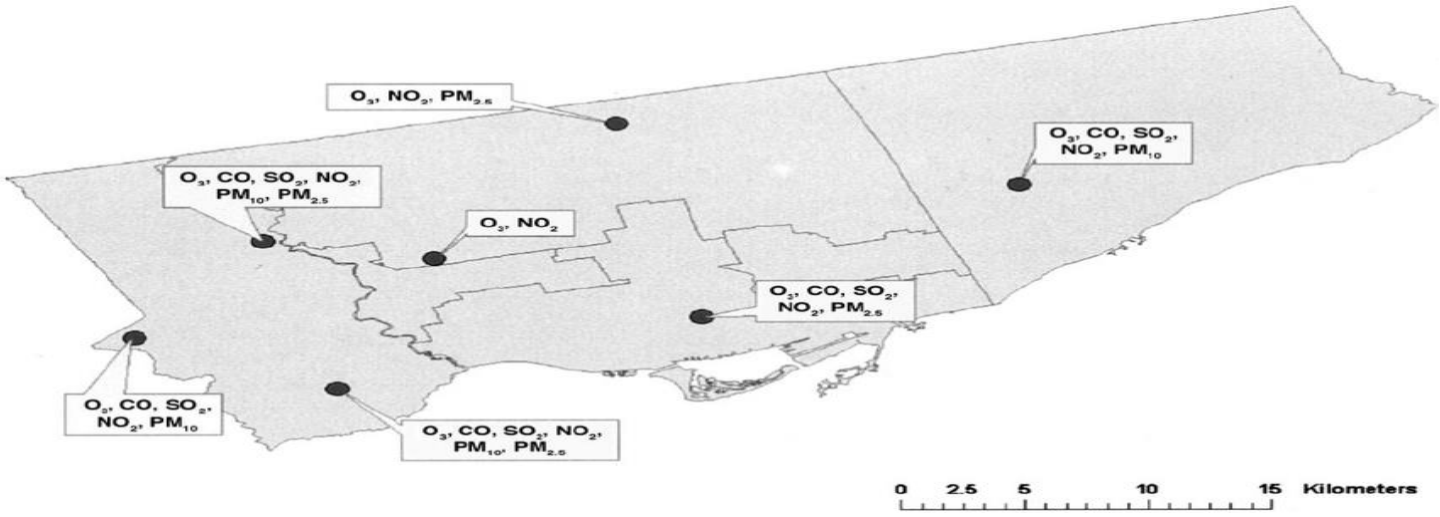

Figure 2: Locations of Toronto Air Quality Stations [5].

§3.2 Pedestrian and Vehicle Counts

The number of pedestrians and vehicles were acquired from ArcGIS Open Data Portal whose extensive time scope spans 12 years and 9 months from 2003-12-01 to 2016-09-07 across 246 monitoring stations throughout the City of Toronto. In terms of the statistical actionability of Vehicles and Pedestrian Counts, Vehicle Counts comprised a healthy 2.15 Signal-to-Noise Ratio (SNR) in contrast to the Pedestrian Counts which consisted of 50% Noise which was repeatedly captured by Gradient Boosting Machine (GBM) in the Auto Machine Learning Layer consuming a considerable portion of time allocated to Auto ML duration due to the excessive noise and lesser precision of counting pedestrians than vehicles (Figure 3).

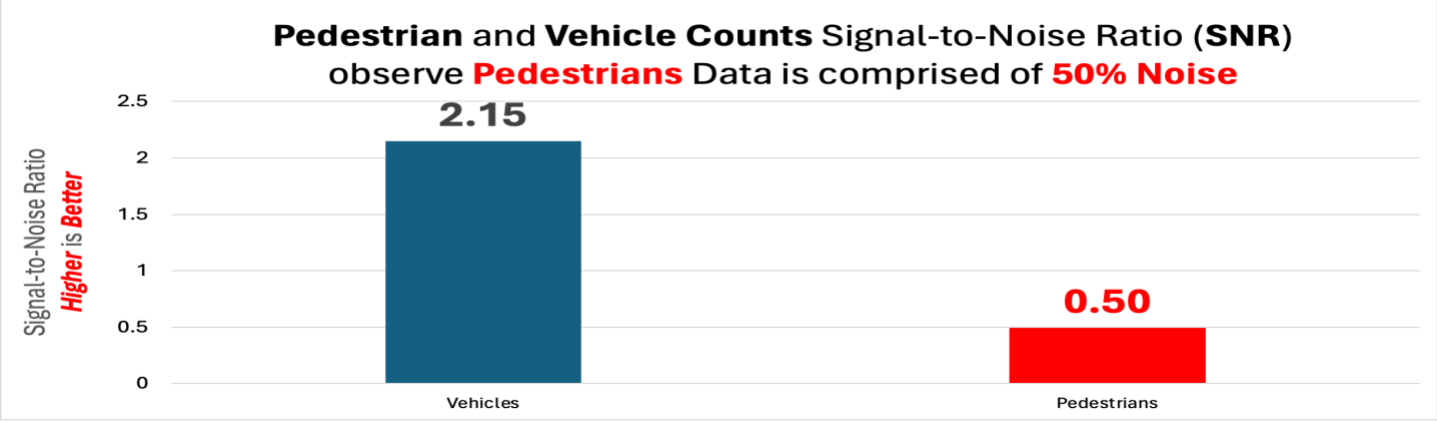

Figure 3: Signal-to-Noise Ratio (Mean ÷ Standard Deviation) of Pedestrians and Vehicles Counts. Due to a lesser precision of measuring pedestrians than vehicles, Pedestrian Data is comprised of 50% Noise consuming most of the time allocated to the Auto ML Phase.

§4 Execution and Data Model Performance

§4.1 Code Structure with Object-Oriented Focus

The entire dataset in both production and staging schemas constitutes a total of 1,857,572 records. The procedurally sequential programming of instantiating and calling functions that repeatedly connect to the database inflated the execution time to surpass seven minutes. However, the pipeline was restructured to run on only two objects: **`configs_obj`** and **`dfs_obj`**.

The **`configs_obj`** is an object that holds the runtime conditions parsed from the user-modified **`config.ini`** file and retains the database connection engines which activates a **single** connection to the database solving the problem of slow database authentications and making a parameter that gets passed to all function calls and terminates only once at the end of the program (Figure 4).

