Kaggle BikeSharing Competion

Chris, Rahul, Raja

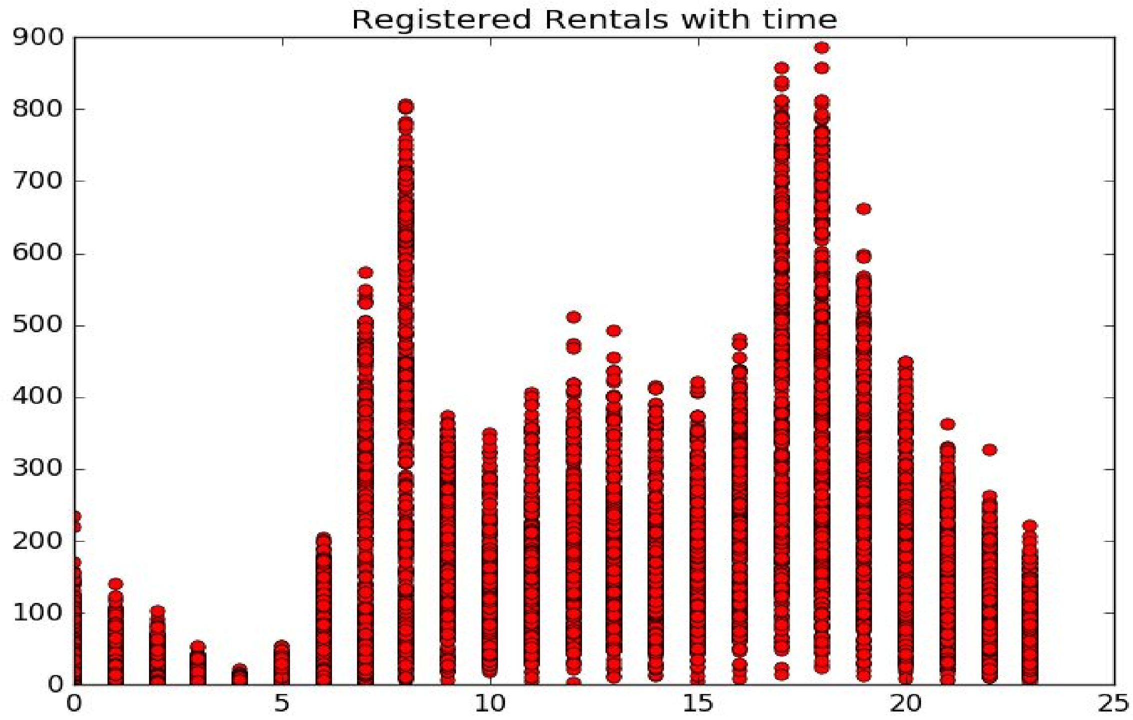
Overview

- Predict bike rental demand in Capital Bikesharing Program in Washington D.C.
- Use historical usage pattern and Weather data to predict future bike rental demand.
- Two kinds of users. Registered and Casual renters.
- Rental kiosks throughout the city. Rent at one location and return bike in a different location.

