# Rental Car Demand Prediction

## Introduction :

ABC is a car rental company based out of Bangalore. It rents cars for both in and out stations at affordable prices. The users can rent different types of cars like Sedans, Hatchbacks, SUVs and MUVs, Minivans and so on.

In recent times, the demand for cars is on the rise. As a result, the company would like to tackle the problem of supply and demand. The ultimate goal of the company is to strike the balance between the supply and demand inorder to meet the user expectations.

The company has collected the details of each rental. Based on the past data, the company would like to forecast the demand of car rentals on an hourly basis.

## Objective :

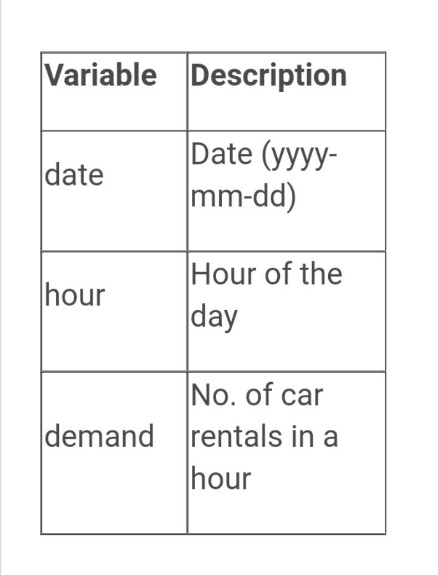
The main objective of the problem is to develop the machine learning approach to forecast the demand of car rentals on an hourly basis.

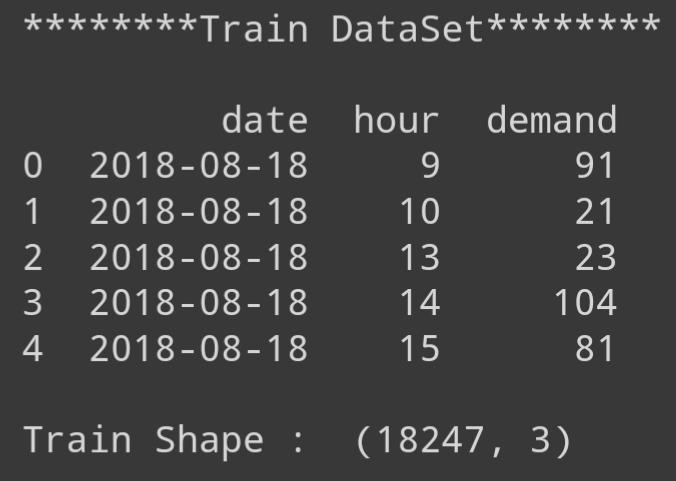
## Context :

1. **Data Analysis**
2. **Data Pre-processing**
3. **Metrics Definition**
4. **Model Prediction**