Similarly, **`dfs_obj`** holds all the database tables in **`public`** schema as Pandas, GeoPandas, and H2O Data Frames that are utilized by **`maps_creator`** for fast HTML creations that averaged 1.74 seconds after removing the slow data reads from the database. Furthermore, memory consumption by **`dfs_obj`** was also checked against the size which consumed only 48 bytes of memory after the final step of inserting H_2O Data Frames.

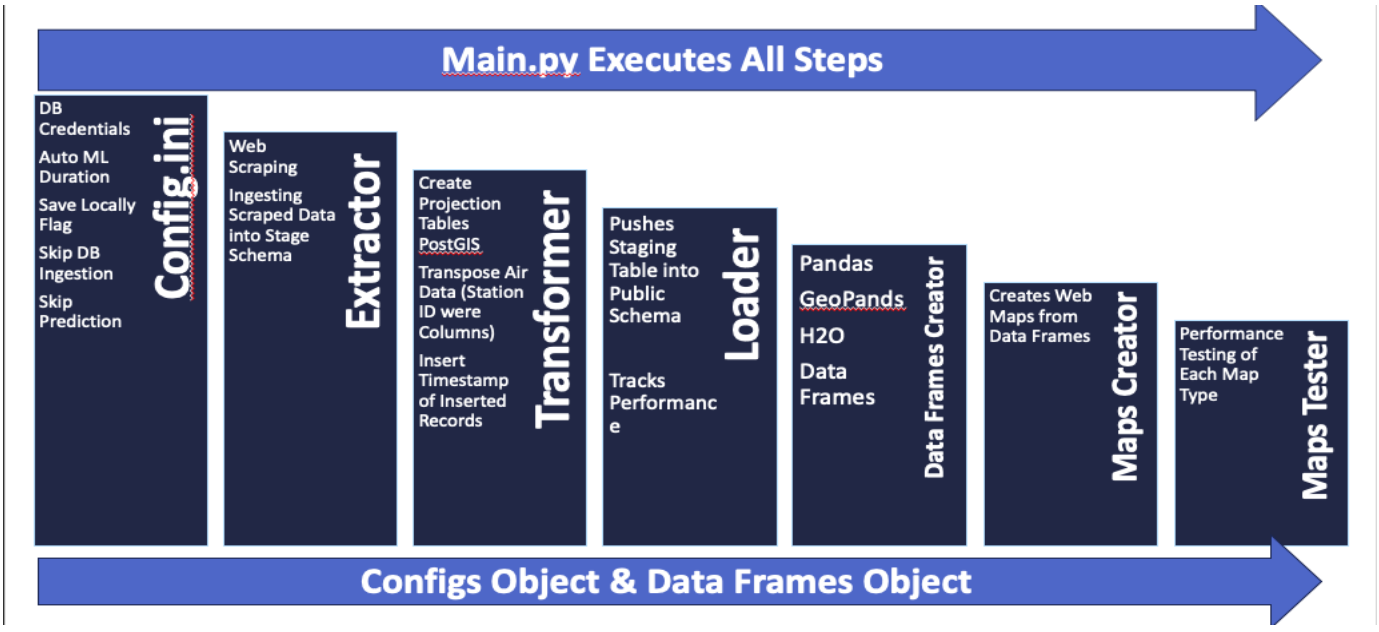

Figure 4: Main steps of the pipeline. There are only two objects that get passed into all functions to avoid slow database operations.

§4.2 ETL Design

The extraction phase utilizes web scraping to extract the CSV files via parsing through HTTP links on the Government of Canada Weather portal for the monthly air data. The monthly air data files amounted to

26 files that are ingested into the first staging layer, **`stg_monthly_air_data`** whose associated tables have the prefix stg to indicate untreated, unfiltered, and only extracted staging data stored in stage schema as they are not in a production-ready state which is exclusive to public schema. Similarly, the geographical stations' metadata including latitude, longitude, and other pertinent information are web-scraped from a separate HTTP portal and stored in the **`stg_geo_names`** staging table.

Three additional columns are added to the staging tables; namely**,** **`last_updated_timestamp`**, **`file_source`** HTTP link, and **`filename`** to trace any duplication, data corruption, or anomalous payload to isolate its source and calculate the time between ingestion into production schema and time of acquisition from the government portal to track per-datum lifespan. Similarly, the daily forecasts are also web scraped, capturing 533 files as a possible secondary resource, and stored in **`stg_monthly_forecasts`**. However, the Traffic Volume Dataset's Python REST API was not as mature as their R's library **`opendatatoronto`** which was integrated into the data extractor procedures.

In terms of data size in the staging layer, there were a total of 1,430,988 records ingested out of which 426,584 records are retained in the production schema (Table 1). The staging layer comprised 68% or 279 seconds of the total execution time at 411 seconds.

Table 1: Number of rows in stage schema totalled 1,430,988 rows in total.

Schema	Table	Rows	Schema Proportion
stage	stg_monthly_air_data_transpose	763,559	53.36%
stage	stg_geo_names	360,254	25.18%
stage	stg_monthly_forecasts	279,921	19.56%
stage	stg_monthly_air_data	18,912	1.32%
stage	stg_traffic_volume	6,073	0.42%
stage	stg_gta_traffic_arcgis	2,269	0.16%

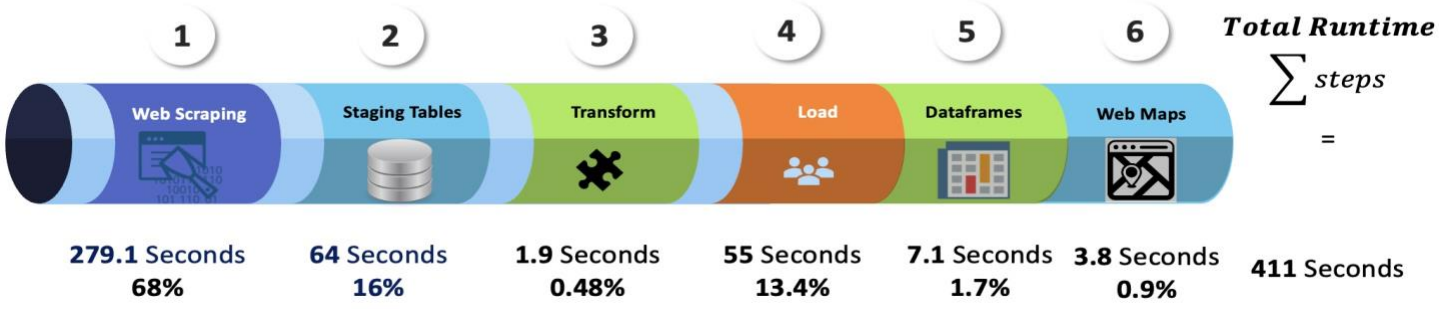

Figure 5: Execution Steps Time Duration where maps creation averaged 3.8 seconds or less than 1% of total execution time.

