Explorative Data Analysis Task

Blue Yonder, Data Science Consultant Position

Zeynep Vatandas

Part 1

Short Analysis of the Dataset

I try to answer the question of "*How natural factors affect using rental bike service for Capital Bikeshare System in Washington DC?***"** by analyzing the dataset. Following points represent the outcome of the analysis, for more detailed analysis, please refer to the Jupyter Notebook.

- ✓ Casual and registered users have a different pattern of using the bike rental service throughout the years of 2011 and 2012 as can be seen in Figure 1.

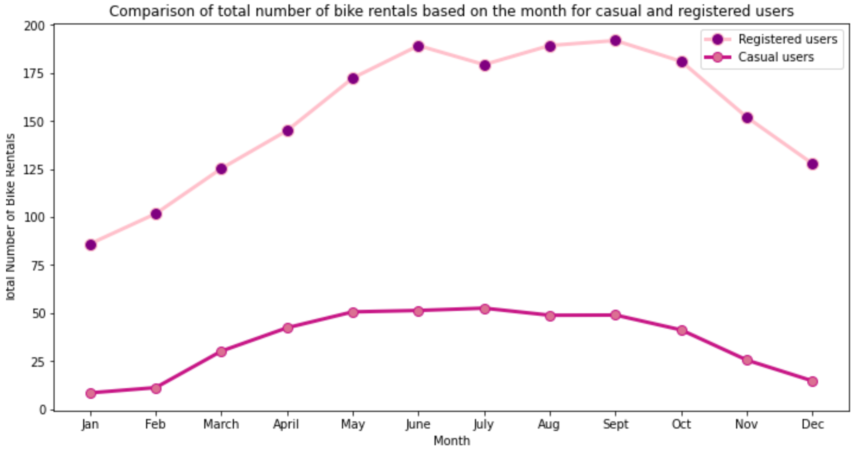

Figure 1. Comparison of total number of bike rentals based on the month for casual and registered users

- ✓ As Figure 2 indicates, registered users have a peak time in the morning and evening while casual users have increasing activity from the morning on until the noon and remain almost stable during the day until the evening. (1 pm to 5 pm). Considering the pattern during the days, registered users might be mostly office or school commuters whereas casual users may be tourists and local people having outdoor activities. This idea is derived based on my own assumptions. To find out more, we would need a different dataset to identify the origin of the users.

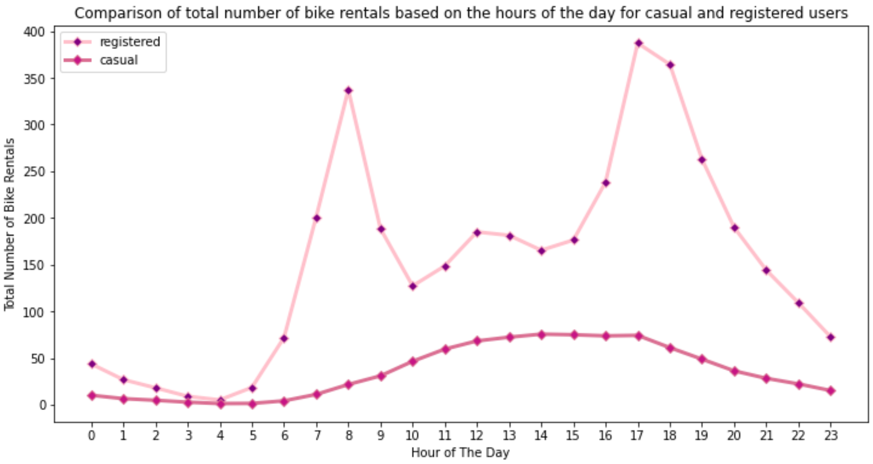

Figure 2. Comparison of hourly total number of bike rentals for casual and registered users

- ✓ Activity level is higher in weekdays for registered users while casual users appear to be more active in weekends as can be seen in Figure 3. High weekday activity shows that registered users follow a regular riding pattern. For the casual users, high weekend activity indicates that casual users might mainly consist of tourists who use rental bike services for local sightseeing and recreation activities.

