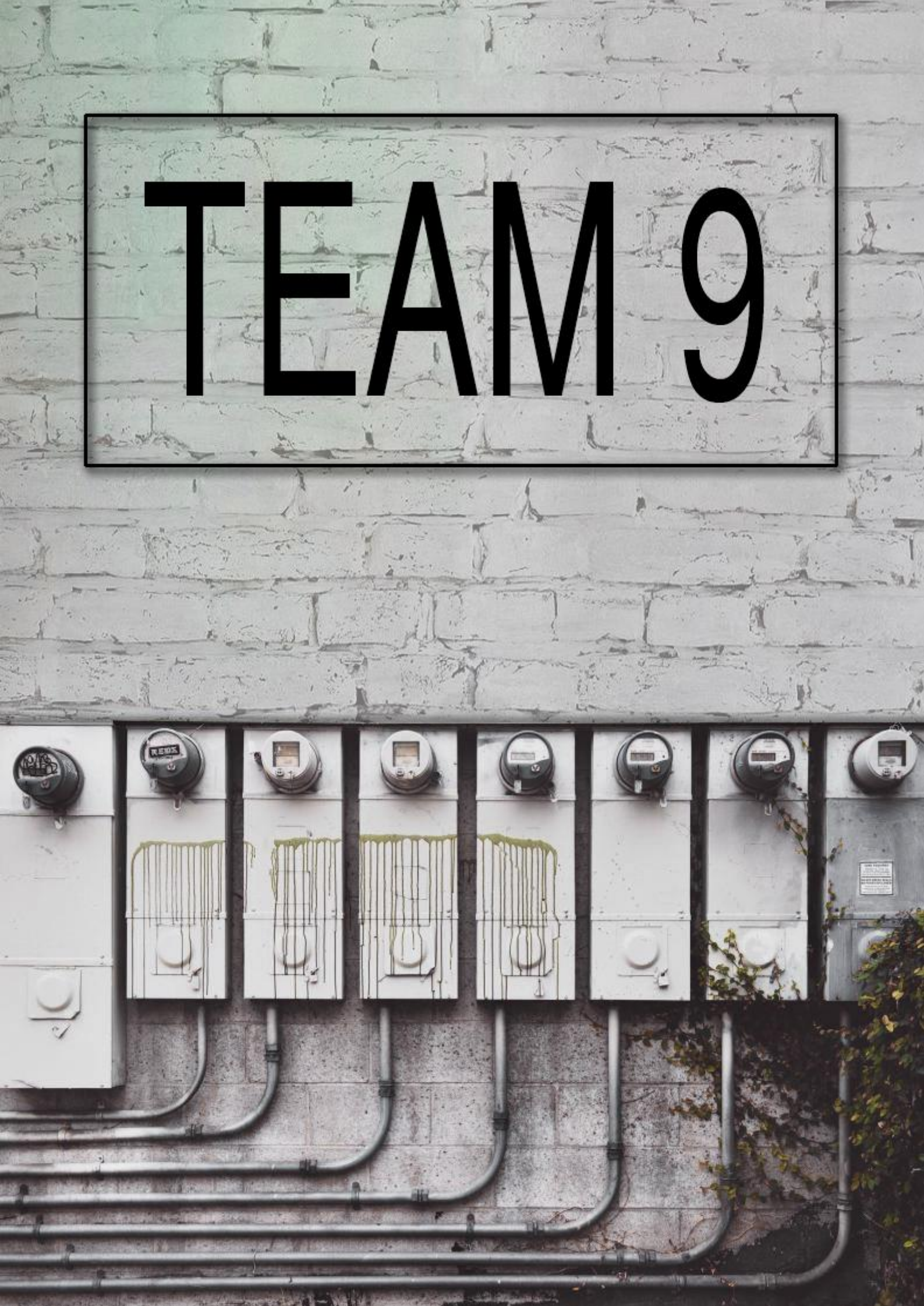

TEAM 9

INDEX

INTRODUCTION

Most of the real-life problems are based upon the prediction of future which is totally oblivious to us such as stock market prediction, future sales prediction and so on.

Time series problem is basically the prediction of such problems using various machine learning tools.Time series problem is tackled efficiently when first it is analyzed properly (Time Series Analysis) and according to that observation suitable algorithm is used (Time Series Forecasting).

The report outlines the analysis of electricity consumption of 5 corporate buildings located in Delhi NCR.

The objective of the analysis is to determine the future demands of electricity so that the overproduction of electricity may be reduced and the resources can be used sustainably.

The dataset was found out to be stationary using Dickey Fuller Test. We separated the dataset according to the buildings and each dataset was passed on to different forecasting models like ARIMA, Simple exponential smoothing, Holt Winter Exponential Smoothing etc..

DATA DESCRIPTION

The dataset contains the value of electricity consumption of 5 buildings across Delhi-NCR noted in 3 different electrical meters during the year 2017.

We are trying to predict the future consumption of electricity in those buildings so that we can reduce the overproduction of electricity and the resources required for producing electricity can be used elsewhere.

The training data is a pretty balanced dataset containing approximately 26,400 entries for each building.

The evaluation metric is given in such a way to weight down the predictions farther in the future, giving more priority to the initial days.

TESTING STATIONARITY OF TIME SERIES DATA

Dickey Fuller Test

In statistics, the **Dickey–Fuller test** tests the null hypothesis is that a unit root is present in an autoregressive model. The alternative hypothesis is stationarity.

p >= 0.05 process is not stationary.

p < 0.05 process is stationary and null is rejected.(Dataset: building 1, main_meter,log_transformation)

Results of Dickey-Fuller Test:	
Test Statistic	-31.375553
p-value	0.000000
#Lags Used	45.000000
Number of Observations Used	26350.000000
Critical Value (1%)	-3.430598

Kpss Test

The KPSS test, short for, Kwiatkowski-Phillips-Schmidt-Shin (KPSS), is a type of Unit root test that tests for the stationarity of a given series around a deterministic trend.

