**General Assembly SF 17 – Data Science Class Project (Predicting Bikeshare demand)**

**Project Milestone 2. Due Date Dec 2, 2015.**

Submitted by Anil G.

## Problem statement and hypothesis

The goal of the project was to predict the bike rental demand for the bikeshare program in Washington DC. This is completed data science competition on *kaggle*. Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. In this competition, participants are asked to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C. The problem is divided in predicting the use by two distinct set of users – *Registered* users of the system and *Casual* users of the system.

## Description of your data set and how it was obtained

The data was provided by the kaggle as part of the competition (Please see the reference below for the origin of the data). The Training Set consists of two years of hourly bike rental data along with the weather conditions during the time. The training data is from January 2011 till December 2012. The data was from 1st day of the month till the 19th day of the month. There were a total of 10866 rows of observations. The data quality was very good, with no missing values. The predictor (features) values included the time of the day, whether it was a working day or a holiday, and weather conditions such as season, temperature. The output consisted of the count of registered, casual and total bike rental during the hour.

The test set was the again the two years of hourly data from January 2011 till December 2012. It included the hourly predictor values for last days of the month from 20th till 28, 29, 30 or 31st of the month based on the number of days in the month. The output values were not provided.

The following table describes the predictor data provided for the training and test set.

| **Predictor** | **Description** |
| --- | --- |
| datetime | Date and hour of the day |
| season | 1 = spring, 2 = summer, 3 = fall, 4 = winter |
| holiday | whether the day is considered a holiday –(1 – holiday, 0 – not a holiday) |
| workingday | whether the day is neither a weekend nor holiday ( 1- working day, 0 – not a working day) |
| weather | 1: Clear, Few clouds, Partly cloudy, Partly cloudy  (Nice to bike on) 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds  4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog |
| temp | temperature in Celsius |
| atemp | "feels like" temperature in Celsius |
| humidity | relative humidity |
| windspeed | wind speed |

The following table describes the output data provided for the training set.

|  |  |
| --- | --- |
| **Output** | **Description** |
| casual | number of non-registered user rentals initiated |
| registered | number of registered user rentals initiated |
| **count** | **Total number of rentals (casual + registered)** |

## Description of any pre-processing steps you took

Since the data quality was extremely good and complete, there was no pre-processing of data needed.

## What you learned from exploring the data, including visualizations

The data was plotted using scatterplots to understand the features that impacted the output variables and the pattern of impact. The data was plotted for the total number of rentals and features and also for casual users and features and registered users and features.

One of the facts I was surprised to find out that the number of rentals were higher during the Fall and Winter season. I was expecting the rentals to be higher during the summer and spring months, when weather is good and days are longer and also there are a lot more tourists or casual users in the town.

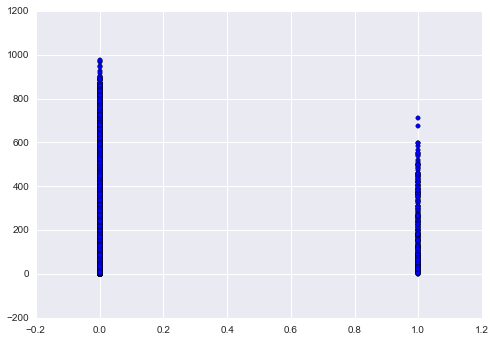
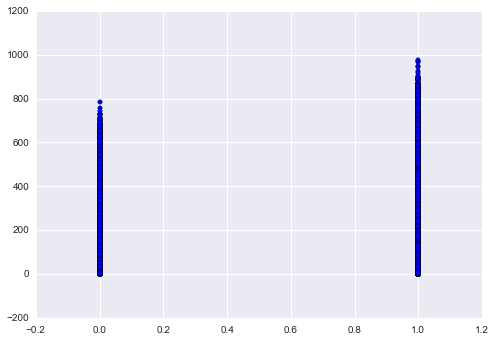
## 

Season versus Count of Rentals

1-Spring, 2-Summer, 3-Fall, 4-Winter

Fall & Winter have more rentals than Summer & Spring

## Additional surprising fact that was discovered that the rentals were higher during the working days as compared to holidays.



Holiday versus Count of Rentals

0 – Not a Holiday, 1 – Holiday

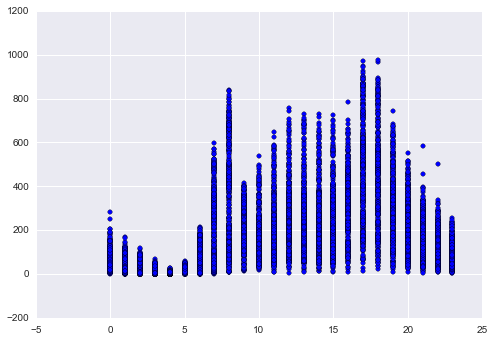
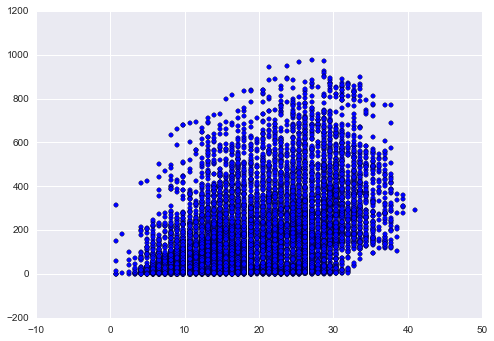
More rentals on non-holidays

Working-Day versus Count of Rentals

0 – Not a Working Day, 1 – Working Day

More rentals on Working Days

## The weather and time of the day had expected impact on the number of rentals. The bike rentals were high when the temperature was pleasant. Also most of the rentals were during work hours, evenings and early nighttime. The rentals would taper down during the late nights and early morning hours.



