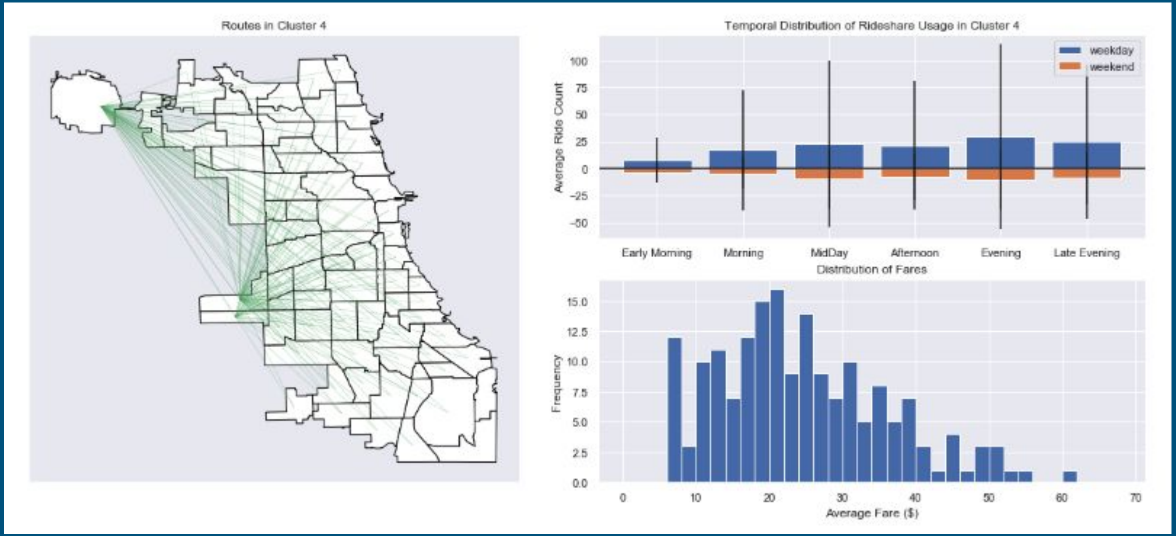
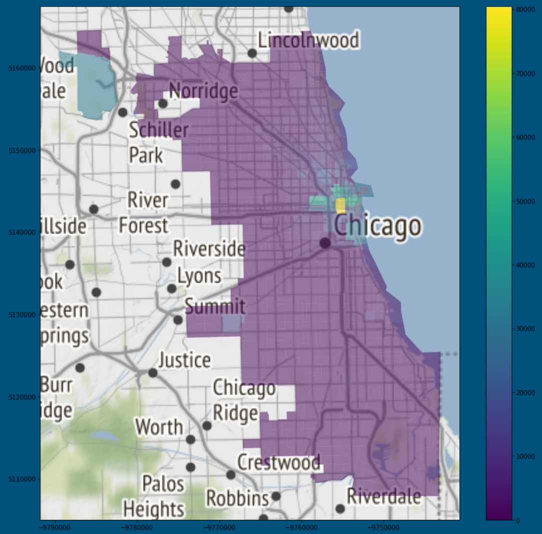




Machine Learning with the Chicago Ridesharing Trips Data Set

Lisa Taylor
Completed as Capstone Assignment 1
Springboard Data Science Bootcamp

Chicago Rideshare Dataset

Chicago requires “Transportation Network Providers” (Lyft, Uber, etc.) to report basic rideshare information

Trips Dataset:

- > 70 million individual trips.
- November 2018 through March 2019
- Anonymized: Locations generalized to the nearest census tract. No specific driver/rider info.
- Fields: start/end census tract, start/end XY coords, start/end datetime, fare, distance, duration, tip, whether pooling authorized, number of trips in pool.

Business Case

Why does Chicago collect this data?

- Gain insight on travel patterns within the City.
- Target improvements in public transit offerings
- Improve infrastructure planning
- Ensure that access to rideshare is provided equitably
- Quantify costs incurred by the City by allowing rideshare services

Other Potential Data Users:

- Rideshare Drivers: Where can I find rides?
- Environmental agencies/citizen groups: How much ridesharing occurs? What are air quality/traffic impacts?

Big Questions

- How do the residents of Chicago use Ridesharing?
- What usage patterns can we see?
- Can we predict usage?

Process

- Cleaning, Wrangling
- Data Enhancement
- EDA
- Statistical Tests
- Machine Learning
    - Clustering
    - Demand Prediction (Regression)

Cleaning and Wrangling, Enhancement

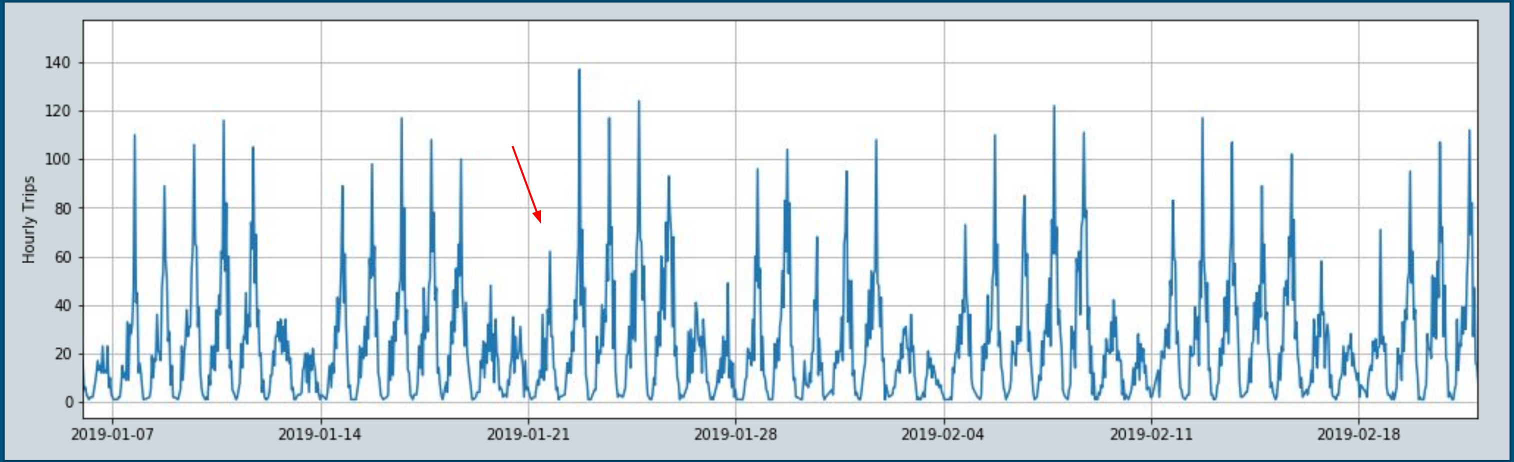
- Download and import
  - 12 GB file
  - Use 5% random subsample for evaluation
- Remove unusable records (zero fare, no census tract)
- Aggregate by hour
  - Datetime manipulation
  - Calculate ride counts, average trip distance, duration, fare by census tract, hour
- Integrate weather data
  - Hourly precipitation, temperature, wind speed from Midway AIrport (NOAA)
  - Merge on ride date time after synchronizing to nearest hour.