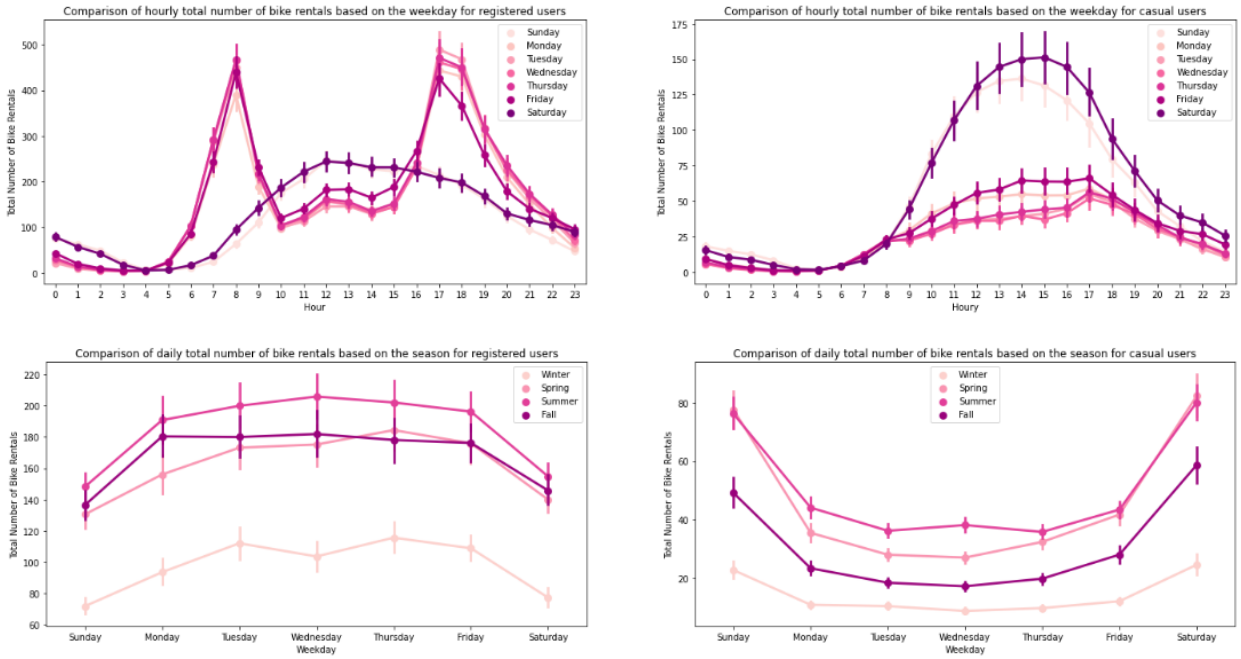

Figure 3. Activitiy level of casual and registered users during weekdays and weekends

- ✓ Figure 4 depicts that the number of registered users is higher in working days compared to the casual users. They show higher activity in non-holiday time as well, which support the idea that registered users use the service for going regular places while casual users prefer the service more when it is holiday time or a non-working day as they use rental bikes for recreation activities.

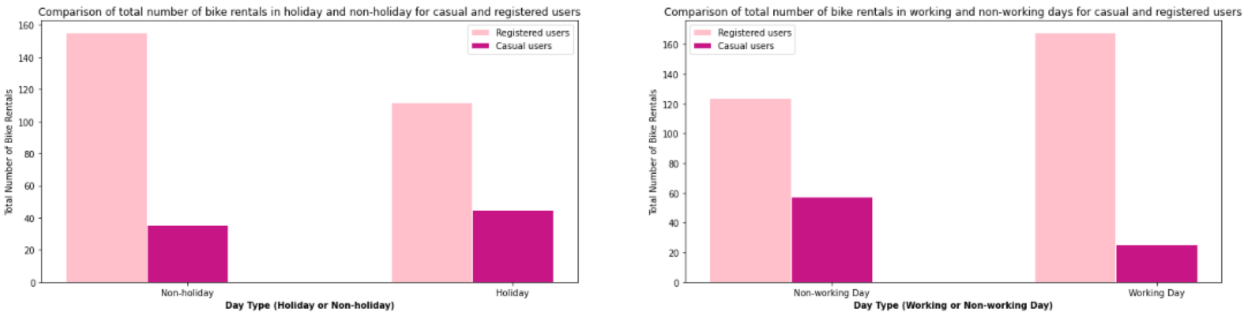

Figure 4. Activitiy level of casual and registered users during holiday, non-holiday, working and non-working day´times

- ✓ Figure 5 suggests that both casual and registered users prefer better weather conditions such as warm weather, convenient humidity, low wind speed and clear sky for using the rental bike service. For both users, the higher the temperature the more number of bike rentals observed as warmer weather let people do outdoor activities more. Humidity graph suggests that around 20% humidity is the best case for both users. When the humidity level gets higher, the number of rental bikes starts decreasing. Wind speed affects the decision of riding a bike as well. Higher the wind speed, the less number of rental bikes as it gets more difficult to ride a bike under the strong wind speed condition.