If p-value is < signif level (say 0.05), then the series is non-stationary. (Dataset: building 1, main_meter,log_transformation)

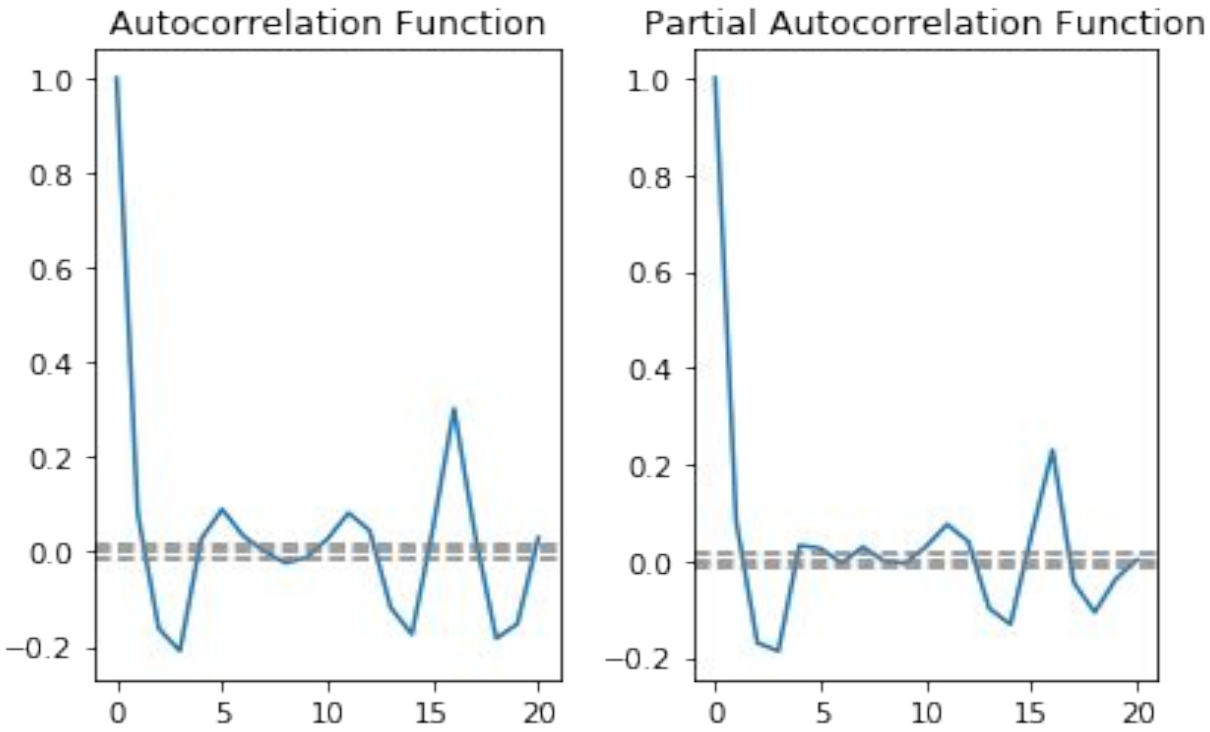
KPSS Statistic: 0.004009712611338216
p-value: 0.1
num lags: 45
Critical Values:
10% : 0.347
5% : 0.463
1% : 0.739
Result: The series is stationary

EVALUATING ORDER OF ARIMA

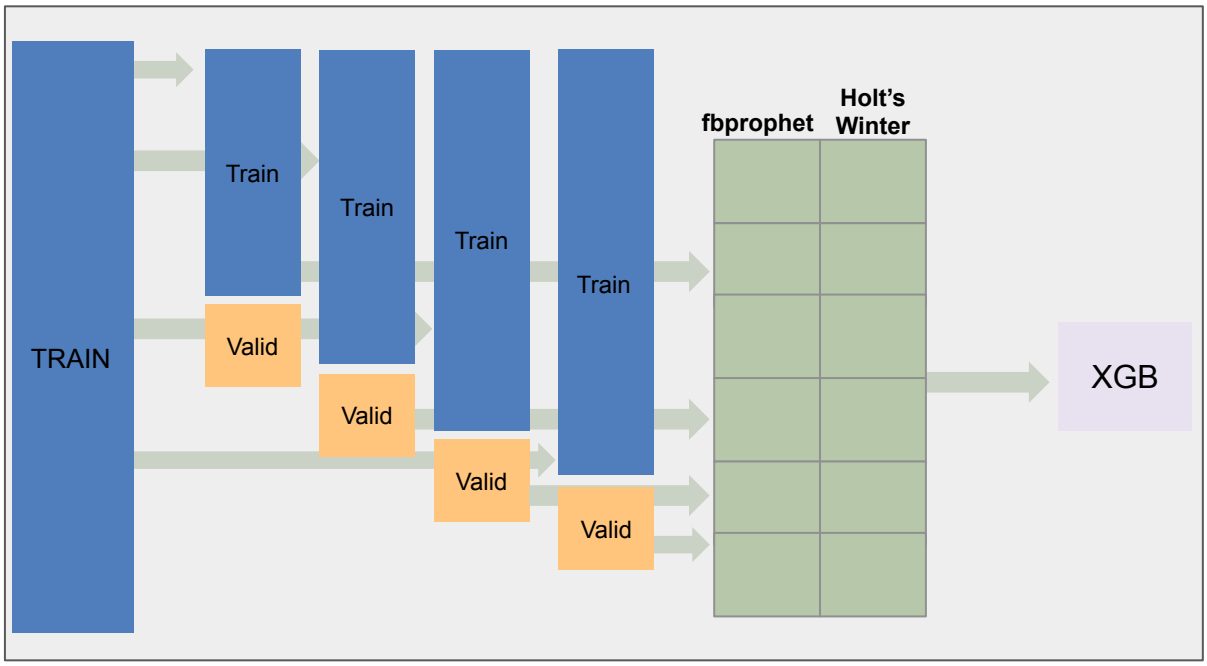
pacf plot:
The **PACF plot** is a **plot** of the partial correlation coefficients between the series and lags of itself. We used the pacf test to determine the value of order of Autoregressive(AR) process i.e. p.(building 1 main_meter)
In the example given below the lag value for which the chart crosses the upper confidence interval for the first time is 1, so we get p=1

acf plot:
The plot of autocorrelation in of a time series data and the lags of itself is known as the Autocorrelation Function. We used the acf plot to determine the value of order of Moving Average(MA) process i.e. q. In the example given below the lag value for which the chart crosses the upper confidence interval for the first time is 1, so we get q=1

