RENTAL BIKE PREDICTION

Part 1

1. Dataset view

day.csv → contain daily data.
hour.csv → contain hourly data.
[As the requirement is to predict hourly utilization "cnt", for this we use only hour.csv as day.csv contain the role up data hour.csv]

2. Understanding of dataset and its variable

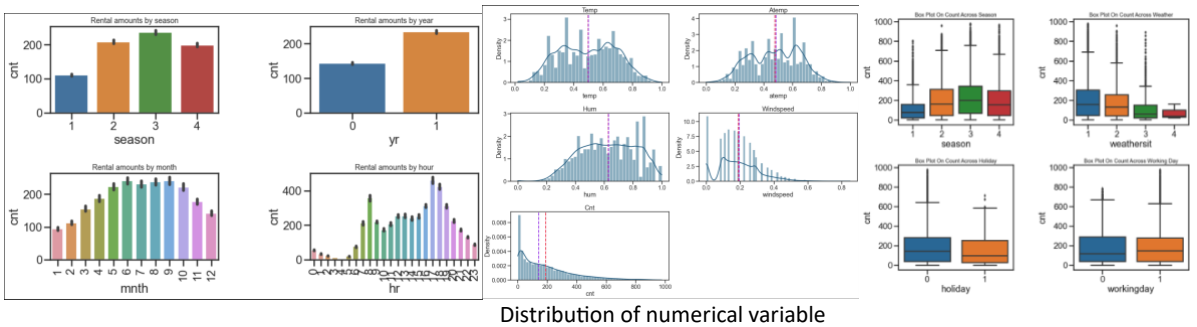
a. Duplicates and Null check: The data has neither duplicate not null values.
b. Dataset description:

```
### Dataset description:
* Total no. of days: 731
* season      : 4 (1:springer, 2:summer, 3:fall, 4:winter)
* yr          : year (0: 2011, 1:2012)
* mnth        : month ( 1 to 12)
* hr          : hour (0 to 23)
* holiday     : binary (yes/no) - weather day is holiday
* weekday     : day of the week (0:Sun, 1:Mon, 2:Tue, 3:Wed, 4:Thu, 5:Fri & 6:Sat)
* working day : binary (if day is neither weekend nor holiday is 1, otherwise is 0)
* weathersit  : - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
                - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
                - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
                - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
* temp        : Normalized temperature in Celsius. The values are divided to 41 (max)
* atemp       : Normalized feeling temperature in Celsius. The values are divided to 50 (max)
* hum         : Normalized humidity. The values are divided to 100 (max)
* windspeed   : Normalized wind speed. The values are divided to 67 (max)
* casual      : count of casual users
* registered  : count of registered users
* cnt         : count of total rental bikes including both casual and registered
```

- *dteday is in object type that should be change in datetime*
- *casual & Registered variable is dependent on target variable cnt, which will not include in final analysis and prediction*
- *instance is just for index which will not include in final analysis and prediction*

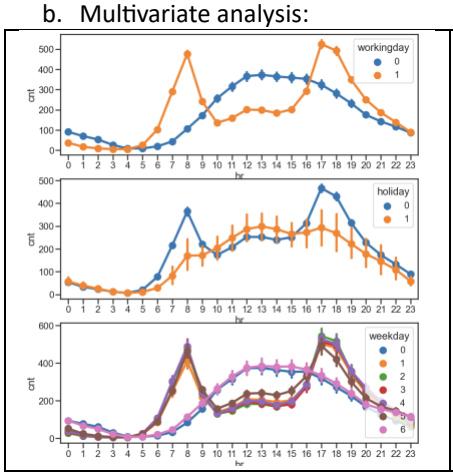
3. Exploratory data analysis (Data Visualization)

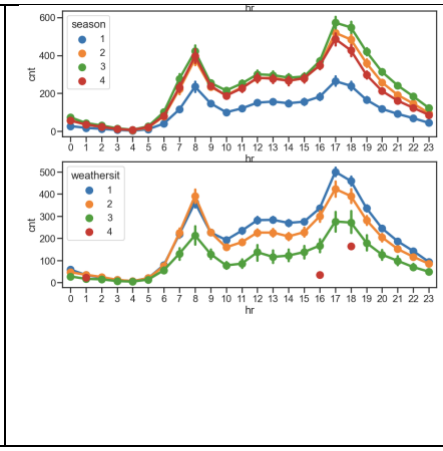
a. Data insights:
- *2012 has more rental than 2011*
- *People prefer to ride bike in summer (June-September) than winters.*
- *Bike rental is higher at 0800 and 1700-1800 Hrs. which means people used bike rental for school/office go and return.*
- *Weather conditions impact the bike rentals as there is very limited bike rental in bad weather where the rentals are highest in good weather condition.*
- *People rent bike less on holiday.*
- *People rent more on working day than holidays or weekend.*
- *People prefer to ride bike in low windspeed and moderate temperature.*