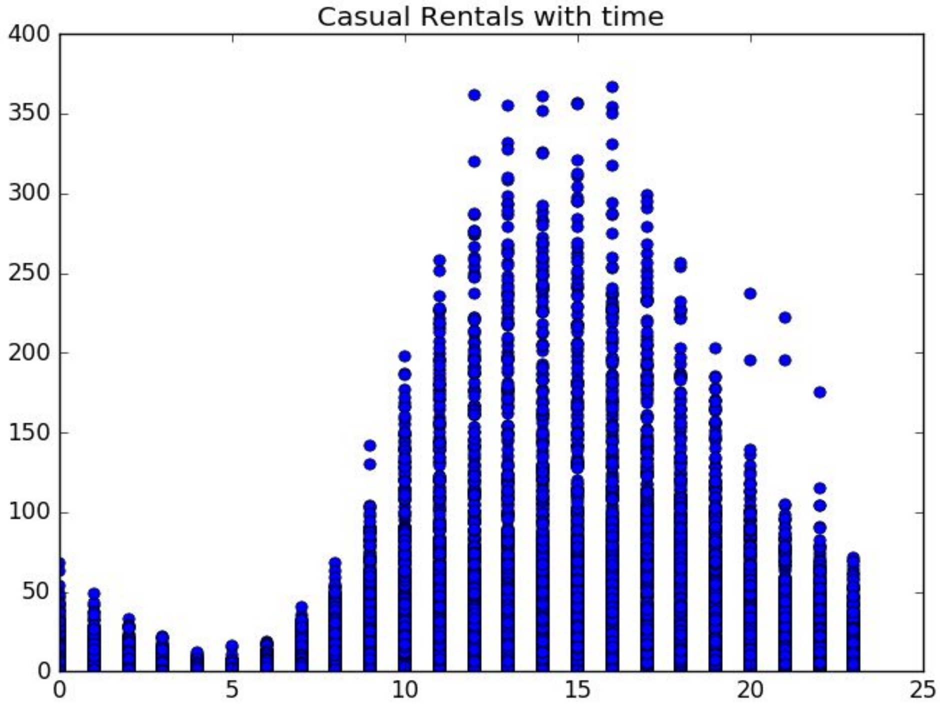
Training Data

Datetime	time of the day
Season	spring, summer, fall, winter
Holiday	neither weekend nor workingday
Workingday	1/0
Weather	clear, misty, cloudy, light rain/snow, heavy ...
Atemp	normalized feels like temperature
Temp	normalized temperature
Humidity	relative humidity
Windspeed	wind speed
Count = registered + casual	total rentals is sum of reg, non reg rentals

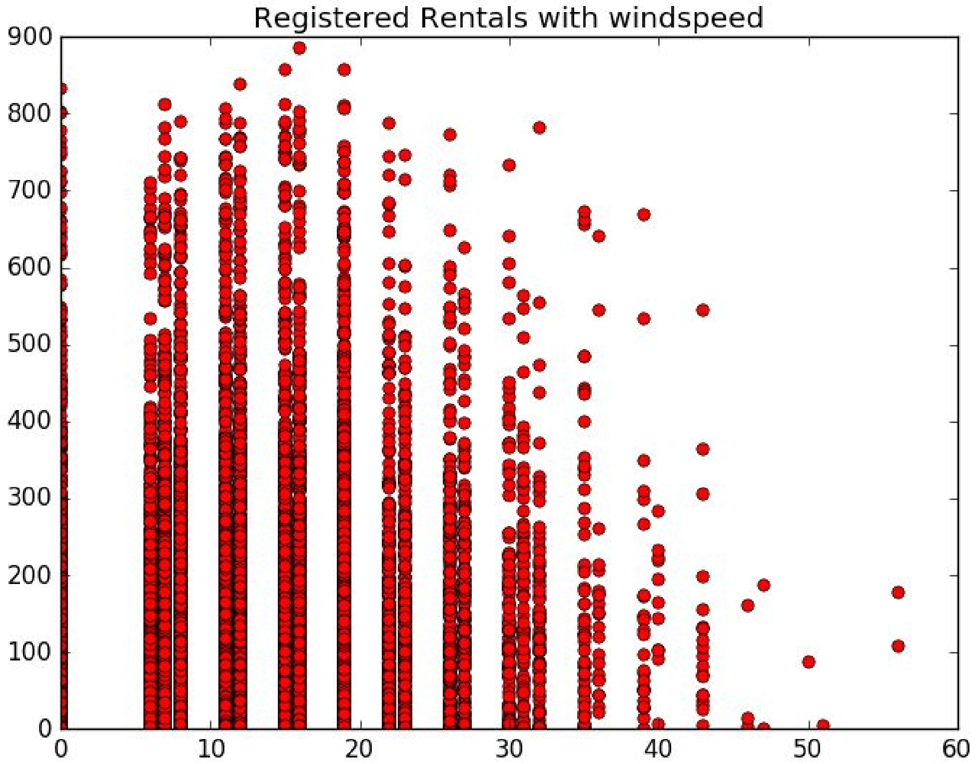
Data Visualization: Registered rentals peak during commute times.