More Enhancement

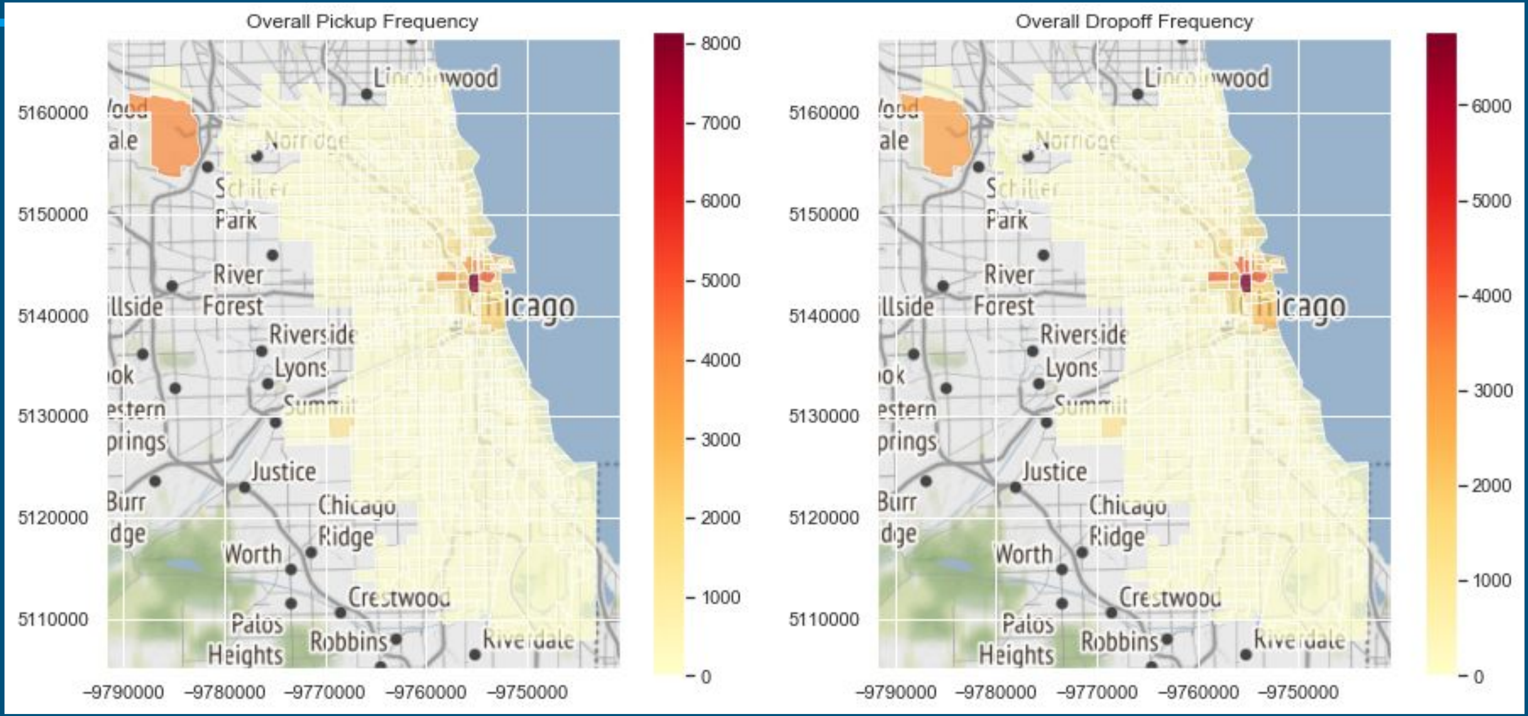
- Census geodata
  - Import census tract geojson to geopandas geodataframe object
  - Enables mapping of data by census tract and geospatial operations on the data (distance, area, etc.)
  - Join rideshare data (multiindex on census tract, time) to census geodataframe with pandas merge
- Census population data
  - Use Census API to acquire population and income data, merge on ride pickup census tract
- Derived fields (per tract):
  - Distance from downtown
  - Direction relative to downtown
  - Population density
  - Median Income

EDA Findings: Temporal Pattern

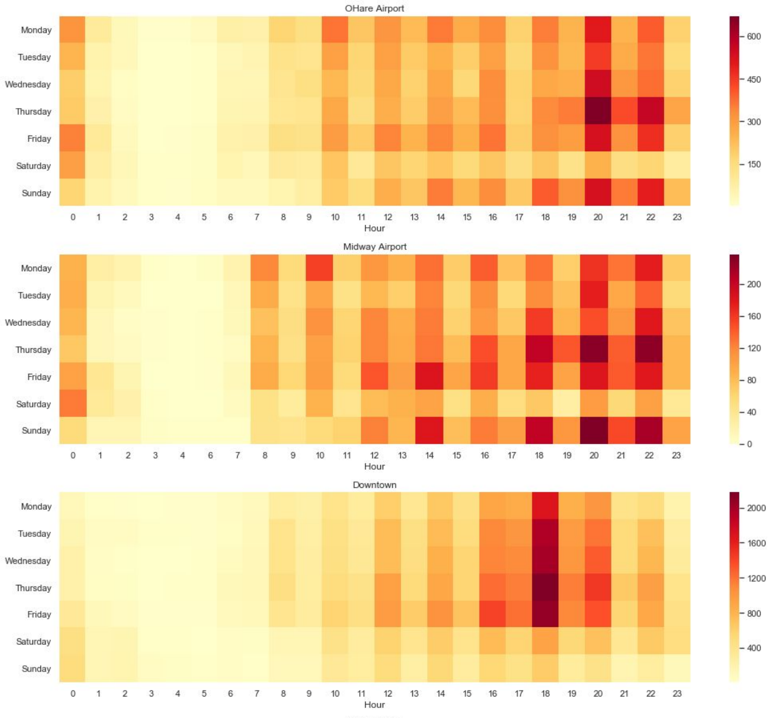
- Usage is cyclical
- Hourly pattern superimposed on weekly pattern
- Holiday effects

