New York Taxi Demand Prediction

By: Rosamund Lim

Business Problem

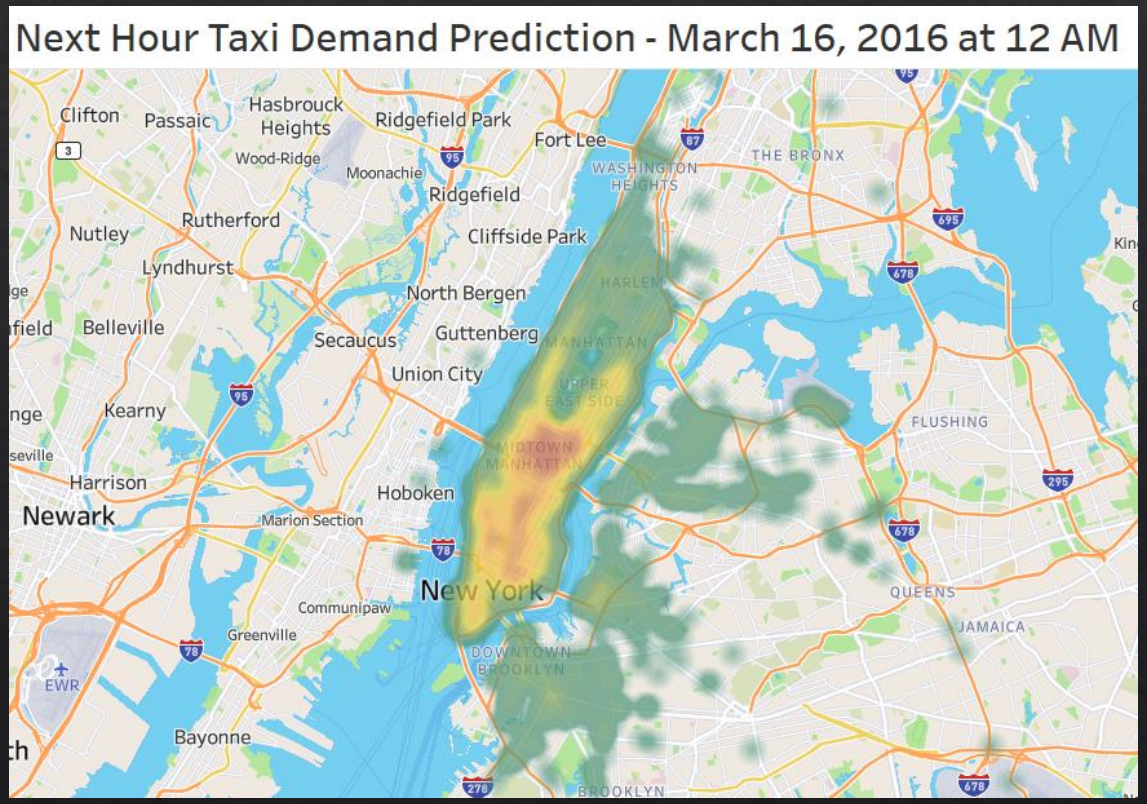
- Given New York Taxi Dataset, show areas where drivers could get a higher chance of getting a ride; resulting in increased earnings
- From Business Problem to Machine Learning Problem
  - Time Series
  - Predict Next Hour’s Pickup Demand for each location

Dataset

- 2016 January to 2016 March
- **~34.5 million** trip records (large dataset taking around 15 sec to read 1/3 of it in Pandas)
- Granularity of each record is YYYY-MM-DD HH:MM:SS

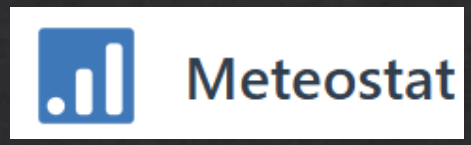
Approach

1. For each unique location, predict the pickup demand (total no. of pickup count) for the next hour, comparing Linear Regression & XGBoost Regression (Poisson)
2. Perform analysis on model: Feature Importance, LIME
3. Create Tableau Visualizations on next hour's predicted demand



Tech Stack

Preprocessing &
Feature Engineering



Modelling

Prediction Analysis
& Visualization

Version Control

Cleaning and Preprocessing Data

Basic Cleaning and Aggregation

- Remove “null island” coordinates (0.0, 0.0) and observations outside of New York
- Down sample `tpep_pickup_datetime` to hourly intervals, aggregated by location
- Down sample precision of `pickup_longitude` & `pickup_latitude` to 3 dp (neighbourhood/street)

