Capstone Project
Taxi Demand Prediction

Team1

Great learning – PGPBDA

Introduction

- Taxi-supply planning requires efficient management of existing taxis and optimization of the decisions concerning additional capacity.

- Demand prediction is an important aspect in the development of any model for taxi planning. The predictions can help in optimizing taxi supply at a given location and time.

- The form of the demand depends on the type of planning and accuracy that is required; like daily, weekly etc.

- In the short run, the taxi demand is mainly influenced by seasonal effects (daily and weekly cycles, calendar holidays) and special events. Weather related variation is certainly critical in predicting taxi demand for lead times beyond a day ahead.
- Prediction using 24 weeks data for the New York city. We consider lead times up to 1 week ahead.

Dataset Details

- Data Set: 2015 Yellow Taxi Trip Data(City of New York, Jan 2015-Jun 2015)

- Actual Data Size: 10.8 GB, 78 Mn rows (Each row represents 1 taxi pickup)

- Data Sources:
  - Taxi Pickups Data: https://data.cityofnewyork.us/Transportation/2015-Yellow-Taxi-Trip-Data/ba8s-jw6u
  - Hourly Weather Data: https://www.wunderground.com/history/airport/KJFK/2015/12/1/MonthlyHistory.html

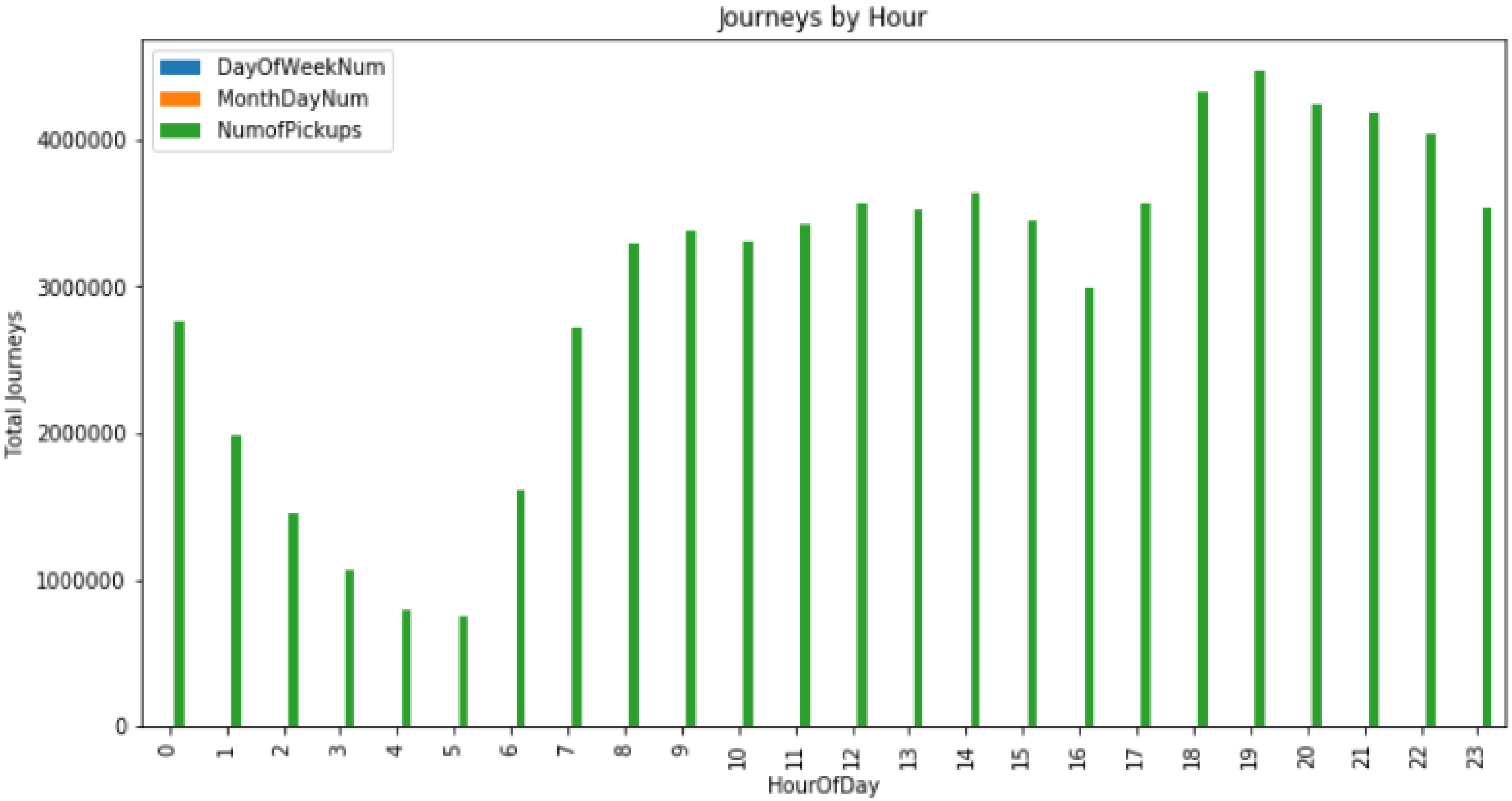
Data Attributes Used

- Although dataset has attributes that capture various details about the Taxi trip, only 3 attributes were used in model building:
  - Pickup time stamp
  - pickup_latitude
  - Pickup_longitude