We used these values of p and q from the pacf and acf plot as parameters while using ARIMA model..

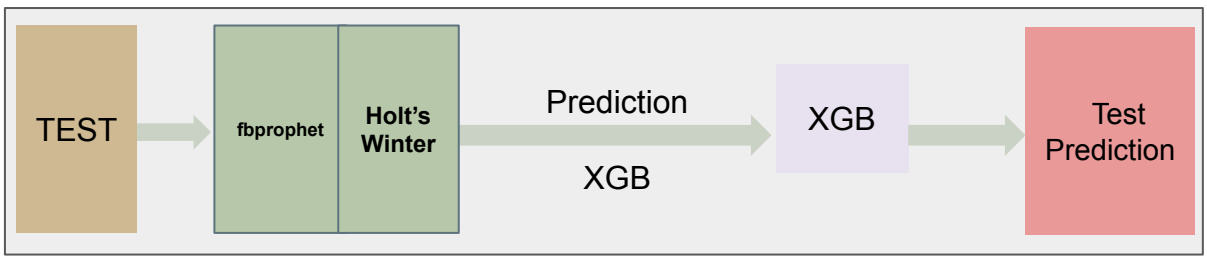


MODEL ARCHITECTURE



k-fold cross validation training for base models

Training for stacking regressor



Prediction basel models

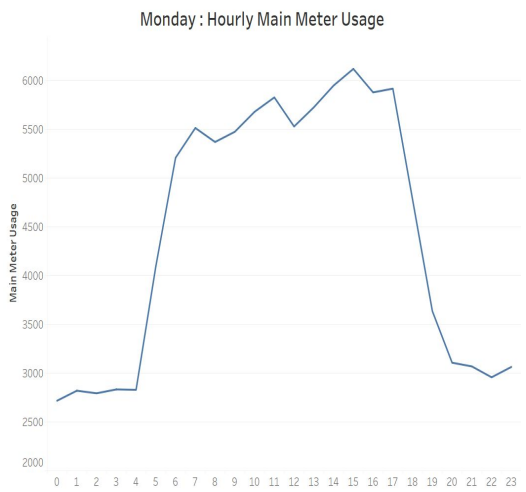
Prediction stacking regressor

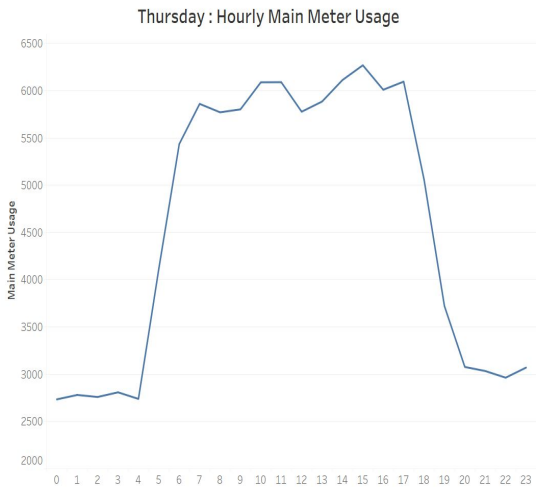
EXPLORATORY DATA ANALYSIS

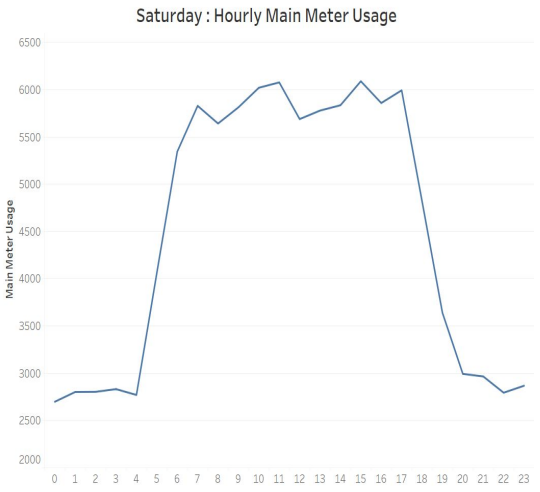
In primary EDA ,we can see that time series is repeating after every 96 timestamps. The difference between two consecutive timestamp is 15 minutes , implying a strong daily seasonality in the timeseries. But this seasonality is not consistent throughout the time series due to anomalies present. These anomalies are found out and normalized using various methods.

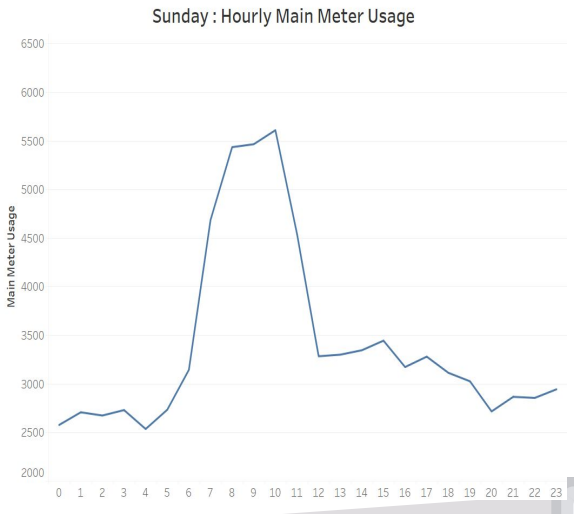
Reduced Usage on Sundays

From the raw plot of time series , it is evident there some anomaly is repeating in definite intervals. So the time series is grouped by day and mean usage across whole time series is plotted. From the plot it is quite clear that overall usage is less in sunday.The peak usage only last for fewer hours sunday than other days.