Feature Engineering

- `is_weekend` (int): 1 if instance is weekend, 0 otherwise
- `day_of_week` (int): 1 to 7 corresponding with Mon – Sun
- `month` (int): 1 to 3 corresponding with Jan to Mar
- `day_of_month` (int): 1 – 31 corresponding with the day of the month
- `hour_of_day` (int): 0 to 23 corresponding with 00:00 to 23:00
- `temp` (float): hourly temperature (degree Celsius) for New York
- `prcrp` (float): hourly precipitation (mm) for New York
- **`target_pickup_count` (int):** next hour’s pickup count; leading variable
- `pickup_count_1h_ago` (int): pickup count 1 hour ago; lagging variable
- `pickup_count_2h_ago` (int): pickup count 2 hours ago; lagging variable

Temporal

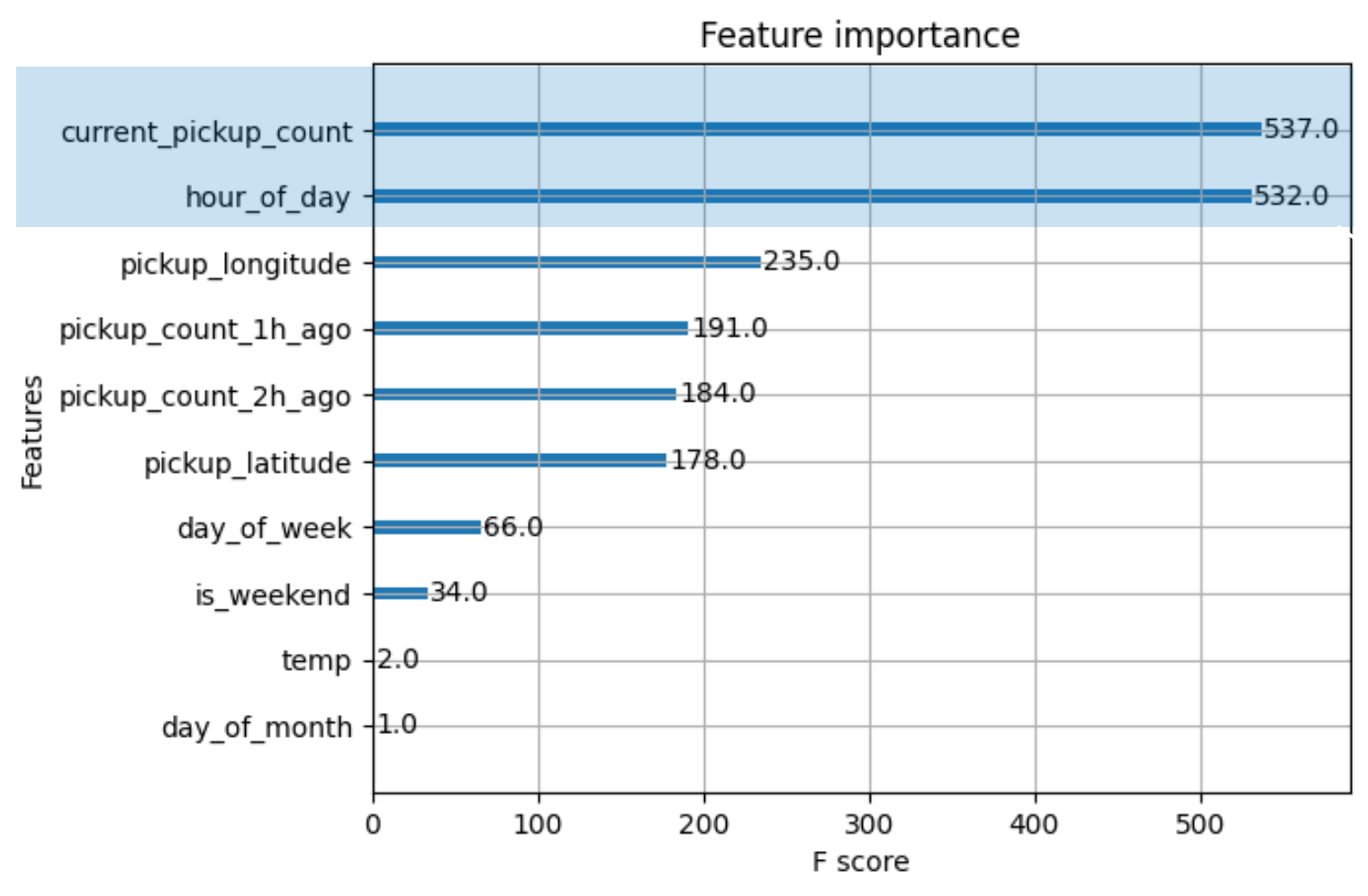
Weather

Leading/Lagging

Modelling