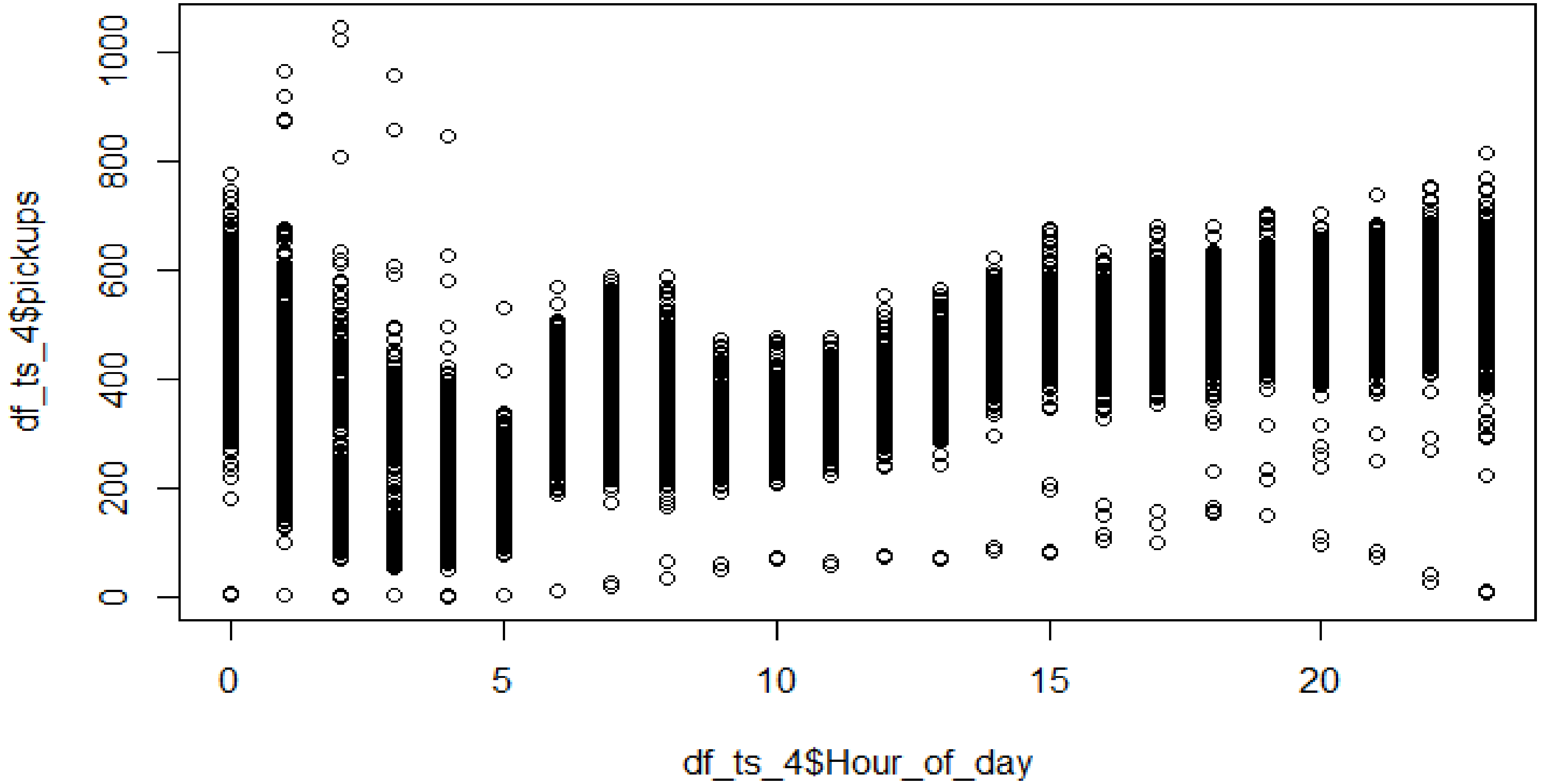
Data Dictionary

http://www.nyc.gov/html/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf

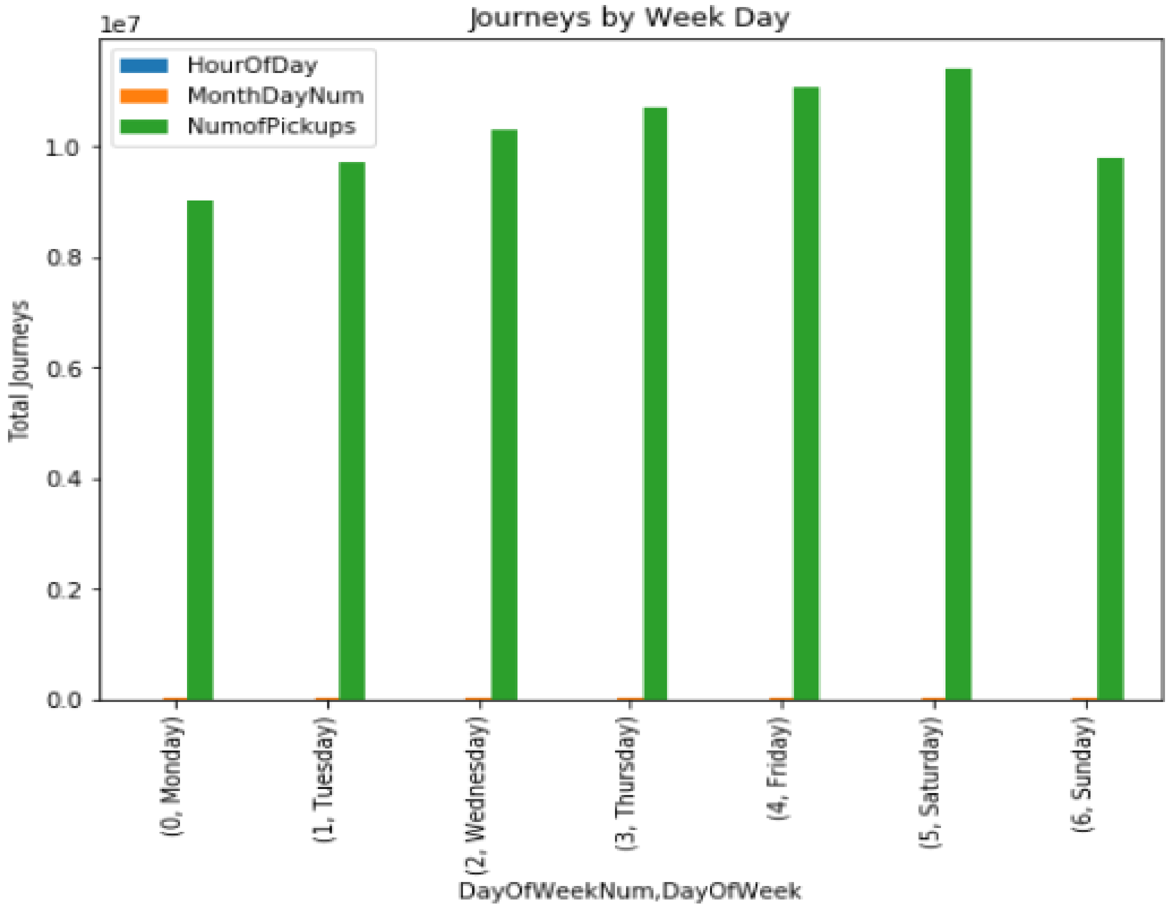
Exploratory Data Analysis –Hourly Demand Trend



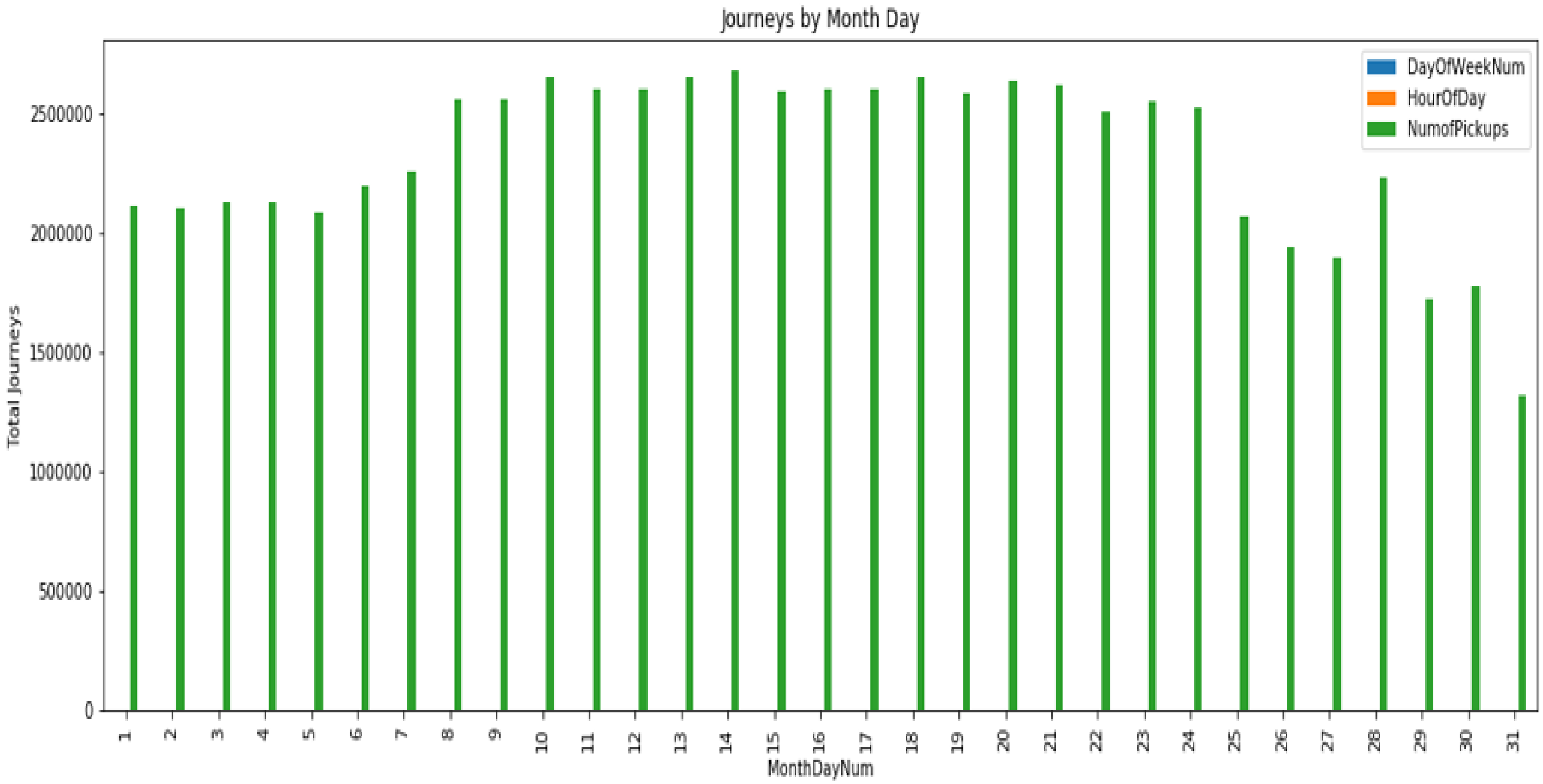
Hourly variation of Pickups



Demand Analysis on Week Day basis



Demand Analysis on Month Day basis

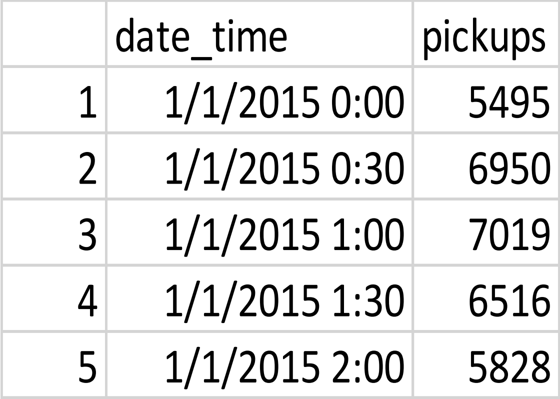


Data Preparation Steps

- Data Imputation: Replace the missing values by average of pickups at 1 previous and 1 ahead time stamps
- Round the location (latitude and longitude) to 1 decimal place.
- Group the data based on concatenated latitude and longitude. This will result in 4 datasets one for each combination of latitude and longitude
- Replace outliers by 5th or 95th percentile using boxplot
- Group the data for each time stamp in each dataset and count rows as no of pickups for that timestamp
- Aggregate taxi pickups for each dataset on half hourly basis

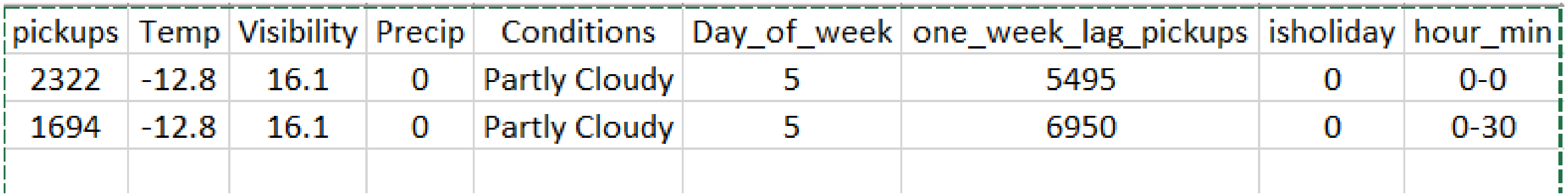
Final Data Set

The dataset prepared is univariate data used for time series models and LSTM model



	date_time	pickups
1	1/1/2015 0:00	5495
2	1/1/2015 0:30	6950
3	1/1/2015 1:00	7019
4	1/1/2015 1:30	6516
5	1/1/2015 2:00	5828

To apply Supervised learning models further data enrichment followed by feature engineering is performed.



pickups	Temp	Visibility	Precip	Conditions	Day_of_week	one_week_lag_pickups	isholiday	hour_min
2322	-12.8	16.1	0	Partly Cloudy	5	5495	0	0-0
1694	-12.8	16.1	0	Partly Cloudy	5	6950	0	0-30

Data Division

23 weeks of data has been used to train all the models and 24th week pickups have been used as test dataset.

