IBM Advanced Data Science Program

November 2020 Capstone Project Presentation

Talking Points

- Use Case
- Data Set
- Data Quality Assessment
- Data Exploration (e.g. correlation between columns)
- Data Visualization (e.g. value distribution of columns)
- At least one Feature Engineering (e.g. imputing missing values) applied
- Selection and justification of Model Performance Indicator (e.g. F1 score)
- At least one traditional Machine Learning Algorithm and one Deep Learning Algorithm applied and demonstrated
- Model performance between different feature engineering and models compared and documented

Use Case

- Use case is collision location prediction. This translates into providing a data product used by Seattle law enforcement, first responders and planners to show the predicted location of future collisions.
- The data product features a map where predicted hotspots are displayed by Seattle Micro-Community Policing Plans (MCCP) regions.
- In Seattle, each community has is different characteristics and rather than attempting to predict collisions at across the entire city the data product attempts to focus on individual MCCPs providing a unique lens to view collision activity for its stakeholders.
- The goal is to leverage a combination of data mining, clustering and regression to offer the best possible interpretation of what could happen in a particular MCCP on a given day, time, weather conditions and historical data.
- The data product will use the Seattle Department of Transportation (SDOT) collision dataset, weather station data, The data product will not be interactive but will show a map covering all of Seattle along with separate maps for each MCCP.
- This use case is ambitious considering my skills, time and resources but should be a lot of fun and a great learning experience. Let's see how it goes!