Hour of the day (00-23 hrs) versus Count

(Rental taper down in late nights and early morning)

Hour of the day (00-23hrs) versus Count

Expected – more rentals during work hours, evening and early night hours. Rentals taper down during late nights and early morning hours

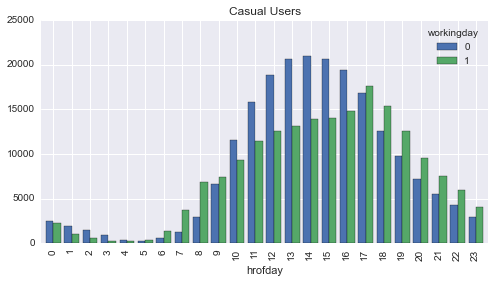
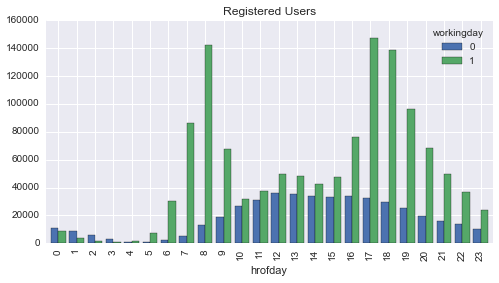
Temp (in Celsius) versus Count

Expected – more rentals when temperature is pleasant

Hour of the day (00-23 hrs) versus Count

(Rental taper down in late nights and early morning)

A different pattern was observed between casual and registered users. The casual users rented more on non-working days as compared to working days. Also most causal rentals would occur during late morning, afternoon and evening hours. The registered rentals are comparatively high on working days and seem to point to the fact they are used a lot of office commutes.



Hour of the day versus *Casual* Rentals on WorkingDay and non-WorkingDay

Hour of the day versus *Registered* Rentals on WorkingDay and non-WorkingDay

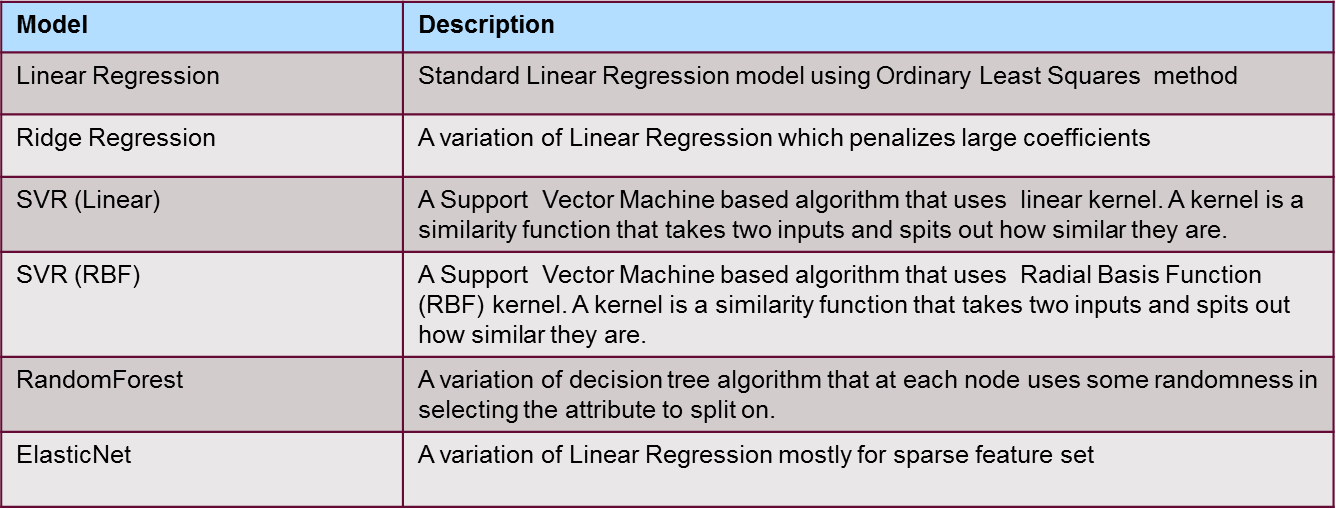
## How you chose which features to use in your analysis

Based on the initial data analysis it was clear that we need to have two different models – one for the registered renters and the other one for the casual users. The features that impacted the two models were different. The two features that seem to impact the most were the type of renter, time of the day of the rental and the type of the rental day - holiday versus working-day. The atmospheric conditions (temp, atemp, weather, windspeed, humidity) had an expected impact on the demand, however the season (summer, spring, winter or fall) had a different impact on the rental demand. The analysis was done with all the above features.