	Linear Regression (Baseline)	XGBoost Poisson Regressor
Params	Default	objective='count:poisson’, n_estimators=280, max_depth=3, learning_rate=0.05, verbosity=2, subsample=0.75, colsample_bytree=0.8
Preprocessing	OHE categorical features	Convert categorical features to `category` dtype
Train Test Split	Train: 1 Jan 2016 – 15 Mar 2016 (~80%) Test: 16 Mar 2016 – 31 Mar 2016 (~20%)	
Performance (Training)	RMSE: 5.00 MAE: 2.66	RMSE: 4.78 MAE: 2.45
Performance (Validation)	RMSE: 4.99 MAE: 2.67	RMSE: 4.76 MAE: 2.45

Model Interpretation (XGBoost Feature Importance)

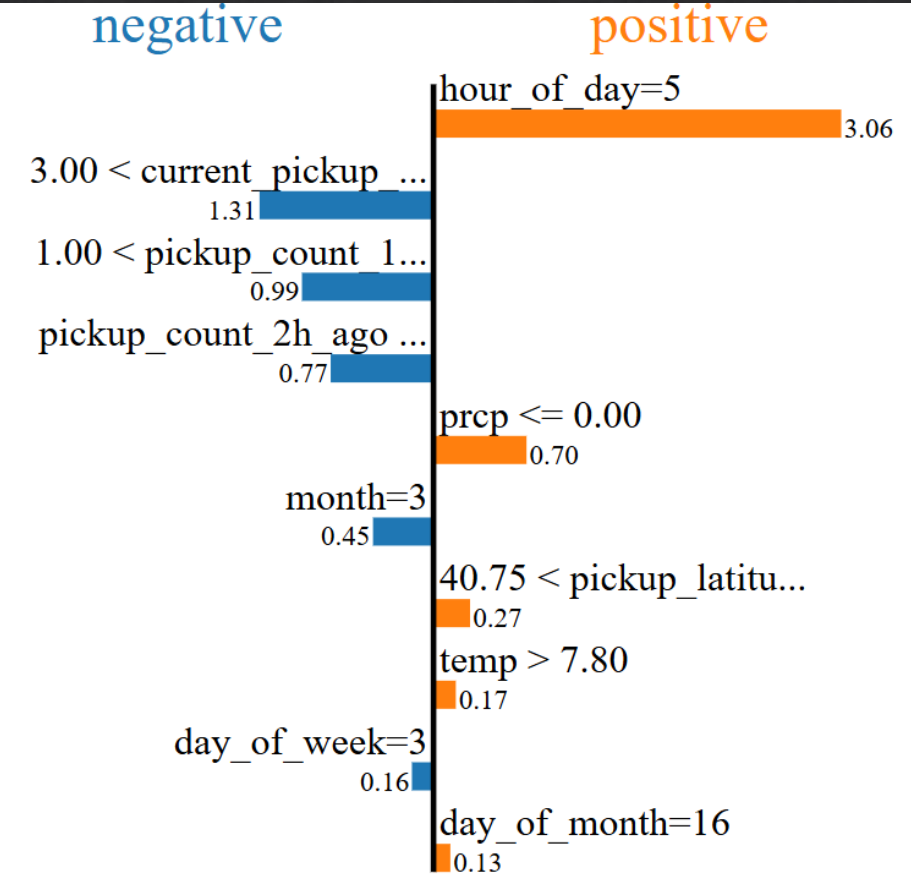


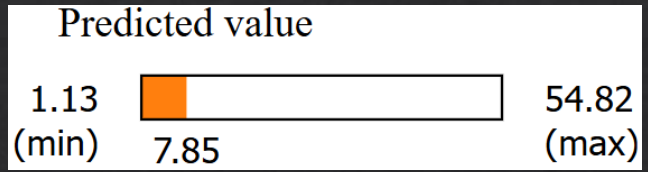
F-score is derived from the number of times a feature is used to split the data across all trees within the model.

