# Analytics Vidhya Jobathon - Apr2022

## Problem Statement

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 in order 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.

## Data Dictionary

We are provided with 3 files - train.csv, test.csv and sample\_submission.csv

### Training set

**train.csv** contains the hourly demand of car rentals from August 2018 to February 2021.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| date | Date (yyyy-mm-dd) |
| hour | Hour of the day |
| demand | No. of car rentals in a hour |

### Test set

**test.csv** 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.

|  |  |
| --- | --- |
| **Variable** | **Description** |
| date | Date (yyyy-mm-dd) |
| hour | Hour of the day |

### Evaluation metric

### The evaluation metric for this hackathon is RMSE score.

## Approach

Below are the 3 main steps followed for solving this problem statement:

1. Data pre-processing and Feature engineering
2. Searching for best machine learning algorithm
3. Final model

### Data pre-processing and Feature engineering

Below steps are carried out during this stage:

* Read the data.
* Create cartesian product of unique list of date and hour column. Create dataframe from this cartesian product. Merge demand from train dataset into this new training data considering date and hour as key.

This will ensure zero demand for all the entries which are not mentioned in training dataset.

* Convert date and hour as datetime variable.
* Generate new features from datetime. These features are:
  + Year, Month, day, and hour
  + Weekend
  + Week of year
  + Day of week. Perform One hot encoding on this feature and drop Day of week.
  + Year\_start, Year\_end, Month\_start, Month\_end, Quarter\_start, Quarter\_end
* Generate same features in test data as well.

## Searching for best machine learning algorithm

Below steps are performed for identifying best machine learning algorithm:

* Split training data into train and validation using time series split. For this. First 90% rows are considered as train dataset and rest dataset considered as validation dataset.
* Separate Features and target for both train and validation dataset.
* Identify best machine learning algorithm from below set and perform hyperparameter tuning of best model.

|  | **Model Name** | **train\_score** | **val\_score** |
| --- | --- | --- | --- |
| **9** | Random Forest Regressor | 22.43 | 37.19 |
| **10** | Extra Trees Regressor | 21.36 | 37.81 |
| **11** | XGBoost Regressor | 20.36 | 44.09 |
| **8** | Decision Tree Regressor | 22.22 | 44.41 |
| **5** | K-Neighbors Regressor | 27.92 | 44.55 |
| **0** | Linear Regression | 42.62 | 47.19 |
| **2** | Linear Regression with L2 regularisation | 42.62 | 47.22 |
| **7** | Support Vector Regressor with gaussian kernel | 41.41 | 47.33 |
| **4** | Poisson Regressor | 43.24 | 47.93 |
| **3** | Huber Regressor | 42.76 | 47.95 |
| **1** | Linear Regression with L1 regularisation | 42.73 | 47.96 |
| **6** | Linear Support Vector Regressor | 42.94 | 47.96 |

## Final model

After identifying best model, we retrained again on whole training dataset using Final tuned **Random Forest** model.

Random Forest Regressor found as most effective algorithm for this problem statement and below is the final hyper-parameter for this model.

| **Hyperparameter** | **value** |
| --- | --- |
| n\_estimators | 600 |
| criterion | squared\_error |
| max\_features | 5 |
| max\_depth | 14 |
| min\_samples\_leaf | 3 |
| min\_samples\_split | 7 |
| n\_jobs | -1 |
| random\_state | 123 |

## Importance of feature

We found that hour, week of year, Day, Month, Year, and weekend and features are most important features. We created this feature during feature engineering process as standard set of feature generation for time series data.