§4.4 Data Model Performance

As the web-scraping process involves multiple HTTP requests, it consumed 64.9% of execution time; however, transformation and loading into production took 81.2 seconds or only 21% of the total execution time. Similarly, 77% of all duplicated and null records in staging schema were eliminated leaving 23% of analytics-ready data for production. Also, using objects massively reduced execution time and expanded the scope and usability of the object's attributes.

Table 1-A: In terms of number of records, nearly a quarter or 23% of staging data was analytics-ready state in public schema for web maps and Machine Learning utilization.

Schema	Rows	Proportion
Stage	1,430,988	77.0%
Public	426,584	23.0%
Grand Total	**1,857,572**	**100.00%**

Table 2: Primary steps execution time where maps creation spanned only 14% or 58.5 seconds in total.

Step Name	Total Duration (Seconds)	Duration Proportion	Files (or Data Frames) Processed
stage	271.33	64.90%	562
production	81.22	21.10%	60
Web Maps	58.53	14.00%	47
Grand Total	**411.08**	**100.00%**	**669**

§4.5 Extended Documentation on User Environment Setup:

A public Git Repo was created to detail how to configure **`Config.ini`** to match the end user's environment settings and to reproduce the backend tables to generate most up-to-date frames via Github.

§5 AI-Based Machine Learning

Given this is a near-live pipeline capturing the most updated datasets available it becomes prone to data size inflation and fundamental changes in the statistical structures of the forecasted counts of pedestrians and vehicles. At the data sources, new monitoring stations for weather, traffic, and pedestrians could be added by the City of Toronto, Government of Canada, or ArcGIS who are the owners of the data portals. More importantly, **`Config.ini`** provides the user with the flexibility to request different forecast frequencies that are chosen from hourly, daily, monthly, quarterly, or yearly options combined with specifying the horizon from the last reported date per monitoring station.

A flexible and autonomous H_2O Machine was the most befitting tool to use under the context of growing data sizes and customizable forecast requirements. H_2O compression ratio is, on average, 1.4% of the original Pandas data frame size; thus, it allows for the accommodation of the potential increase in the

data size from the sources. "Using in-memory compression, H2O handles billions of data rows in-memory, even with a small cluster" [6]. More importantly, it provides cross-validation folds, Root Mean Square Error (RMSE) to select the super learner, ROC Curves for classifications, hyperparameter tuning, and enough overhead on the best forecast models injected into the maps.

§5.1 Anecdotal Evidence versus Statistical Truth

In the era of big data, trends may not be statistically significant and hotspots for underactive or performant regions could be due to anomalous spikes not from consistent, meaningful, and robust change. Thus, web maps could be misleading as they are categorized as descriptive methods lacking the needed statistical rigour to validate the observed phenomena.

When forecasting Traffic Conditions, according to H_2O Variable Importance Plot (Figure 6 – Top), there is a congestion at Allen Road that is of extreme significance. This is confirmed by the descriptive web where the largest hotspot for vehicular activity does indeed occur on Allen Road (Figure 6 – Bottom).