Project Architecture

nbviewer
Voilà
Jupyter Hub

Crash Data

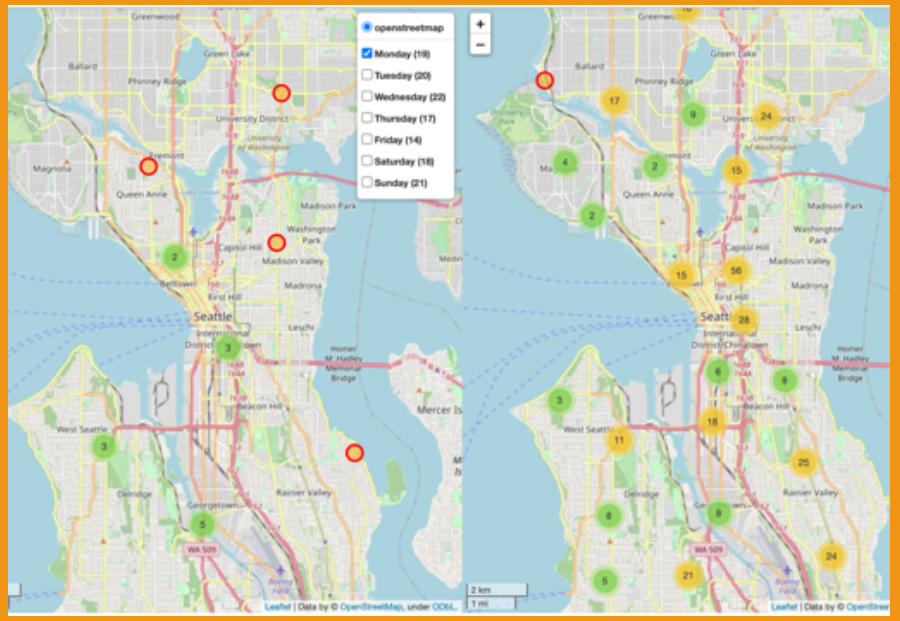
Met Data

Preprocess, Features, Evaluation

Time Series Prophet

Clustering Ball Tree

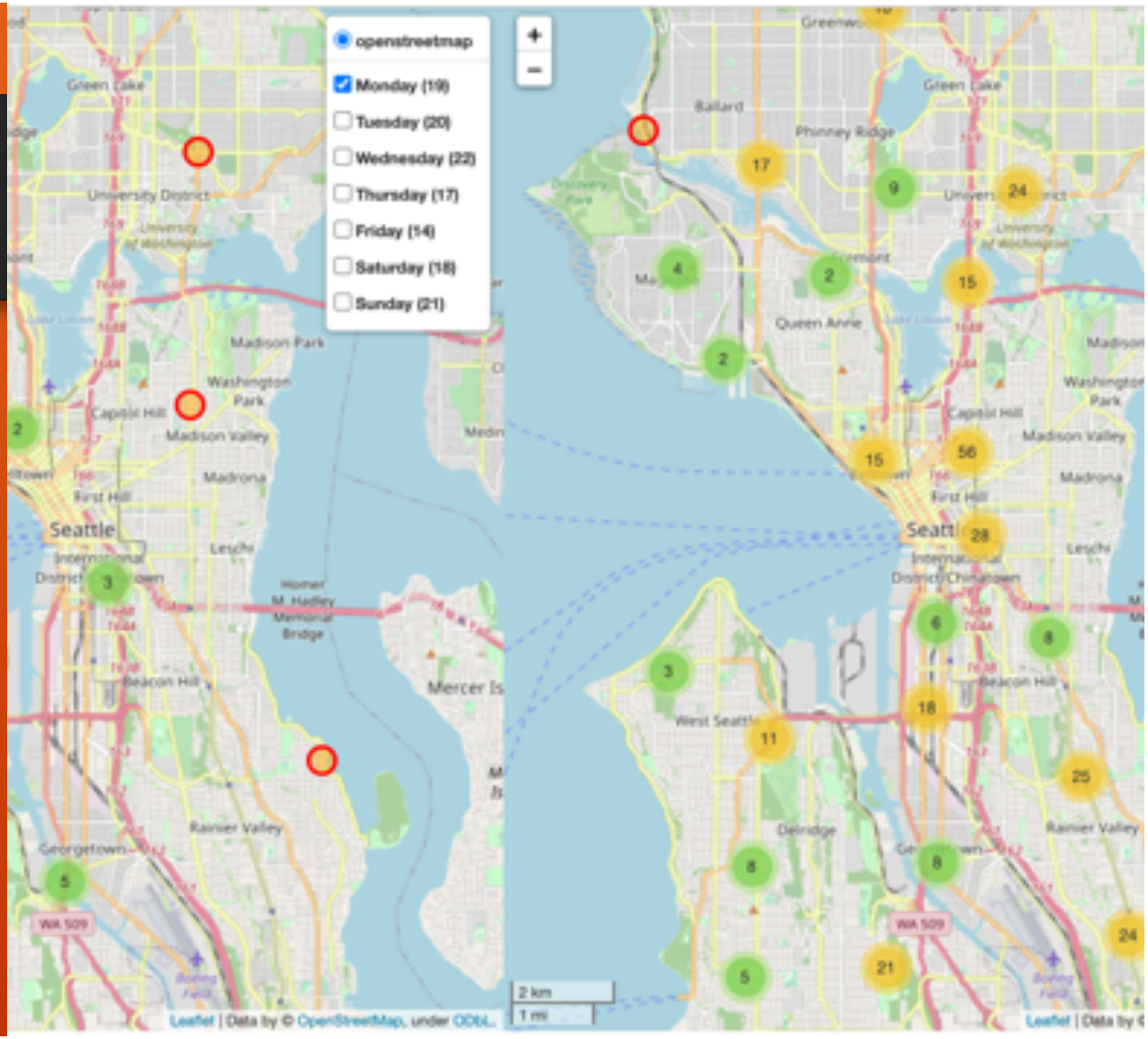
ASUM-DM



User Application – JupyterLab, Folium, ipywidgets

Application Map View

- End “data product”
- Map-based interface
- Presents side-by-side maps to the user
- Left map displays predicted collisions locations
- Right map displays actual
- Predictions are based on the Seattle Dept. of Transportation dataset
- Runs in a JuypterLab Notebook



Project Assets

- seattle_collisions.data_exp.python.1_0.ipynb
- seattle_collisions.etl.python.1_0.ipynb
- seattle_collisions.feature_eng.python.1_0.ipynb
- seattle_collisions.model_def.python.1_0.ipynb
- seattle_collisions.model_train.python.1_0.ipynb
- seattle_collisions.model_evaluate.python.1_0.ipynb
- seattle_collisions.model_deployment.python.1_0.ipynb
- Architectural Decision Document (ADD)

Data Set

- Publicly available from the Seattle Department of Transportation (SDOT)
- Includes collision data from 2004 recently updated to October 2020
- This project uses a subset of that data from 2015 to 2020
- Data is collected by the police and curated by the traffic management authorities
- Data is limited and does not include information about the driver(s), vehicles, detailed weather information
- Has issues with quality, although SDOT does a great job with it, on the whole issues can be addressed with feature engineering
- Offline - this data would be great to have in near-time. It had not been updated for many months. The more current the data the better the intelligence
- Lack of traffic flow data – I did discover flow data it was difficult to integrate into the project due to time constraints.

Data Quality Assessment

- Quality was checked heavily during the EDA, ETL and Feature Engineering stages
- During EDA, each field was checked and visually inspected
- Categorical and numeric data verified
- Weather data was analyzed and repaired where possible. The SDOT data was spot checked and the weather was not always correct, e.g. not raining when it was, so it was overridden by a separate data set
- Nulls were replaced with imputed values or removed

Data Exploration

- Available in the notebook
- All attributes where checked, visual inspection and/or plotting
- Assessment of feature quality, and possible new features. The primary focus location data (latitude and longitude) which was solid. It appears that geocoordinates where entered after logging the collision (placed on police location in written form)
- Distributions were reviewed
- Mapping performed to examine existing locations
- Interesting conclusions after a detailed review, e.g. parked cars

Feature Engineering

- Data
  - Type checking, much of the data is plan text, date conversions, nulls, floats needed to be addressed
  - Visualizations used validate distributions (mostly done in EDA)
  - Categorical values managed, handling many missing values
- Geohash
- Geocoordinates
- Augmented data with weather and solar information
- Where applicable one-hot encoding
- Timestamps, dates wrangled

Model Performance Measures

- Time Series
  - MAE/RMSE
  - Visualization
  - Helped by Prophet tools
- Clustering
  - Plenty visual inspection, mapping utility built in Folium to review
  - Utilized various clustering algorithms to cross check results
  - Silhouette scores where used, however were problematic. There are a many dissimilar collisions in small areas, so adjustments were needed

Modeling

- Multiple tools reviewed, none had the right fit
- Ensembles XGBoost and Random Forest were initially considered. Data is ready for it but were not a good for intermittent time series
- Zero-inflated Poisson and Binomial Regression via statsmodels but data sparsity could not be overcome
- SARIMA, Croston Model also reviewed
- Finalized on Prophet, easy to work with and très Pythonic
- LSTM – used to compare against Prophet results. Results were promising but not as good as Prophet. Worth revisiting for further tuning
- Clustering
  - DBSCAN, OPTICS deep dive. Second look at OPTICS from should be considered
  - Ball Tree in SKL used, nearest neighbor was the best choice with good flexibility

Model Performance

- Considerable amount of time spent on the clustering problem
- Evaluating the clustering performance was difficult
- EPS for DBSCAN was tricky to get right for this use case
- Introduced Geohashing, helped dealing with geocoordinates, clustering, analysis
- For time series, work remains but Prophet rocks
- Additional regressors helped modeling and easily integrated, continued tuning could yield a much better results but at a low resolution

Thanks for watching…

- And thank you Romeo, Nikolay, Ilja and the rest of IBM team for a great course!