Distribution of numerical variable

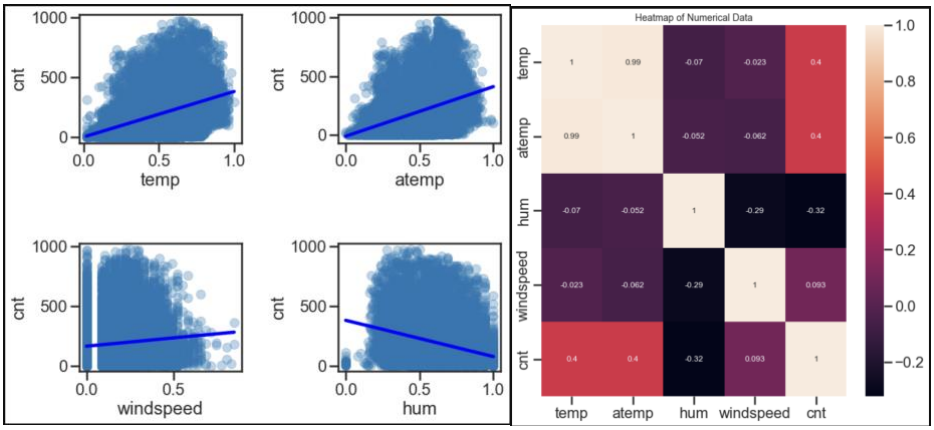
b. Multivariate analysis:





- *During working day the maximum demand is on 0800 and 1700-1800 but it is regular on non-working day or holiday.*
- *the number of rentals is highest in the fall and lowest in the spring.*
- *the number of rentals is highest when the weather is good, but there are a few rentals at 1600 and 1800 during heavy rain and snow. This can be considered as outliers.*

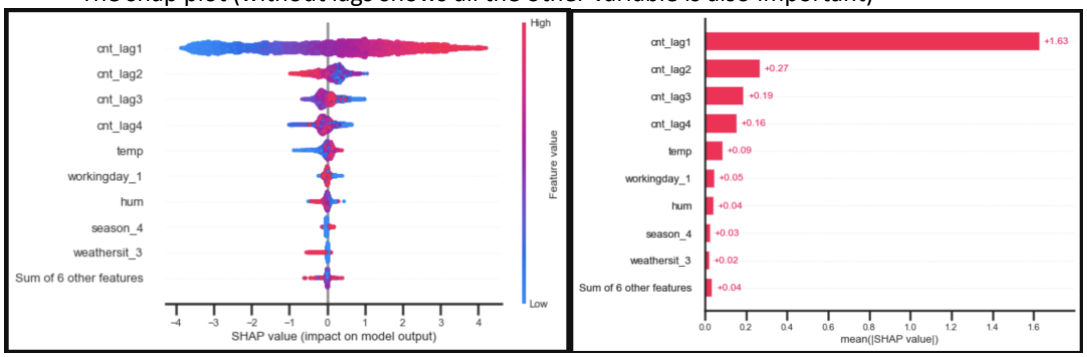
c. Correlation analysis:



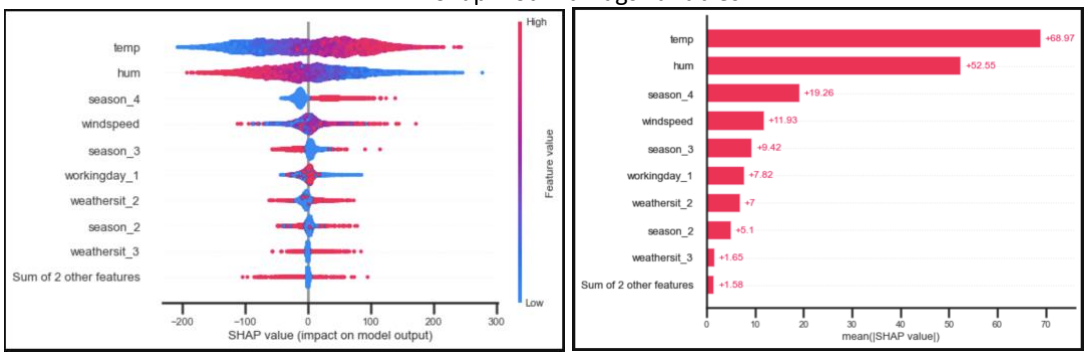
- *Temperature (temp) and feeling temperature (atemp) has +ve correlation with cnt*
- *Humidity is -ve correlated with count (more bike rental in lower humidity).*
- *temp and atemp is highly correlated, we need to drop either one variable for the prediction.*

4. Data preprocessing and feature engineering:

a. Data transformation for target variable – 'cnt' (Cube root has better bell curve)
b. One hot encoding for categorical variable
c. Drop variable due to multicollinearity – atemp, casual and registered.
d. Drop 'yr', 'month' and weekday as workingday and holiday is derived from these variables.
e. Check correlation of lags (t-1, t-2, t-3 and t-4) of 'cnt' using acf and pacf plot
f. Add lags in dataset (t-1, t-2, t-3 and t-4)
g. Check variable importance using XGBRegressor "shap plot" and "featureimportance"
   - Lags as feature are important feature for prediction (based on "shap bar plot")
   - Since lags variable is main driving feature for modeling, to see the other variable importance we plot the shap plot without lags.
   - The shap plot (without lags shows all the other variable is also important)