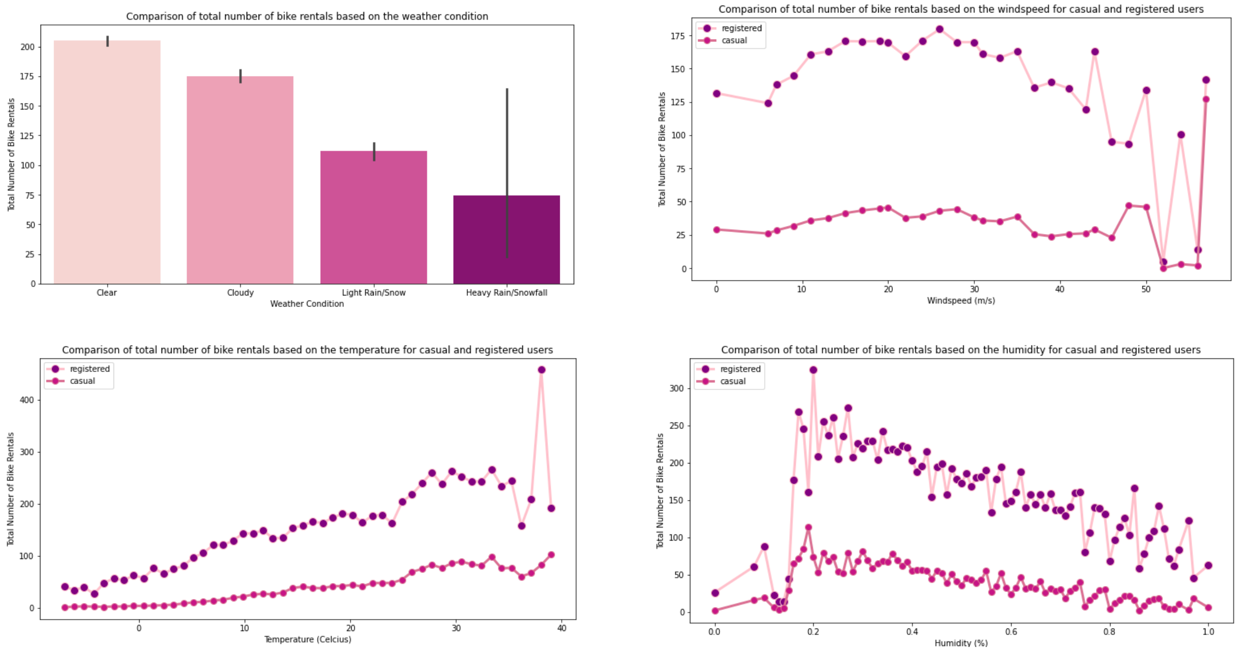

Figure 5. Activitiy level of casual and registered users based on the weather conditions

- ✓ Among the seasons, both users prefer summer time as the weather conditions are more convenient for riding a bike as can be seen in Figure 6.

