TEAM - 7

INTER HALL DATA ANALYTICS

INDEX

DEFINITION

Project Overview

Accurate electricity consumption forecasting has primary importance in the energy planning of developing countries. Long term electricity consumption forecasting is the basis for energy investment planning because overestimation of the consumption would lead to superfluous idle capacity which means wasted financial resources, whereas underestimation would lead to higher operating costs for energy suppliers and would cause potential energy outages. Therefore, modelling electricity consumption with good accuracy becomes vital in order to avoid costly mistakes.

For the purpose mentioned above, Dream Vidyut, which is a modern age electricity generation company, dreams about electrifying entire India in a sustainable manner. As part of their efforts, currently they monitor the electricity consumption of more than a dozen corporate buildings in Delhi-NCR to approximately forecast the demands of these buildings so that over-production of electricity can be reduced. In this project, Dream Vidyut has made this data available for creating a model better than theirs to forecast the electricity demand and thus help them to optimally serve their purpose.

Problem Statement

In this project, we are required to approximately forecast the electricity consumption across three electrical meters of 5 buildings in Delhi-NCR between 1st January and 18th April of 2018 from the data of the previous year i.e. 2017 from April to December.

We have the following files available: -

- train.csv - contains the electricity consumption of ***5 buildings*** measured across ***three electrical meters from April to December in 2017.***
- test.csv - contains the timestamps from the ***year 2018*** and the building ids for which predictions have to be made.
- sample_submission.csv - contains a sample submission file.

Data Description

In this project, we have a training dataset (train.csv), which consists of 1,32,000 rows and 5 columns.

The following fields have been provided in the training dataset:

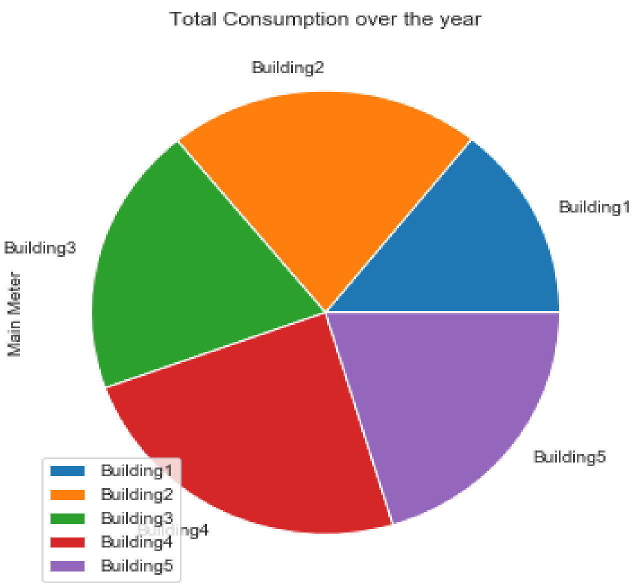
1. *Timestamp*: It is the time at which the data is taken
2. *Main_meter*: It shows the Main Meter reading
3. *sub_meter_1*: It shows the reading of Sub Meter 1
4. *sub_meter_2*: It shows the reading of Sub Meter 2
5. *building_number*: This column tells the building number for which the data is recorded.

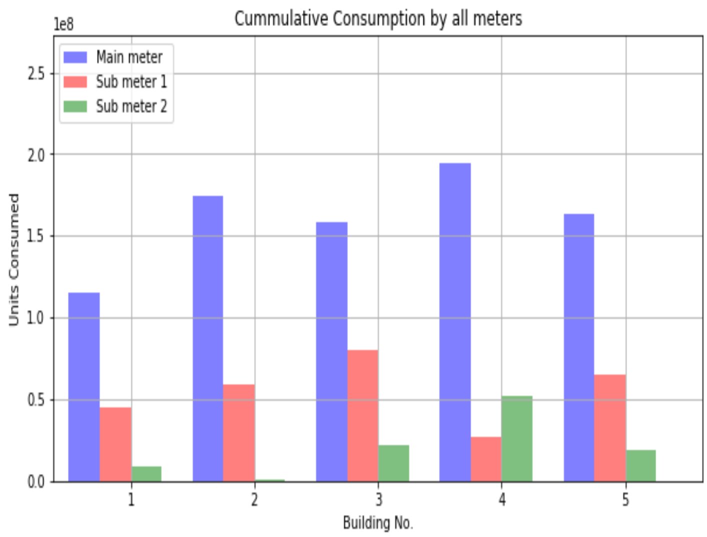
We have 5 buildings namely 1, 2, 3, 4 and 5.
Each building has equal amount of data i.e. (132000)/5 = 26400.

In “train.csv”, we have data from 1st April 2017 to 31st December 2017.
Total Number of Days = 275
Total Number of readings for each building = 275*96 = 26400

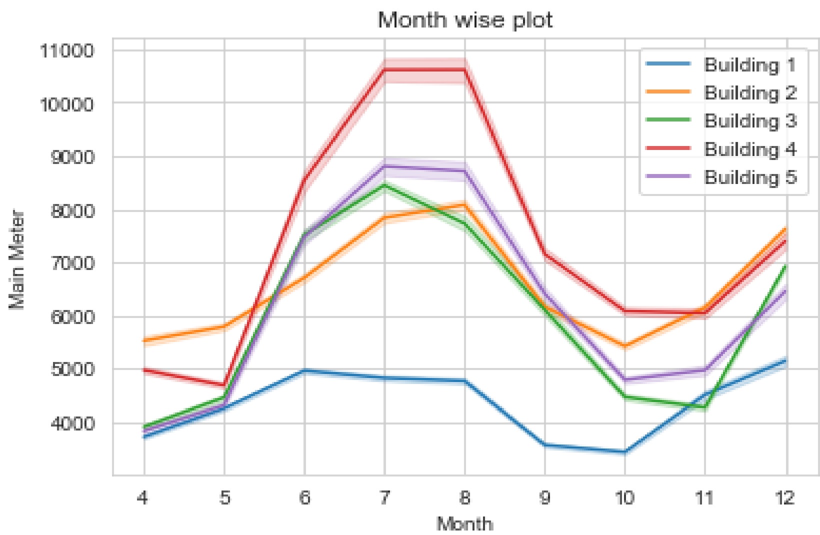
ANALYSIS OF DATA

Data Visualisation

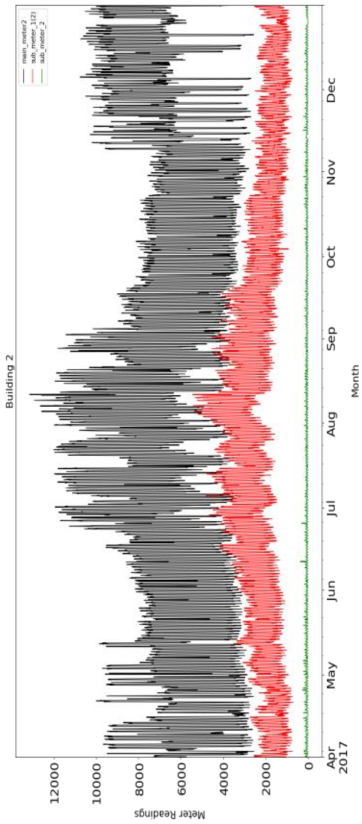


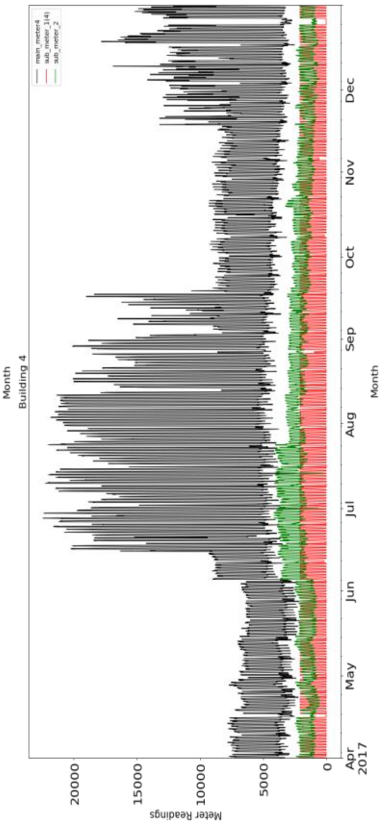


From the above charts (pie chart & bar graph), we can see that **Building 4** consumes the most whereas **Building 1** consumes the least energy units over the whole year.