Shap Plot with lags variables



Shap Plot without lags variables

5. Modeling and prediction

a. Drop first 4 rows contains Nan values due to lag calculation.
b. Transforming the target variable misleads in model performance metrices and doesn't change much in terms of R-square. So, we are not transforming the target variable.
c. Model training and Testing on XGB Regressor. Took last month data for testing the model.

d. Calculation of performance matrices – R-Square, MAE and RMSE (for first 24 hrs and first 7 days)
e. Comparison of results of XGB Regressor with Linear regression results proves the XGB Regressor is best fit algorithm for prediction.

	Model	R2_train	R2_test	Adjusted_R2_train	Adjusted_R2_test	MAE_train	MAE_test	RMSE_train	RMSE_test
0	XGBRegressor_month	0.97346	0.93616	0.96643	0.87787	19.88636	26.41867	29.65325	41.63674
1	Linear_Regression_month	0.78572	0.74477	0.62019	0.43699	55.64334	49.98294	84.26080	83.24949

6. Results and Conclusion

a. The model also has similar performance without transforming the target variable.
b. XGB Regressor gives promising results compare to Linear regression based on performance matrices.
c. XGB Regressor is best for business case as it is fast and efficient on large dataset as well.
d. XGB Regressor is non-linear machine learning techniques which prevents model to overfit.
e. The model performance (in terms of MAE and RSME) is declining when it predicts for entire week. So current model is best fit for daily/hourly demand forecast.

	Model	R2_train	R2_test	Adjusted_R2_train	Adjusted_R2_test	MAE_train	MAE_test	RMSE_train	RMSE_test
0	XGBRegressor_24Hrs	0.97346	0.95284	0.90543	0.74926	19.88636	27.88100	29.65325	35.75752
1	XGBRegressor_week	0.97346	0.91941	0.96386	0.82229	19.88636	33.55442	29.65325	54.34670

7. For daily use on production following steps need to follow

a. For production need separate python script which will run on cron job (scheduler) on hourly/daily basis depend upon the business requirement
b. Only data manipulation, transformation and lag calculation of test data code will be used for production script.
c. Need to schedule model training on latest dataset (once in week/month)
d. Need a live data pipeline which store live data in database (cloud-hosted NoSQL database) that can be used the hourly prediction of rental bike.
e. The predictive model needs to maintain or upgrade by analyzing latest data using same jupyter code.

Part 2

1. Solution on large dataset and Scaling properties:

A scale up solution would need a complementing platform that can scale up with solution as the demand grows up. In today's time there are multiple companies that offers not only infrastructure on demand but also the additional services such as platform to run models as per the requirements. This scalable solution would ideally comprise of following components:

a. A scalable database: This database stores the data in runtime and can expand the storage capacity on demand. In addition to data storage this would also provide data for analysis or model building in real time
b. Computation platform: This is the platform where our model will run from time-to-time basis updated data. This will also be the platform where the scoring/prediction engine will run.
c. Web services for endpoints: The endpoints serve an important role in delivering the model results in run time. It reduces the lag between the time from model prediction to prediction result delivery to the targeted stakeholder.

2. Technologies for data storage and processing used:

In the current era of big data technologies, All the cloud services companies provides the storage and building predictive modelling on large scale data. All the cloud services hosted noSQL database which is best for this data scalability problem. But if the firm doesn't want to take services of cloud platforms like AWS or Azure then they should setup a complete infrastructure on premises that includes High performance computers along with big data storage like HDFS infrastructure etc.

Hands on Experience: I have used AWS Kinesis stream for data pipelining along with DynamoDB for storing and processing live data for a project. For analytics and predictive modelling of big data I have used pyspark on GCP and data bricks for data on MS Azure for 2-3 years.

3. Limitation and drawbacks:

Few of the key problems that model will face once we start feeding in terabytes of data for model building would be:

a. Platform cost: While these cloud platforms do come with a convenience of scalability, it all comes at a cost. As you keep storing the additional data in real time, the data would need additional storage and it would be very long until the platform cost would run in 100 thousand.
b. Processing time: Larger the modelling data, larger would-be processing time and availability of forecasting results
c. Data storage: Cloud platforms provides data scalability and availability at the cost of data storage in remote server. If you're concerned about data availability in your country/continent, you might have to check with data storage location of cloud providers.

Few suggestions to handle these problems:

a. Instead of granular hourly level forecasting, try daily level forecast. This will solve for more accurate model results and will also take care of inconsistencies with hourly weather and temperature data.
b. Shorten the data duration for model building. In time series forecasting, it has been often observed that recent data adds more value than older data. This means, instead of building model on last 3 or 4 years, build model on last 2 years of data.

********** END **********