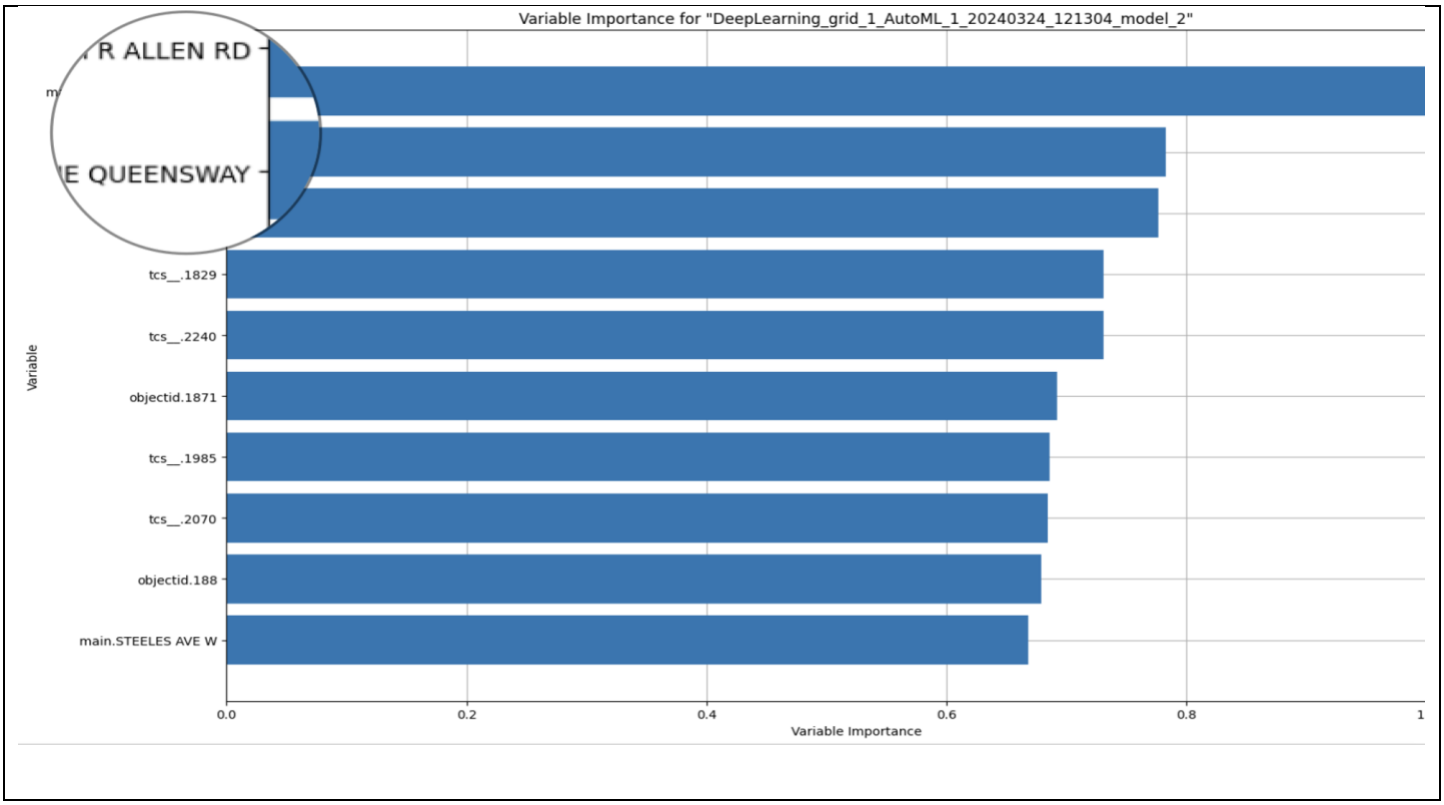

Figure 6: H2O Indicates there is a traffic hotspot at **Allen Road** that is of extreme statistical significance. The web map displays there is a hotspot at Allen Road as H_2O specified earlier in the Auto ML step.

Similarly, to forecast pedestrian flow, according to H_2O Variable Importance Plot (Figure 7 – Top), there is an overactive region for pedestrian activity on Carlton Street Road that is of extreme significance. This is confirmed by the descriptive web where the largest hotspot for pedestrian activity does indeed occur on Carlton Street (Figure 7 – Bottom).

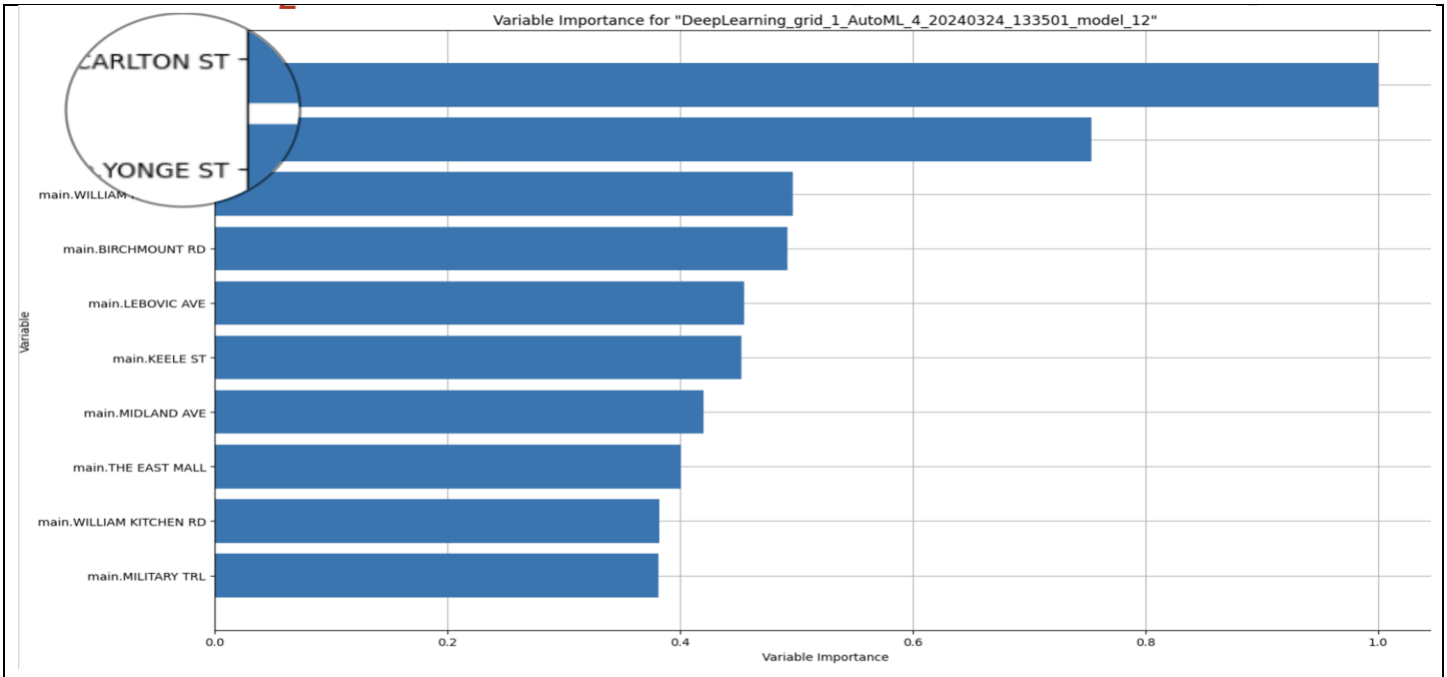

Figure 7: To forecast pedestrian activity, H_2O indicates there is an overperforming region of pedestrian activity on Carlton Street (top) which is confirmed by the web map as it displays a pedestrian hotspot on Carlton Street.

§5.2 Statistical Consensus on Temporality

§5.2.1 Traffic Volume and Vehicle Counts

The Traffic Forecast Models Variable Importance Heatmap for the top 20 models in the leaderboard all jointly indicate Traffic and Vehicles Count are strictly spatial processes heavily dependent on the location identifiers not the temporal variable **`count_date`**. This vital finding dictated that I do not emphasize animated Time-Series Heatmaps as not to mislead the user into believing traffic congestion is a spatio-temporal process when all statistical evidence eliminates the time factor (Figure 8).