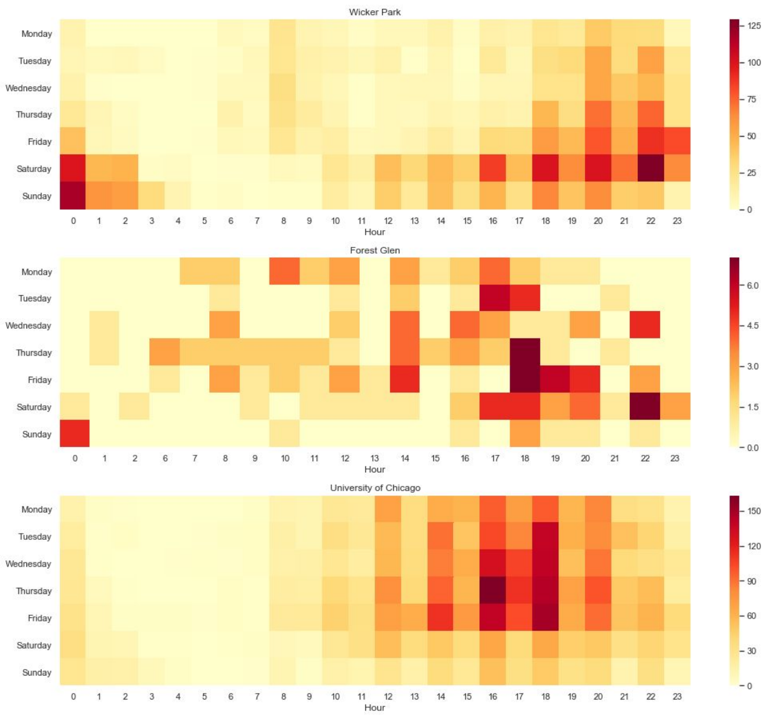


EDA Findings: Spatial Pattern



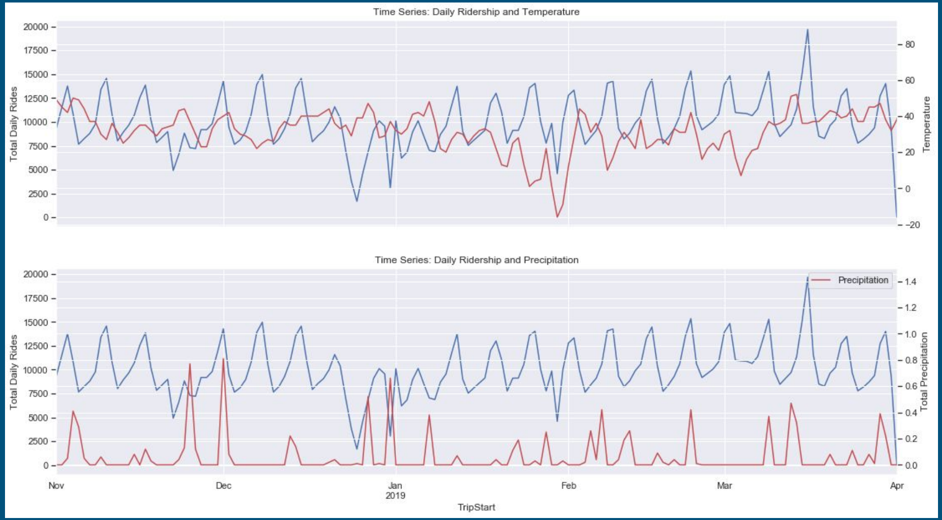
EDA Findings: Spatiotemporal Patterns

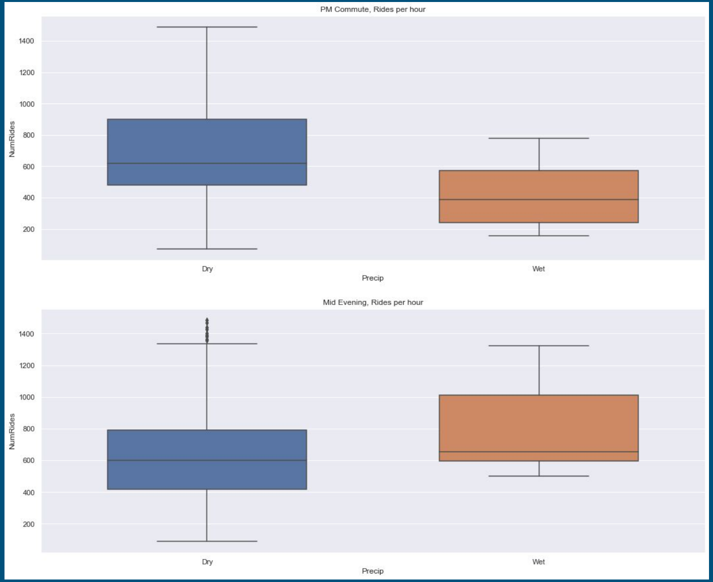




EDA Findings: Weather Effects?

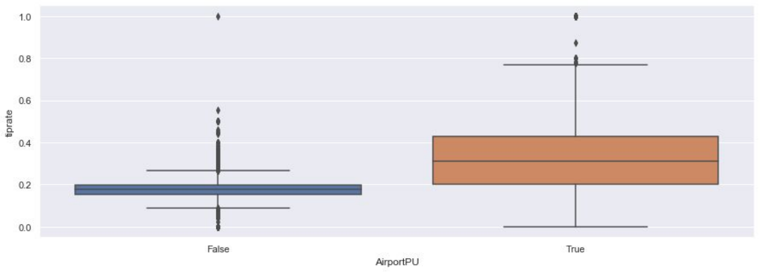
- Not apparent at daily level
- Focused effect within specific time windows

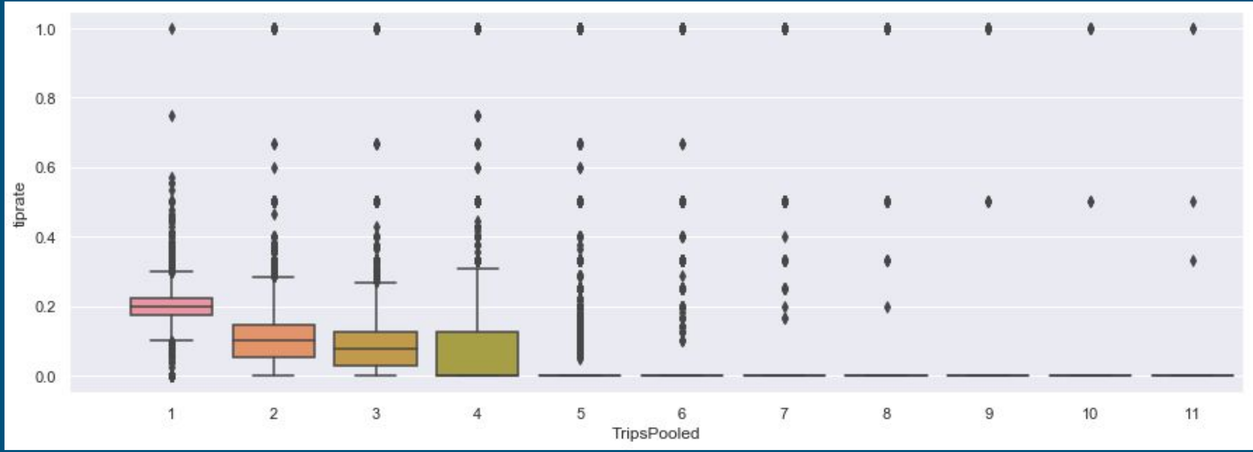




EDA Findings: Tipping

- Overall 20% tipping rate
- Statistically significant increase in probability of tipping for airport pickups.
- Decrease in tipping probability for pooled rides





Machine Learning

Business Goal	ML Application
Target improvements in public transit offerings	Clustering analysis to uncover common utilization patterns
Improve infrastructure planning	Regression to predict rideshare utilization by location and time
Ensure that access to rideshare is provided equitably	Data mining to determine relative effect of income on usage *