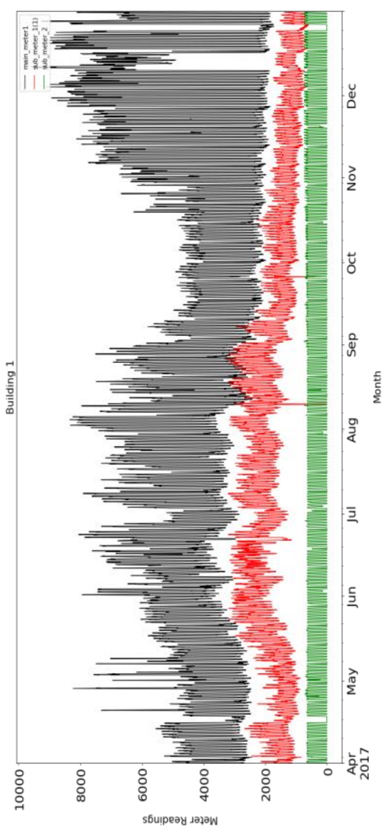
From this plot, we can see that meter readings are **almost constant and comparatively very high** for the period of **mid-June to August** for all the buildings.
This period corresponds usually to the monsoon season.

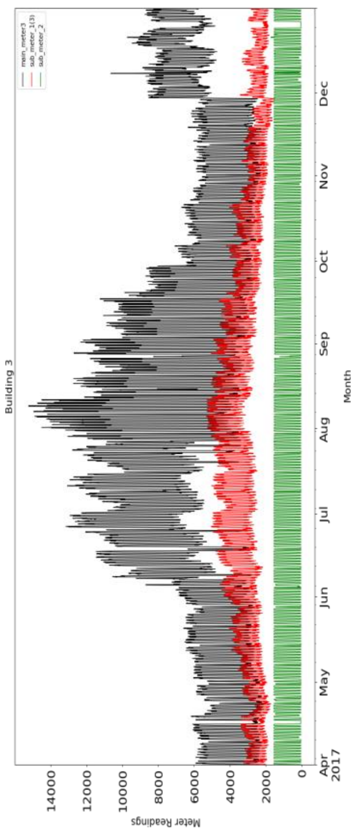


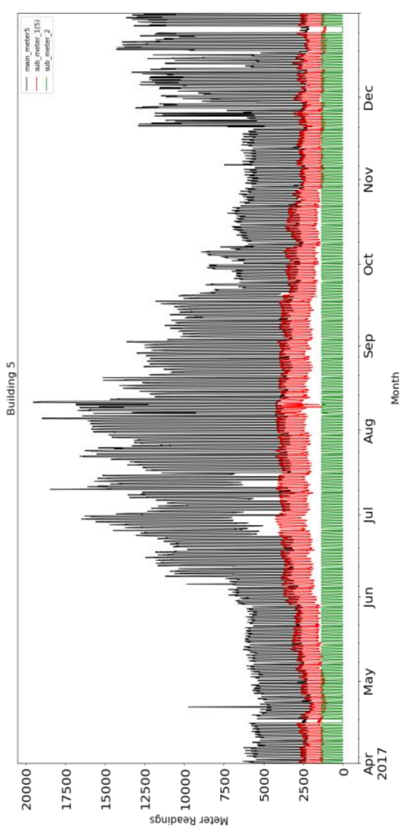


These are the most basic visualization plot, depicting the ‘meter readings’ of all the tree meters (main_meter, sub_meter_1, sub_meter_2)

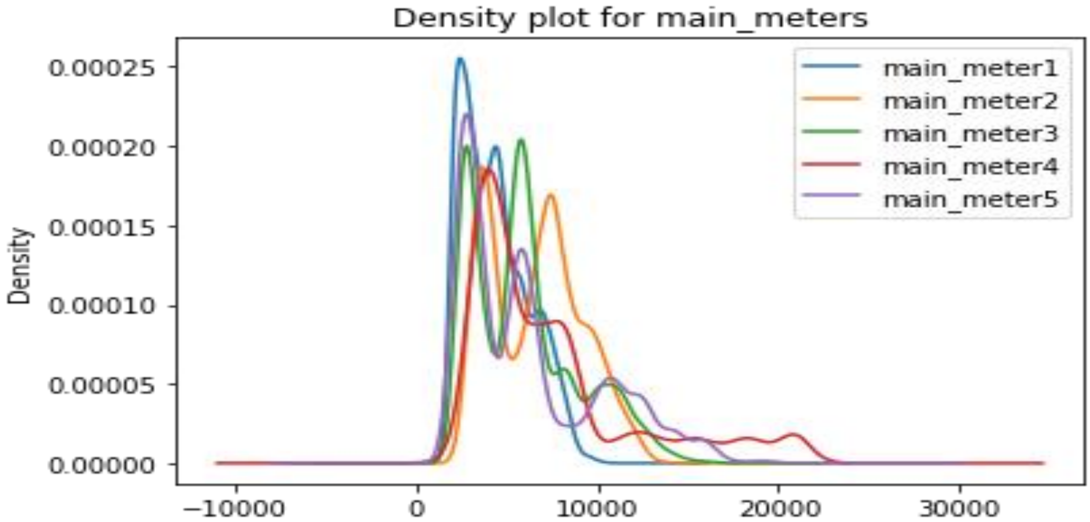
A basic inference that can be drawn from these is that for buildings 2 & 3, during December, the main meter reading increases significantly whereas sub-meter readings remain almost constant. **This shows that any heating elements (Ex: Heaters, Geysers), which are primarily used during the winter season, are not recorded by either submeter 1 or submeter 2** but by some other submeter contributing to this increase in main meter reading.







DENSITY PLOT (for the whole year)

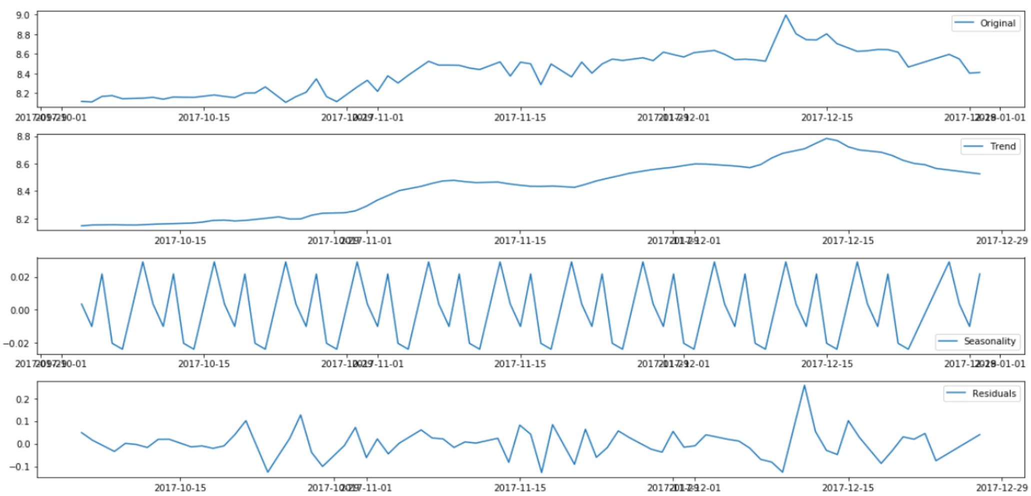


The density plot here depicts the readings of the main meter. Readings of **Building 1** are concentrated in a **small range of values**, which may mean that only some **specific types of work** undergoes in Building 1 over the year (could be an industrial building); whereas in **Building 4**, readings are **scattered over a wide range**, which suggests different types of work are being done in Building 4 and hence it could be a **building engaged in various works** throughout the year.