Most useful features for making splits across the trees

Model Interpretation (LIME)

Observation 6 – Latitude: 40.765 , Longitude: -73.977 , 3/16/2016 5:00:00 AM



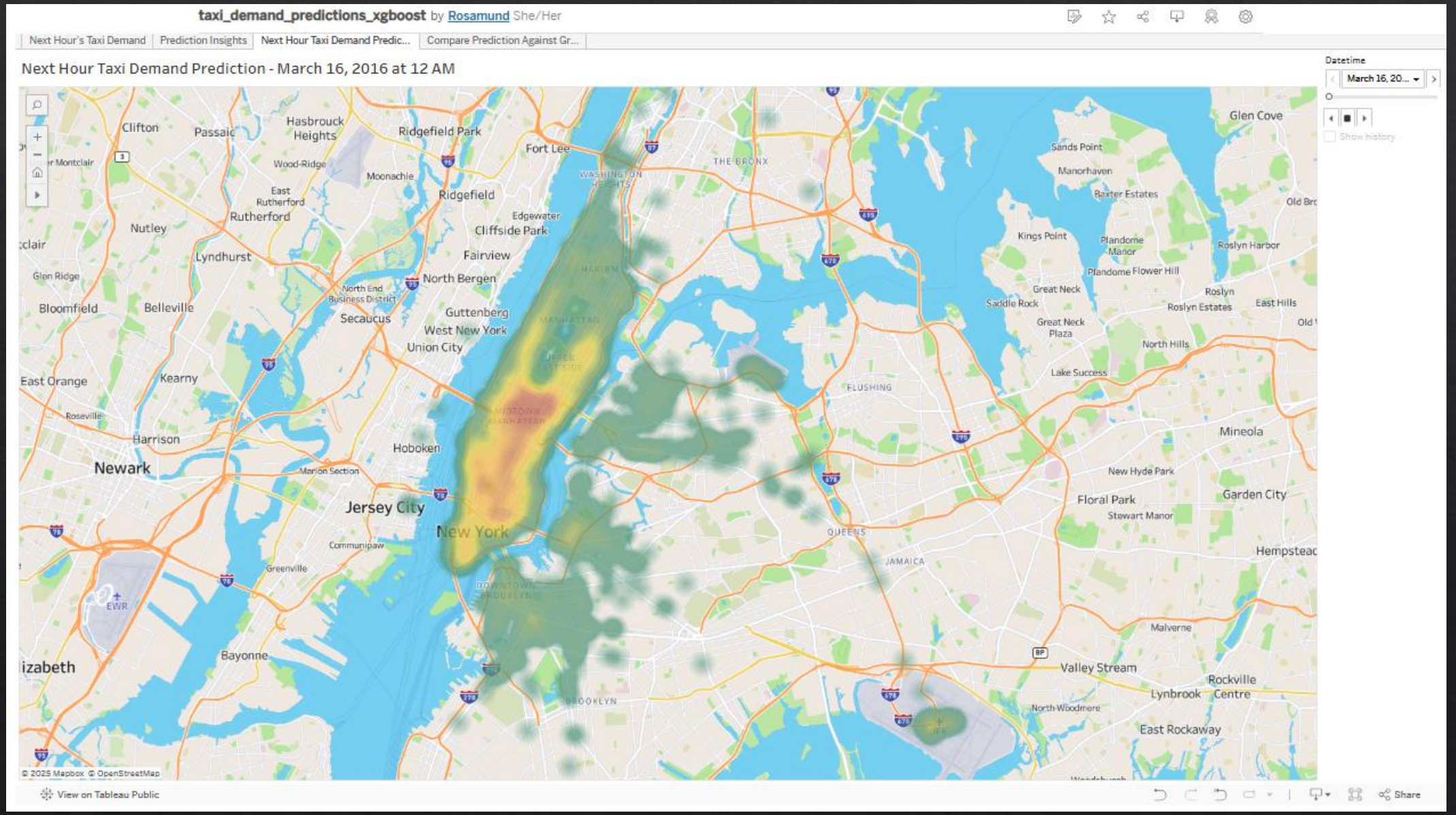


Feature	Value
hour_of_day=5	True
current_pickup_count	5.00
pickup_count_1h_ago	2.00
pickup_count_2h_ago	1.00
prcp	0.00
month=3	True
pickup_latitude	40.77
temp	10.00
day_of_week=3	True
day_of_month=16	True

LIME
(Local Interpretable Model Agnostic Explanations)