* Not part of this evaluation

Clustering Analysis

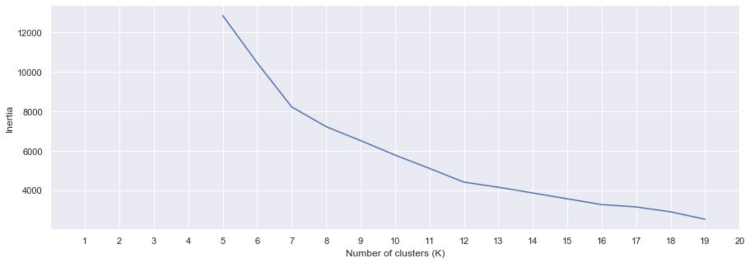
- Customer Segmentation: cluster by behavior
- Data structure:
  - Rows indexed by Route (MultiIndex, unique pairing of Pickup and Dropoff location)
  - Columns of ride counts for each of the following periods:
    - Weekday morning
    - Weekday mid-day
    - …
    - Weekend evening
    - Weekend late evening
    - Average Fare
    - Is Airport (PU or DO)
- All fields scaled

Model Selection

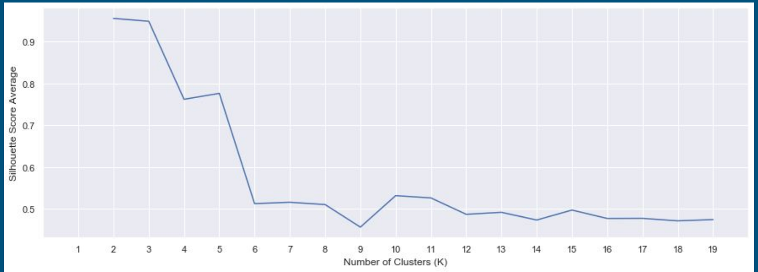
- KMeans ★
- DBSCAN
- Agglomerative Clustering
- Spectral Clustering