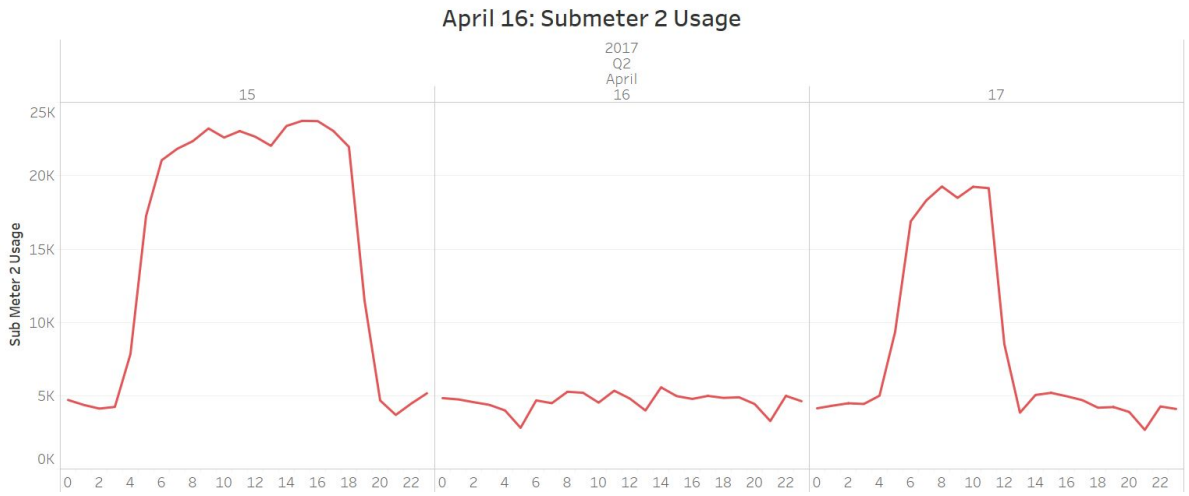


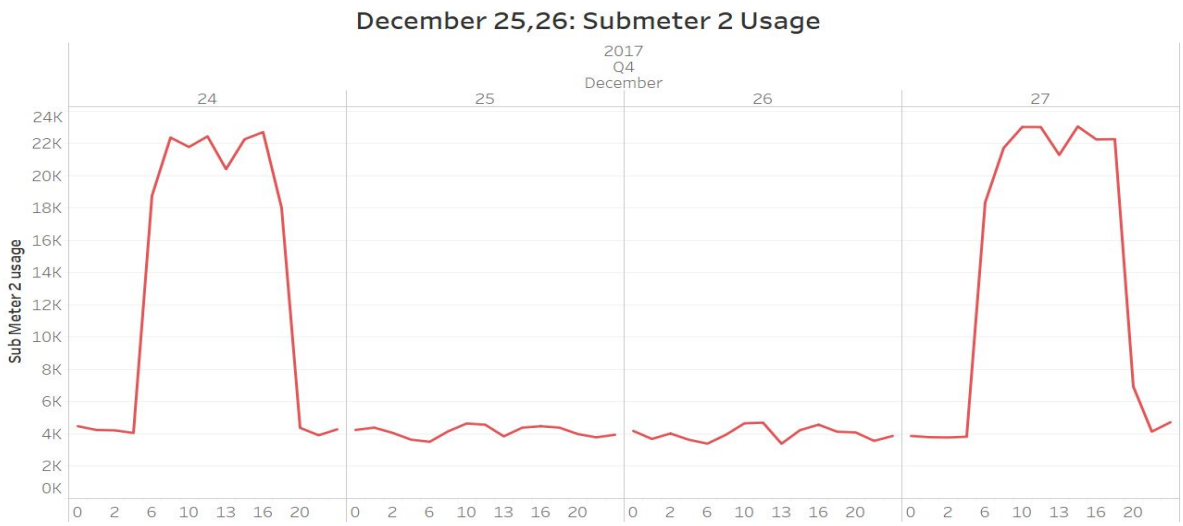




Reduced Usage on December 25,26 and April 16

Time series show daily seasonality with some exception of sundays.But this seasonality is broken in 3 days by a large margin. On these days peak usage during day is absent and usage is residual usage. These days could be holidays for company (Christmas and Easter fall on these days) , though other public holidays do not follow this trend.