Exploratory Data Analysis

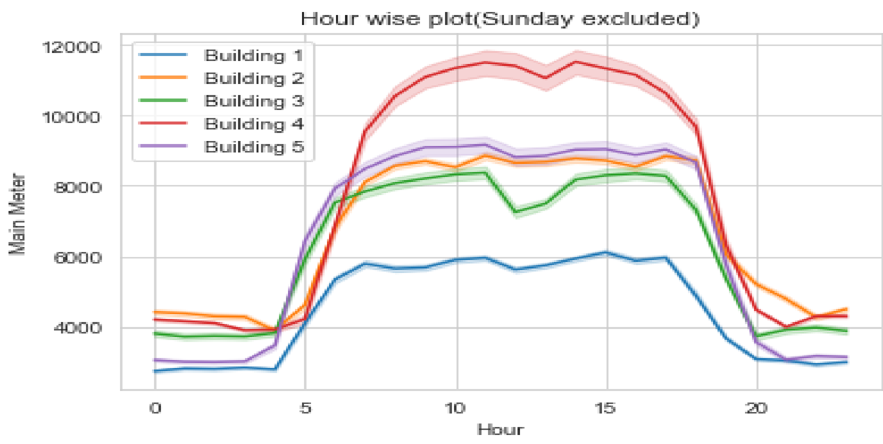
With a focus on summarizing and visualizing the important characteristics of a dataset, Explorat-ory Analysis assists in understanding the data's underlying structure and variables, developing intuition about the dataset and deciding how it can be investigated with more formal statistical methods. After a detailed analysis, some significant results could be observed as follows:

EXPLORING THE TIME SERIES

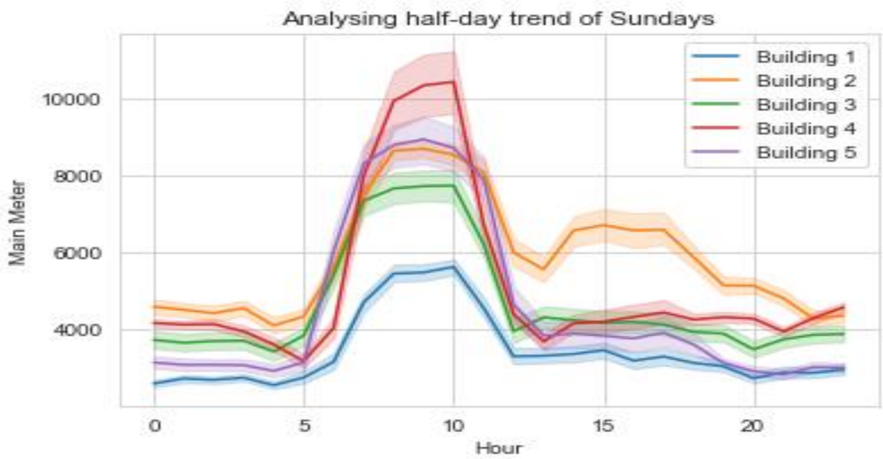


The graph shows the trend, seasonality and residual against the original time series of the electricity demand. The trend shows that the demand increases continuously, reaches peak and then decreases over April 2017 to December 2017.

HOUR WISE ANALYSIS

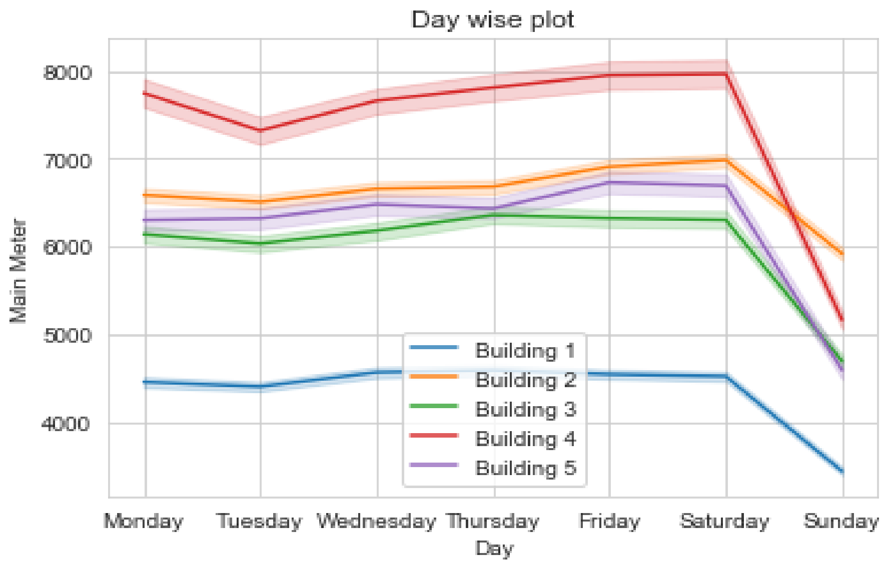


Hour wise analysis (Sundays excluded) over the year suggests that on an average, **Building 4** consumes the **highest number of electrical units** while **Building 1** consumes the **least**. At a glance, we realize that the **active hours** of all the buildings range from **7 AM** to **6 PM**.



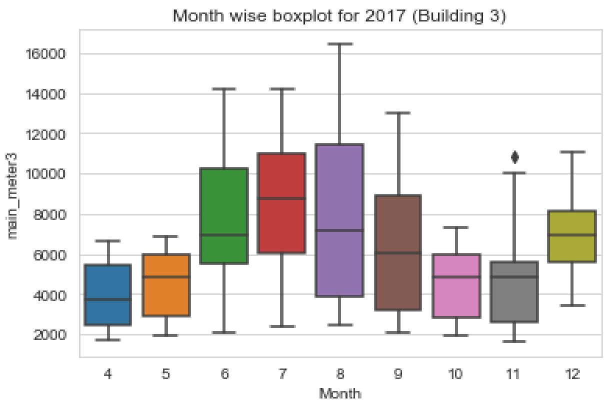
For **Sundays**, unlike the weekdays, the active hours are mostly from **7 AM** to **11 AM** only. This shows that **Sundays** might be serving as **half-days**.

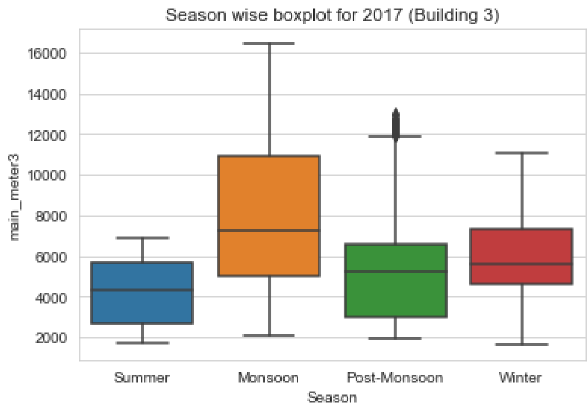
DAY WISE ANALYSIS



Day wise analysis of the main meter readings of the given buildings shows that a **global minimum point** for every building plot lies on **Sunday**. Thus, Sunday might be a day-off in some buildings. But **Building 1 & 2** have readings relatively **comparable** to other weekday readings, this suggests that these buildings are **engaged in certain activities even on Sundays**.

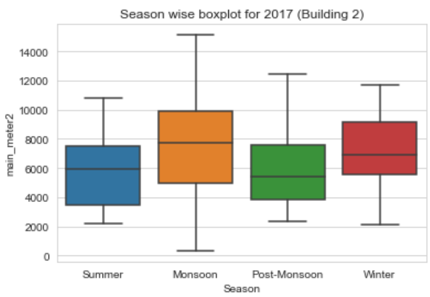
SEASON WISE ANALYSIS

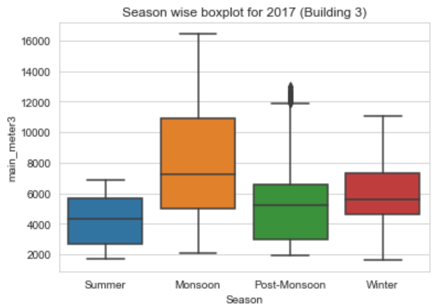


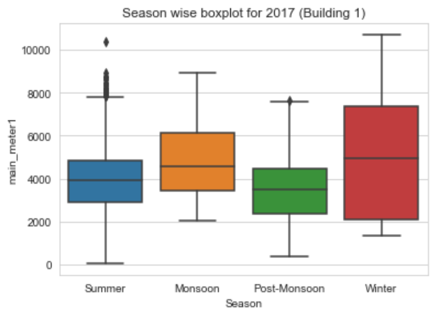


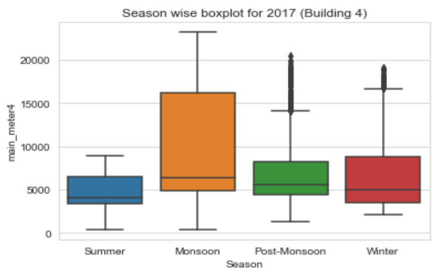
On analyzing the data month wise, we observed a general trend for all the buildings that main meter readings are nearly the same for a particular season, hence we grouped them as -

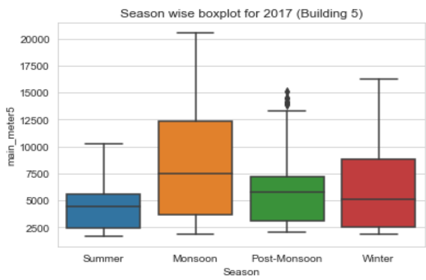
- Summers: - April, May & 1st half of June
- Monsoon: - 2nd half of June, July & August.
- Post-Monsoon: - September & October.
- Winter: - November & December.