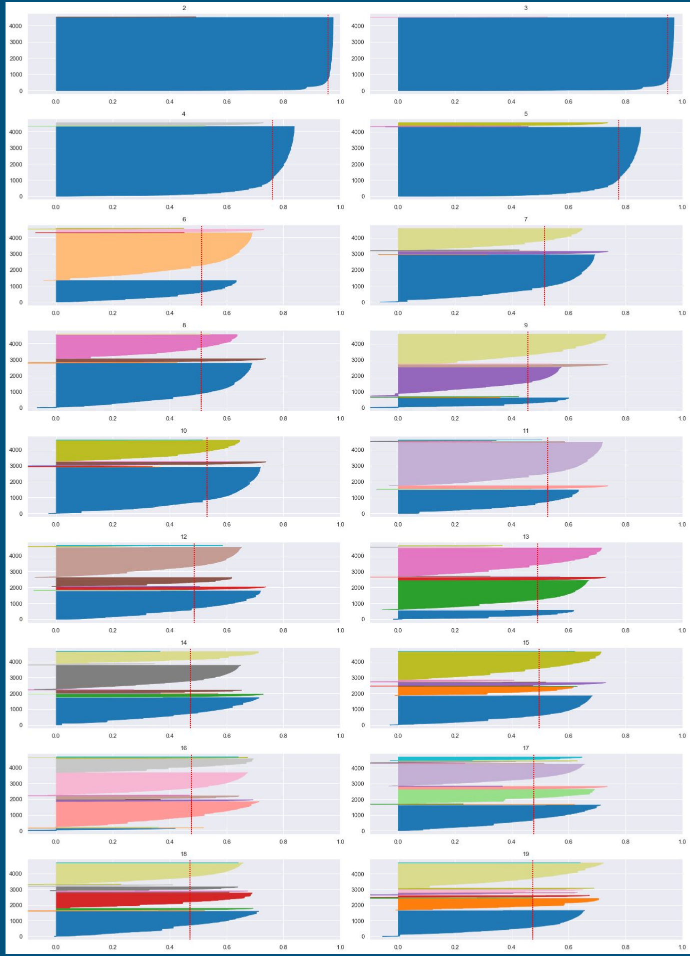
KMeans: Selection of K

- Elbow method

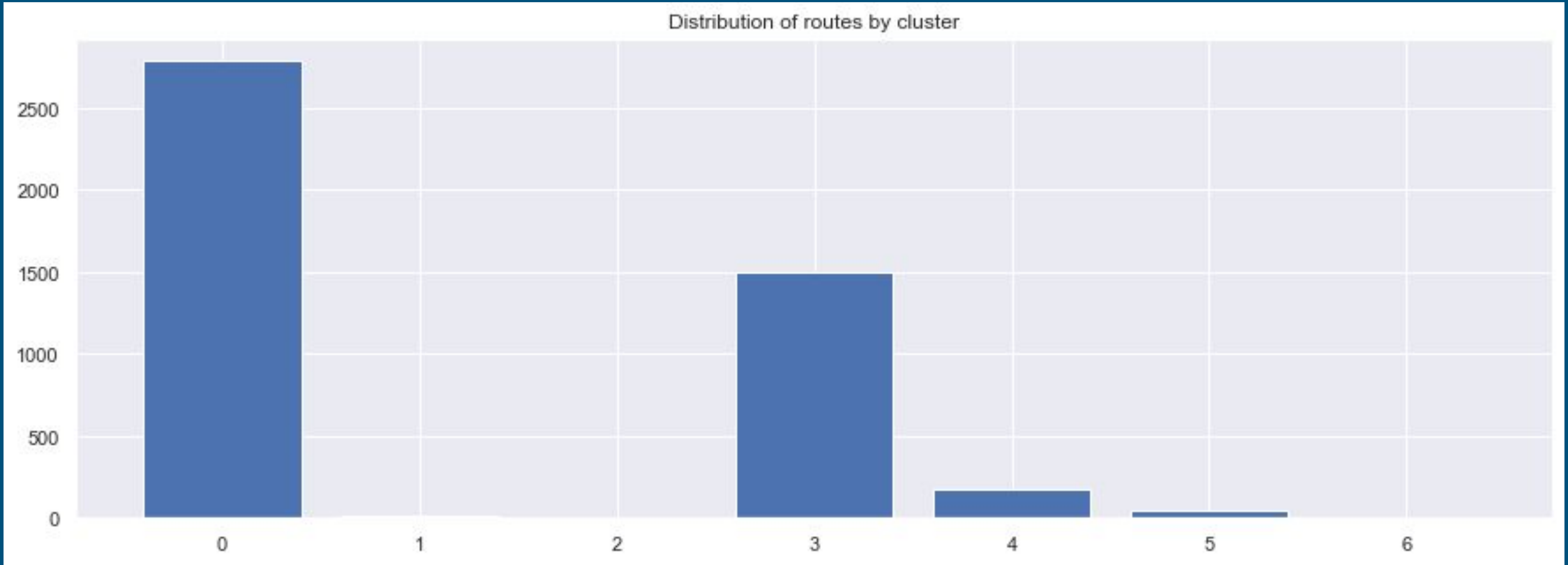


- Silhouette



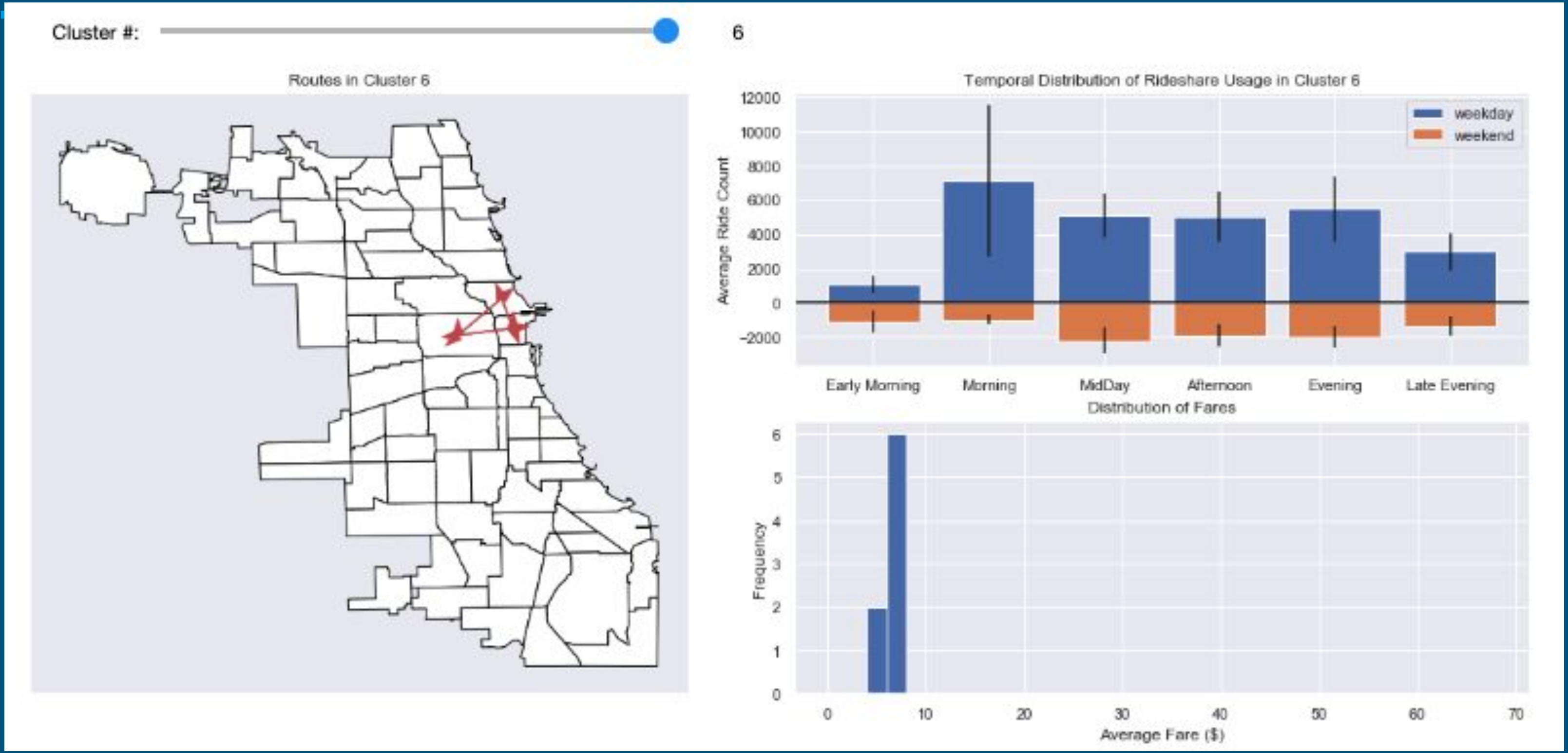


KMeans, K=7

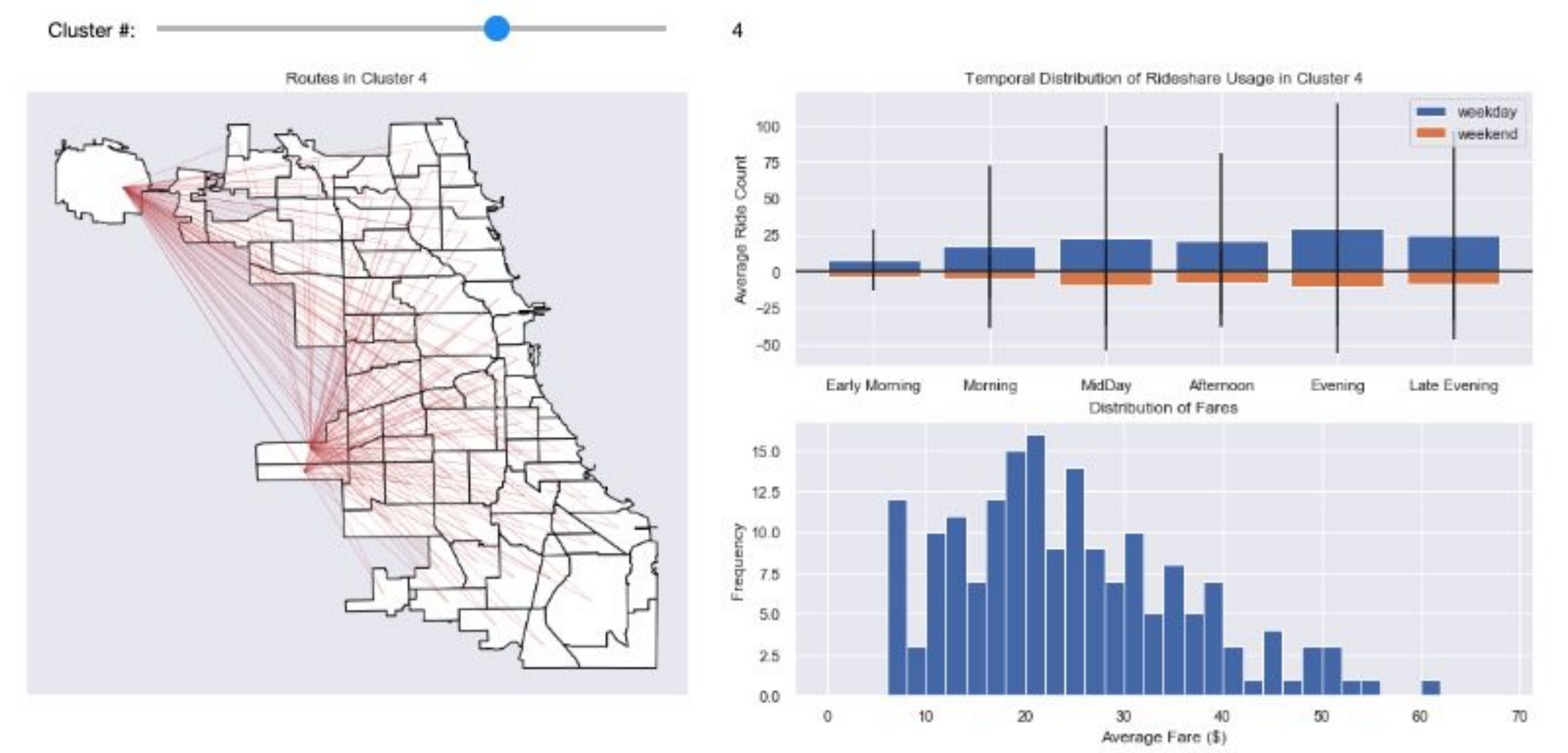


Distribution of routes by cluster
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6

Visualizing Clusters - City Center



Visualizing Clusters - Airport



Clustering - Results

Found clusters representing distinct rideshare patterns:

- Commuting near downtown: morning vs evening patterns
- Rides within the city center
- Airport Pickups
- Longer commute rides
- Low frequency routes with higher fares: commute pattern, random times

Higher K → more difficult to interpret

Using distance, ride time instead of fare → similar outcomes

Data structure has strong influence on the clustering model outcome.