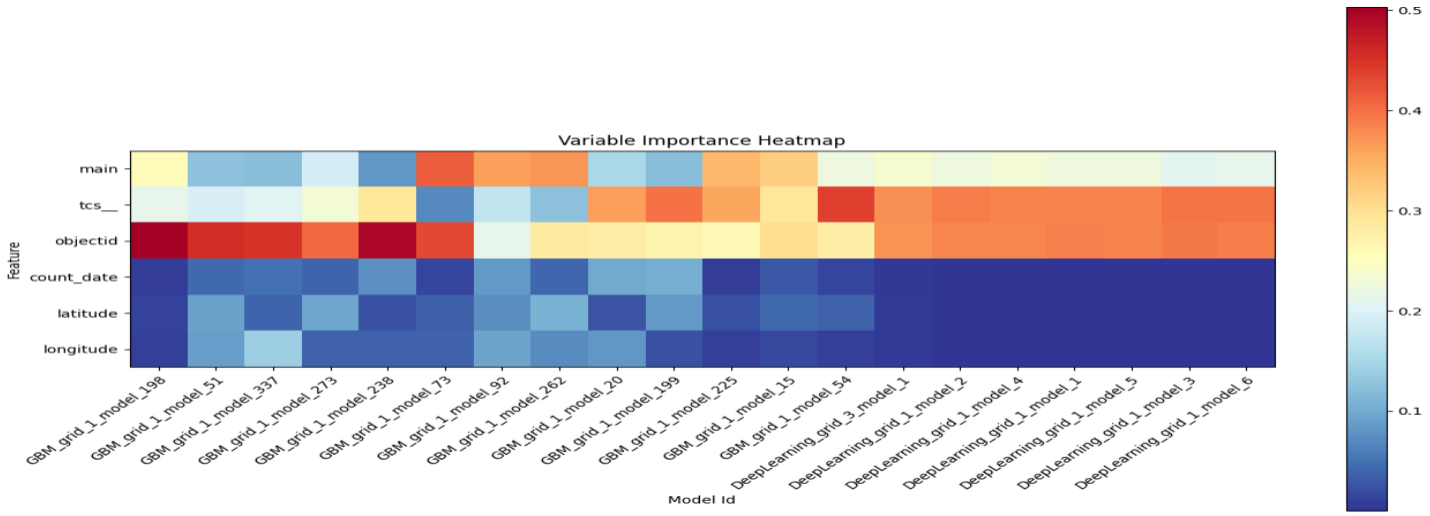

Figure 8: Traffic Forecast Models Variable Importance Heatmap for the top 20 models in the leaderboard all jointly indicate Traffic and Vehicles Count are strictly spatial process heavily depedent on

§5.2.2 Pedestrian Flow

The latter result was also evident in modelling the pedestrian flow. The temporal variable, **`count_date`**, is also not of statistical importance across all attempted machine-learning methods negating the needs to produce time-series heatmaps that may mislead the user into believing pedestrian count is co-dependent on time. Unlike the previous battery of features to predict Traffic, only two predictors were used; namely, **`count_date`** and **`main`** (for main intersection location) to amplify any effect **`count_date`** might have on predicting pedestrian counts, once again the spatial feature main was a very active predictor while **`count_date`** was weak across all the top 20 models (Figure 9). Therefore, animated time series of the pedestrian flow were omitted to warrant false conclusions based on descriptive trends of no statistical significance.

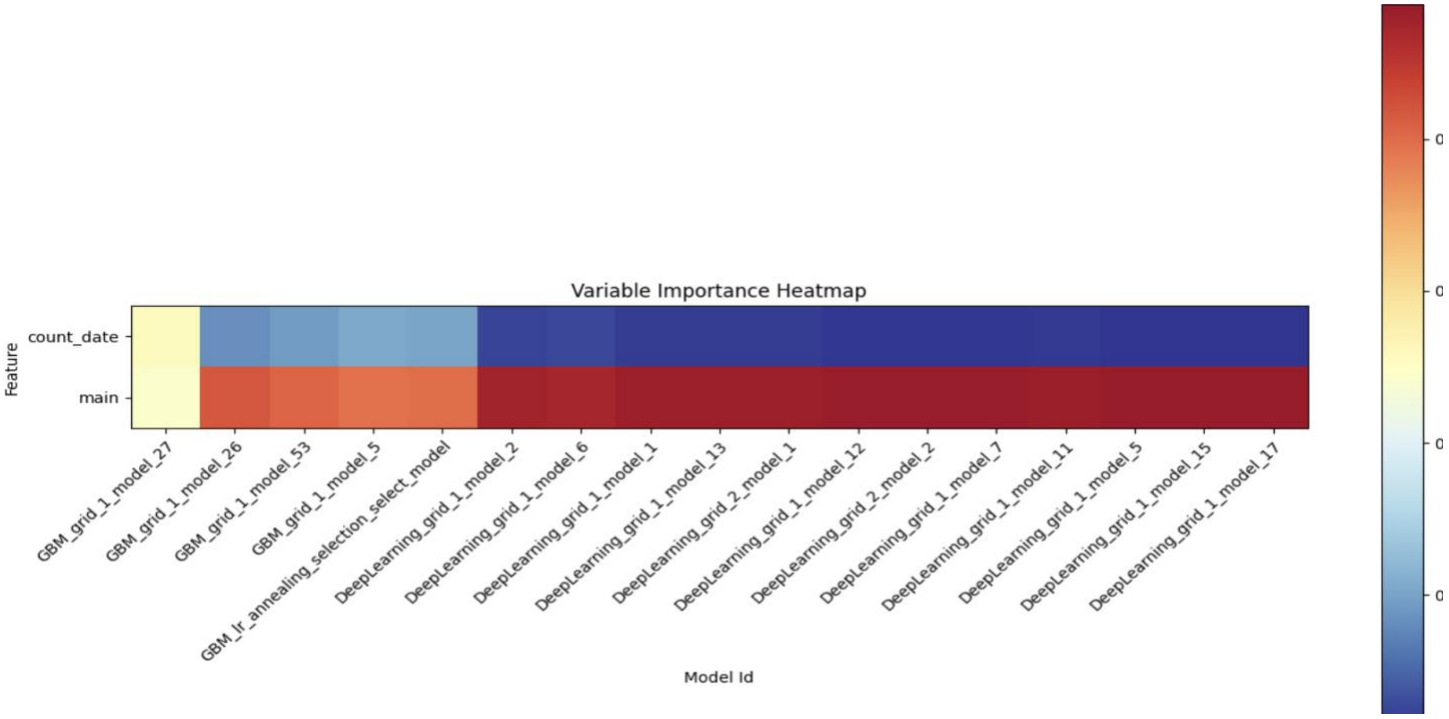

Figure 9: Pedestrian Counts prediction was very dependent on the spatial location of the monitoring station not the date of observation.