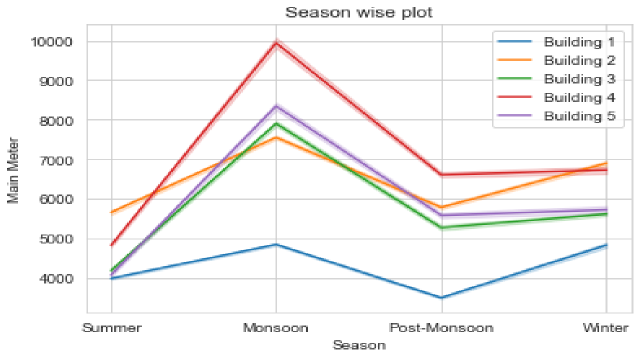






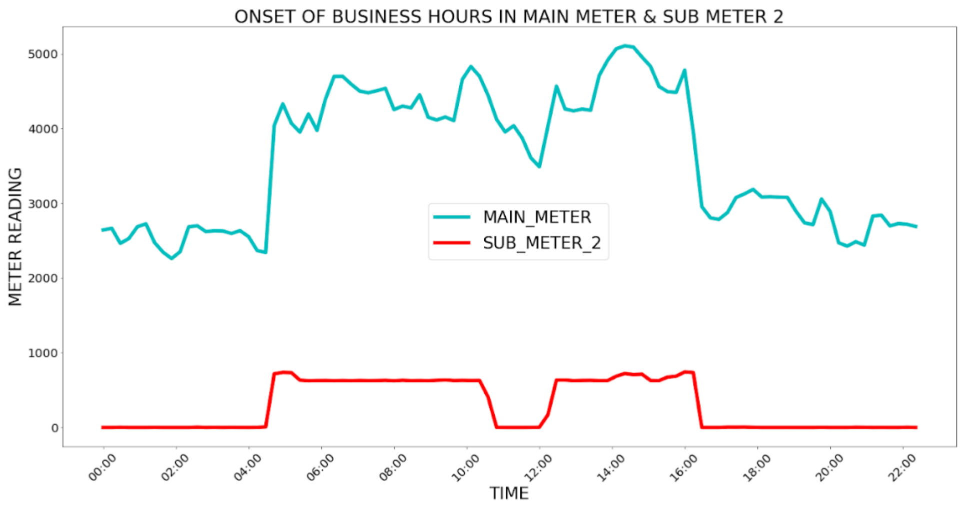






From the seasonal plot for all the five buildings, we can see that there is a general trend in all the buildings i.e. <u>readings increase during the monsoon season</u> and <u>decreases in all the buildings in the post-monsoon season</u> and again increases in the winter season.

RELATIONSHIP BETWEEN MAIN_METER AND SUB_METER_2

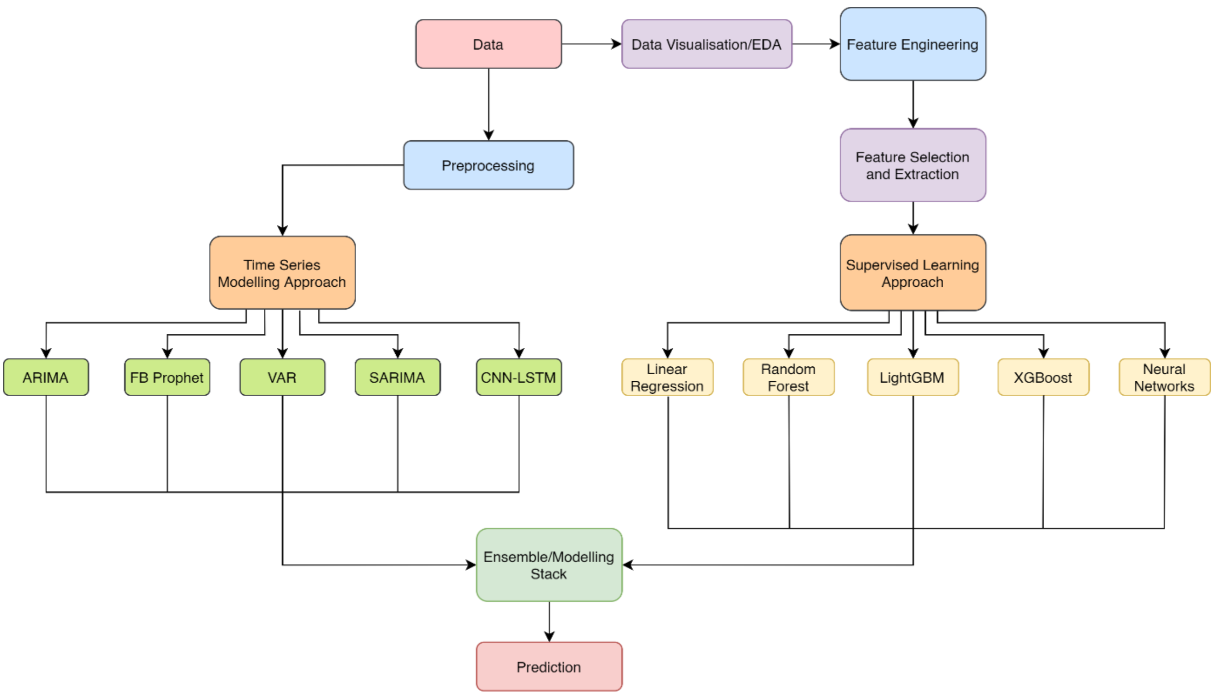


A general trend **observed in all the buildings** is that **the onset of the readings of main_meter and sub_meter_2 occurs simultaneously, i.e. a clear pattern is observed throughout the year.**

FURTHER DATA ANALYSIS

(INCLUDED IN ANNEXTURE)

ALGORITHMS AND TECHNIQUES



Schematic Depiction of Approach and Model applied

Modelling the problem using a Time-Series forecasting approach

I. Feature engineering:

In the given time series, we started by applying some basic feature engineering in order to optimize our results.

a. To begin with, we applied the time series models on the **difference between the two consecutive values** to make the time series stationary (or remove the non-stationarity).
b. Also, the **logarithm of all values** was taken to apply all the time series models so as to normalize the data.
c. **Rolling means** were taken as a feature, and the time series models were applied to it.

II. Model and Approach:

- We started with modelling the time series with the **ARIMA model** as it's the most basic and for time series; but the model, as expected failed to capture seasonality in the data.
- We moved on to work with Seasonal Arima (**SARIMA)** to support the seasonal component of the time series. It failed miserably.
- We also tried the state-of-the-art technique for modelling time series via **Facebook's prophet model.** It performed better than others but was still unacceptable.
- The results using **encoder-decoder** were also analysed but we did not get any betterment on the results of the prior models.
- The model which we tested next was the **Vector Autoregressive model (VAR)** to capture the linear interdependencies among multiple time series of main_meter, sub_meter_1 and sub_meter_2 and the results obtained from it were the best till now thus showing that there is interdependence between the 3-meter readings along with their values with time.
- We next applied the normal **LSTM model** and followed by the time distributed **CNN LSTM** for further improvement.