Because of half hourly aggregation of data

- Lag 24 is equivalent to 12 hours
- Lag 48 is equivalent to 24 hours
- Lag 336 is equivalent to 1 week

Short Term Univariate Prediction

Algorithms Used

Supervised Learning Models

- Linear Regression
- Random Forest
- Conditional Inference Decision tree
- Conditional Inference Random Forest
- XG Boost

Time Series Models

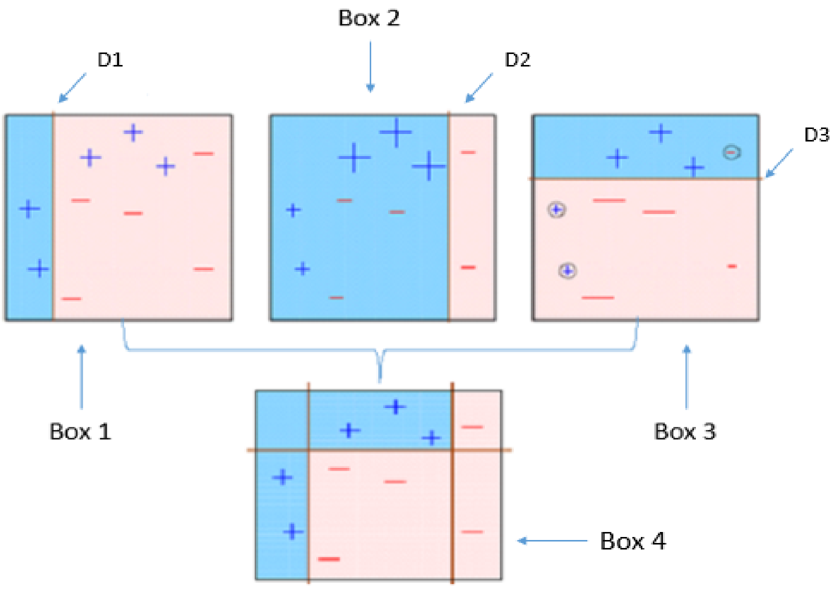
- ARIMA with Fourier terms for seasonality
- Holt – Winter

Deep Learning Models

- LSTM – A type of RNN

XG Boost

Boosting is a sequential process; i.e., trees are grown using the information from a previously grown tree one after the other. This process slowly learns from data and tries to improve its prediction in subsequent iterations. Let's look at a classic classification example:



Four classifiers (in 4 boxes), shown above, are trying hard to classify + and - classes as homogeneously as possible. Let's understand this picture well.

1. **Box 1:** The first classifier creates a vertical line (split) at D1. It says anything to the left of D1 is + and anything to the right of D1 is -. However, this classifier misclassifies three + points.
2. **Box 2:** The next classifier says don't worry I will correct your mistakes. Therefore, it gives more weight to the three + misclassified points (see bigger size of +) and creates a vertical line at D2. Again it says, anything to right of D2 is - and left is +. Still, it makes mistakes by incorrectly classifying three - points.
3. **Box 3:** The next classifier continues to bestow support. Again, it gives more weight to the three - misclassified points and creates a horizontal line at D3. Still, this classifier fails to classify the points (in circle) correctly.
4. Remember that each of these classifiers has a misclassification error associated with them.
5. Boxes 1,2, and 3 are weak classifiers. These classifiers will now be used to create a strong classifier Box 4.
6. **Box 4:** It is a weighted combination of the weak classifiers. As you can see, it does good job at classifying all the points correctly.

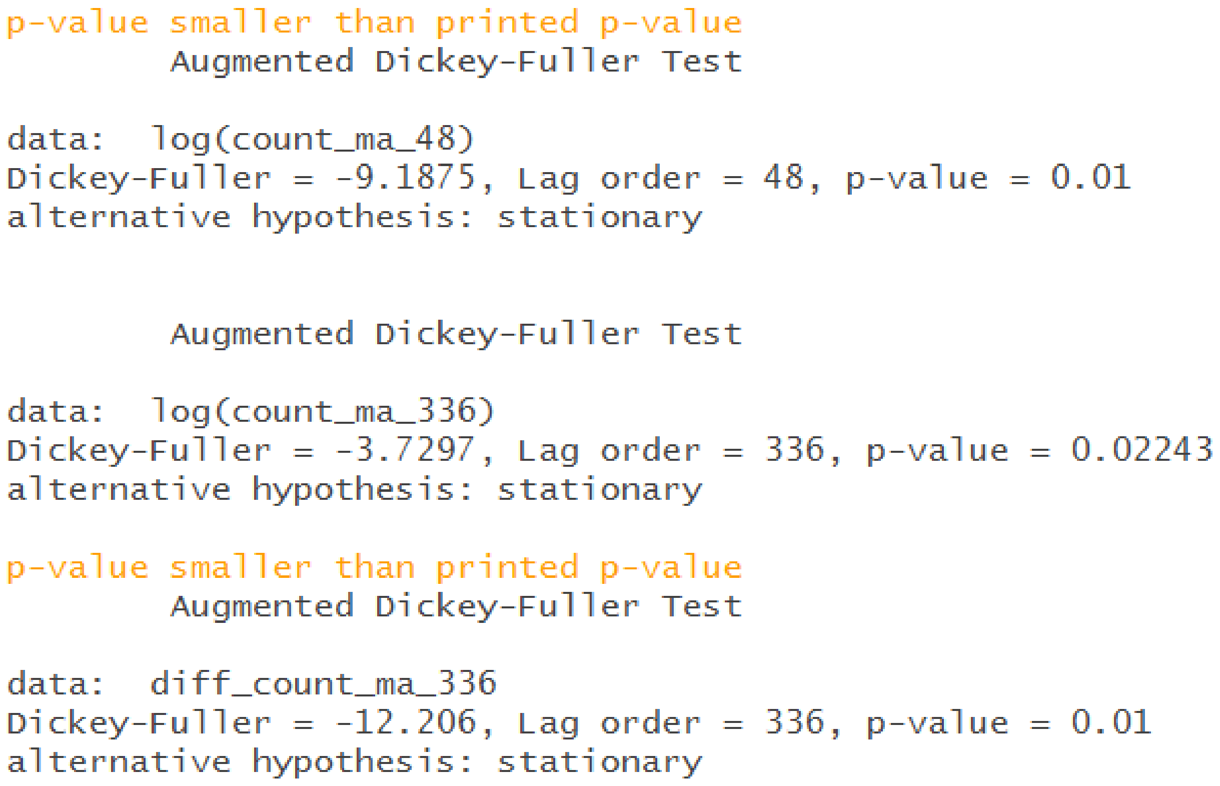
Seasonality Determination using Periodogram

period	spec
48.00000	23656012219
24.00000	7708466828
25.86826	1510237251
332.30769	998244094

ARIMA

- ARIMA model requires time series to be stationary
- Differencing the series once made it stationary

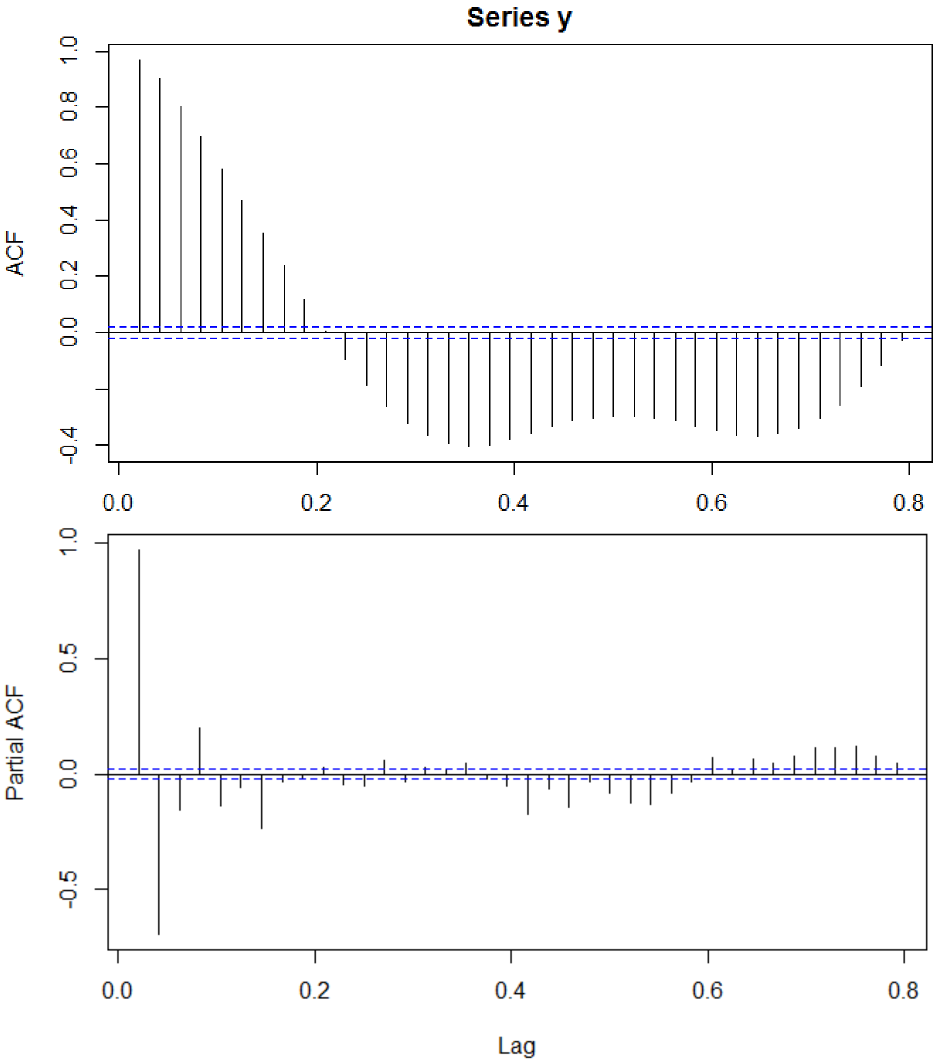
Time Series is stationary at lag 48, but not at 336.