Data Visualization: Casual rentals steady in the afternoon

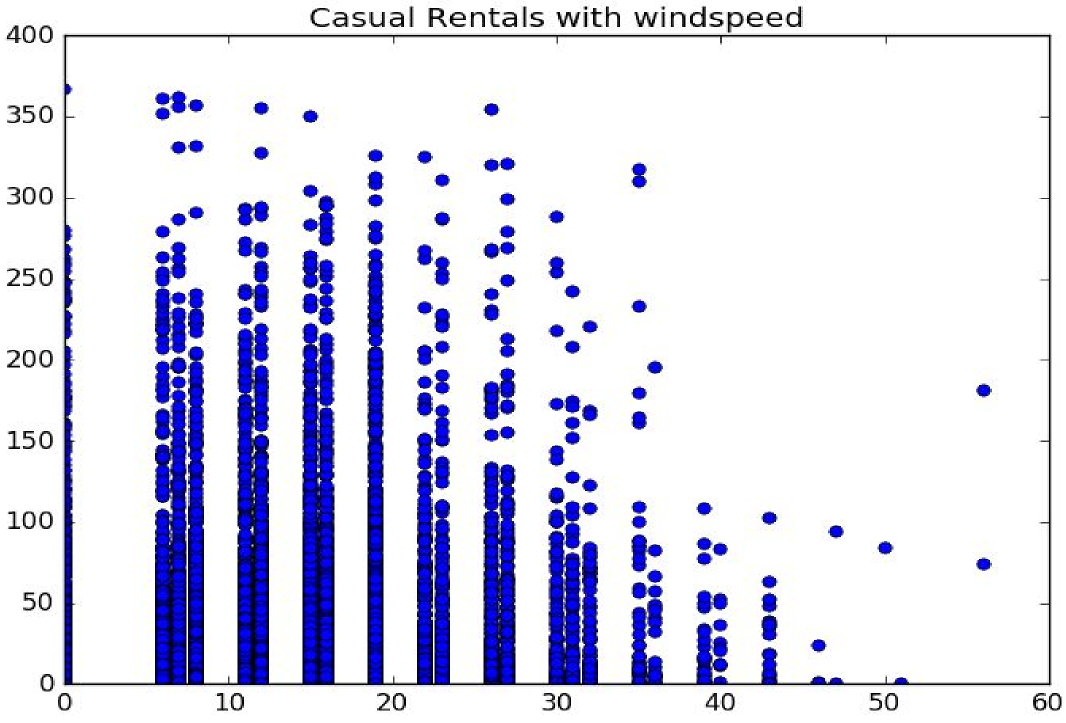


Data Visualization: Registered rentals decrease as windspeed Increases

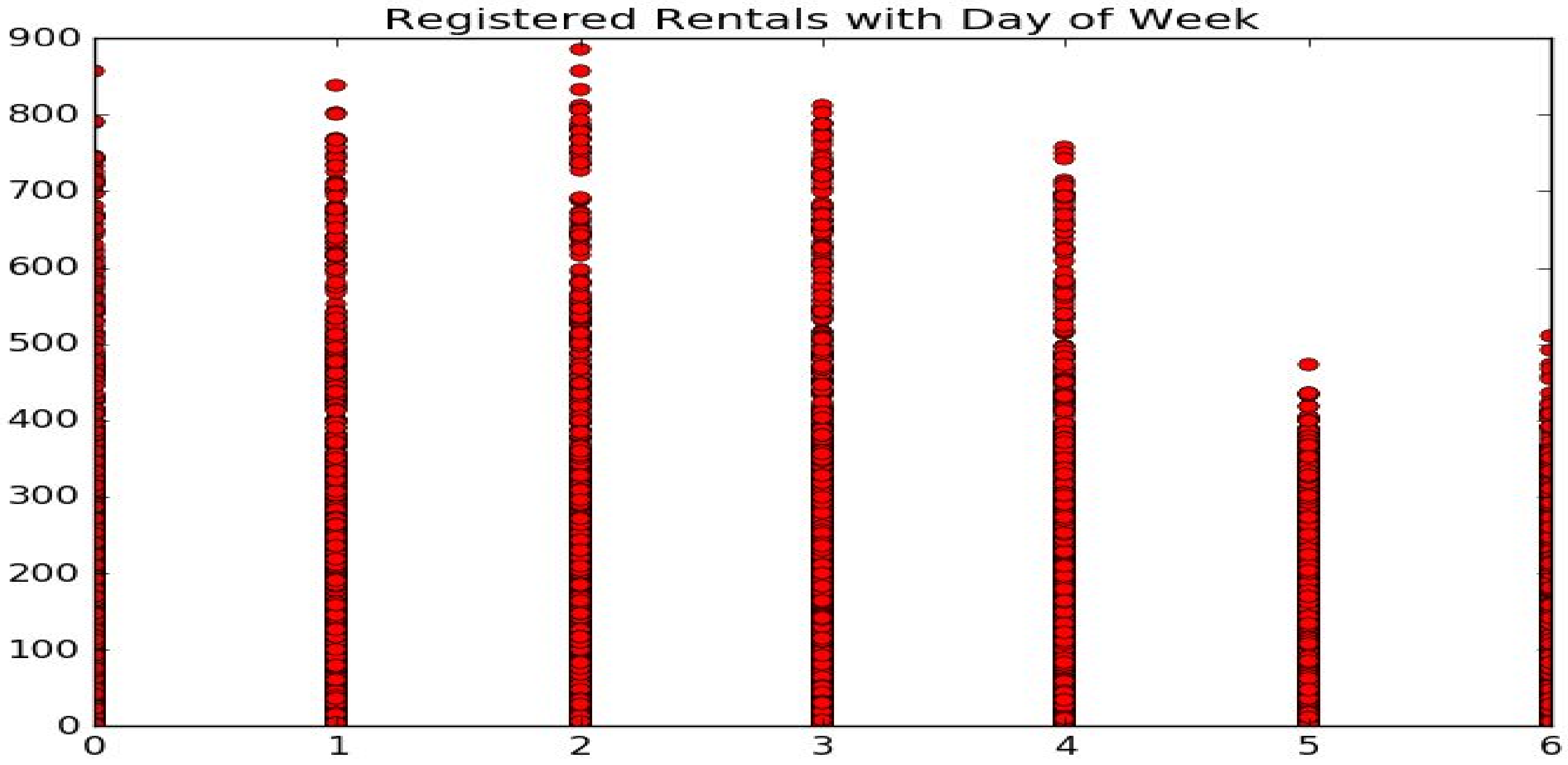


Data Visualization

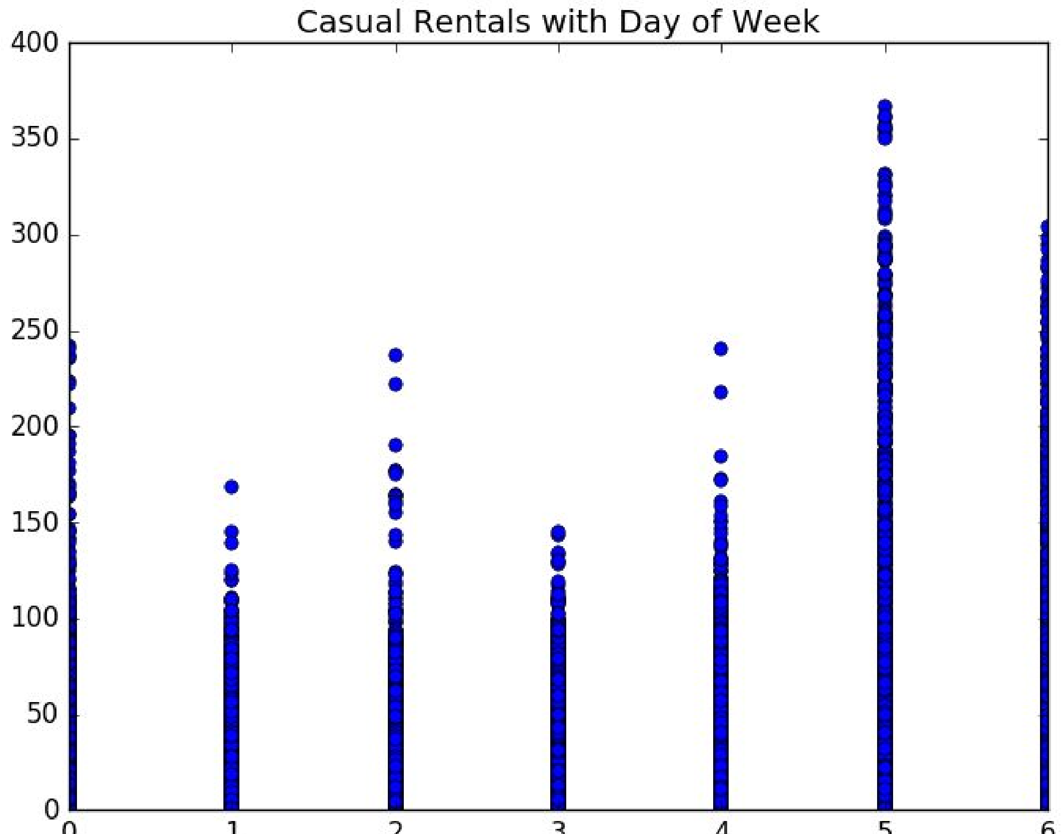
Casual rentals also decrease with windspeed increase



Data Visualization : Busy weekday registered rentals (Mon =0, Sun=6)



Data Visualization : Busy Casual rentals in Weekend. (Mon=0, Sun=6)



Feature Engineering

- Datetime :   Break it into month, day of the week, hour.
- Holiday is redundant once we have day of the week and working day.
- Atemp is correlated with temp. Dropping temp as atemp (feels like temperature) is a more realistic representation of conditions.
- Convert weather = 4 (heavy snow/rain) to 3 (snow/rain).
- Total of 9 features. Month, DayofWeek, Hour, WorkingDay, Weather, Atemp, Humidity, Windspeed, Season.
- Predict casual and registered rentals separately and sum them to get total rental count

Prediction

- KNearestNeighbors used to make baseline submission. Score : 1.02
  Did not predict casual and registered separately. Score improves to with separate prediction.
- Linear Regression. Predicted negative counts were converted to 0. Did not result in a good score.
- DecisionTree Classifier. Improved the score to 0.586.
- RandomForest Classifier. Improves score further to 0.559 with parameters num_trees=100 and max_features=9. Slight improvement if features are ordered by feature_importance_. Important features are hour, humidity, atemp, windspeed.

Prediction Contd

- Random Forest Regressor. Best result of 0.49988.   Num_estimators = 100, max_features=8.

Conclusion

- Random Forest Regressor produced the best predictions.
- Improved the score from our initial submission of 1.02 to 0.49988.

Future Work

- We learnt about feature engineering but did not have time to tune predictors as much as we would like. So we should try to tune the parameters more.