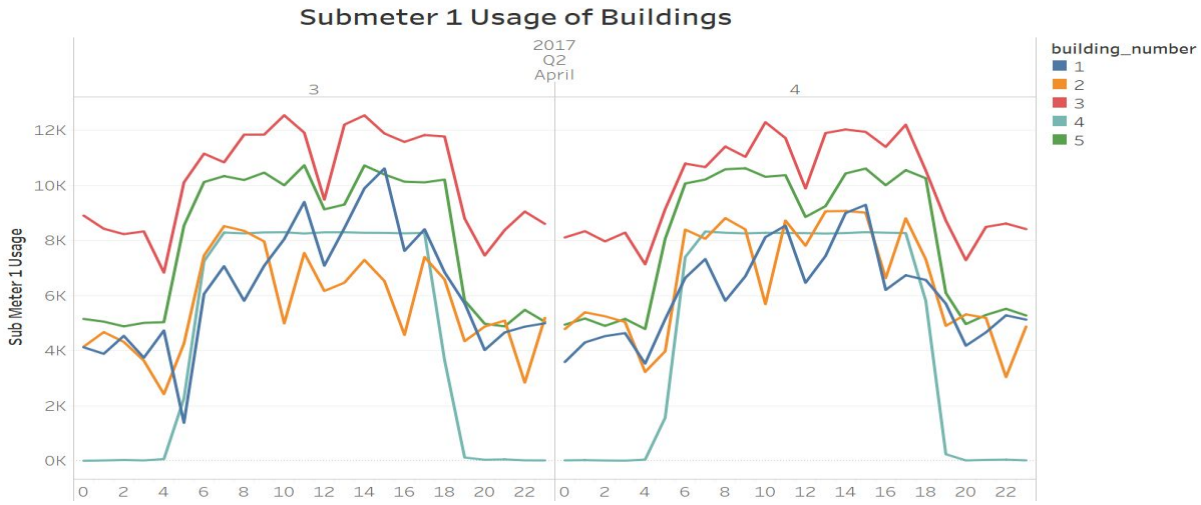


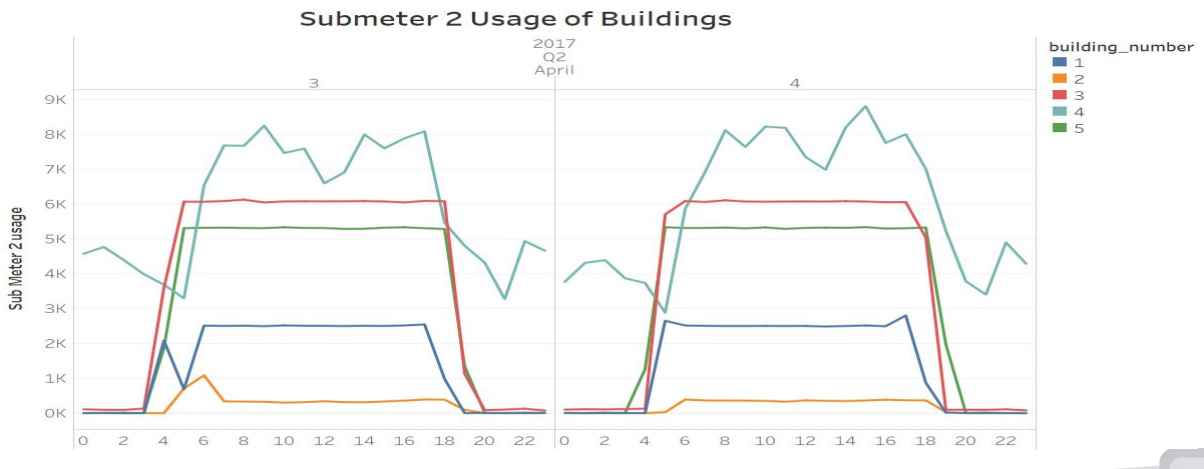
Pulsating Behaviour of Submeter 1 and 2

For sub meter 1 usage,building 4 is showing a pulsating character as plotted on figure below. This character is shown only by building 4 for submeter 1 usage.Pulsating character is defined by near zero usage during night(hours 0 to 5 and 18 to 24) and near constant usage on day (hours 5 to 18).Fluctuations present is negligible considering the range.

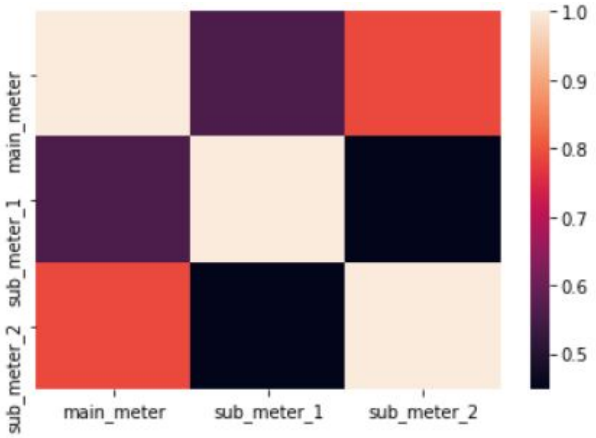
For submeter 2 usage,all building except Building 4 show the pulsating behaviour shown by building 4 in sub meter 1 consumption.That is Building 1,2,3 and 5 shows pulsating behaviour throughout the time series except during anomalies listed before.

Pulse behaviour indicates presence of instruments and equipments which have constant electricity requirement throughout day hours.





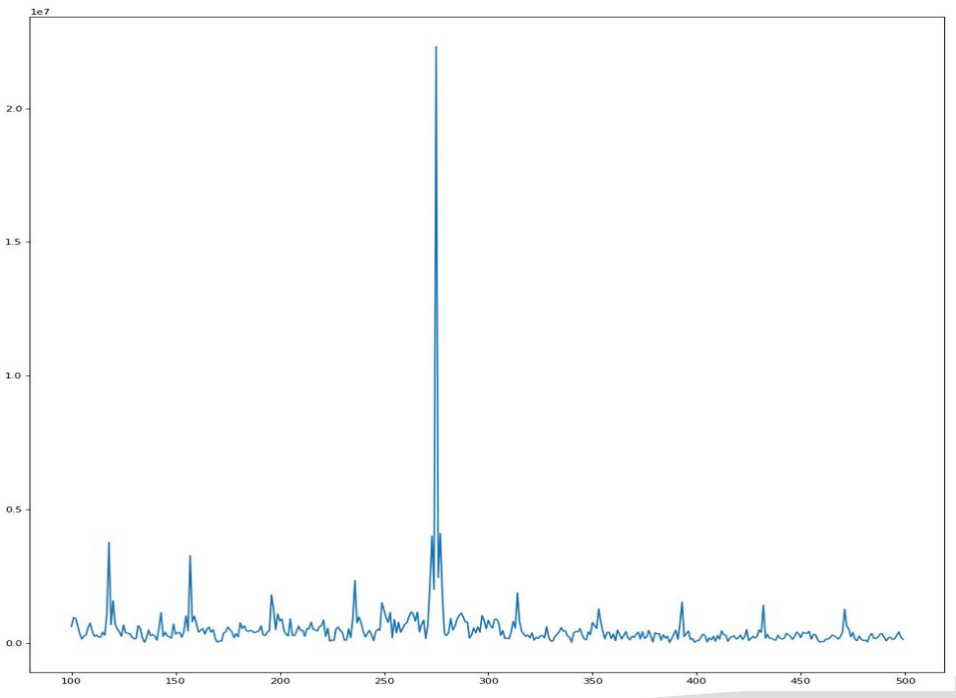
Correlation between meters:



We can see that the correlation value between all the meters are above 0.5 for a building, the increase and the decrease is taking place at nearly same time for all three meters. So we decided to use the same model for all three meters.

Finding Frequency Using Fast Fourier Transform:

We plotted the graphs of fast fourier transform(which is a discretized implementation of fourier transform) of the given dataset for building 1.We saw a sudden increase at a data index around 280.This behaviour was shows by other buildings giving maximum around 2nd or 3rd multiple of 96. This can be explained by anomalies in time series listed before.The anomaly on sundays is causing frequency to shift to higher values.



MODELLING TIME SERIES

1. ARIMA

ARIMA, short for 'Autoregressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.
An ARIMA model is characterized by 3 terms: p, d, q.
p is the order of AR term, q is the order of MA term, and d is the number of differencing required to make the time series stationary.

General equation for ARIMA model:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

If d=1: $y_t = Y_t - Y_{t-1}$

2. Prophet

The Prophet uses the a decomposable time series data mainly comprising of mainly three components namely seasonality, trend and holidays. They all are combined in the following manner:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon t$$

g(t): piecewise linear or logistic growth curve for modeling non-periodic changes in time series.
s(t): periodic changes (e.g. weekly/yearly seasonality)
h(t): effects of holidays (user provided) with irregular schedules

:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

To fit and forecast the effects of seasonality, prophet relies on fourier series to provide a flexible model. Seasonal effects s(t) are approximated by the function given above:

3. Holt-Winters forecasting Method

The Holt-Winters forecasting algorithm allows users to smooth a time series and use that data to forecast areas of interest. Exponential smoothing assigns exponentially decreasing weights and values against historical data to decrease the value of the weight for the older data. In other words, more recent historical data is assigned more weight in forecasting than the older results.

There are three types of exponential smoothing methods used in Holt-Winters:

- **Single Exponential Smoothing** – suitable for forecasting data with no trend or seasonal pattern, where the level of the data may change over time.
- **Double Exponential Smoothing** – for forecasting data where trends exist.
- **Triple Exponential Smoothing** – used for forecasting data with trend and/or seasonality.

There are two main HW models, depending on the type of seasonality:

• **Multiplicative Seasonal Model**

In this model time-series is assumed to be represented as:

$$yt = (b1 + b2t)St + t$$

• **Additive Seasonal Model**

In this model the time series is assumed to be represented as:

$$yt = b1 + b2t + St + t$$

Notations Used:

b1: is the base signal also called the permanent component
b2: is a linear trend component
St: is a multiplicative/additive seasonal factor
εt: is the random error component.

4. Time Series using Tabular Data.

By studying various statistical models we developed a new algorithm to forecast time series. Since the data was periodic in day we use 96 previous data points and arranged them in pandas dataframe. This gave us a tabular data where each columns represents the past values. This was trained using conventional machine learning models like LightGbm, Linear Regression.

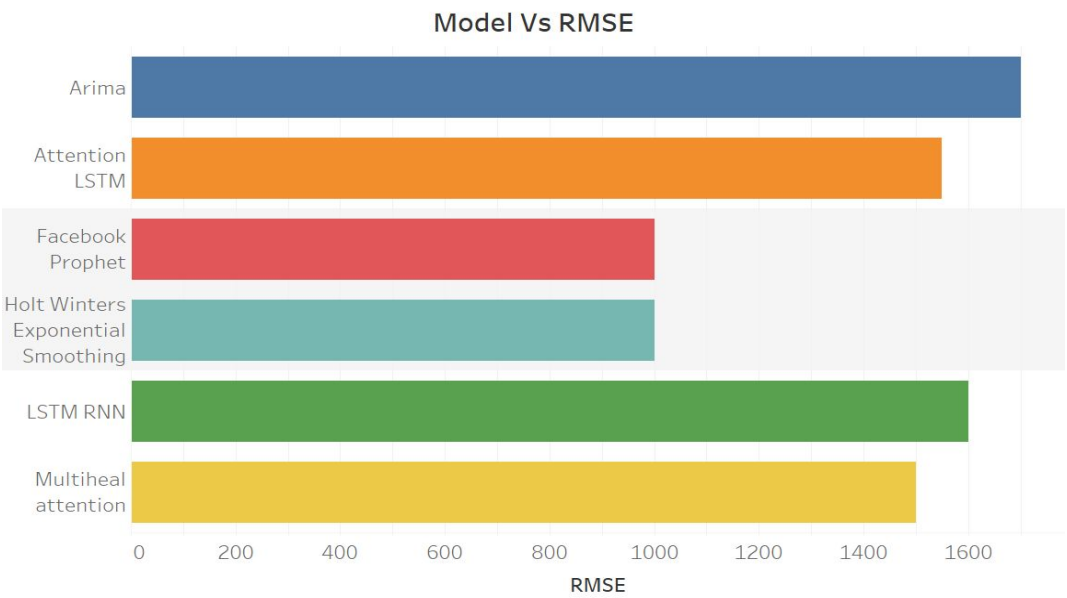
It was found that this model didn't performed well but at the same time it was robust to the anomalies in the data that occured around christmas and easter.The model was also found good at capturing the periodic trends . And due to use of tabular machine learning models the analysis of the weights assigned to previous samples was easy.