Moving towards a Supervised Learning Approach

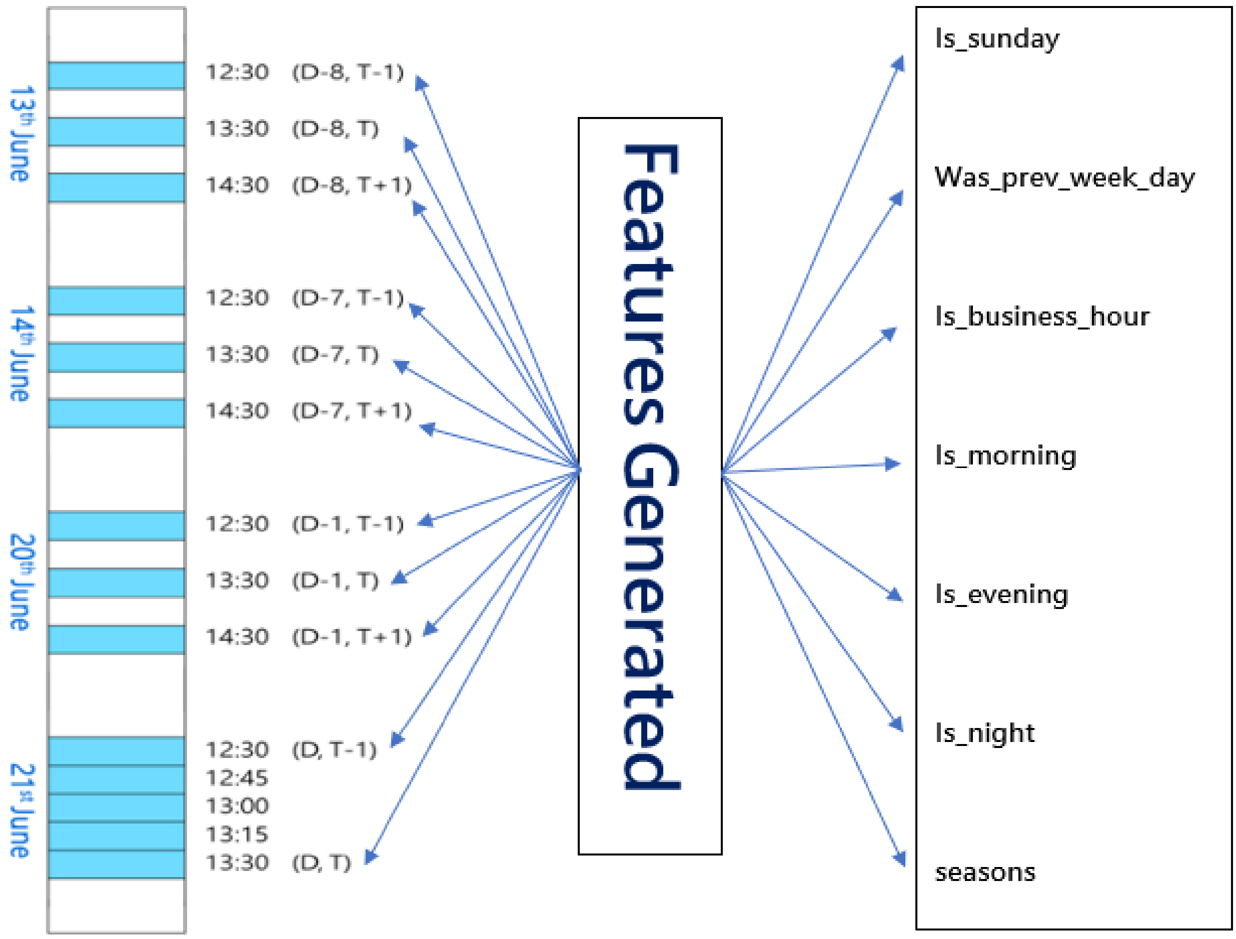
I. Why Supervised Learning

Trends observed during exploratory data analysis were not adequately captured through time-series models. The evaluation metrics showed that the error was large. Thus, we decided to shift to other methods.

Based on the exploratory data analysis, we designed new features that gave better results on supervised learning models.

II. Feature Engineering:

Various new features were designed based on the exploratory data analysis done.
The schematic below depicts the features that we created for each of the timestamp based on differentiating time and date.



The Features are:

Timestamps as a new feature;
Time-15; Time-30; Time-45; Time-60; Previousdaytime; Previousdaytime; Previousdaytime; Previousdaytime; Previousdaytime; Previousweektime; Previousweektime; Previousweektime; Previousweektime; Previousweektime

Apart from the time and date based features, we also created some other features (for all three meter readings) to get the best result and the most optimized predictions; These are: Issunday, Wasprevweekday, Isbusinesshour, Ismorning, Isevening, Isnight, Seasons

For all 3 meters
(These are also depicted in the schematic above)

III. Feature Selection and feature extraction

We applied other methods like Backward Elimination, Auto-Encoder & PCA but didn't get good results, hence they were dropped.

Backward Elimination - involves testing the deletion of each variable using a chosen model fit criterion, and repeating this process until no further variables can be deleted without a statistically significant loss of fit.

Backward Elimination

Building 1: r-square:0.916, s1_pw_pdt_p, S2_pw_pdt_p, Wasprevweekday

Building 2: r-square:0.964, S2_pw_pdt_m, Season, S1_pw_pdt_m, MM_pdt_m

Building 3: r-square:0.966, MM_pw_pdt_p, S1_pw_pdt_p, Isevening

Building 4: r-square:0.972, S2_pw_pdt_p, Season, Isbussinesshour*

Building 5: r-square:0.970, MM_pw_pdt_p, S2_pw_pdt_p, Wasprevweekday, Season

Auto-Encoder - An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner.

Using Auto encoder-decoder 16 features were extracted and the RMSE obtained was 680.

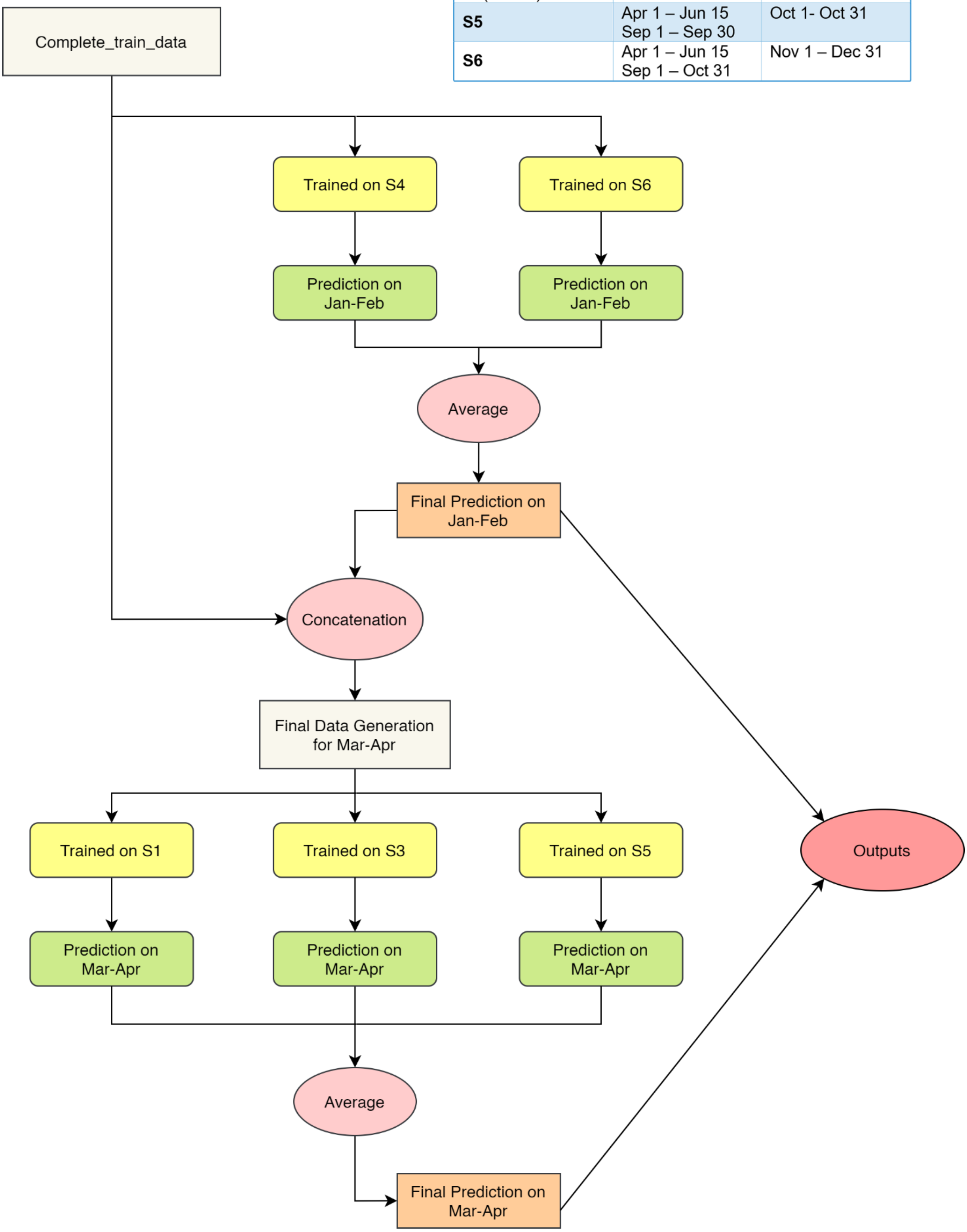
PCA - Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables

Using PCA 16 features were extracted & the RMSE obtained was 900.