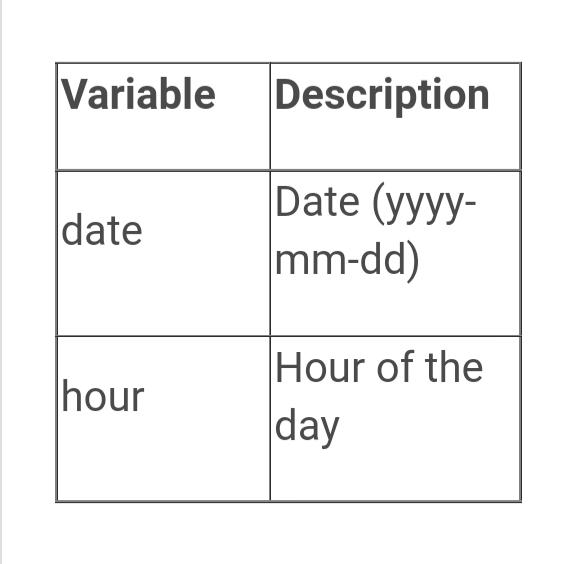
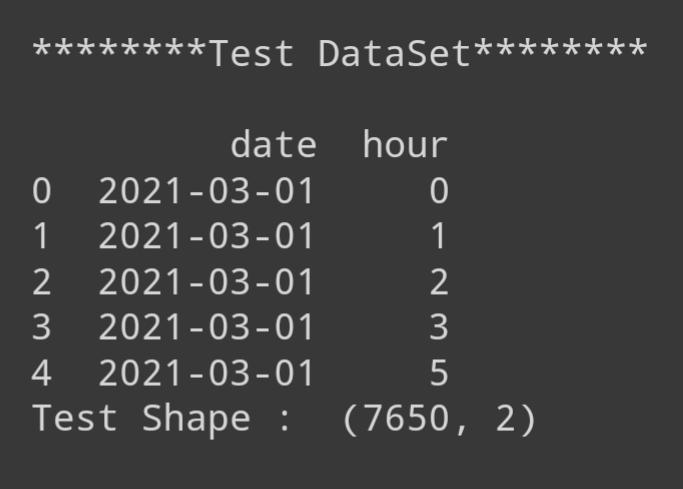
## Data Analysis :

Training Dataset - contains the hourly demand of car rentals from August 2018 to February 2021.

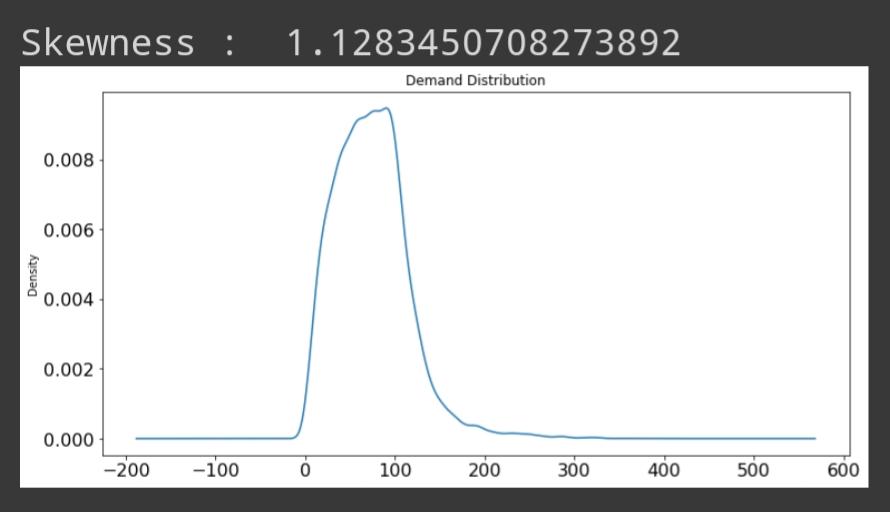




Test Dataset - contains only 2 variables: date and hour. You need to predict the hourly demand of car rentals for the next 1 year i.e. from March 2021 to March 2022.