## Details of your modeling process, including how you selected your models and validated them

As the data was explored, it was discovered that the holiday feature only included the government and other widely observed holidays. It did not include weekends. Therefore a dummy or derived feature called *weekend* was added to the feature set. It was set to 1 for Saturday and Sunday and 0 otherwise. Following features were extracted from the datetime field to run the analysis – hour of the day (*hrofday* from 0-23 ), day of the month(*dayofmth*), day of the week (*dayofweek* 1-7 for Monday through Saturday), peak hours (*peak* – the peak hours varied between working and non-working day). Additional a variable (*rollingmth* to vary from 1-24) was created to account for the incremental monthly increase in registered users as the program become popular.

The modeling process was the classic supervised regression machine learning. The existing set of regression models were used to analysis the data and predict the demand for the test data. The prediction were evaluated by running 10 fold cross validation and comparing the Root Mean Square Error (RMSE) values. I used the suggested path provided by scikit-learn (see reference below) to select the models to run the analysis on. All the modeling was done using the scikit-learn libraries. I used an iterative process to change the feature sets and run the model, cross validate and compare the RMSE with a prior running. In most cases the default values for the model parameters were used. Following are the details of the different models that were run and the best values I obtained during analysis. I used the 75:25 ratio for train\_test\_split to train the model.



|  |
| --- |
|  |

Following is the results from the run for different features and models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Features Used** | **RMSE**  **Casual**  **Renter** | **RMSE Regular**  **Renter** |
| Linear Regression | 'weather', 'temp', 'atemp', 'windspeed',  'workingday', 'season', 'holiday', 'hrofday', 'peak', 'rollingmth', 'dayofweek', 'weekend' | 34.47 | 91.28 |
| Ridge Regression | 'weather', 'temp', 'atemp', 'windspeed',  'workingday', 'season', 'holiday', 'hrofday', 'peak' | 34.97 | 106.23 |
| SVR (Linear) | 'weather', 'temp', 'workingday', 'season','holiday', 'hrofday' | 39.68 | 133.37 |
| SVR (RBF) | 'weather', 'temp', 'atemp', 'windspeed',  'workingday', 'season', 'holiday', 'hrofday', 'peak','rollingmth' | 46.63 | 146.97 |
| RandomForest | 'weather', 'temp', 'atemp', 'windspeed','hrofday','workingday' | 24.70 | 81.02 |
| ElasticNet | 'weather', 'atemp', 'windspeed', 'humidity' | 39.86 | 137.20 |

Following is the variation of the Registered and Casual users in the provided training data set.

|  |  |
| --- | --- |
| bikedatatrain['count'].describe() | bikedatatrain['registered'].describe() |
| count 10886.000000  mean 191.574132  std 181.144454  min 1.000000  25% 42.000000  50% 145.000000  75% 284.000000  max 977.000000 | count 10886.000000  mean 155.552177  std 151.039033  min 0.000000  25% 36.000000  50% 118.000000  75% 222.000000  max 886.000000 |

## Your challenges and successes

The main challenges I ran into was with the tuning of the models, it was an iterative process with running with different features and comparing the RMSE values. Initially it was also difficult the order in which model should be run. I do not believe that I have optimized the models to the best values. In addition the competition called out to only use the training data that was available up to that point to run the prediction. I however used the complete training set to fit and predict the model and also to cross validate.

## Possible extensions or business applications of your project

After the project models are tuning, we could add a web interface for the use of the Bikehare management and users. The project can be developed for the use of the bikeshare management to understand the peak and patterns in the usage of the bike systems, so they can optimize the number of bike they need to maintain and order for future. The users of the bikeshare system can use the project to understand when the peaks demand occur and if they can be sure if a bike will be available for them to rent.

## Conclusions and key learnings

During the analysis I discovered that none of the model seems to be very accurate. This leads me to believe that I need to identify more derived or dummy features that predict the bike rental demand. The features need to be able to incorporate the influence of additional registered renter signing up with the program over a period of time. In addition it was suggested I should include inputs from other sources such as the DC Subway system performance, whether congress is in session etc. to include features that may be impacting the demand for bike.

Since RandomForest seems to be more accurate than the other algorithm – I should use the model parameters to tune the model and also run the model with different feature sets. Additional algorithms based on decision trees and polynomial regression should also be tried to see if the model performance improves.

## References

* Kaggle competition (<https://www.kaggle.com/c/bike-sharing-demand>)
* Problem Definition, Data Set and related information:

Fanaee-T, Hadi, and Gama, Joao, Event labeling combining ensemble detectors and background knowledge, Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg.

* Class Notes and python programs
* Choosing the right estimator (<http://scikit-learn.org/stable/tutorial/machine_learning_map> )