§6 Performance Testing

§6.1 Map Type and Browser Interaction

The advantages secured by H_2O for autonomous ML and phenomenal compression rates do not prevent and have no impact on, the generated HTML file size or the browser's responsiveness and speed to rendering the maps. For example, the user may request hourly forecasts for the next 90 days that are also embedded with the curated data from the pipeline. Therefore, the need arises for Performance and Load Testing on the generated HTML files.

There are three map types are considered: Turf, Folium and Mapbox which use different Python libraries to construct the map. Providing identical data structures to all map types does not mitigate the fundamental problem each Python library used to generate the map uses different approaches and some can be more efficient than the others. They do not respond identically to the tested popular browsers: Safari, Google Chrome, and Firefox. as some are slower than others. The final step, **`map_tester`** is to measure the loading time, in milliseconds, of the web map on each browser type. The interaction effect between the map type and loading time is very pronounced.

If using the Folium library, the first choice for the browser is Safari with an average of 481 milliseconds followed by Firefox with 946 milliseconds and finally the slowest is Google Chrome with 1,566 milliseconds. Mapbox loaded the fastest in Safari with 1,264 milliseconds followed by the second choice of Google Chrome at 1,870 MS and finally Firefox as the slowest browser for Mapbox.As for Turf, Safari loaded the fastest with 273 MS followed by Google Chrome at 755 ms and finally Firefox was the slowest at 866 MS (Figure 10).

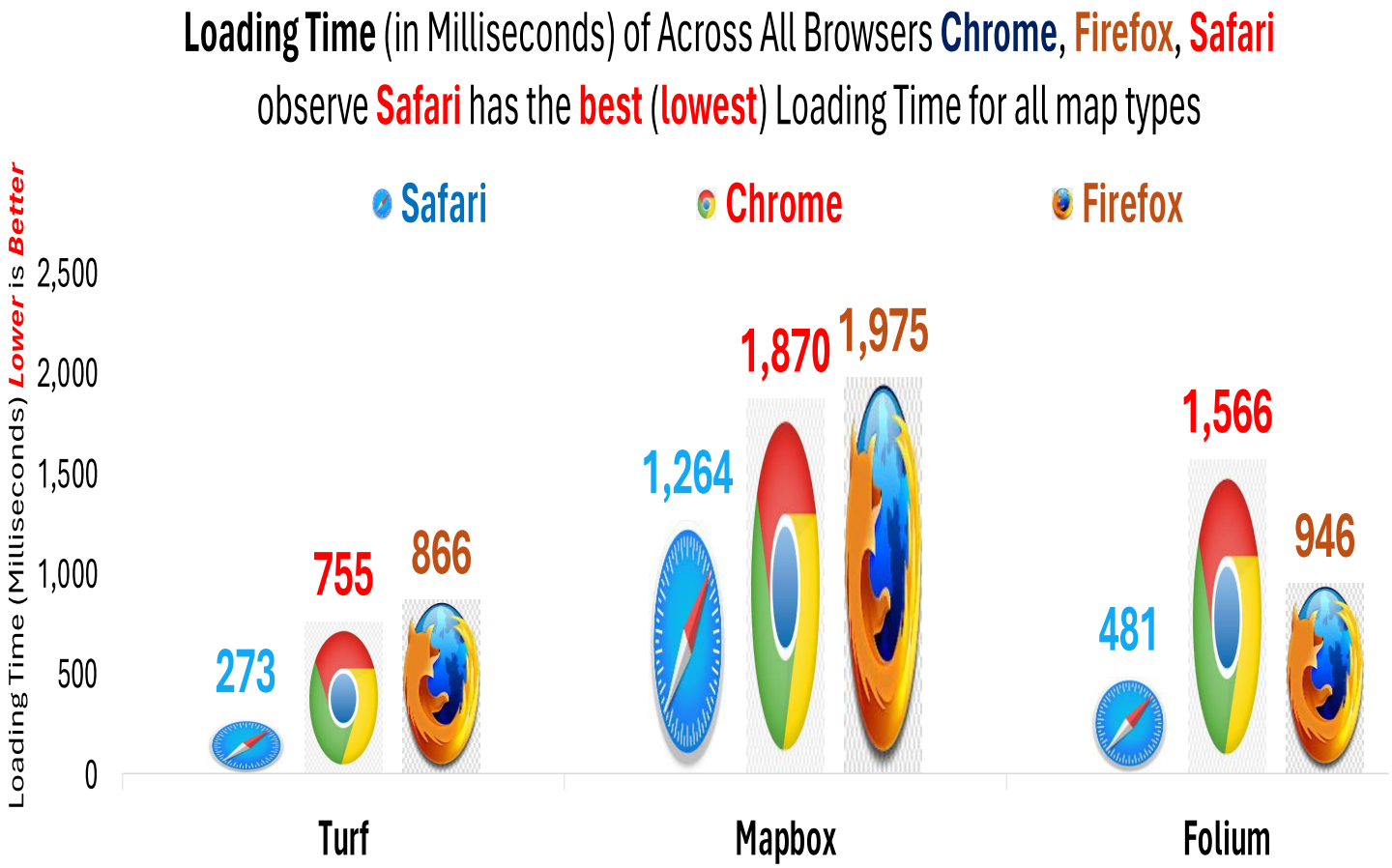

Figure 10: Browser Loading Time in Milliseconds. Observe Safari is the fastest browser irrespective of the map type. However, Google Chrome and Firefox loading speeds vary depending on the map type.

§6.2 Selection of the Optimal Browser for Each Map

The user can specify if maps should launch in the browser after they are built as indicated in the **`run_conditions`** of **`Config.ini`**. Since browser type exhibits an interaction effect with HTML map as evidenced by the loading times (Figure 11), the **`maps_tester.py`** module selects the optimal browser of the minimal loading time for each HTML file; thus, streamlining the user experience by providing the most responsive browser available for each HTML file.