**Target Variable – “demand”**

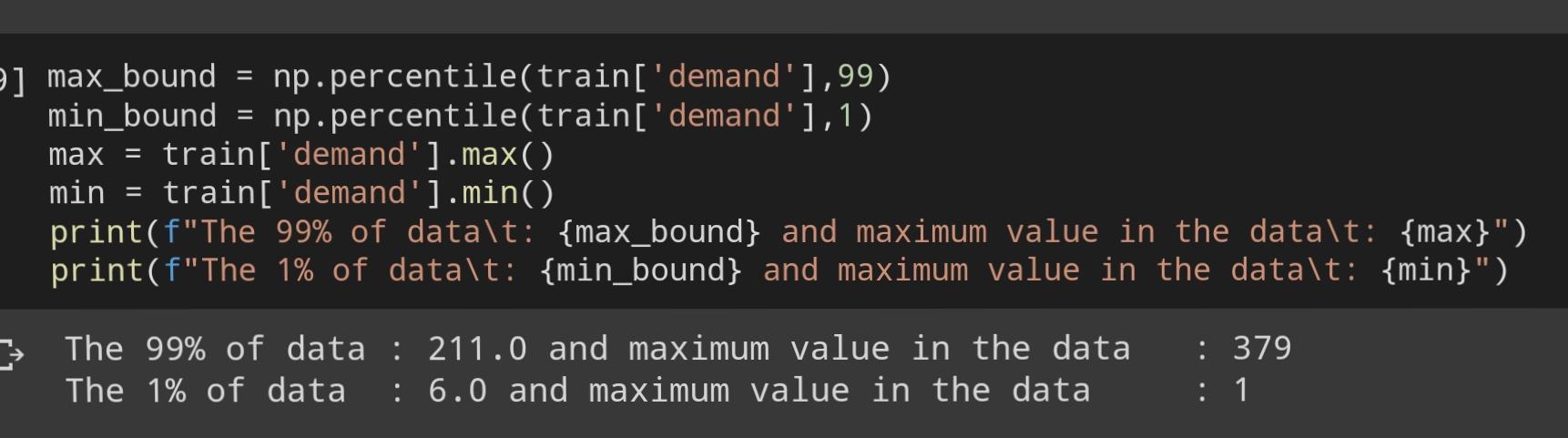


From the above and density distribution plot and skewness value we can see that demand column is right skewed (skewness >1)

To reduce skewness we can try removing outliers or transform the variables.

**Handling Outlier :**

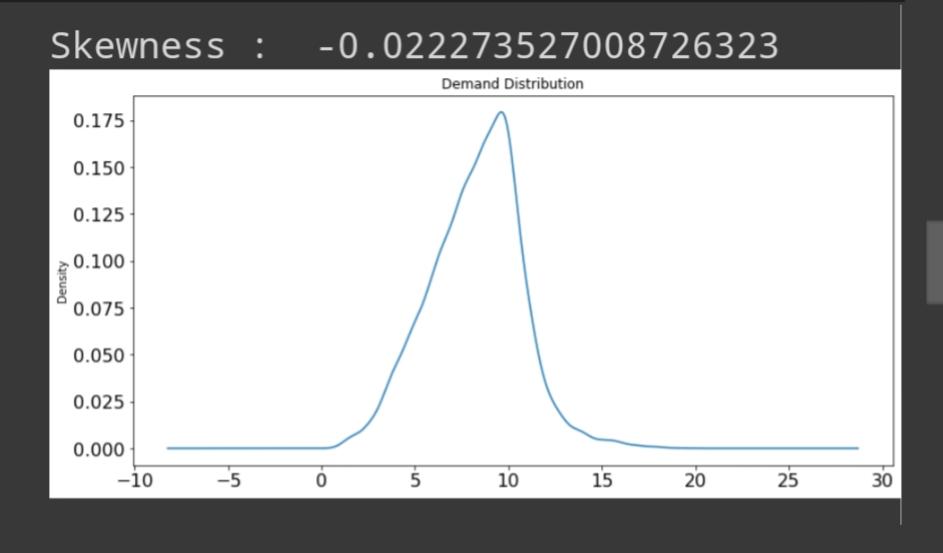
Considered every value above 99% percentile and below 1% of data as outliers.



Replaced every value above 99% of data and below 1% of data with the max\_bound and min\_bound value respectively.

After handling outliers skewness reduced to 0. 71

To further more reduce skewness tried transforming the data, took the square root of the data and found significant reduce in the skewness.: -0.02



## Data Pre-Processing :

Tried transforming datas into different for inorder to achieve standard normal distribution.