5. RNN-LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural networks capable of learning dependence in sequence prediction problems. We used the same for our time series sequential data. We tried with various model architectures to minimise the RMSE value for our data.

6. Multi Head Attention LSTM

Attention is the idea of freeing the encoder-decoder architecture from the fixed-length internal representation.This is achieved by keeping the intermediate outputs from the encoder LSTM from each step of the input sequence and training the model to learn to pay selective attention to these inputs and relate them to items in the output sequence.Multi-head self attention is used to capture the relationship for each pair of words within a sentence. Multi-head map the input many times to capture richer information



As we can see that Prophet and Holt Winter models work best for the given time series data..

MODEL ARCHITECTURE

Our main focus was on Holt's Winter Exponential Smoothing and facebook's Prophet, since the results for these two better than other models. In Holt's Winter Exponential Smoothing the seasonal_periods parameter was set to 672 (96*7) to capture trends for whole week. This was later verified and found to be best parameter by using grid search.
For facebook's Prophet we did grid search and found that peak values were obtained for setting seasonality parameters to multiples of 96. Prophet also has facility for additional regressor variable on which time series might depend. Two variables were added :

1) One variable representing the day of week
   This was done to properly capture the different behavior of data in sunday (peak values were obtained for lesser amount of time). All other days were set as 0 and sunday was set as 1.
2) The Other variable was hours
   Hours were label encoded. For hours 0, 1, 2, 19, 20, 21, 22, 23 the value was set to 1, for hours 6, 7, 8, 9, 10, 11, 12 , 13, 14, 15 value was set to 2, for hours 3, 4, 5 which indicated the upward transition was set to 3 and for 16, 17, 18 which indicated the downward and less steeper transition of value was set to 4 .

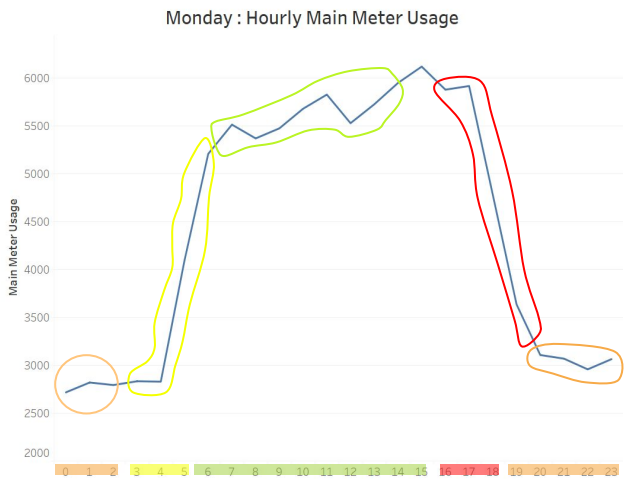
The custom seasonality was set to match the daywise periodicity.
It was also found model was predictions were more precise when periodic parameters were set to values obtained from fft(fast fourier transform).

: Category 1

: Category 2

: Category 3

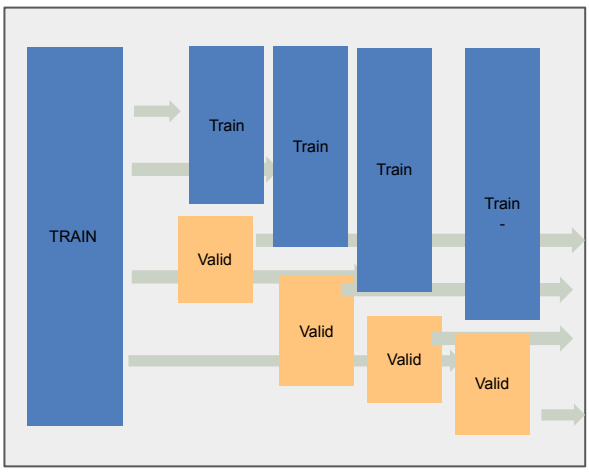
: Category 4



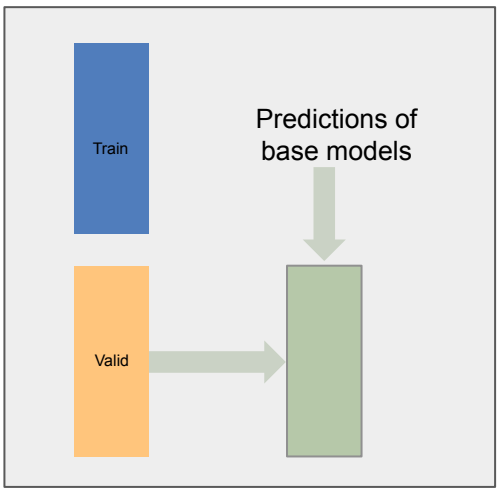
MODEL ARCHITECTURE

We used these two tuned models as base models and XGBoost as our stacking regressor. The parameters of xgboost were tuned using grid search and bayesian optimisation using rmse as metric and best of two was chosen for final model.
Stacking was done as follows :

1) Each building and meter was selected using a for loop.
2) Training data was divided into training and testing dataset with 80/20 split ratio.