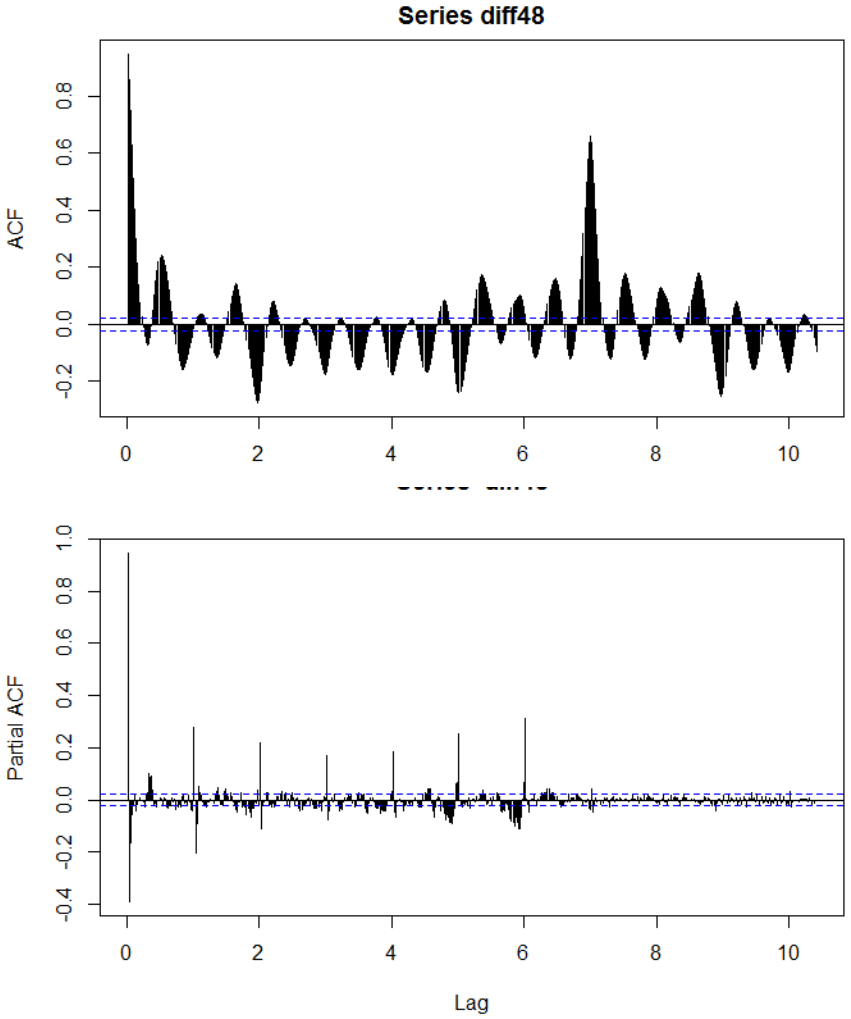
```
p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  log(count_ma_48)
Dickey-Fuller = -9.1875, Lag order = 48, p-value = 0.01
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  log(count_ma_336)
Dickey-Fuller = -3.7297, Lag order = 336, p-value = 0.02243
alternative hypothesis: stationary

p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  diff_count_ma_336
Dickey-Fuller = -12.206, Lag order = 336, p-value = 0.01
alternative hypothesis: stationary
```

ACF and PACF plots for 1st time series to identify non Seasonal and Seasonal AR and MA factors

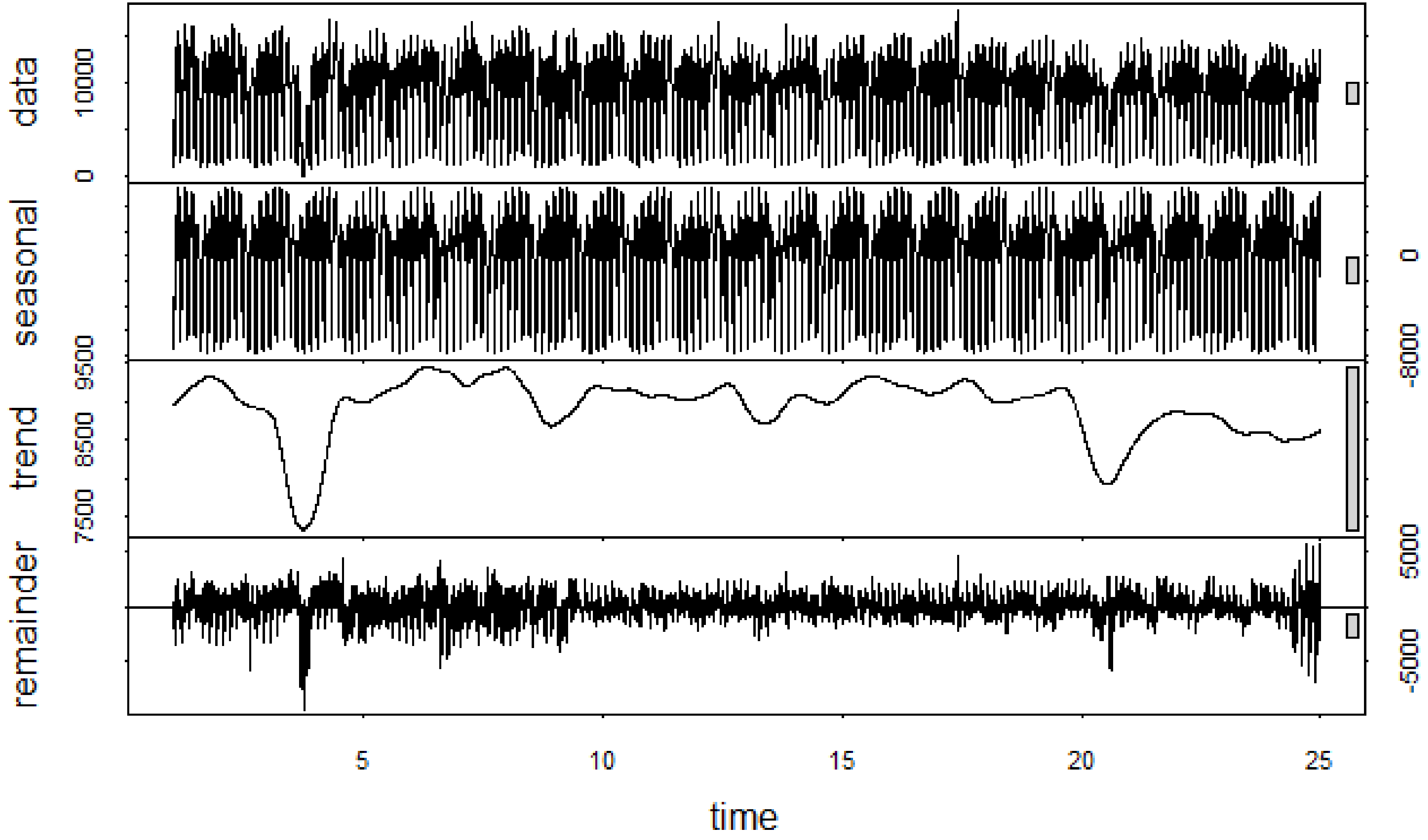
Trend – (p,d,q) = (7,0,0)



Seasonality – (P,D,Q) = (7,0,0))



Decomposition of Time Series for lag 336



Triple Exponential Smoothing/Holt-Winters Method

- The idea behind triple exponential smoothing is to apply exponential smoothing to the seasonal components in addition to level and trend.
- The smoothing is applied across seasons, e.g. the seasonal component of the 3rd point into the season would be exponentially smoothed with the the one from the 3rd point of last season, 3rd point two seasons ago, etc.
- In math notation we now have four equations

- $\ell x=\alpha(yx-sx-L)+(1-\alpha)(\ell x-1+bx-1)$ - Level
- $bx=\beta(\ell x-\ell x-1)+(1-\beta)bx-1$ - trend
- $sx=\gamma(yx-\ell x)+(1-\gamma)sx-L$ - Seasonal
- $y\hat{}x+m=\ell x+mbx+sx-L+1+(m-1)$ - Forecast

Holt Winter – Parameter Tuning

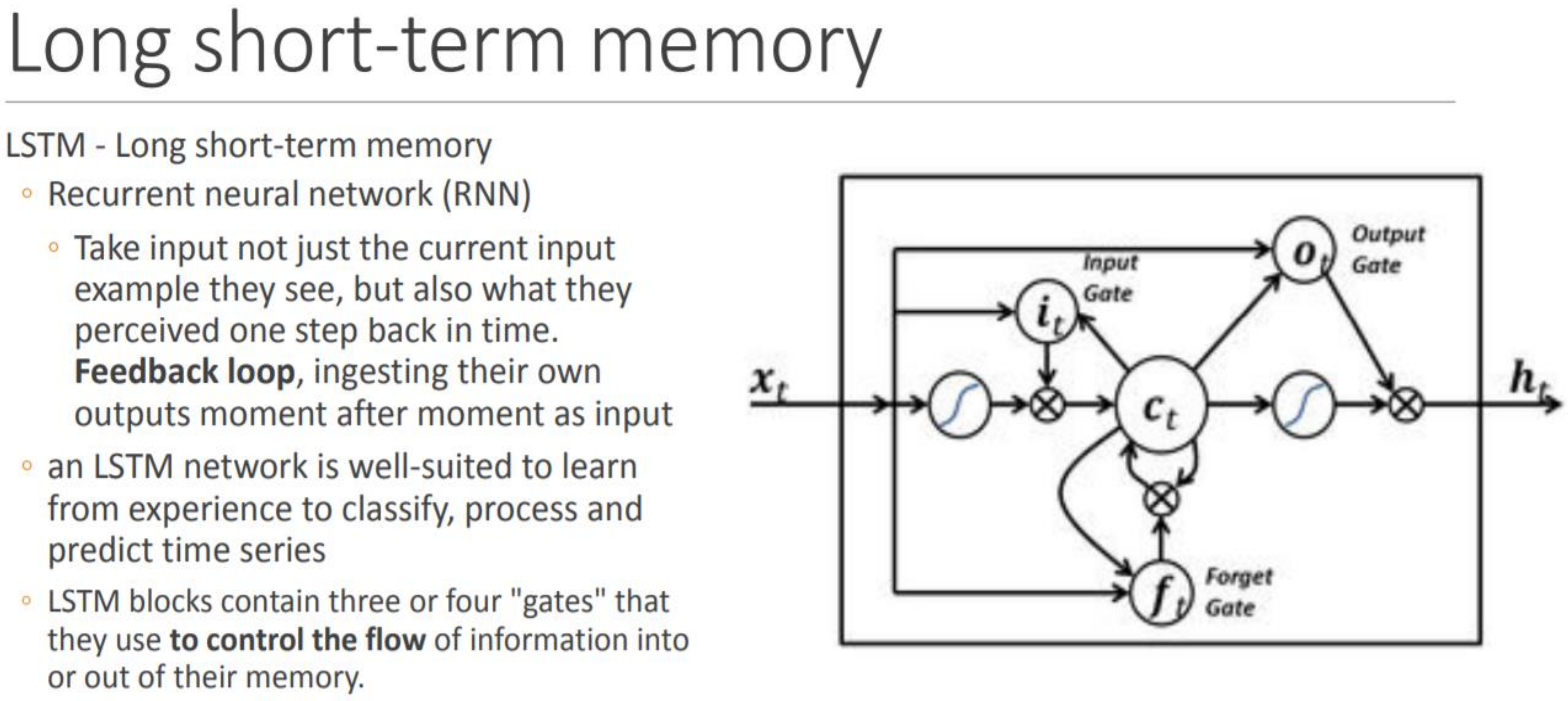
To optimize alpha(level), beta(trend) and gamma(seasonality) parameters in Holt Winter model, 3 nested loops were used as below:

```
y = train time series
For alpha in 0.1 to 1
  For beta in 0 to 1
    For gamma in 0 to 1
            model = HoltWinters(y, alpha,beta,gamma,start.periods = 336)
            forecast = predict 336 (1 week ahead values) using computed model
            result = combine forecast and test data
            calculate MAPE
      If computed MAPE < last MAPE then accept the solution
```