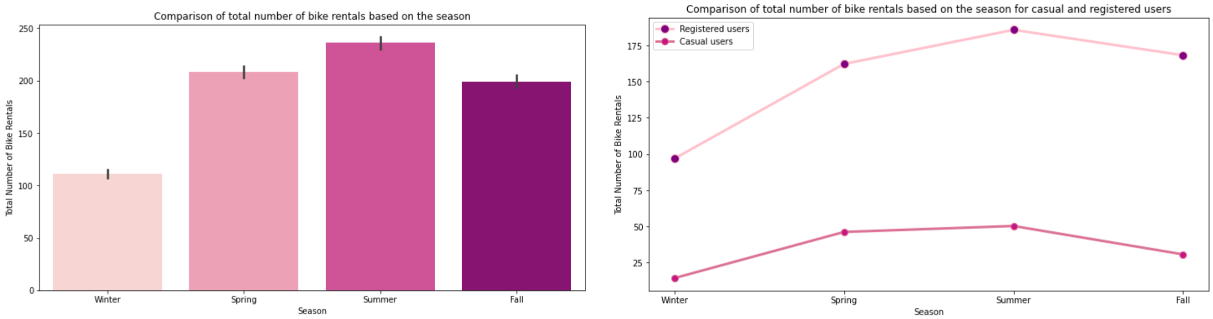

Figure 6. Activitiy level of casual and registered users based on the season

- ✓ Correlation matrix helps us to understand how the dependent variable which is total number of rental bikes is influenced by the features. I dropped the irrelevant data which are ID and Date, as well as highly correlated values feeling temperature (correlated with temperature), month (correlated with season), number of casual and registered users. We also perform one hot encoding on categorical columns which are weather and season. Temperature and hour of the day seem to have an influence on the total number of bike rental the most as Figure 7 depicts.

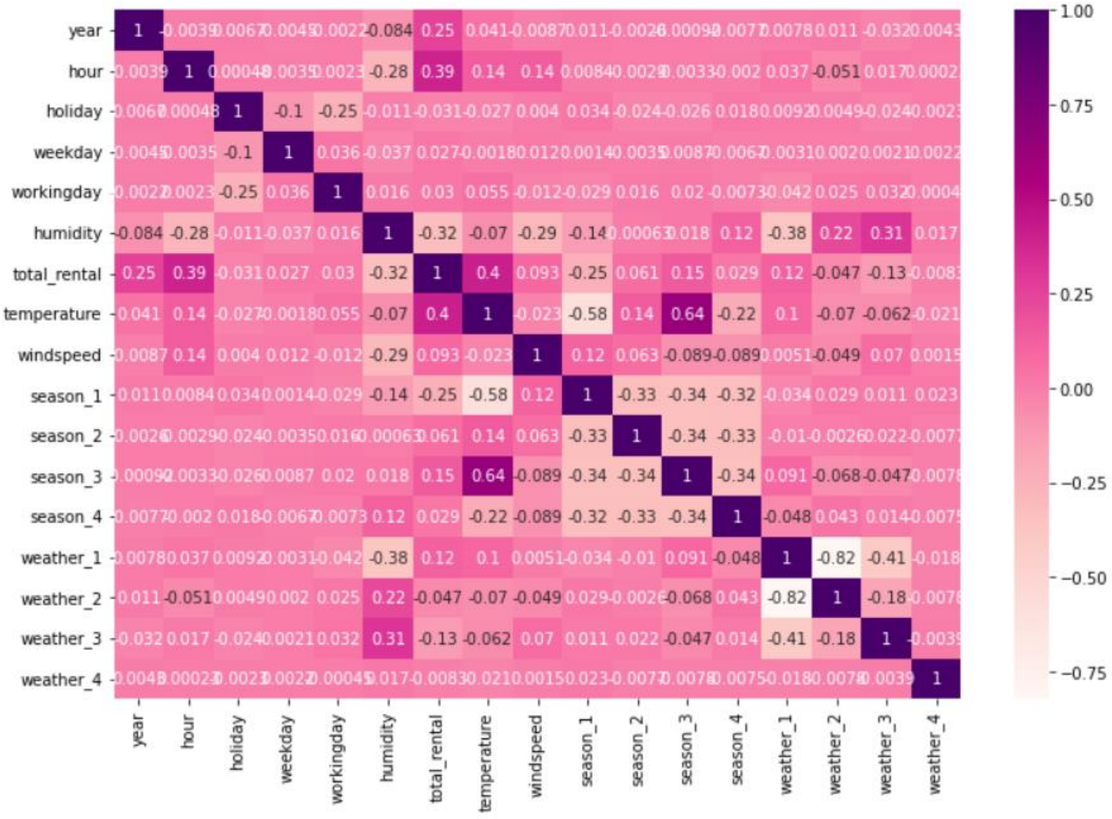

Figure 7. Correlation Matrix

- ✓ *Recommendations:* Capital Bikeshare system may consider having more stations around the areas where offices or schools are mainly located as most of the users are the registered users who commute to workplaces and schools on a daily basis. Supplying extra bikes to the stations in the peak rental hours may be considered to increase the usage. To increase the number of users in off-seasons (fall and winter), discounted prices for bike renting may be considered. Maintenance work for the bikes can be done at night as the demand is very low compared to the other times of the day. The company may continue to apply the prediction so the changes in rental bike demands can be observed.