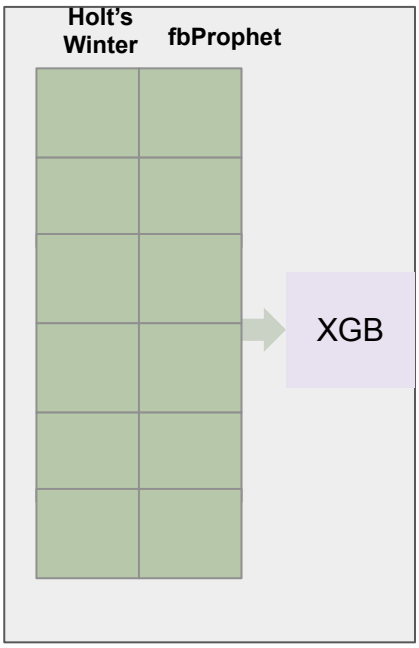
Then the time-series-k-fold-split from sklearn was used to further divide training data into train data and validation data.



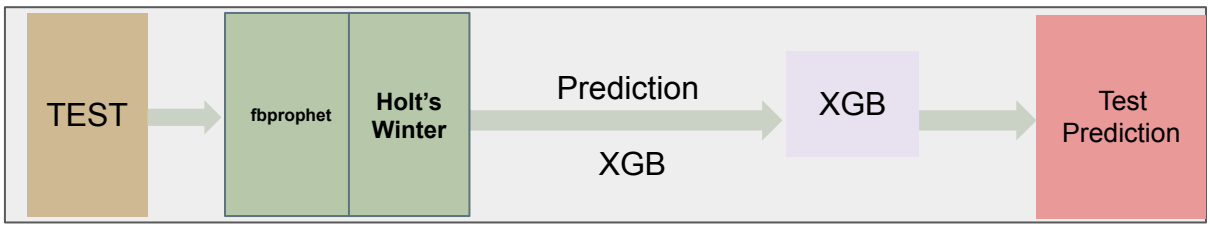
For each fold both the models (Holt's Winter Exponential Smoothing and Prophet) that were tuned previously, were trained on the training dataset and their prediction on validation set was stored in pandas dataframe along with ground truth valued of validation set.

MODEL ARCHITECTURE

1)



- These provided us with a tabular dataset with 2 columns and also ground truth values on which xgboost was trained.
- Now the based models were again trained on whole dataset. And predictions were obtained on new test dataset(i.e is 20 split). This gives dataframe again with 2 columns on which XGBoost gives final prediction. This completes the stacking procedure. Evaluation metric was calculated using final predictions by xgboost and ground truth values of 20%split.
- For the final test results we use the whole training dataset for traini g base and stacking regressors. And 20%split voice was replaced with original test set.



- The final predictions were again clipped to make all values positive and especially for submeter 2 which was like an impulse.

EVALUATION METRIC

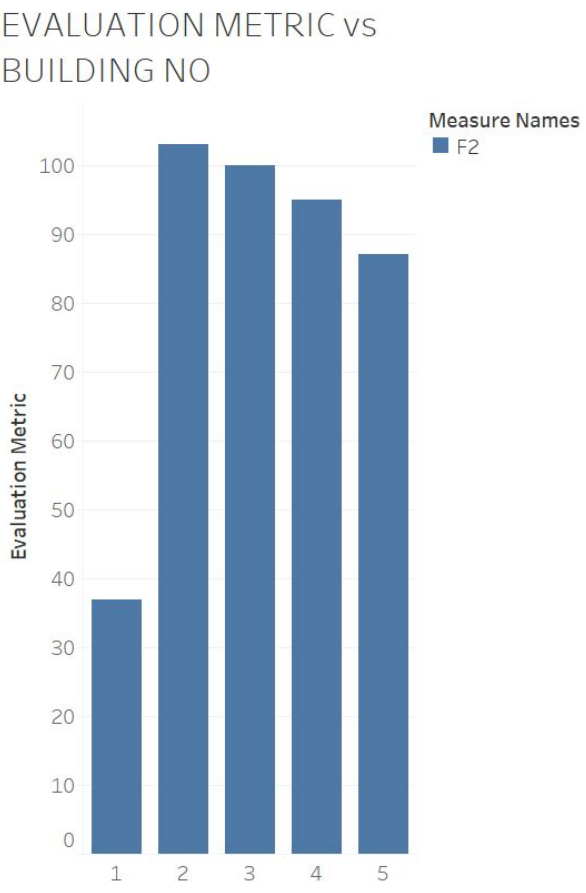
Simple Average across errors of all 5 buildings where error for a single building is calculated as

$$1/3\sum_{i=A}^{i=C}(1/\overline{m_i})\sqrt{\sum_{t=1}^{t=T}(m_{it}-\widehat{m_{it}})^2.e^{-kd(t)}}$$

$e^{-kd(t)}$ Is a basically added to down weight prediction further in future. When k is substituted in the equation and evaluated, this constant will reduce magnitude of squared error.This term doubles the tolerance for error every hundred days.

RESULTS

As all correlation values between meters are greater than 0.25, we were using the same model for all the meters.
Using **facebook prophet** we are getting lower rmse value for the final parts of the data after november 2017.
Using **Holt Winters Exponential Smoothing** we are getting lower RMSE values for the data before oct 2017.
We stacked the predictions into a single dataset and used regresor to predict the optimal value near to the true value. We are getting lower rmse for **XGBoost regressor.**
Then finally after tuning to optimal values of parameters we evaluated using the given evaluation metric and we are getting evaluation scores between **30 and 60** for nearly all the buildings and meters
We are getting the lower values of evaluation metric for the prediction of main meter and higher values of evaluation metric for prediction of sub meter 2 values.



ANNEXURE

```
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
figure(num=None, figsize=(15, 15), dpi=80, facecolor='w', edgecolor='k')
f = np.fft.fft(main_meter)
f_mod = np.abs(f)
plt.plot(range(100, 500), f_mod[100:500])
plt.show()
np.argmax(f_mod[10:])
```


```
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
figure(num=None, figsize=(15, 15), dpi=80, facecolor='w', edgecolor='k')
f = np.fft.fft(main_meter)
f_mod = np.abs(f)
plt.plot(range(100, 500), f_mod[100:500])
plt.show()
np.argmax(f_mod[10:])
```