Conditional Inference Decision Tree

- Conditional inference trees estimate a regression relationship by binary recursive partitioning in a conditional inference framework.
- Advantage: avoids the variable selection bias of CART algorithms: They tend to select variables that have many possible splits
- The algorithm works as follows:
  - Test the global null hypothesis of independence between any of the input variables and the response (which may be multivariate as well).
  - Stop if this hypothesis cannot be rejected.
  - Otherwise select the input variable with strongest association to the response.
  - This association is measured by a p-value corresponding to a test for the partial null hypothesis of a single input variable and the response.
  - Implement a binary split in the selected input variable.
  - Recursively repeat the above steps 1 and 2.

LSTM – A type of Recurrent Neural Network



Long short-term memory

LSTM - Long short-term memory

- Recurrent neural network (RNN)
  - Take input not just the current input example they see, but also what they perceived one step back in time. **Feedback loop**, ingesting their own outputs moment after moment as input
- an LSTM network is well-suited to learn from experience to classify, process and predict time series
- LSTM blocks contain three or four "gates" that they use **to control the flow** of information into or out of their memory.

LSTM Network for Regression

- To model LSTM data is re-arranged in the following manner
  - Every value starting from position 337 is put into 1-D array. This is Y-variable
  - 336 values prior to the value added to Y-variable are added to X variable
  - X variable is an array with dimensions 8046,336
  - Y variable is an array with dimensions 336,336
  - Layers used 1 input, a hidden layer with 4 LSTM blocks and an output layer that makes a single value prediction, followed by a dense layer
  - The default sigmoid activation function is used for the LSTM blocks. The network is trained for 20 epochs and a batch size of 1 is used.

Random Forest

- Random forest algorithm can be used for both classification and regression problems.
- Random Forest creates forest with number of trees.
- Advantages of Random Forest are:

1. Same forest classifier can use for both classification and the regression task
2. Random forest classifier will handle the missing values
3. Random forest won’t overfit the model
4. Can model the random forest classifier for categorical values also

Forecasting Accuracy

- Let Y_t is the actual value of Y at time t and F_t is the corresponding forecasted value
- Assume that there are n (for example $n = 100$) observations in total
  - Mean absolute error(MAE)

    $$MAE = \frac{1}{n}\sum_{t=1}^{n} |Y_t \ - F_t|$$

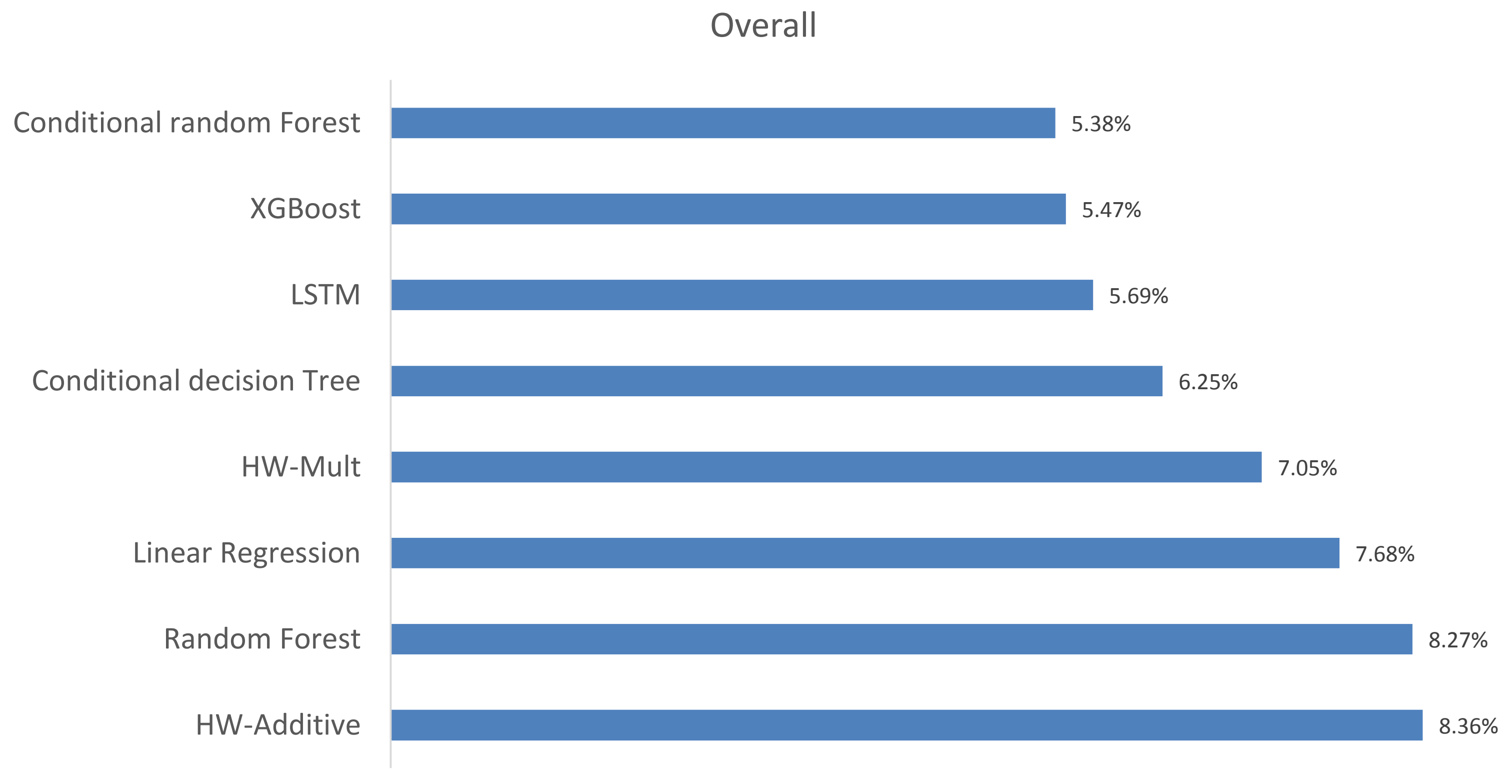
    - MAE is the average absolute error and should be calculated on the test data set
  - Mean absolute percentage error(MAPE)

    $$MAPE = \frac{1}{n}\sum_{t=1}^{n} \frac{|Y_t - F_t|}{Y_t} \times 100\%$$

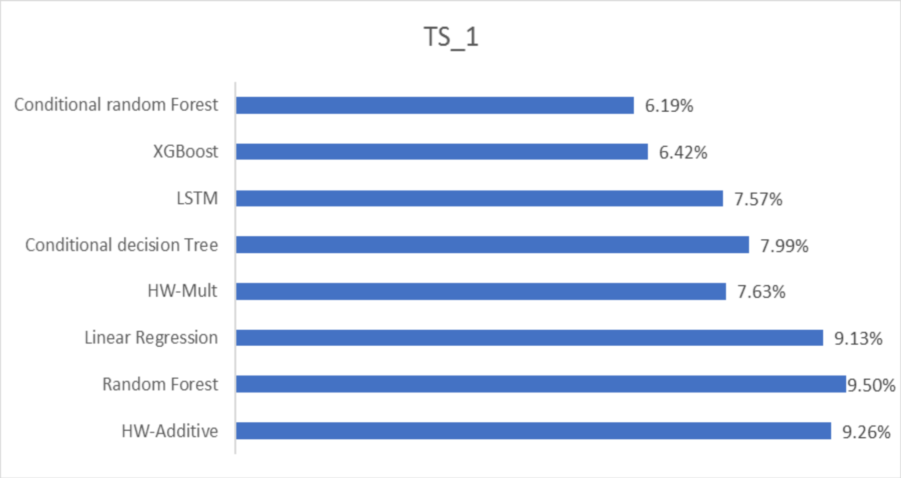
    - MAPE is one of the popular forecasting accuracy measures used by practioners since it expresses the average error in percentage terms and is easy to interpret
    - Since MAPE is dimensionless it can be used for comparing different models with varying scales

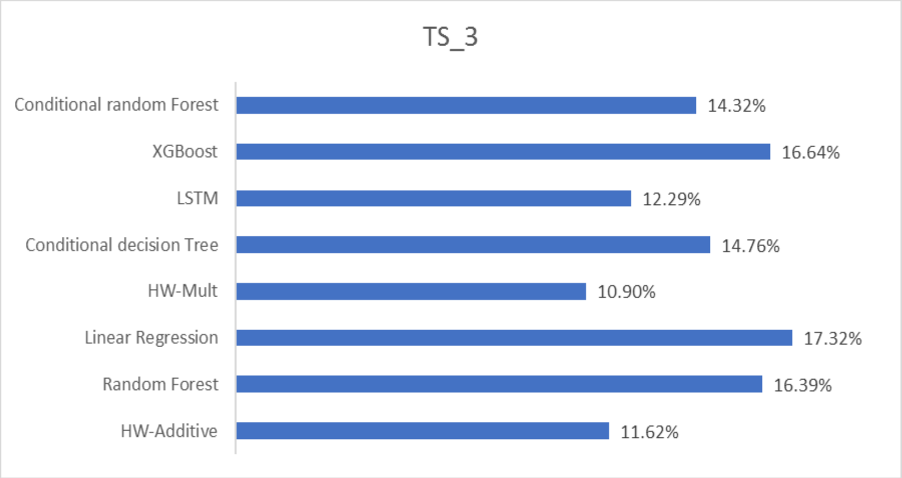
Performance Comparison of Different Models Used