Prediction Model

Considering a scenario where The Capital Bikeshare can use the collected data from past years to predict next year's demand, Random Forest Regression is used as it is a good prediction model for a data set with categorical variables and when a company needs to predict a continuous value which is correlated with other features that affect the outcome. In our case, we would like to predict total number of rental bikes in relation to the other parameters such as temperature, season, hour of the day etc. As our target value is not an outcome that can only take a value from a defined finite set or do not represent a value that is either 0 or 1, we can say it is continuous. The random forest algorithm provides a higher level of accuracy by applying decision tree algorithm. The algorithm is stable, in case of having a new data point introduced in the dataset, the overall algorithm is not affected much since the new data may impact one tree, but it is very hard for it to impact all the trees. After using Random Forest Regression in Capital Bikeshare dataset, following Mean Absolute Deviations from the test data and the predictions of the model have been obtained to compare.

- ✓ Mean Absolute Deviation of Test Data: 142.66409997338314
- ✓ Mean Absolute Deviation of Predictions: 137.0116341421385

Comparing two deviations, we may say that they do not deviate from each other much. However, to justify more if the prediction of the model is good enough, we can check the error loss functions below:

- ✓ Mean Absolute Error (MAE): 27.66410208340183
- ✓ Root Mean Squared Error: 47.15407169138791
- ✓ Root Mean Squared Log Error (RMSLE): 0.35811094742534183

Considering the mean of total number of rental bikes which is 189.46, Mean Absolute Error of 27.66 does not look like a very good prediction, as it roughly makes around 14%. Having a prediction 27 more or less than the actual target value could be tolerable in case the mean of total rental bikes is quite high such as in thousands. Therefore, using another prediction model and comparing the error loss of each model to decide can be a good approach.

Assume that the code you are writing is used in production in a daily prediction service and maintained by your colleagues (what could that mean?)

A way to monitor the model continuously and check if the prediction model is still valid for the service would be necessary. Deployment of updates should be done with CI/CD pipeline as automated process handles the repetitive build, test and deployment tasks and alerts you about any issues. Beside all, the code should be clean, understandable and easily changeable by the team.

Part 2:

What are the scaling properties of your model, if you assume that the amount of data you need to handle go up to several terabytes? Do you see any problems?

Random Forest Algorithm is good for large data sets and usually they do not have scaling issues. However, the basic idea in Random Forest algorithm is to pool a lot of very deep trees and growing deep trees can result in taking up a lot of resources. Playing with parameters like *number of trees* can help to reduce computational time, but it is not very ideal solution. Because these parameters also have a huge impact on the accuracy of the prediction. Regardless of the prediction model used, depending on the size of the data, bottlenecks like data storage, network bandwidth, computation power can become a problem.

How would you address these problems? Are there technologies for data storage/predictive modelling you can build upon? Describe how the technologies you mention solve the scaling problems you see with model.

Training data can be stored in traditional databases (e.g. Microsoft SQL, Oracle, Postgres) until they grow to 1TB. If the data grows beyond this, it's time to look for other solutions, for example cloud storage systems like AWS.

Spark is a good processing framework that supports SQL queries, streaming data, machine learning and graph processing. Because it is very fast for in-memory processing and APIs make it easy to use to manipulate semi-structured data and to transform data. The Input Pipeline for the solution also needs to be structured correctly to not become the bottleneck. Reading from the source requires a lot of I/O power such as disk speed and network bandwidth. Eventual transformation on the data can require additional CPU processing power. Loading the data for training, is relying on GPUs/ASICs assuming we have them available. The input data should be broken into batches to parallelize the steps and utilize all resources at the same time and not blocking each other from working.

What are the limits and drawbacks for your new approach?

Even though using a cloud data storage can be a good idea for big data, the money it costs and its limitations should be taken into account as well. AWS, besides its cost, has some limits on size of the training data, batch prediction and number of variable in a data file.

Do you have hands-on experience with such technologies? Which ones? For how long?

I have basic knowledge of the mentioned technologies. Unfortunately, I do not have a hands-on experience yet. In my academic work experience, I did not have an opportunity to work with these technologies. However, I am very keen on having a hands-on experience with them.