**Independent variable :**

**Variable 1 - hour** – Integer

hour\_sqrt – square root of hour variable

hour\_inv – inverse of hour variable (1/hour)

hour\_log – log of hour

**Variable 2 – date –** Date

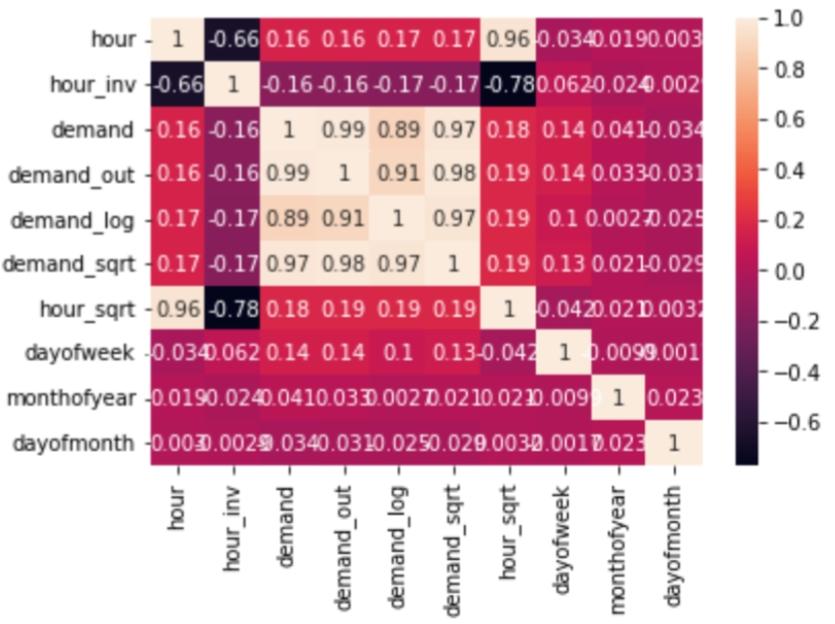
dayofweek – week of the month, range (1,4)

monthofyear – month of the year, range(1,12)

dayofmonth – day of the month, range (1,31)

**Bivariant Analysis :**

Finding the correlation between the independent and dependent variables using heatmap.