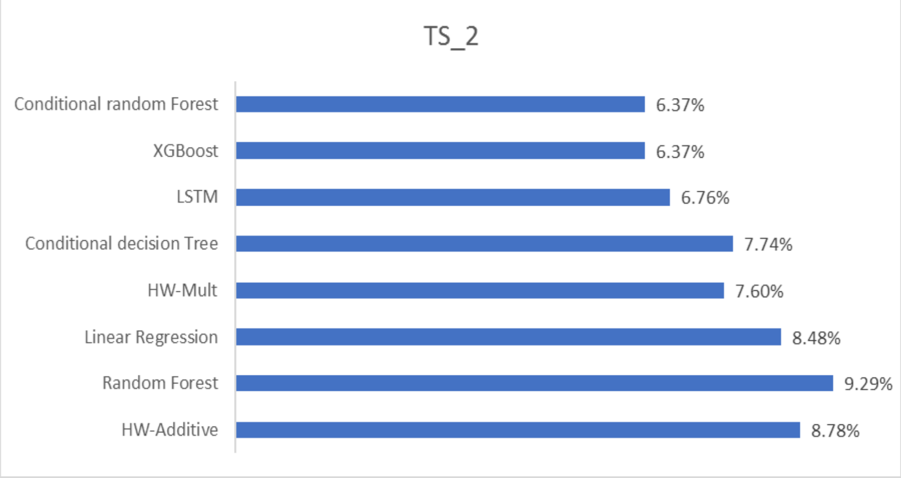
Conditional RF outperforms XGBoost and LSTM in terms of MAPE

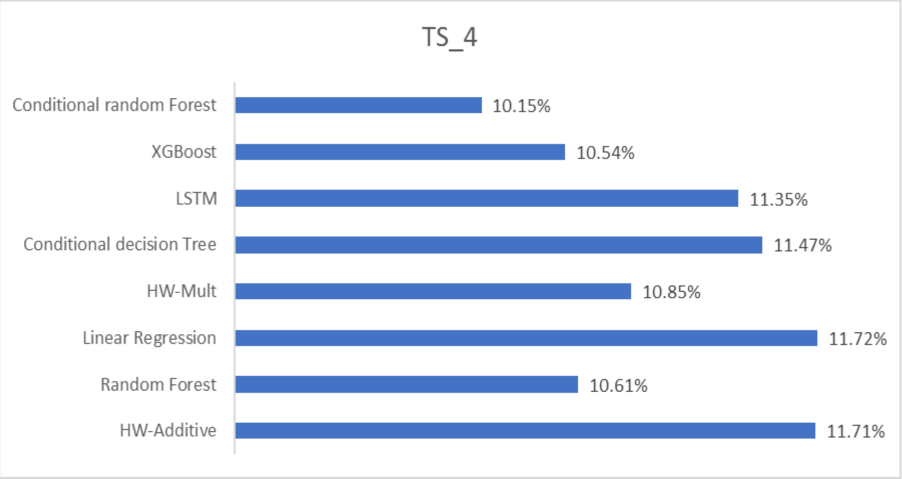


MAPE for Individual Time Series









Forecasting Accuracy

- Mean squared error(MSE)

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t - F_t)^2$$

  - Lower MSE implies better prediction
  - However, it depends on the range of the time-series data
- Root mean square error(RMSE)

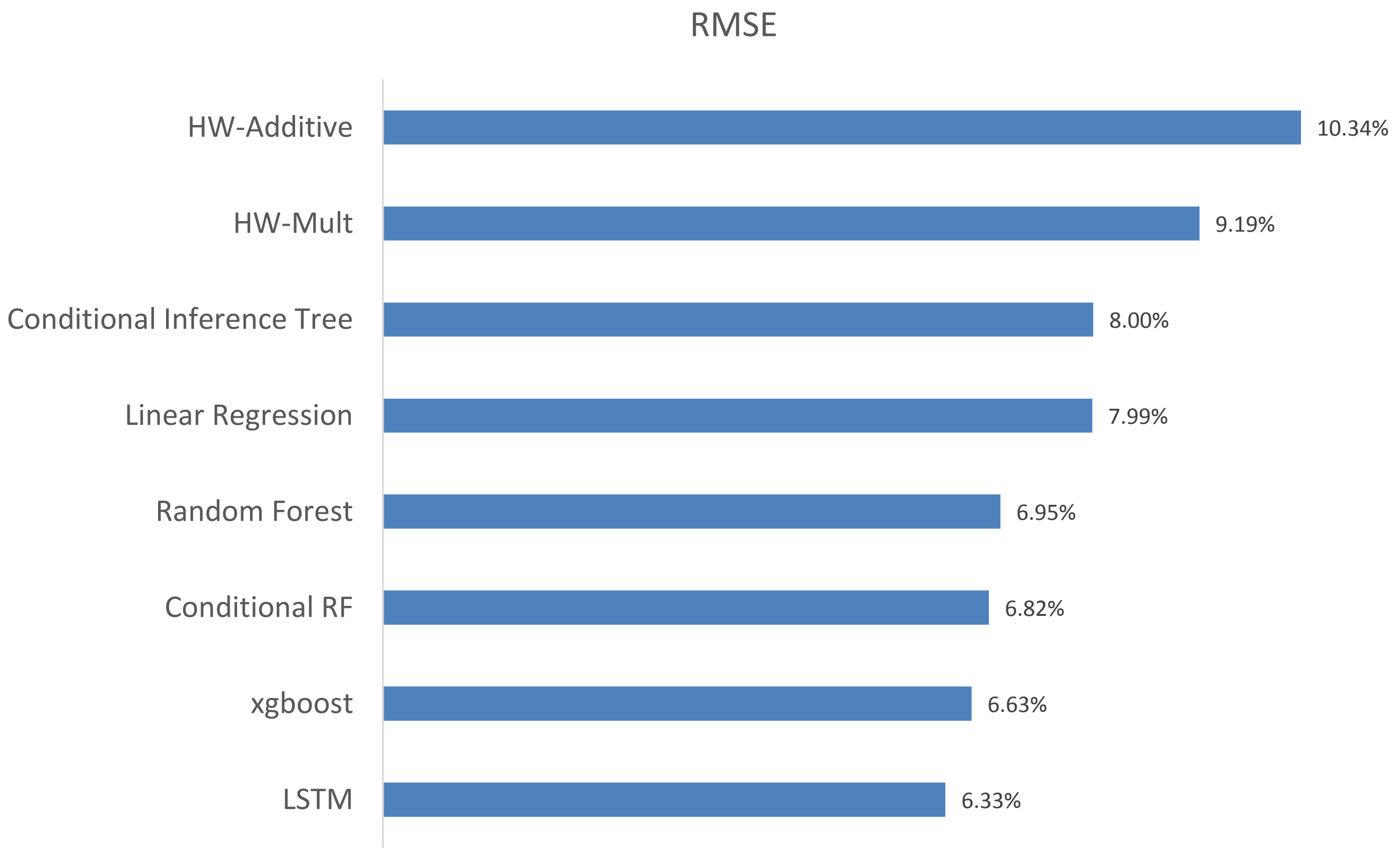
$$\text{RMSE} = \sqrt{(\frac{1}{n}\sum_{t=1}^{n}(Y_t - F_t)^2)}$$

  - RMSE and MAPE two most popular accuracy measures of forecasting
  - RMSE is the standard deviation of errors or residuals

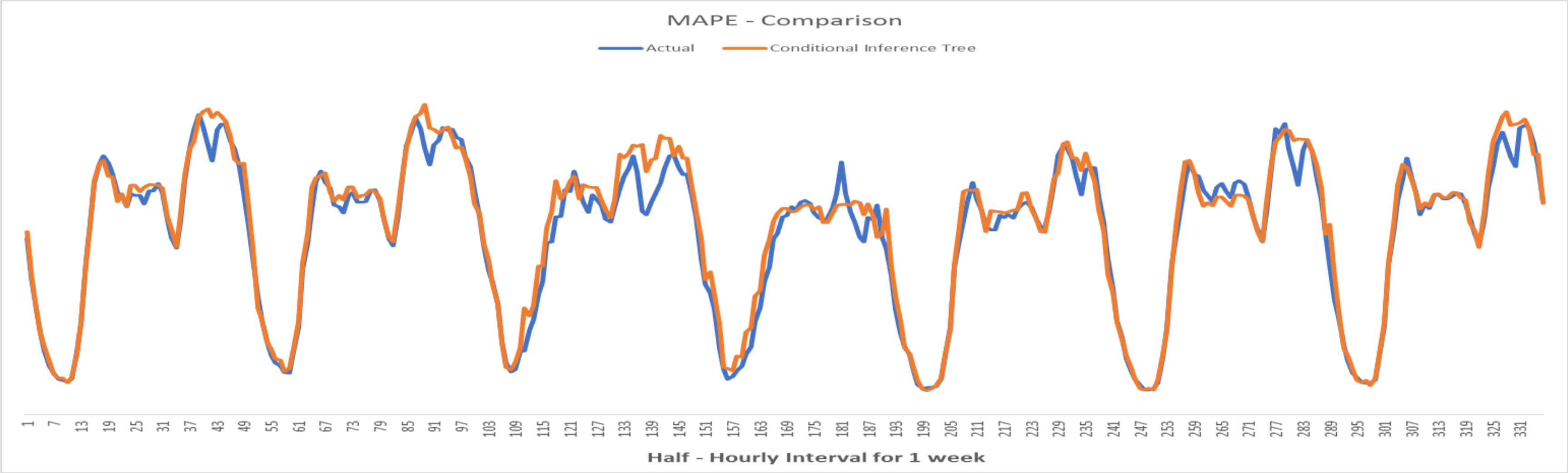
- Example: In 2006, Netflix, the movie portal, announced a competition with a prize money worth one million dollars to predict the rating on a 5-point scale likely to be given a customer for a movie. The participants were given a target RMSE of 0.8572 to qualify for the prize (source: https://en.wikipedia.org/wiki/Netflix_Prize)