Clustering - Future Work

- Weighting parameters
- Finer spatial scale
- Incorporate demographics

Demand Prediction by Regression

- Goal: predict ride counts per pickup location given DOW, hour
- EDA: cyclical, focus on downtown, airport
- Method: Assume future rideshare usage will be consistent with historical pattern.

Dataset

- Group on pickup tract, ride start time (rounded)
- Average fare, tip, distance per grouping
- Count rides per grouping (~5% of total)
- Derived fields: Hour, DOW, IsHoliday, IsAirport
- Enriched fields: distance to downtown, income, population density, temperature, precipitation

Linear Regression - Statsmodels

Advantage: Explainable model, infer parameter effects

Potential Issues:

- Correlated features (distance, time, fare)
- Is underlying model linear?
- Y (Number of Rides) not normally distributed.

OLS Regression Results

```
results=smf.ols('NumRides ~ DistToDowntown + TripTotal + C(Precip)
        + C(IsAirportPU) + C(DayPeriod) + C(IsWeekday)'
        ,data=agg_hourly_all).fit()
```

Dep. Variable:	NumRides	**R-squared:**	0.149
Model:	OLS	**Adj. R-squared:**	0.149
Method:	Least Squares	**F-statistic:**	9429.
Date:	Fri, 25 Oct 2019	**Prob (F-statistic):**	0.00
Time:	14:30:18	**Log-Likelihood:**	-1.5281e+06
No. Observations:	537380	**AIC:**	3.056e+06
Df Residuals:	537369	**BIC:**	3.056e+06
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.5931	0.022	210.937	0.000	4.550	4.636
C(Precip)[T.Wet]	0.1839	0.031	6.003	0.000	0.124	0.244
C(IsAirportPU)[T.1]	9.2190	0.060	153.052	0.000	9.101	9.337
C(DayPeriod)[T.morning]	0.3546	0.020	17.369	0.000	0.315	0.395
C(DayPeriod)[T.midday]	0.4401	0.021	21.235	0.000	0.399	0.481
C(DayPeriod)[T.afternoon]	0.7442	0.022	34.596	0.000	0.702	0.786
C(DayPeriod)[T.evening]	1.6513	0.021	79.566	0.000	1.611	1.692
C(DayPeriod)[T.lateevening]	0.9898	0.022	44.611	0.000	0.946	1.033
C(IsWeekday)[T.1]	-0.2841	0.012	-22.974	0.000	-0.308	-0.260
DistToDowntown	-0.3834	0.001	-280.178	0.000	-0.386	-0.381
TripTotal	0.0166	0.001	18.116	0.000	0.015	0.018

Non-Parametric Regression

Does not assume functional form for Y=f(X)

Candidates:

- KNN Regression
- RBF-kernel SVM
- Tree-Based Models

Baseline Models

NumRides = F(DistToDowntown,Hour,IsWeekday,TripMiles,Precip)

TrainTestSplit: (Time series) Before: 80%, After 20%

Remove airport pickups and holidays

Test Results:

Baseline Model	**R-squared**	**RMSE**
Random Forest	0.81	1.9
Gradient Boosting	0.68	2.5
Bagging	0.81	1.9

Data Enhancement

- New Fields
  - Bearing - direction relative to downtown
  - IsWeekday —>DOW
  - TripTotal —> Mean Fare
  - Add demographics
  - Remove precip
- Fit RandomForestRegressor

NumRides = F(DistToDowntown, Bearing, Hour, DOW, MeanFare, PopDensity)

Test Results: R-Squared: 0.84 (was 0.82), RMSE: 1.8 (was 1.9)

Hyperparameter Tuning

Cross-Validation with Time Series Split applied to training data, 4 partitions:

- Number of Estimators

n_estimators	10	20	30	50
R-squared	0.825	0.831	0.832	**0.834**

- Grid Search: Parameter Selection Method, Tree Depth

For max_depth=16:

max_features	Log2	Square Root	None (Bagging)
R-squared	0.830	0.829	**0.839**

Final Model

Hyperparameters:

N_estimators = 50

Max_depth = 16

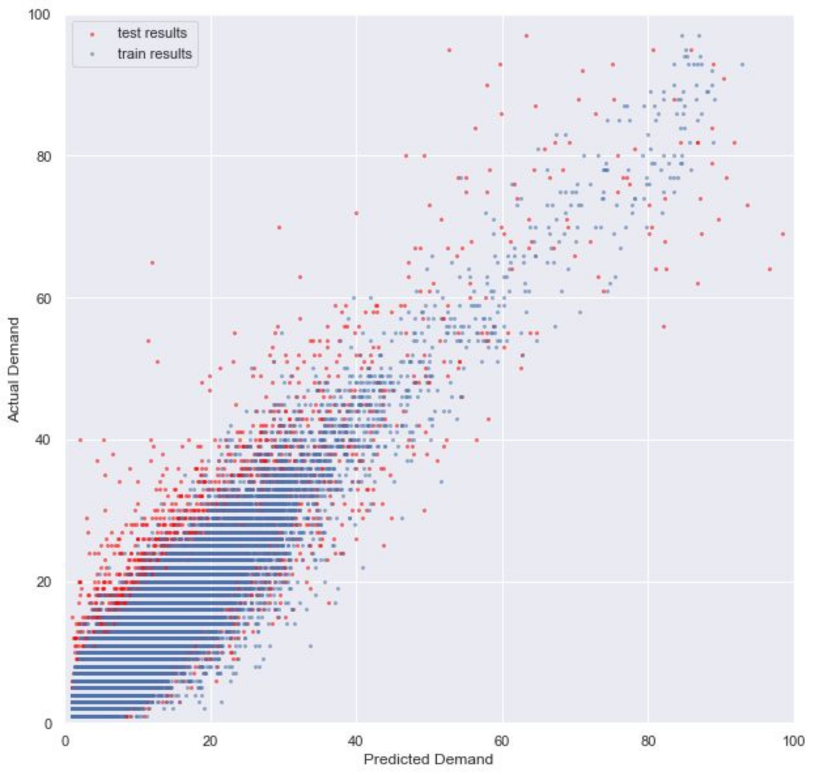
Feature_Selection = None

Fit to training set and score on test:

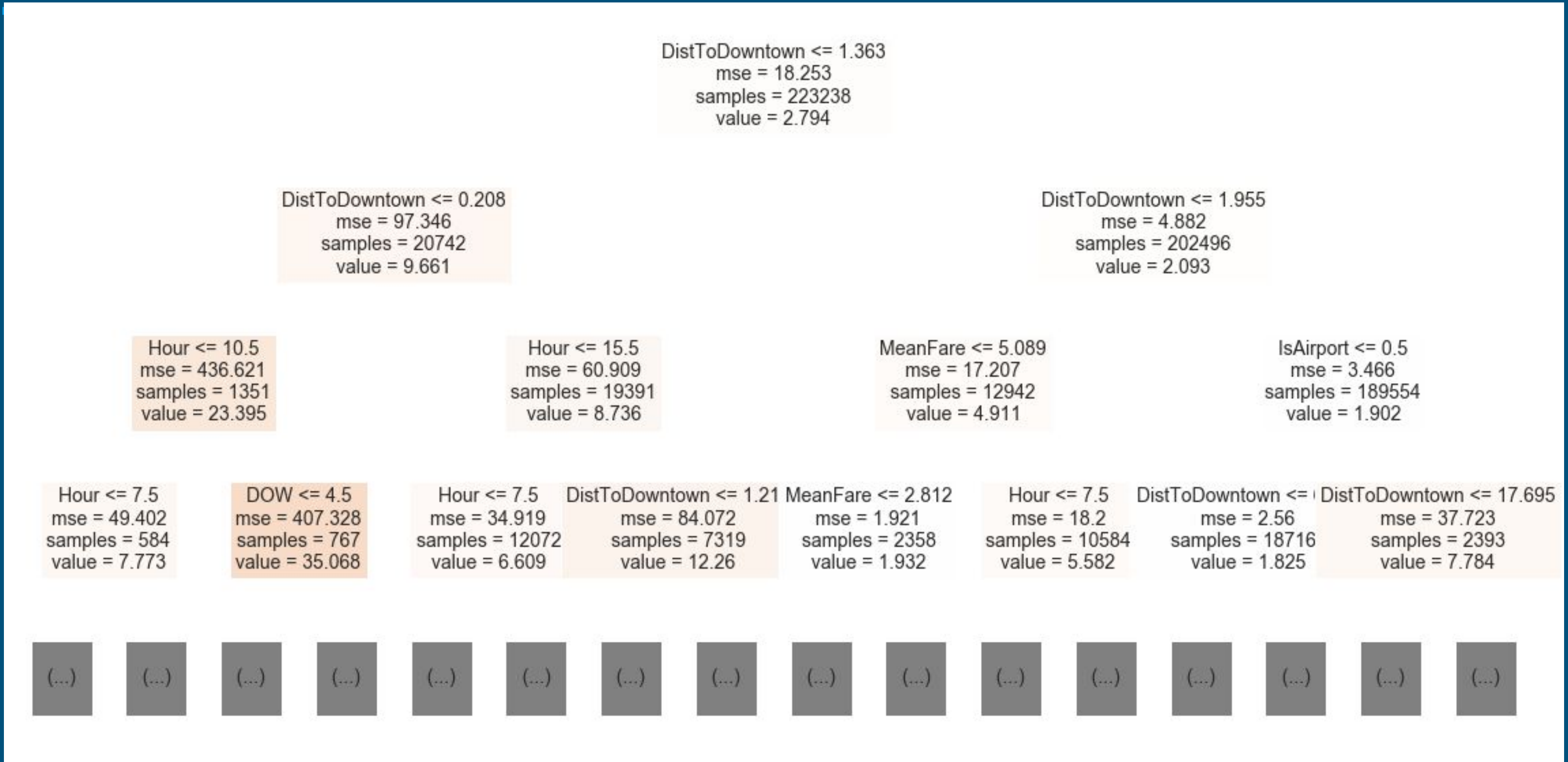
R-squared: 0.85

RMSE: 1.69

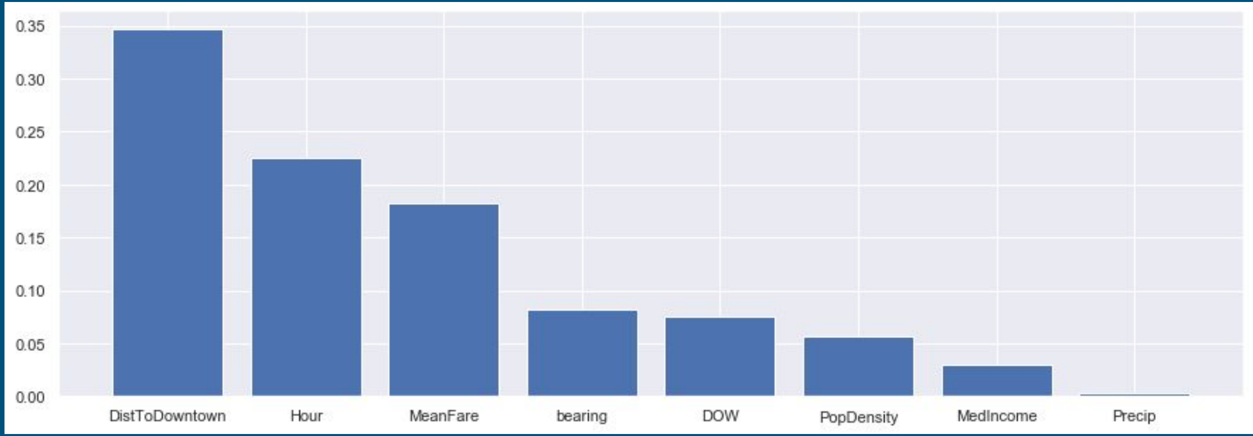
Train Vs Test



Plot Tree


DistToDowntown <= 1.363
mse = 18.253
samples = 223238
value = 2.794
DistToDowntown <= 0.208
mse = 97.346
samples = 20742
value = 9.661
DistToDowntown <= 1.955
mse = 4.882
samples = 202496
value = 2.093
Hour <= 10.5
mse = 436.621
samples = 1351
value = 23.395
Hour <= 15.5
mse = 60.909
samples = 19391
value = 8.736
MeanFare <= 5.089
mse = 17.207
samples = 12942
value = 4.911
IsAirport <= 0.5
mse = 3.466
samples = 189554
value = 1.902
Hour <= 7.5
mse = 49.402
samples = 584
value = 7.773
DOW <= 4.5
mse = 407.328
samples = 767
value = 35.068
Hour <= 7.5
mse = 34.919
samples = 12072
value = 6.609
DistToDowntown <= 1.21
mse = 84.072
samples = 7319
value = 12.26
MeanFare <= 2.812
mse = 1.921
samples = 2358
value = 1.932
Hour <= 7.5
mse = 18.2
samples = 10584
value = 5.582
DistToDowntown <=
mse = 2.56
samples = 18716
value = 1.825
DistToDowntown <= 17.695
mse = 37.723
samples = 2393
value = 7.784
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)

