Predicting streetcar delays

General Machine Learning

Andrei Cheremukhin
cheremuk@stanford.edu

In this project we are going to solve a problem of predicting streetcar delays in Toronto, Canada. The problem to solve is gridlock: how to predict delays in the streetcar system so they can be avoided. The streetcar network has many advantages over other modes of transit (last longer, no emission, cheaper to build and maintain etc.). But they have one big disadvantage: when a streetcar gets blocked, it can cause compound delays in the streetcar network and contribute to overall gridlock on the city's busiest streets. This project is practical application. I am going to use a publicly available dataset that describes every delay encountered in the streetcar system in Toronto since January 2014.

"TTC Streetcar Delay Data"
https://open.toronto.ca/dataset/ttc-streetcar-delay-data/

This dataset is live. It gets updated monthly, so there is an opportunity to test different models with data that they have never seen before. The dataset represents an XLS file for each year, in each file XLS file, a tab for one month and contains the following fields:

- Report Date
- Route
- Time
- Day
- Location
- Incident
- Min Delay
- Min Gap
- Direction
- Vehicle

The dataset currently has over 90 thousand records and between one thousand two thousand new records are added every month. That dataset is not too big and too small which makes it's interesting for methods we are going use in this project: logistic regression and neural networks.

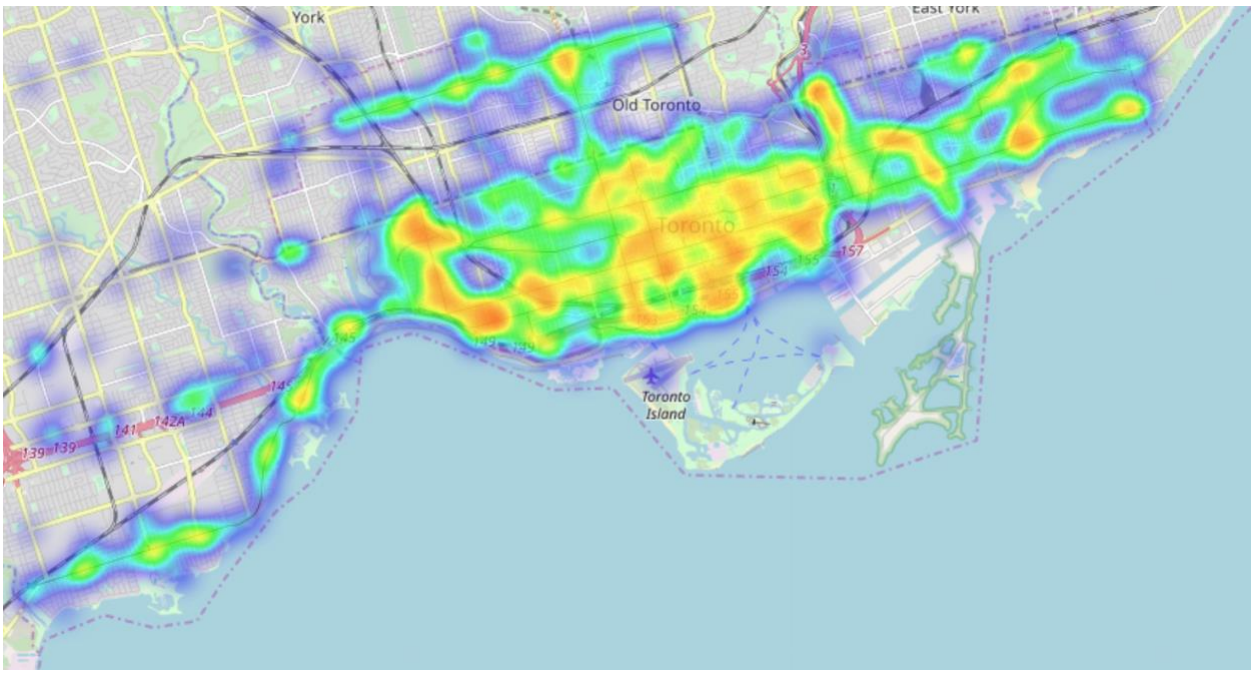
Preparing the Data.

The very first thing we have to is cleaning the data set. The available data contains a lot of anomalies: e.g. the April 2019 tab has columns called **Delay** and **Gap** rather than **Min Delay** and **Min Gap** like all the other tabs in the original dataset.