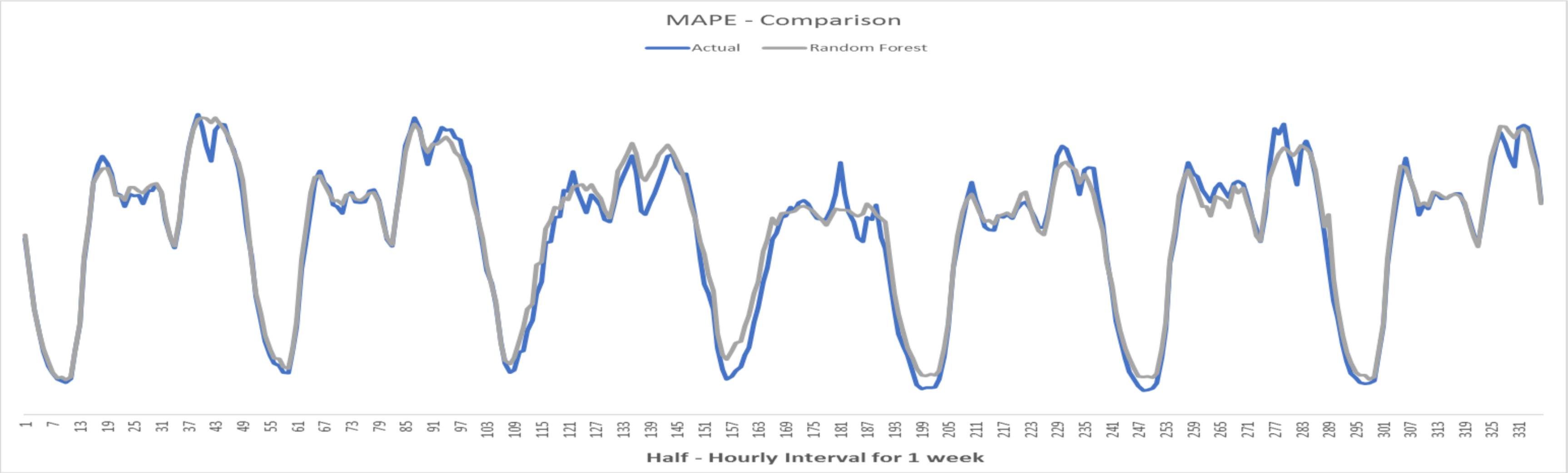
Overall RMSE

LSTM outperforms XGBoost and Conditional RF in terms of RMSE

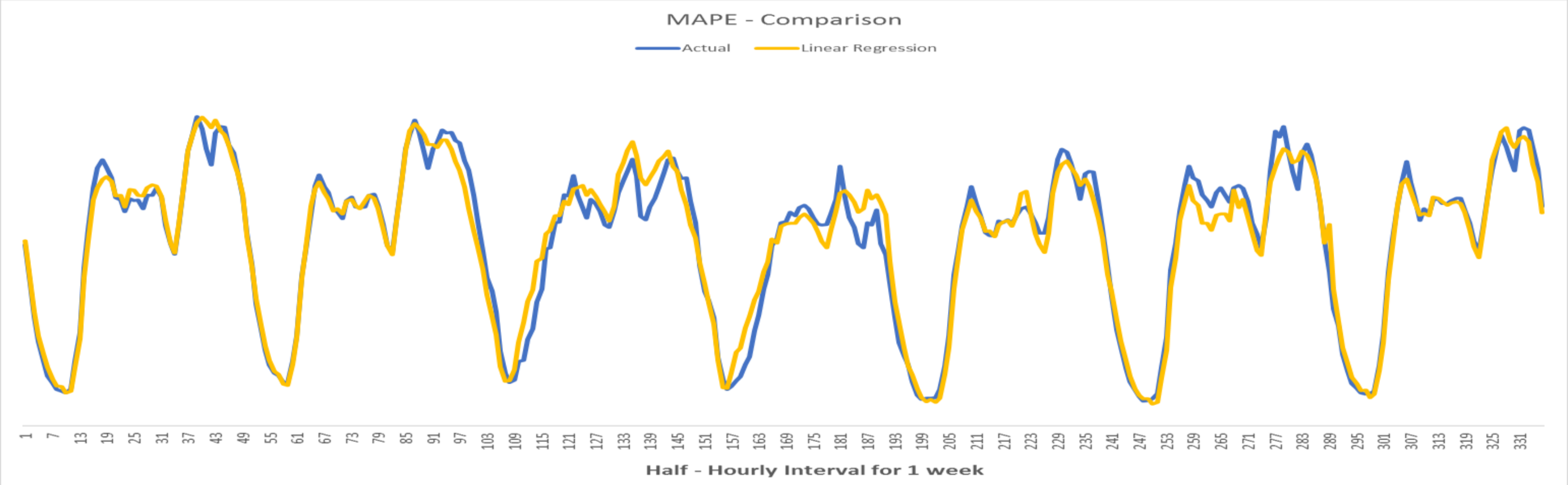


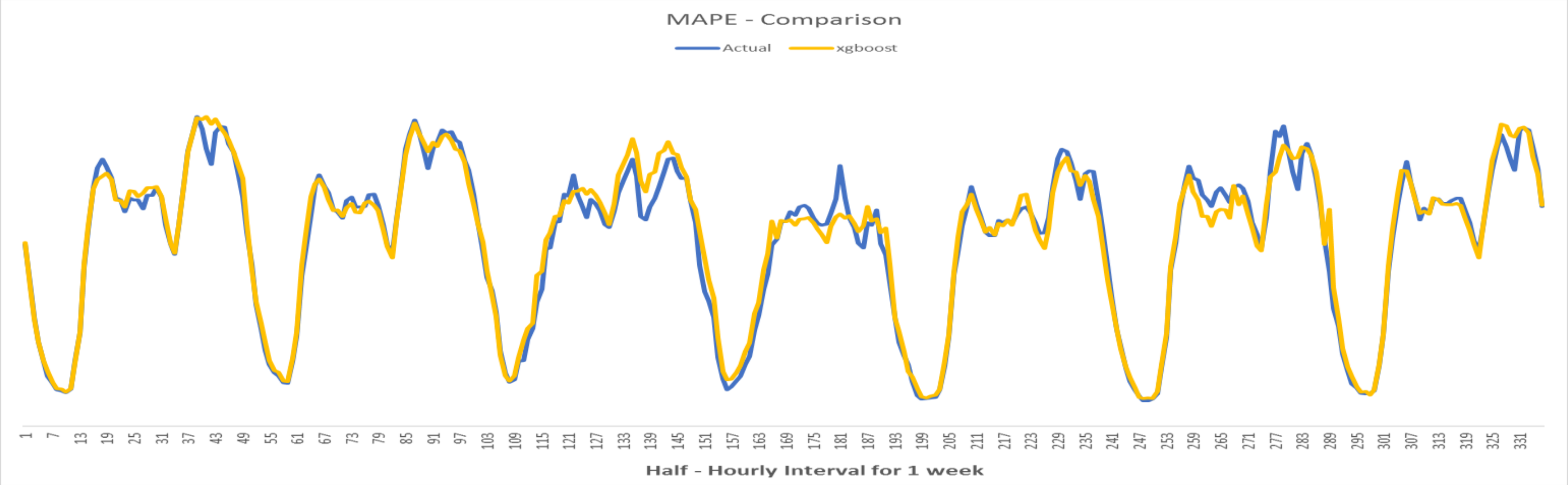
MAPE Comparison on Test Dataset



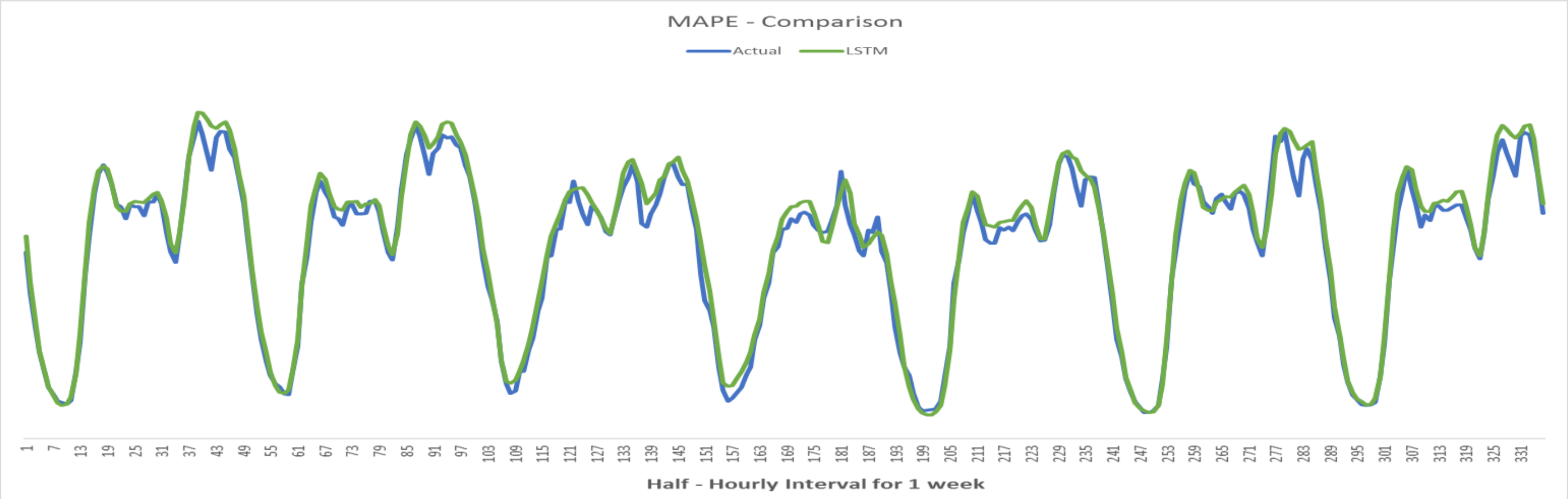


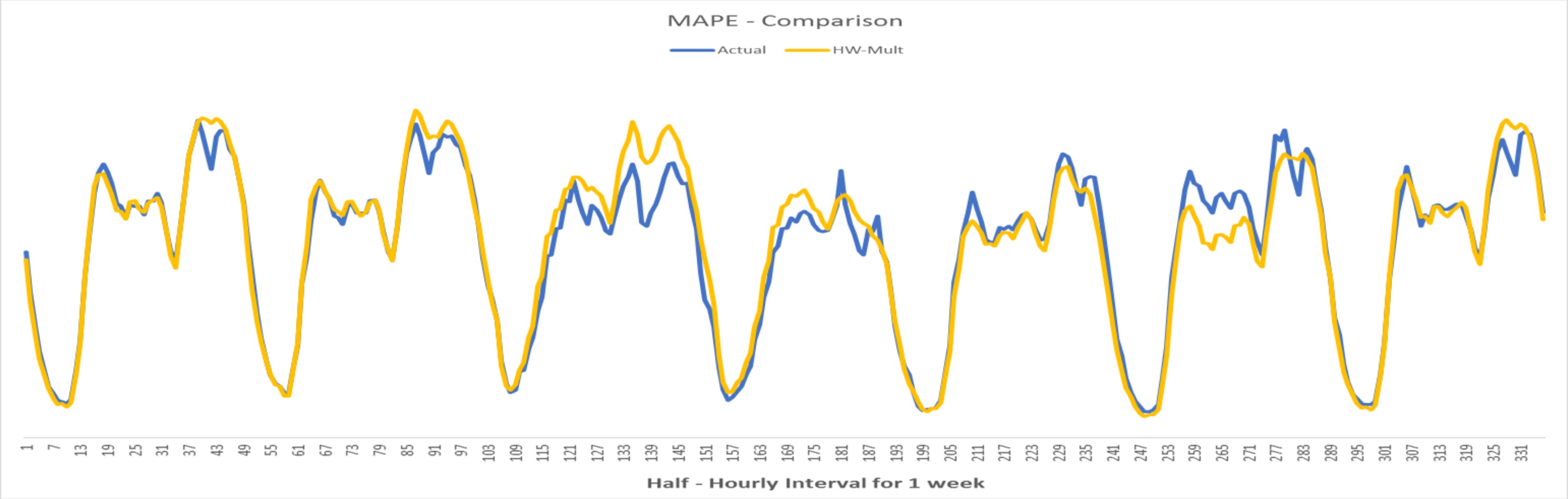