IV. Model and Approach

- Firstly, we applied the basic **Linear Regression model** and then moved on to a more advanced **Random Forest model**.
- Since both of the models account only for the linearity of the data we moved towards bagging and boosting based models and tried more complex model **XGBoost** so that we can consider non-linear relations too, but we didn't get a better result. This may be because XGBoost doesn't work well on a large dataset as it isn't able to capture all the parameters.
- Therefore, we tried **Light GBM** but still, our model didn't improve much. This may be because it uses some kind of modified mean encoding for categorical data which caused overfitting.
- So, we also worked with the use of neural networks and applied the **DNN (Deep Neural Network) model.**

Label	Train Data	Validation Data
S1(Summer)	Apr 1 – May 31	Jun 1 – Jun 15
S2(Monsoon)	Jun 16 – Aug 15	Aug 16 – Aug 31
S3(Post Monsoon)	Sep 1 – Oct 15	Oct 16 – Oct 31
S4(Winter)	Nov 1 – Dec 15	Dec 16 – Dec 31
S5	Apr 1 – Jun 15 Sep 1 – Sep 30	Oct 1- Oct 31
S6	Apr 1 – Jun 15 Sep 1 – Oct 31	Nov 1 – Dec 31



Schematic depicting approach for Prediction

RESULT

Model Performance on Evaluation Metric & its Interpretation

We got the best outputs/prediction by analysing the results of various models; by creating features which will enhance our results; by using the best model's season-wise as we interpreted from our exploratory data analysis that this will optimize our results.
We now evaluated our results and have tabulated it for better interpretation

The evaluation metrics that we used for testing our models is a **Simple Average** across errors of all 5 buildings where error for a single building is calculated as -

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

Where A, B and C represent Main meter, Sub-meter 1 and Sub-meter 2 respectively,

- T is the total number of timestamps in the test.csv for a particular building,
- t is the timestamp for which prediction is being made,
- k = (ln 2) / 100
- d(t) is the value of the day in which timestamp t falls
- m(it) is the actual value of meter i at time step t
- Caret(m(it)) is the predicted value of meter i at time step t
- Bar(m(i)) is the mean value of meter i - This value is used for normalization

For the evaluation metric provided, it's evident that for a given month the prediction error of the prior days will amount for higher value rather than later days of the month because of the exponential term in the error function. This can be seen very clearly from the plot as well.
The metric provides equal weightage to each of the three meters thus showing that correctly predicting each meter's reading is of utmost importance despite the main_meter having the highest reading values throughout.
The observed values of evaluation metric are tabulated as follows:-

USING TIME SERIES APPROACH

MODEL	Building-1	Building2	Building3	Building4	Building5
ARIMA	54.23057583	58.93669177	50.6636733	57.89242747	60.03572692
SARIMA	124.3849556	134.571593	116.664206	132.3112276	136.9505128
FB Prophet	23.09	25.52	14.40666667	19.97333333	16.62333333
VAR	20.18133333	20.29633333	29.2	32.37	19.87
CNN_LSTRM	24.731	33.82	17.52766667	22.913	19.91266667

*seasons as indicated on Page no. 13

USING SUPERVISED LEARNING APPROACH

Building 1

Models	Season 1	Season 3	Season 4	Season 5	Season 6
Linear Regression	3.575	4.368666667	4.653666667	4.362	8.856
Random Forest	4.397666667	5.608	6.812666667	5.608	12.14366667
LightGBM	4.034666667	3.428333333	6.252333333	3.428333333	10.707
Xgboost	7.263333333	5.873333333	5.754666667	1.083	16.44

Building 2

Models	Season 1	Season 3	Season 4	Season 5	Season 6
Linear Regression	8.4514	7.065666667	9.621666667	12.13866667	16.825
Random Forest	8.493333333	9.76	10.02066667	12.03533333	19.293
LightGBM	8.648666667	9.526666667	9.463333333	11.27366667	17.95
XgBoost	10.73	10.95	9.654666667	3.013333333	23.80066667

Building 3

Models	Season 1	Season 3	Season 4	Season 5	Season 6
Linear Regression	5.9718	3.5721	4.502	4.667866667	8.066666667
Random Forest	8.793333333	3.962	5.402333333	3.217666667	11.1
LightGBM	8.383333333	3.876666667	4.776666667	4.123803333	8.9564
XgBoost	6.955066667	4.2	4.965	2.423333333	11.63366667

Building 4

Models	Season 1	Season 3	Season 4	Season 5	Season 6
Linear Regression	4.1106	3.923676667	6.120333333	5.460466667	9.996
Random Forest	7.806666667	5.603333333	6.955	4.209	13.40666667
LightGBM	8.224333333	5.38	7.4199	5.236533333	10.04333333
XgBoost	7.263333333	6.85	7.446666667	1.566666667	14.74333333

Building 5

Models	Season 1	Season 3	Season 4	Season 5	Season 6
Linear Regression	3.754133333	3.718333333	5.787333333	5.181666667	9.166666667
Random Forest	6.023333333	3.74	5.863333333	3.492666667	10.95666667
LightGBM	5.781333333	3.632333333	5.564333333	4.567666667	9.809
XgBoost	7.17	4.01	5.561666667	1.210933333	10.858

