Indicators of Collision Severity

UBC MDS Capstone Project with UrbanLogiq

June 2019

Authors
Sayanti Ghosh
Jes Simkin
Evan Yathon
Mike Yuan

UBC MDS Mentor
Vincenzo Coia

High-Level Summary

In line with a Vision Zero framework, UrbanLogiq seeks to identify contributors to severe traffic collisions, summarize this information and pass it to relevant city stakeholders. With this information, city stakeholders are able to influence their public policy to continue working towards a Vision Zero commitment. This project prioritizes inference, interpretability, and reproducibility while identifying indicators of collision severity using logistic regression and XGBoost with SHAP values. Products of this project are a scientific report detailing the project's analysis, results, and recommendations, in addition to a complementing GitLab repository consisting of Python scripts, notebooks, and visualizations.

Introduction

UrbanLogiq offers data analytics services for government bodies to '[build] better communities with data'. Some of their traffic analysis work incorporates a Vision Zero framework. Vision Zero is a global project to "eliminate all traffic fatalities and severe injuries while increasing safe, healthy, equitable mobility for all". The initiative advances the idea that traffic collisions resulting in fatal or major injuries are preventable.

In line with Vision Zero, UrbanLogiq seeks to understand factors contributing to major and fatal collisions for a client city. UrbanLogiq's goal is to communicate these factors in an interpretable manner to aid their clients' public policy and planning initiatives towards building better communities. As such, the main scientific objective of this project is to identify collision severity indicators in a city. Maintaining UrbanLogiq's mission in mind, a successful approach for achieving the main scientific objective requires prioritizing inference, interpretability, and reproducibility.

This report will outline the project's data, methods, data product, results, and recommendations for future work. Due to confidentiality, the city examined for this project cannot be disclosed in this report.

The Data

UrbanLogiq provided eight datasets, including seven geospatial sets describing geographic city attributes and one collisions set containing traffic collision records. To complement the data provided by UrbanLogiq, we obtained five additional open source datasets.

The primary dataset in this study is the collisions dataset provided by UrbanLogiq which contains collision records from 2008 to 2018. Each row is a collision and contains information such as time, date, location, lighting conditions, vehicle type, and injury severity. While geolocation for each collision was not initially provided with the dataset, UrbanLogiq later provided this information.

The seven geospatial datasets provided by UrbanLogiq include city data for sidewalks, buildings, intersections, road centerlines, zoned parcels, parcels and road signals. Each dataset contains different features with geolocation data. For example, the road centerlines dataset contains speed limits for each road in the city, and the buildings dataset contains building heights for each building in the city.

We obtained five additional open source datasets to complement the data provided by UrbanLogiq. These datasets include economic and demographic data at the neighbourhood level, geolocation of bus stops and bikeways, as well as statutory holidays. Most of these datasets were downloaded from local government open data portals (transit, health departments).

Data Science Methods

The Observational Unit

An important first step was to define an observational unit that aligned with UrbanLogiq's business objectives and a Vision Zero framework. Our observational unit is defined as a collision related to an intersection and having occurred between 2008-2018. Furthermore, our observations were encoded with a binary response variable. A positive binary response was defined as a collision resulting in a major or fatal injury, while a negative response was all other collisions. In turn, any result would be conditional on a collision having occurred at an intersection. This means that findings would be interpreted as influencing collision severity, not influencing a collision occurring.

This choice of observational unit resulted in a very specific interpretation, but one that was confirmed by UrbanLogiq as useful. Knowing which factors can influence a collision to be more severe is in line with Vision Zero and is useful for policymakers. A binary response also aligns with logistic regression, a technique fulfilling the interpretability requirement.

An alternative could have been to define an intersection as an observational unit with a response of incident count, regardless of whether a collision had occurred. Compared to the chosen observational unit, this alternative would instead emphasize static intersection surroundings. While this focus would have yielded a different yet useful interpretation, it was decided against as it would exclude collision-specific data such as time of day or roadway conditions. This method was also rejected due to project time constraints.

Exploratory Data Analysis

Exploratory data analysis was conducted to gain familiarity with the data. Interactive visualizations were created with Folium, a Python package harnessing the interactive mapping power of the open-source Javascript package, Leaflet.js. Static visualizations exploring data distributions were created with software including Matplotlib, Seaborn, Plotnine, and Kepler.gl.

Static visualizations for features across various datasets presented initial avenues for further analysis. Particularly insightful visualizations described the response variable's distribution with regard to features such as time, lighting, and being located at an intersection. Also useful were static visualizations for geospatial features and collisions. While packages like Matplotlib were useful to start gaining an understanding of geospatial data distributions across the city, they were limited to only conveying very high-level insights.

Folium was used to create richer geospatial interactive visualizations. These interactive visualizations allowed for zoom interaction, data layer toggling, street-level context with OpenStreetMap data (including street names, amenities, buildings, etc), and more. With Folium, interactive HTML maps were made displaying collisions with other geospatial data overlain (eg. speed limits, neighbourhoods, etc). Moreover, these maps provided street-level and city-wide distributional insights of collisions and features, aiding with work related to geolocation validation and feature engineering.