MAPE Comparison on Test Dataset



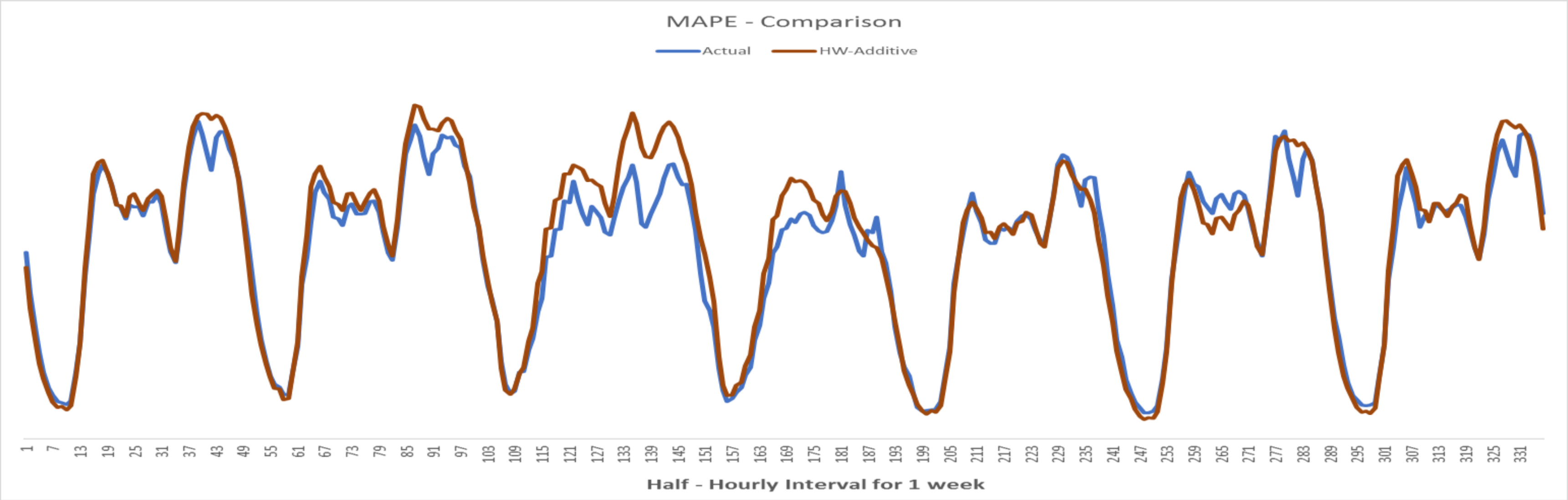


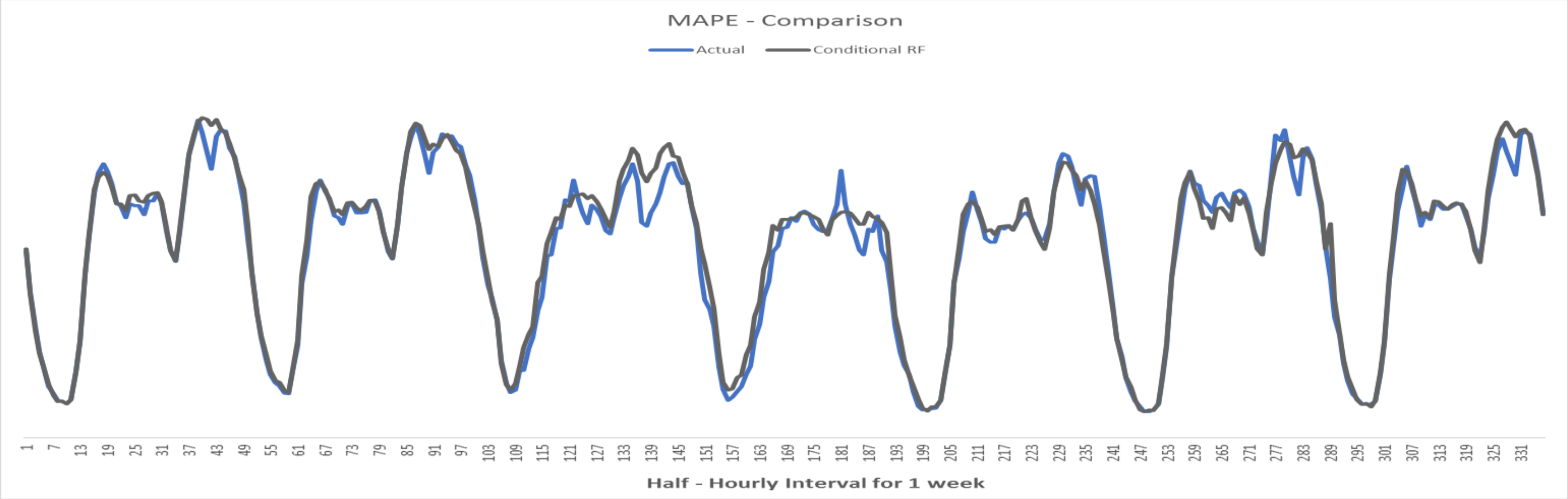
MAPE Comparison on Test Dataset



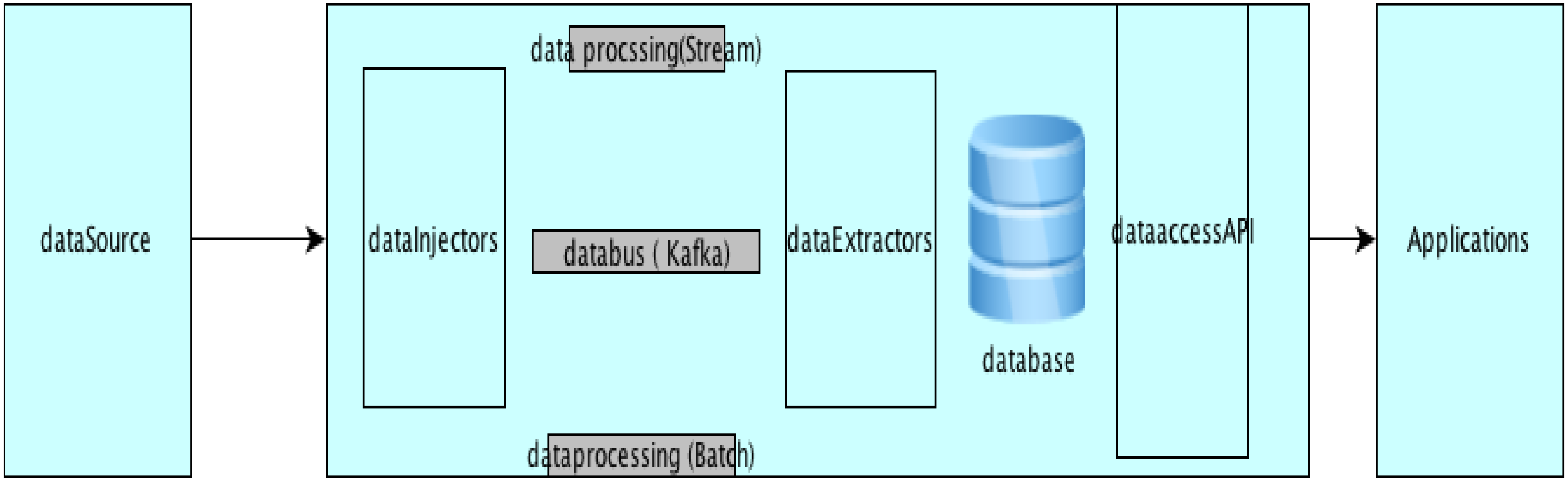


MAPE Comparison on Test Dataset





Integration into Production System



Companies which may benefit of this Algorithm