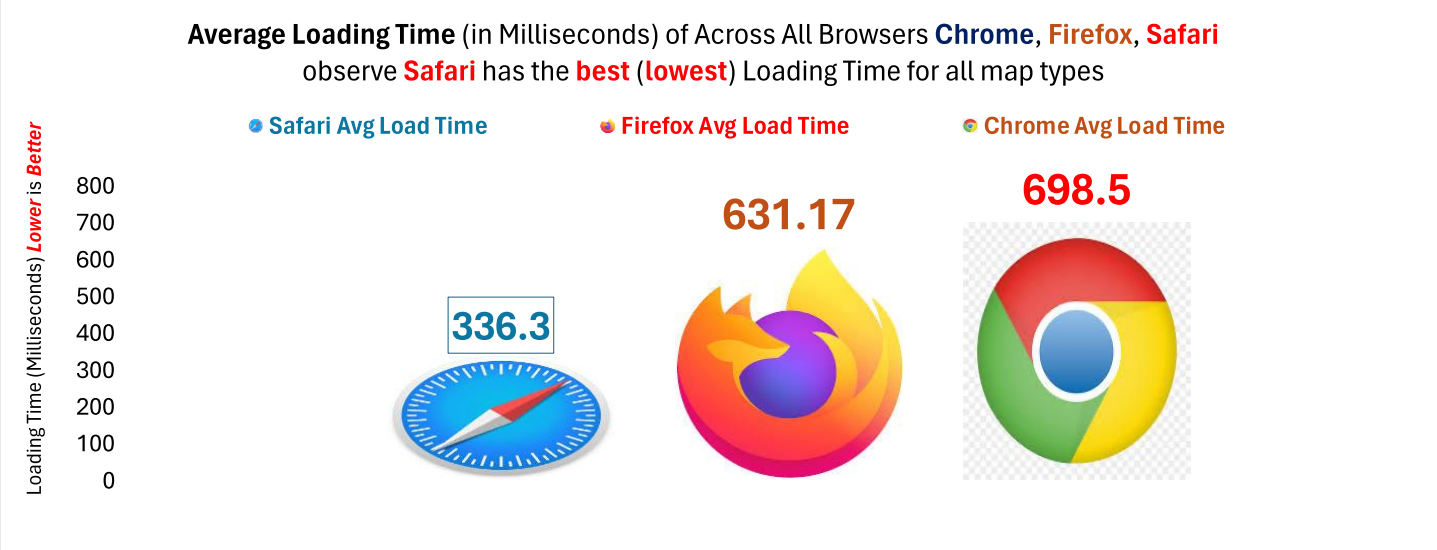

Figure 11: Safari with an average loading time of 336 milliseconds is a clear winner when the deciding factor is the average loading time it takes to render any web map file.

§7 Discussion

The findings of the statistical analysis revealed that the displayed hotspots in the descriptive web maps carry strong statistical power to deem the hotspots in pedestrians, traffic, and air quality as true indicators of hyperactivity in the region. The latter validation on the descriptive data also carries a high degree of assurance for the generated predictions based on the user's requested forecast horizon and frequency. However, the lack of feature importance of the temporal variables dictated excessively embedding time-series animations since traffic volume, vehicular, and pedestrian counts are not dependent on time but on the physical locations of the monitoring stations.

Furthermore, the nearly certain event of the increase in the data size on both the user's requested forecast horizons and the data sources was treated by H_2O's outstanding data compression and Load Testing that enabled the autonomous selection of the most responsive browsers to each map type. Those actions jointly give the user the best user experience possible without sacrificing omitting map layers or dropping records from the curated data in the production schema which may lead to a loss in statistical power.

It is equally important to emphasize each step's duration is tracked and logged in the **`data_maps_performance_tbl`** and **`data_model_performance_tbl`** production schema for easy debugging, optimization, and flagging any step that might be taking too long to execute.

§7.1 Scalability and Adaptability

The project focused on creating a unified pipeline that binds data ingestion to curation to build interactive maps in a clear, cohesive, and readable code. The application limits user input only to **`Config.ini`** and only one script to executable script, **`main.py`**, that triggers the entire process. Though one map might have been sufficient, I opted for three different map types with different implementations to dive deeper into the differences in performance between those different GIS libraries and shed light on performance and load testing before presenting the user with the optimal browser for individualized experience based on user's forecast requirements and the utilization of H_2O to safeguard against unexpected increases in data size giving robustness to python implementation.

§8 Areas of Improvement

§8.1 Improvements of Machine Learning Methods

The forecast model is very thorough in making predictions, not globally, but for every monitoring station with a customized forecast horizon based on its own last reported date. However, it can be improved by unifying all data sources into one master table. The latter step would bring all the records in the production schema totalling 426,584 records which are more than sufficient to enhance the robustness of the prediction model as opposed to three different forecast models in each section: traffic, vehicle count, and pedestrian flow. This was observed as most of the super learners were a Gradient Boosting Machine (GBM) which is indicative of excessive variability and nearly identical feature space among those three predictors. One unified dataset will give H2O the chance to select other classes of forecast models from Distributed Random Forests (DRF), Artificial Neural Networks (ANN), or Generalized Linear Regression (GLM).

There is a likelihood of the event of a drastic change in the statistical structures of the response variables where temporal predictors become statistically significant. Accordingly, there needs to be a sub-algorithm that checks the statistical significance of temporal covariates in the Auto ML Leaderboard and embeds more layers of animated time series heatmaps since time-dependent web maps convey statistical truth and instill statistically sound conclusions with the user.

§8.2 More Summary Charts Embedded into Markers

Given the expansive time scope of some of the datasets such as ArcGIS from 2011-09-08 to 2016-09-07 and Environment Canada's meta information on the monitoring stations from 1850-01-01 to 2024-03-05, more summary statistics such as histograms and violin plots need to be embedded into the markers. Currently, there is only one layer of Air Traffic Charts where each marker carries a histogram of the Air Quality Index for the lifetime of the selected location (Figure 12). More layers such as these are needed.