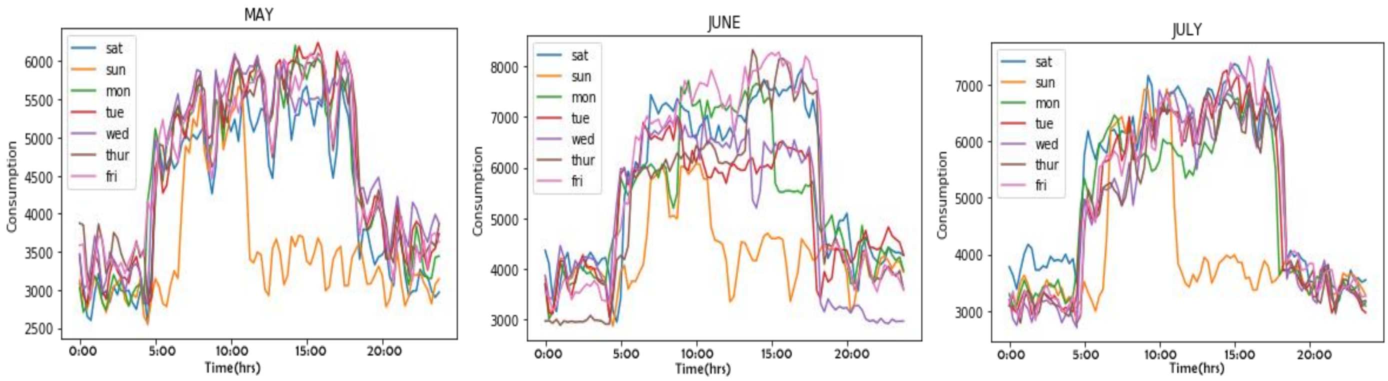
Conclusion and Recommendation

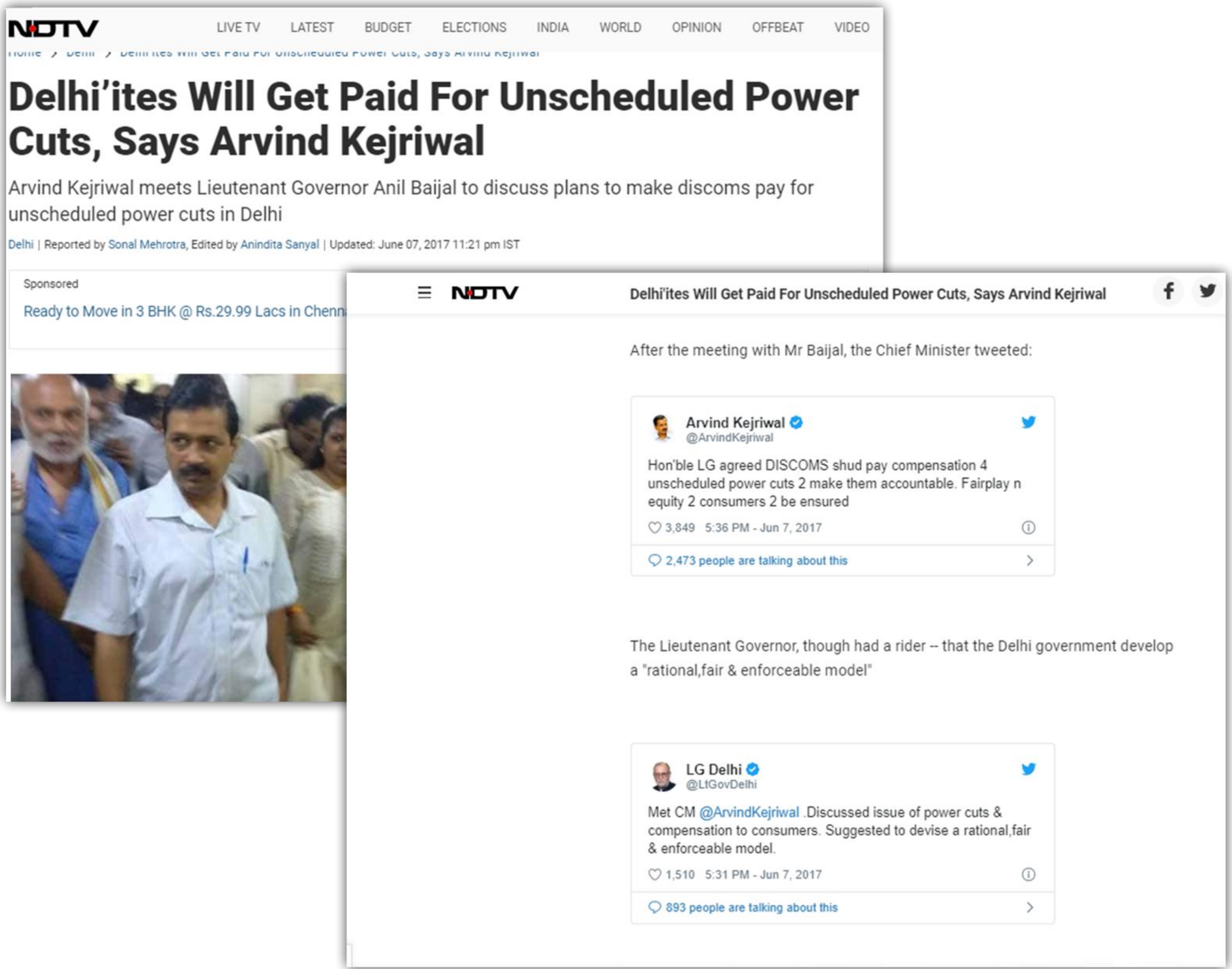
- The industrial sector accounted for 44% (532 billion KWh) of total electricity consumed (1,208 billion KWh) in India for the year 2015. Even considering 30% of this energy input being wasted by industry amounts to 160 billion KWh annually, equivalent to 20,000 MW of coal-based power generation capacity.
- Draft says as much as 20-50% of the energy used by industry is wasted, being released into the environment in the form of exhaust gases and liquids that flow out of plants.
- The potential for waste heat power from industrial and non-industrial processes for countries such as the US is 7,000-10,000 MW of generation capacity.
- India’s potential for power generation through waste heat is significant as its industries are much less energy efficient when compared with advanced countries.
- It is the height of paradox for a country where blackouts are more a norm than a exception.
- Over 3 billion units of electricity, or a day's national consumption, were wasted in 2014-15 as congestion in the transmission highways blocked trading between surplus and deficit regions.
- After applying various models on the dataset, we concluded that ensembled model gives the best result and so we are deploying that for our results.
- Through exploratory data analysis and various plots, we concluded that each building consumes **maximum** electrical units during the **monsoon season** and **minimum** during the **summer** and **post-monsoon** season.
- Also, Building 4 consumes the most electrical units while building 1 consumes the least.
- During the weekdays, buildings consume more electrical units compared to Sundays.
- As electricity wastage is of negative effect to both our economy and environment, via this analysis we have tried to predict he demand to that near about exact amount of electricity can be generated thus minimizing the wastage. From our through analysis throughout the problem solving, we think that the 5 buildings maybe some exhibition buildings or buildings of a tech park.
- Hence the company should produce less electricity during the summer and post-monsoon season for these buildings so that overproduction of the electricity can be reduced and use the resources elsewhere and carry forward their dream about electrifying entire India in a sustainable manner.

ANNEXURE

Further Data Analysis

1. In the month of June, a lot of electricity consumption fluctuations were observed for all buildings, maybe because of the reasons mentioned below in the article.
We have plotted variations for Building 1.





NDTV LIVE TV LATEST BUDGET ELECTIONS INDIA WORLD OPINION OFFBEAT VIDEO

Delhi'ites Will Get Paid For Unscheduled Power Cuts, Says Arvind Kejriwal

Arvind Kejriwal meets Lieutenant Governor Anil Baijal to discuss plans to make discoms pay for unscheduled power cuts in Delhi

Delhi | Reported by Sonal Mehrotra, Edited by Anindita Sanyal | Updated: June 07, 2017 11:21 pm IST

Sponsored

Ready to Move in 3 BHK @ Rs.29.99 Lacs in Chenn

NDTV Delhi'ites Will Get Paid For Unscheduled Power Cuts, Says Arvind Kejriwal

After the meeting with Mr Baijal, the Chief Minister tweeted:

Arvind Kejriwal @ArvindKejriwal

Hon'ble LG agreed DISCOMS shud pay compensation 4 unscheduled power cuts 2 make them accountable. Fairplay n equity 2 consumers 2 be ensured

3,849 5:36 PM - Jun 7, 2017

2,473 people are talking about this

The Lieutenant Governor, though had a rider -- that the Delhi government develop a "rational,fair & enforceable model"

LG Delhi @LtGovDelhi

Met CM @ArvindKejriwal .Discussed issue of power cuts & compensation to consumers. Suggested to devise a rational,fair & enforceable model.

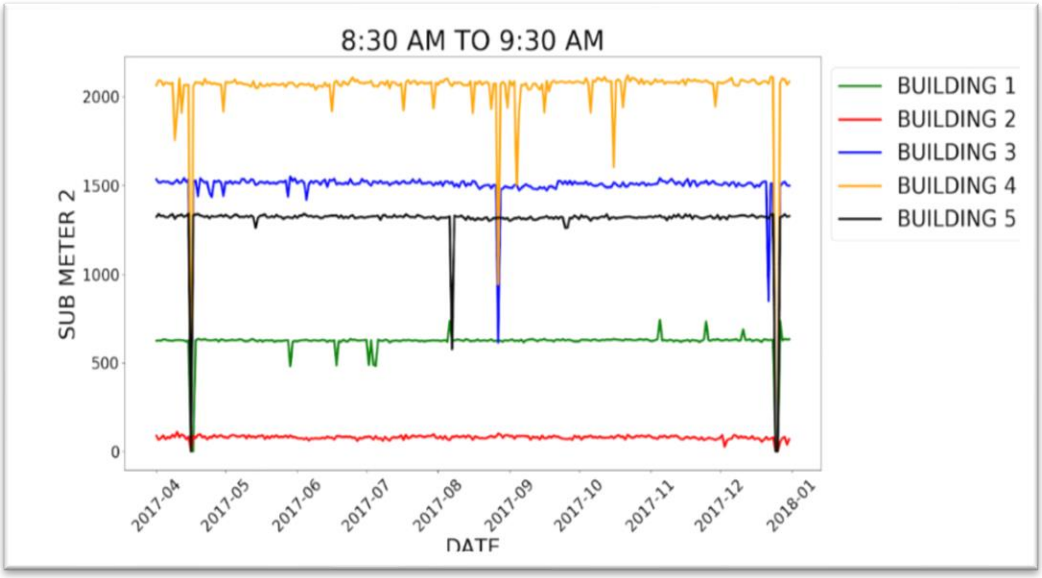
1,510 5:31 PM - Jun 7, 2017

893 people are talking about this

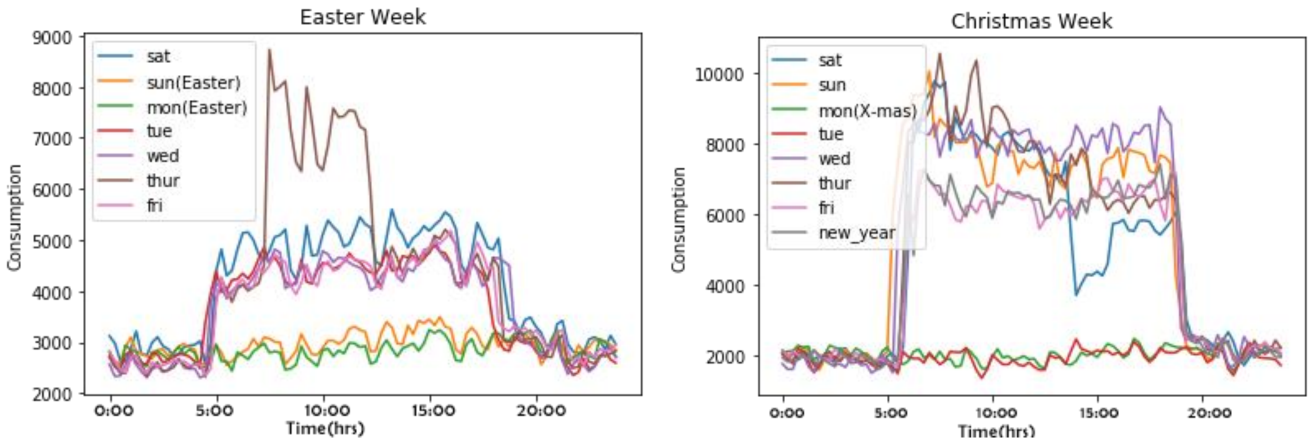
2. **A trend observed in sub_meter_2:**

The reading of sub_meter_2 from **8:30 AM to 9:30 AM** is almost constant over the year for all buildings except building 4.