	Report Date	Route	Time	Day	Location	Incident	Min Delay	Min Gap	Direction	Vehicle	Report Date Time	latitude	longitude
1	2016-01-01	511	02:14:00	Friday	fleet st. and strachan	Mechanical	10.0	20.0	e	4018	2016-01-01 02:14:00	43.636298	-79.409635
2	2016-01-01	301	02:22:00	Friday	queen st. west and roncesvalles	Mechanical	9.0	18.0	w	4201	2016-01-01 02:22:00	43.645335	-79.413184
3	2016-01-01	301	03:28:00	Friday	lake shore blvd. and superior st.	Mechanical	20.0	40.0	e	4251	2016-01-01 03:28:00	43.614962	-79.488658
5	2016-01-01	505	15:42:00	Friday	broadview station loop	Investigation	4.0	10.0	w	4187	2016-01-01 15:42:00	43.677135	-79.358208
6	2016-01-01	504	15:54:00	Friday	broadview and queen	Mechanical	6.0	12.0	e	4181	2016-01-01 15:54:00	43.659363	-79.347697

We applied the following cleaning procedures to the original dataset:

- removed unexpected columns from XLS files: Delay, Gap, Incident ID. These columns are present only in some tabs for 2019 year. The majority of dataset doesn't have them.
- removed all rows with missing values in some columns. Location, Min Delay, Min Gap and Direction are not present for approximately 10% of the dataset.
- unified directions. Upon investigating the original dataset, we figured out that direction should be categorized and can have 5 valid values: 'e' for Eastbound, 'w' for Westbound, 's' for Southbound, 'n' for Northbound, and 'b' for Both Ways.
- removed invalid routes and vehicles. We used publicly available information to get full list of currently used routes and vehicles by the Toronto street car system and removed everything which is not being used anymore. https://www.ttc.ca/Routes/Streetcars.jsp https://en.wikipedia.org/wiki/Toronto_streetcar_system_rolling_stock - CLRVs_and_ALRVs
- converted location to latitude and longitude. Location presents us a unique problem: it's given in text-free format with typos, errors and in inconsistent form. Initially, we tried to unify all locations by applying text transformation functions (change all values to lower case, remove punctuation, use consistent tokens, make the order of streets in junction values consistent) and categorize that column. After applying above transformation, we still got over 8000 unique values. Instead, we decided to convert all locations to latitude and longitude with Google's geocoding API https://developers.google.com/maps/documentation/geocoding/start
  We managed to convert over 80% of the original locations to a tuple (latitude; longitude). The below figure shows the heatmap of delay counts.



Deriving the Data.

In order to apply the chosen machine learning techniques, we have to modify our original data set: create a data frame with records for every date, hour, route, direction combination since Jan 1 2014 and join with the original data frame generating target column that is 1 if there was a delay in that date / hour / route / direction combination and 0 otherwise. After applying that transformation, we can apply our classification machine learning algorithms and could use that model in order to predict delays in the future.
The resulting data set is somewhat imbalanced: the ratio is 1:48 between examples with labels 0 and 1, respectively.

	Report Date	count	Route	Direction	hour	year	month	daym	day	Min Delay	target
0	2014-01-01	0	301	e	0	2014	1	1	2	0.0	0
1	2014-01-01	0	301	e	1	2014	1	1	2	0.0	0
2	2014-01-01	0	301	e	2	2014	1	1	2	0.0	0
3	2014-01-01	0	301	e	3	2014	1	1	2	0.0	0
4	2014-01-01	0	301	e	4	2014	1	1	2	0.0	0

Deriving data set contains 2540912 examples; approximately, 50000 of them are positive which means during that time slot there was a gridlock.

Training the Model.

We split the data set in the following proportions:

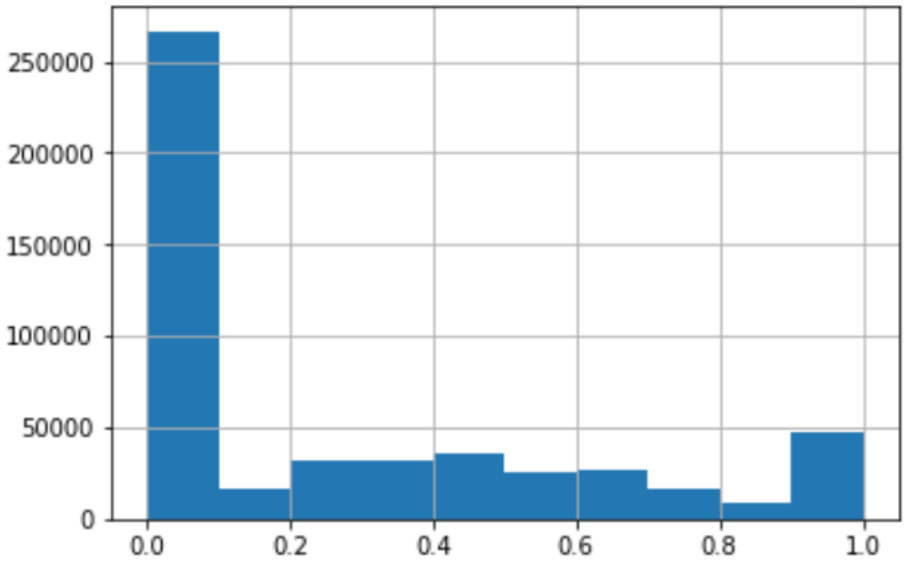
- 70% for training

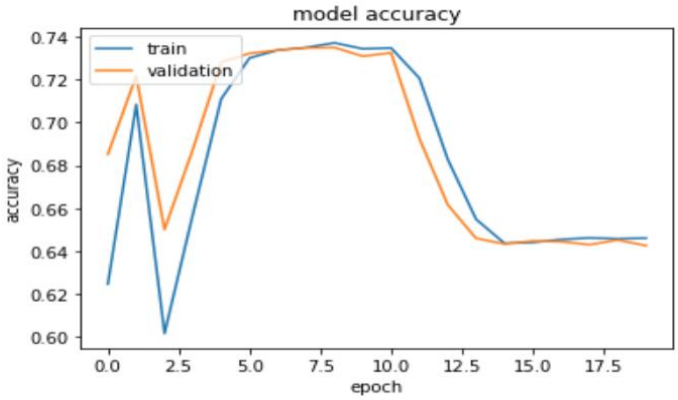
- 10% for validation
- 20% for testing

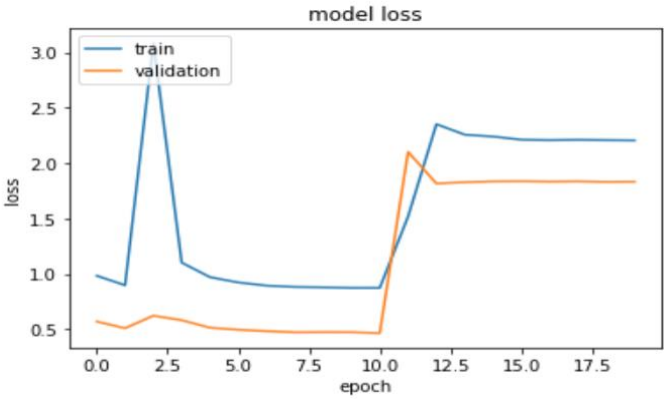
After preparing the dataset we trained vanilla regression with scikit-learn's **LinearRegression** implementation. We used the following features:

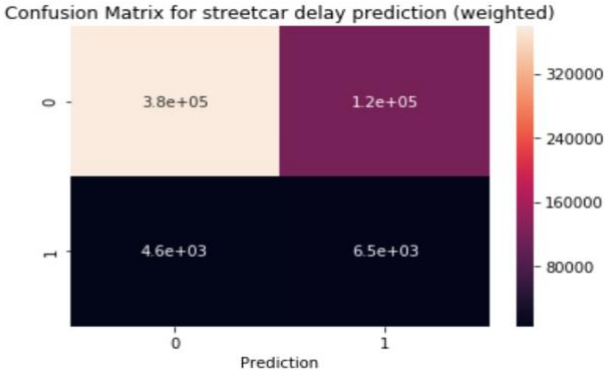
- Route
- Direction
- Hour
- Month
- Day of Week
- Time of Day: morning, midday, overnight

As a target feature we used Boolean target **Min Delay** > 0. We trained the logistic regression and tried to use a different threshold for predicting gridlocks (*0.5, 0.6, 0.7, 0.8, 0.9*) but results were still quite poor: the best accuracy on test set was only 0.37.
Going through the incorrectly predicted examples and visualizing the test set with PCA we clearly could see the dataset is not linearly separable. Next, we applied neural network with sigmoid activation layer and using binary cross entropy as loss function. We used deep learning framework **Keras** in order to train and validate our model. Results look significantly better.









Future work.
An interesting step would be to add open source historical weather data to the streetcar dataset to make it possible to data-driven analysis to determine if there is a correlation between the number of delays and unsettled weather.
https://www.canada.ca/en/environment-climate-change/services/climate-change/canadian-centre-climate-services/display-download.html

Summary.
In this project we successfully applied the following classification machine learning algorithms in order to predict streetcar delays in Toronto: logistic regression and neural networks.
Code.
https://drive.google.com/file/d/1S4NeaHB10R0rSIPUQCpZ9KWXzd1-96mB/view