Feature Engineering

Features from relevant geospatial and open source datasets were merged into the collisions dataset to enhance observational units with more contextual data, especially with regard to intersection surroundings. Spatial joins were performed using the Geopandas package. A 200 foot radius was defined around each intersection occurring in the collisions data, and relevant features from other geospatial datasets that fell within this radius were joined to the collisions data. A 200 foot radius was chosen based on the estimation of an average size of a city block size as well as to facilitate interpretability. For example, a 'total number of buildings within 200 feet' feature was added from the buildings dataset to each row of the collisions dataset using geolocation data.

Inaccuracies were found and fixed for some geolocations in the collisions dataset. These inaccuracies imply a possible loss of precision for some engineered features at some intersections. This was confirmed with UrbanLogiq to be an existing issue, and identified as a future improvement but one that was beyond the scope of this project.

Using solely the collisions dataset implied finding crash severity indicators for the entire city. To complement the city wide strategy, a community-based approach was implemented. As it was recognized that different areas of the city may have different crash severity indicators, unsupervised learning techniques were used with neighbourhood-level demographic data and zoning data to cluster similar neighbourhoods. Demographic data was used for insight into resident populations and zoning data was used for insight into land usage. Clustering allowed for the creation of three data subsets for modelling, in addition to the city-wide primary dataset.

An alternative local approach to consider would be clustering similar intersections together rather than similar neighborhoods. We decided to continue with neighbourhood-level clustering

instead, with the assumption that neighbourhood-level insights would be more useful for policymakers.

Modelling

Logistic Regression and XGBoost were used to select important features before modelling. Only features selected by both logistic regression and XGBoost were then used for modelling. With Logistic Regression, features were selected using L1 regularization with various regularization strengths. With XGBoost, we used methods such as forward selection, selectKbest and SHAP values to identify important features.

The criteria to choose the most suitable models for feature selection was to prioritize recall performance for predicting major incidents. Recall was the chosen metric to penalize misclassification of major and fatal collisions and align modelling with Vision Zero. It was confirmed by UrbanLogiq that over predicting severe incidents was preferable when compared to underpredicting severe incidents. Further feature selection was performed manually using our intuition and UrbanLogiq's domain expertise, selecting only features that could be useful to policy makers.

After features were selected for each neighbourhood cluster and the whole dataset, logistic regression was used with bootstrapping to provide confidence intervals for the coefficient of each important feature. An advantage of bootstrapping over the central limit theorem is not relying on the normality approximation. Some estimates were not normally distributed, and bootstrapping confidence intervals captured this variation to provide a wider confidence interval. Combining the output of sklearn's logistic regression with bootstrapping provided more informative coefficient information. In addition to logistic regression, XGBoost with corresponding SHAP values were used to rank features by importance.

Logistic Regression and XGBoost with SHAP values were chosen as models as they both aligned with our response variable while providing interpretability. Logistic regression was favoured for providing interpretability when paired with effect coding. XGBoost with SHAP values, being a tree-based method, was recommended by UrbanLogiq and chosen as it is more robust when dealing with class-imbalanced data while also providing interpretability.

Alternatives considered were mixed effect models and SVCs. These alternatives were not chosen due to time-constraints and in order to prioritize interpretability.

Data Product & Results

The data product is comprised of a scientific report complemented by a Gitlab repository. The goals of the data product are two-fold. Firstly, that the data product complements UrbanLogiq's existing work and serves as a springboard for future work. Secondly, that the data product enables UrbanLogiq to reproduce this study. Both the scientific report and the GitLab repository were designed with usability, reproducibility and interpretability in mind.

The scientific report details the project's analysis, results and recommendations for future work. Consideration was taken to design the report's formatting and structure in a way that best fit the analysis and methods, time-constraints, and UrbanLogiq's reporting best practices.

As described in the scientific report, the results of the study indicate that accounting for certain factors and degrees of uncertainty, there appear to be specific factors contributing to major and fatal collisions at the intersection level in the city. Furthermore, it appears that different types of neighbourhoods have different types of indicators. This suggests that geospatial context matters when identifying contributing factors of major and fatal collisions at the intersection level.

The GitLab repository contains Python scripts, custom functions, Jupyter notebooks, visualizations (static and interactive) and pickled models. In prioritizing reproducibility and useability, the repository's directories and notebooks have clear and informative documentation and usage instructions throughout.

While UrbanLogiq determined the data product format from the outset, alternative product formats to be considered for future projects with the same data and objectives could include a predictive model or dashboard tool.

Conclusion & Recommendations

Overall, this project identified crash severity indicators for a city while incorporating both city-wide and community-based approaches aligning with Vision Zero. The methods prioritized inference, interpretability, and reproducibility. Our results indicate that there are different factors associated with severe collisions across different neighbourhoods, suggesting that incorporating a local approach to this research is important. Logistic regression and XGBoost with SHAP values proved to be useful models for providing interpretable crash severity indicator effects at the intersection level. The project's scientific report and GitLab repository will provide UrbanLogiq with a framework to reproduce the project findings or explore the research objectives with data from a different city.

Recommendations are outlined in the scientific report and include possible improvements for feature engineering and modelling. One recommendation is to use GoogleMap geolocation API for collisions to provide better geographic coverage and precision. Another recommendation is to incorporate amenity data to account for population generators throughout the city such as amusement parks or arenas and perhaps to account for neighbourhood walkability. Regarding modelling, another tree-based model such as LightGBM can be used instead of XGBoost. LightGBM tends to provide better performance, accuracy and memory efficiency.