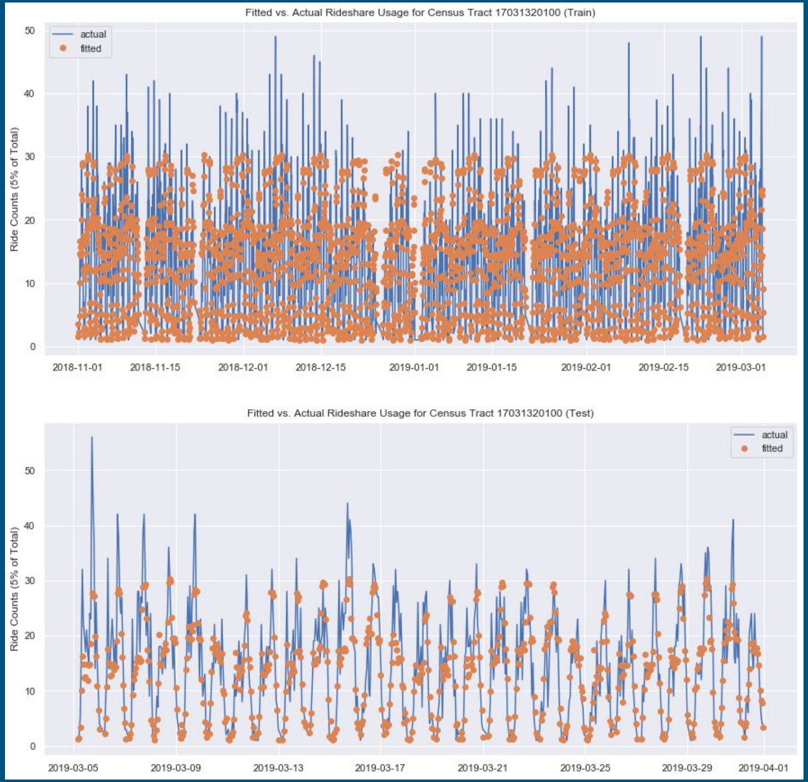
Feature Importances



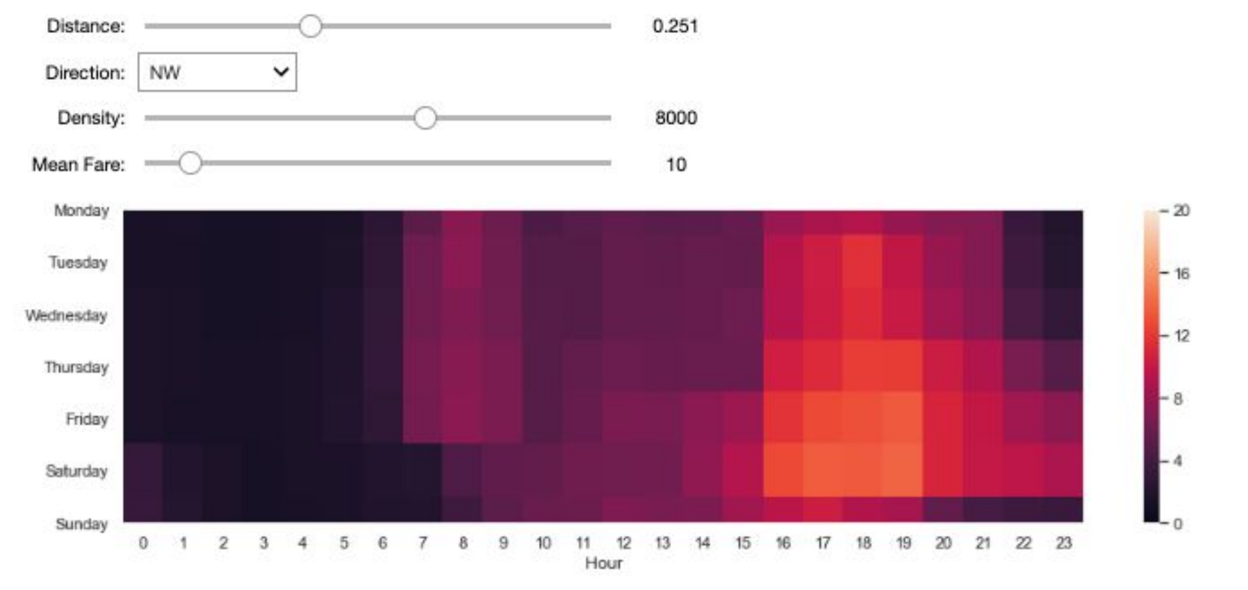
Example of Results - Single Census Tract

Test

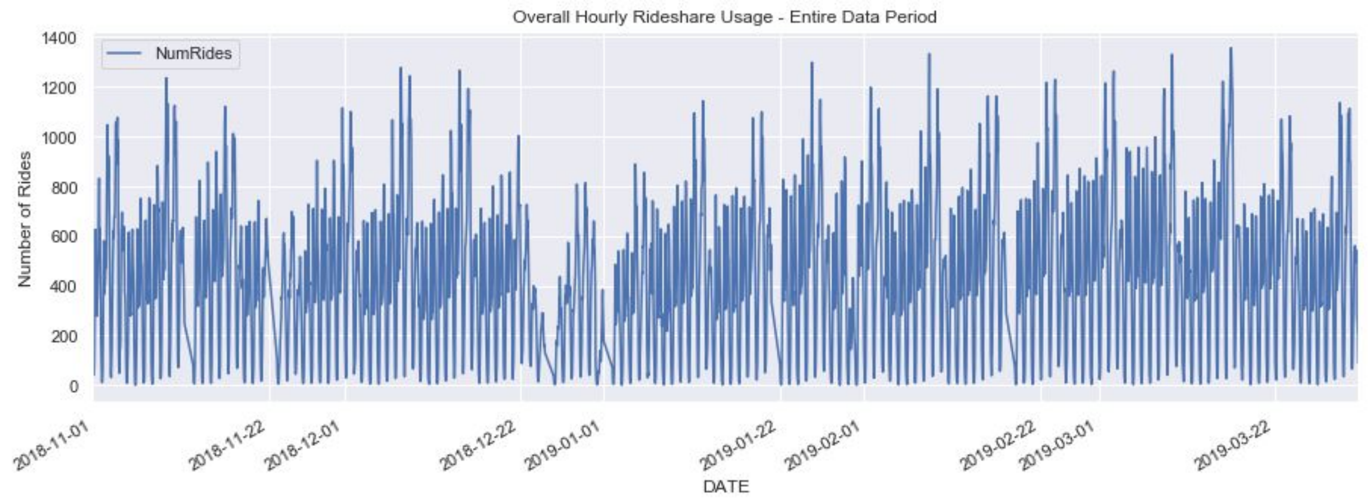
Train



Visualize Results



Why does this work?



Why was random forest/bagging successful at fitting a spatiotemporal prediction model?

- Time series stationary (no trends) and very repeatable.
- Spatial variables explained enough spatial variation in usage.
- Same time period resampled each week resulting in multiple measurements of dependent variable
- Model predicts future behavior from an average of past behavior, with some randomization due to the bootstrapping performed by the algorithms.

Future Work - Usage Prediction

- Use Full Trips Dataset
- Spatiotemporal Visualizations
- Demographic Effects
- Other City Features
- ARIMA model for each census tract

Conclusion

- Developed rideshare customer segmentation and usage prediction tools
- Built dashboards to enable inspection of results and prediction
- Demonstrated methods for gaining insight from rideshare datasets