For building 4, the same trend was observed in sub_meter_1 in place of sub_meter_2.



3. It was observed that consumption of electrical units on Christian holidays (like Easter & Christmas) were very low compared to other days (or holidays like Dussehra or Muharram).



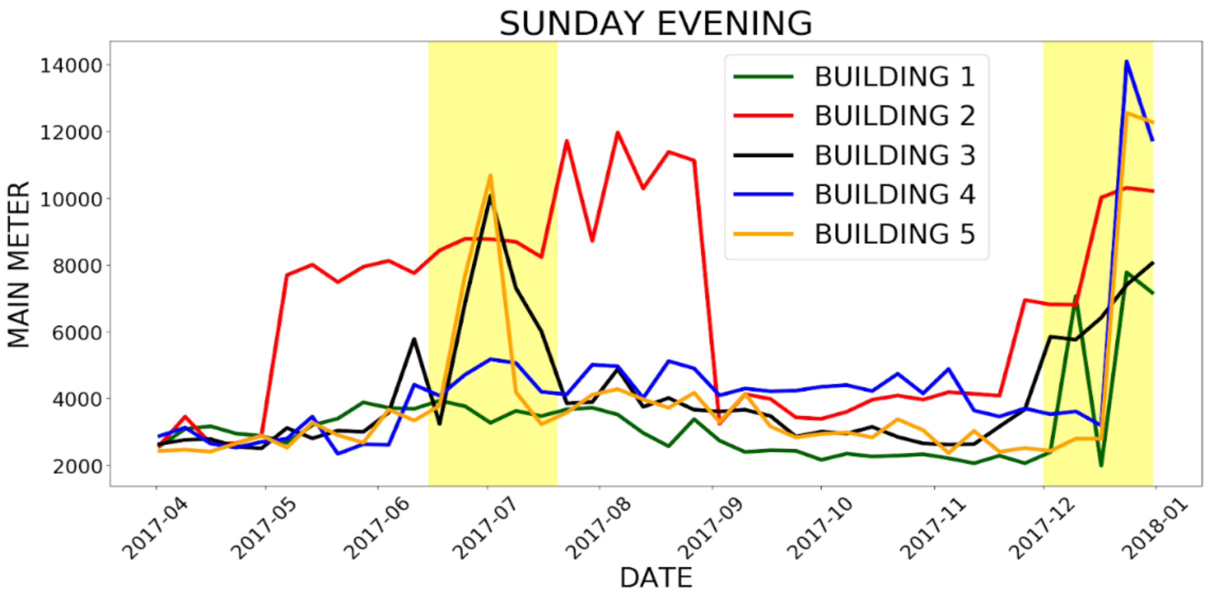
Plot for Building 1 on Christian holidays

4. **Weekend analysis in winter (for Building 3):**

We plotted day-wise trends every week at a particular time (17:00) and observed the following:

- On Sunday, from mid-June to July there was a sharp increase in value (to about 10000) and from July onwards decreased to some lower range (to 5000 range).

- At the end of the year, the work done on weekends increased a lot (at 17:00) Sunday values reached around 8000 but the reading decreased as the year ended for the rest of the days.



Model Description

1. FB PROPHET

Prophet is open source software released by Facebook's Core Data Science team. Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

2. ARIMA

ARIMA, short for '**Auto-Regressive 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. Any 'non-seasonal' time series that exhibits patterns and is not a random white noise can be modelled with ARIMA models. An ARIMA model is characterized by 3 terms: p, d, q where,

p is the order of the AR term

q is the order of the MA term

d is the number of differencing required to make the time series stationary.

3. LIGHT GBM

Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm. It splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. So, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Also, it is surprisingly very fast, hence the word 'Light'.

4. LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture. Unlike standard feedforward neural networks, LSTM has feedback connections. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell.

5. XG BOOST

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting that solves many data science problems in a fast and accurate way.

6. VAR

Vector autoregression (VAR) is a stochastic process model used to capture the linear interdependencies among multiple time series. VAR models generalize the univariate autoregressive model (AR model) by allowing for more than one evolving variable. All variables in a VAR enter the model in the same way: each variable has an equation explaining its evolution based on its own lagged values, the lagged values of the other model variables, and an error term.

7. SARIMA

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

8. Linear Regression

The linear regression algorithm is one of the fundamental supervised machine-learning algorithms due to its relative simplicity and well-known properties. It is a linear approach to modelling the relationship between a dependent variable and one or more independent variables.

9. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

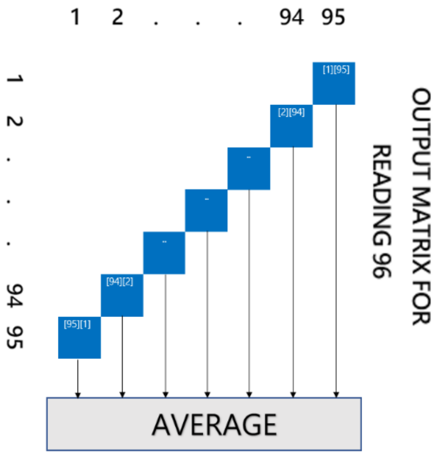
10. Deep Neural Network

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

11. CNN LSTM

The CNN Long Short-Term Memory Network or CNN LSTM for short is an LSTM architecture specifically designed for sequence prediction problems with spatial inputs, like images or videos.

This involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction.



Software Stack

1. **Python 3.6** - Language of choice.
2. **Pandas** - For handling files.
3. **Numpy** - For complex numerical analysis.
4. **Seaborn** - Plotting advanced visualizations.
5. **Matplotlib** - Plotting visualizations.
6. **Sklearn** - For making machine learning models.
7. **XGBoost** - For using boosting models.
8. **Datetime** - Handling date-time data types.
9. **LightGBM** - Tree-based gradient boosting framework
10. **FBProphet** - Forecasting time series data

References

- XGBoost Classifier:
  https://xgboost.readthedocs.io/en/latest/index.html

- Gradient Boosting Classifier:
  https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

- Linear Regression:
  https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

- Label Encoder:
  https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

- One Hot Encoder:
  https://medium.com/@contactsunny/label-encoder-vs-one-hot-encoder-in-machine-learning-3fc273365621
  https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

- Random Forest Classifier:
  https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

- Grid Search:
  https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
  https://medium.com/@senapati.dipak97/grid-search-vs-random-search-d34c92946318

- Box Plot:
  https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html

- Line plot
  https://matplotlib.org/3.1.0/api/_as_gen/matplotlib.pyplot.plot.html

- Pie Chart
  https://matplotlib.org/api/_as_gen/matplotlib.pyplot.pie.html

- Bar Graph
  https://matplotlib.org/api/_as_gen/matplotlib.pyplot.bar.html

- Density Plot
  https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.density.html

- FB Prophet
  https://facebook.github.io/prophet/docs/quick_start.html

- Arima
  https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

- LSTM
  https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

- Light GBM
  https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

- VAR
  https://www.machinelearningplus.com/time-series/vector-autoregression-examples-python/

- SARIMA
  https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

- Auto-Encoder
  https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f

- PCA
  https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial

- Backward Elimination
  https://towardsdatascience.com/backward-elimination-for-feature-selection-in-machine-learning-c6a3a8f8cef4

- Deep Neural Network (DNN)
  https://www.sciencedirect.com/topics/engineering/deep-neural-network