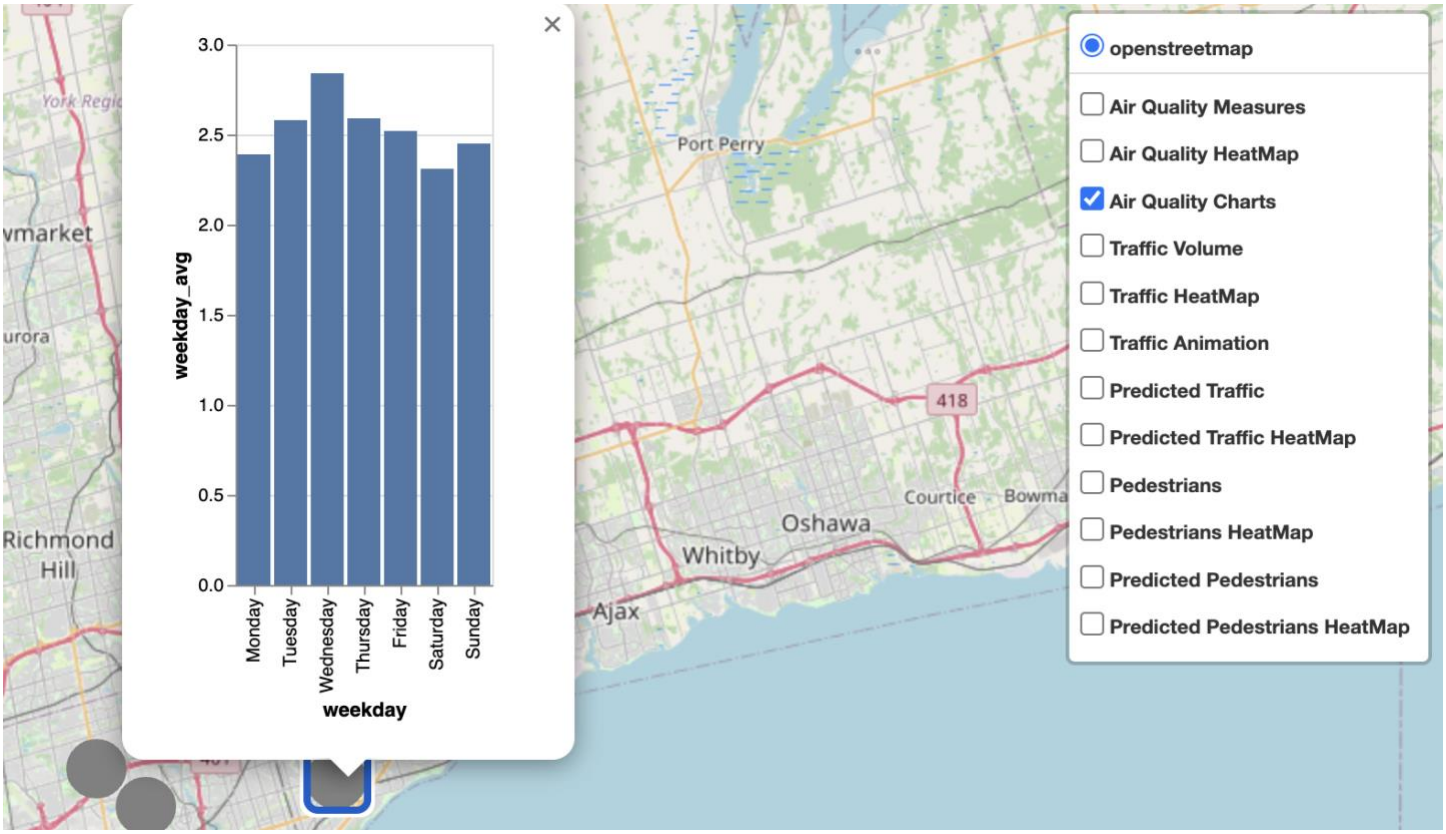

Figure 12: Given the many measurements taken at each monitoring station, more descriptive summary charts are needed.

§8.3 One Color Gradient for both Markers and Heatmaps

If the measurement falls on the global mean, it is coloured **black**. Should it fall below the global mean, it is coloured red, and if it **exceeds** the global mean, it is coloured **green** (Figure 13). The colours need to be assigned the same **colour gradient** as the heatmap such that the colour intensity of all the embedded layers correlates to the statistical relationship to the grand mean of the select user-selected layer.

Figure 13: The markers are assigned **black**, **red**, or green in the event they **fall** on, **fall below**, or exceed the grand mean respectively.

§9 Conclusion

This project constituted a serious undertaking of web map development that is customizable to the user's requirements and resistant to any drastic changes in both data size and the probabilistic structure of the target variables. It unifies machine learning, SQL procedures, and interactive plotting that rigorously predicts user experience by simulating stress-loading the HTML files before they are presented to the user. Even though 426,584 records that are used to produce the maps may not constitute 'Big Data' in the order of terabytes; however, given the fast responsiveness of the maps whose average load time is 336 milliseconds and file size never exceeding 10 megabytes provide a good starting point to further improve upon statistical modelling and diversify the data sources for better data curation and more elaborate visualizations.

References

1) Shamsi H, Munshed M, Tran M-K, Lee Y, Walker S, The J, Raahemifar K, Fowler M. Health Cost Estimation of Traffic-Related Air Pollution and Assessing the Pollution Reduction Potential of Zero-Emission Vehicles in Toronto, Canada. *Energies*. 2021; 14(16):4956. https://doi.org/10.3390/en14164956

2) *Toronto and GTA traffic: Ongoing construction closures*. CityNews Toronto. (2024, March 15). https://toronto.citynews.ca/traffic/

3) van der Laan, Mark J.; Polley, Eric C.; and Hubbard, Alan E., "Super Learner" (July 2007). *U.C. Berkeley Division of Biostatistics Working Paper Series.* Working Paper 222. https://biostats.bepress.com/ucbbiostat/paper222

4) Canada, E. and C. C. (2021, April 28). *Government of Canada*. Canada.ca. https://www.canada.ca/en/environment-climate-change/services/air-quality-health-index/about.html

5) Locations of air-pollution–monitoring stations in Toronto. (n.d.). https://www.researchgate.net/figure/Locations-of-air-pollution-monitoring-stations-in-Toronto_fig1_7685480

6) Aiello, S., Eckstrand, E., Fu, A., Landry, M., and Aboyoun, P. (Jan 2018). Machine Learning with R and H2O. http://h2o.ai/resources/