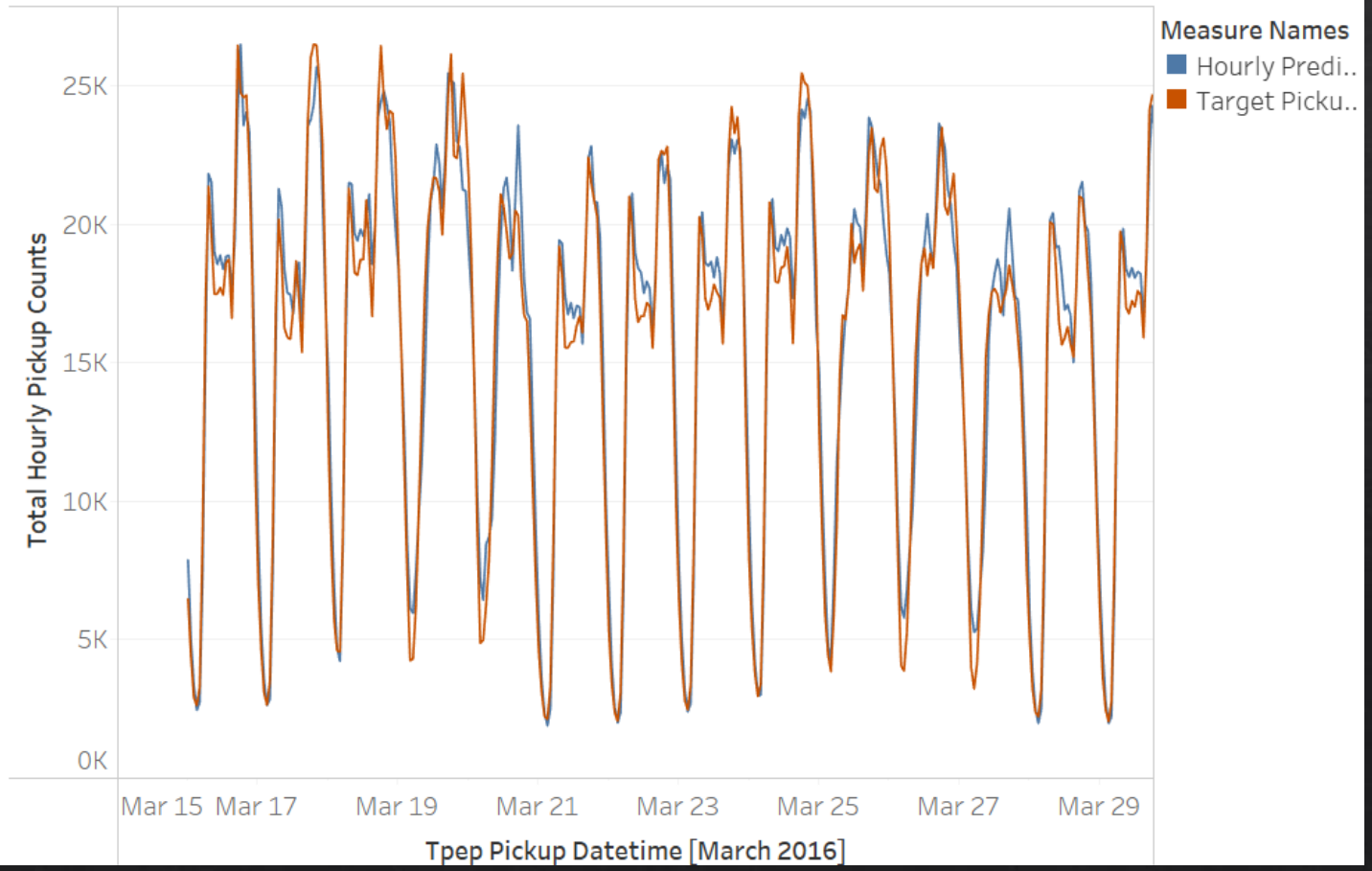
Modifies a single data sample by tweaking the feature values and observes the resulting impact on the output

Serving predictions



Serving predictions

Hourly Predictions represent the predicted values for next hour demand for each location, hourly timestamp. Target Pickup Count represents the actual (ground truth) values for next hour demand for each location, hourly timestamp. My XGBoost model was able to capture the hourly periodic trends for example, the spikes and troughs at different timings of the day. Spikes tend to be between 5 - 7pm and troughs between 2-4 am



Future work

Hyperparameter Tuning
- Bayesian Optimisation

Deployment
- Containerisation with Docker
- Fast API to serve predictions in real-time