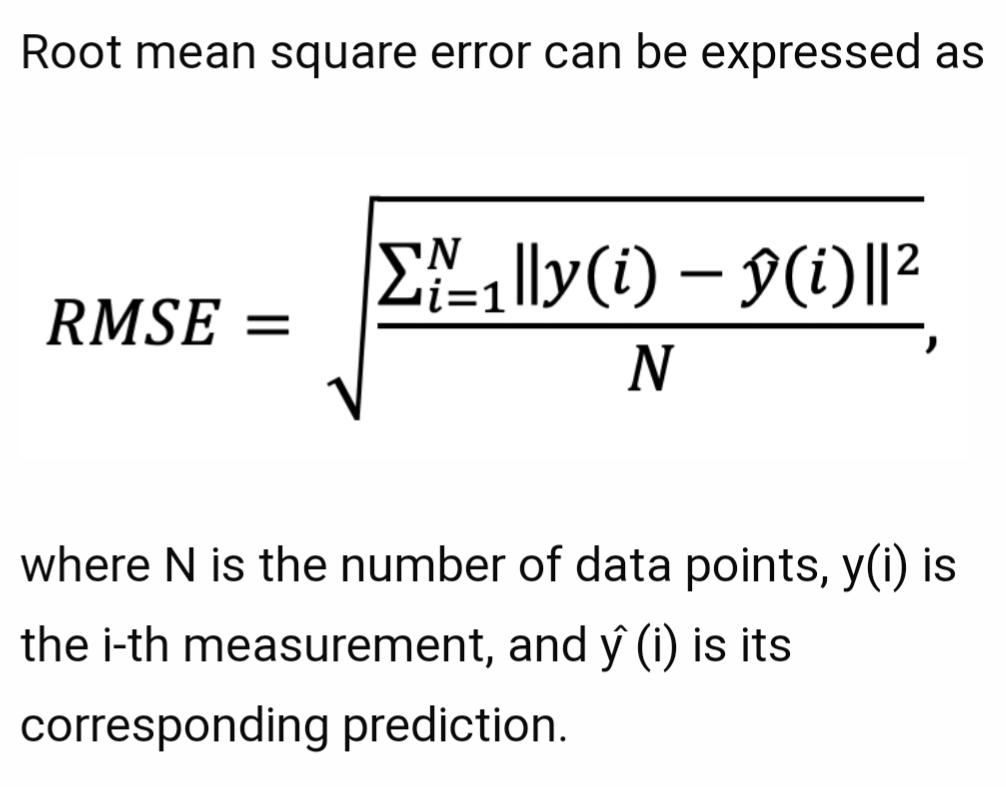


From the above heatmap we could see that hour, hour\_inv, hour\_sqrt, dayofweek is perfoeming good w.r.t demand, demand\_sqrt, demand\_log

## Metrics Definition :

As stated, used **Root Mean Squared Error** as the Metrics.

To compute RMSE, calculate the residual (difference between prediction and truth) for each data point, compute the norm of residual for each data point, compute the mean of residuals and take the square root of that mean.



## Train Test Split :

As we don’t have target variable in test data set, splitting train data set into train and validation 75 : 25 ratio.

## Model Selection :

As the target variable is continuous, tried with regression models.

**Models tried :-**

1. Linear Regression
2. LGBMRegressor
3. CatBoostRegressor
4. XGBRegressor

**Linear Regression –**

**Model 1A Linear Regression** with hour, hour\_sqrt, monthofyear, dayofmonth, dayofweek and target as demand.

RMSE score – 39.59

**Model 1B Linear Regression** with hour, hour\_sqrt, monthofyear, dayofmonth, dayofweek and target as demand\_out.

RMSE score – 38.58

**Model 1C Linear Regression** with hour, hour\_sqrt, monthofyear, dayofmonth, dayofweek and target as demand\_log.

RMSE score – 42.42

**Model 1D Linear Regression** with hour, hour\_sqrt, monthofyear, dayofmonth, dayofweek and target as demand\_sqrt.

RMSE score – 39.72

From the above four models Model 1A (target – demand), Model 1B (target – demand\_out), Model 1D (target – demand\_sqrt) are having similar less RMSE score.

**Cross validation –**

Cross-Validation is a resampling technique with the fundamental idea of splitting the dataset into 2 parts – training data and test data.

In simple terms, cross validation is a technique used to access how well our machine learning models perform on unseen data.

**LGBM Regressor –**

After performing cross validation and hyper parameter tuning finally achieved better accuracy in the model,

**cross\_val\_score(LGBMRegressor(n\_estimators=76),train[['hour','monthofyear','dayofmonth','dayofweek']], train['demand'])**

From the above model got the **RMSE** score of **32.78**

**CatBoost Regressor –**

After performing cross validation and hyper parameter tuning finally achieved better accuracy in the model,

**cross\_val\_score(CatBoostRegressor(n\_estimators=29),train[['hour\_inv','monthofyear','dayofmonth','dayofweek']], train['demand'])**

From the above model got the **RMSE** score of **34.56**

**XGB Regressor –**

After performing cross validation and hyper parameter tuning finally achieved better accuracy in the model,

**cross\_val\_score(XGBRegressor(n\_estimators=200),train[['hour\_inv','dayofweek','monthofyear']], train['demand'])**

From the above model got the **RMSE** score of **34.80**

## Conclusion :

The major step in the prediction process is pre-processing of the data.

Data transformation were done to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms.

Applying single machine algorithm on the data set accuracy was very less. Therefore, the ensemble of multiple machine learning algorithms has been proposed and this combination of ML methods gains better accuracy . This is significant improvement compared to single machine learning method approach.

From the above models, we conclude that the **LGBM model** is good fit for prediction with **RMSE** score of **32.78** , better than other models.