CITY OF BOSTON BUILDING POWER CONSUMPTION DATASET

Website Link: http://vishwashah.pythonanywhere.com/

Assignment 3: City of Boston Power Consumption

Contents

About Dataset:

The dataset contains power consumption for various buildings in the City of Boston. The dataset has 5 minute interval data for each day of the year 2014.
The building is identified by its account number.
The unit for power is kwh,kvarh and power factor.
Data also contains the Channel name for the power consumption values.

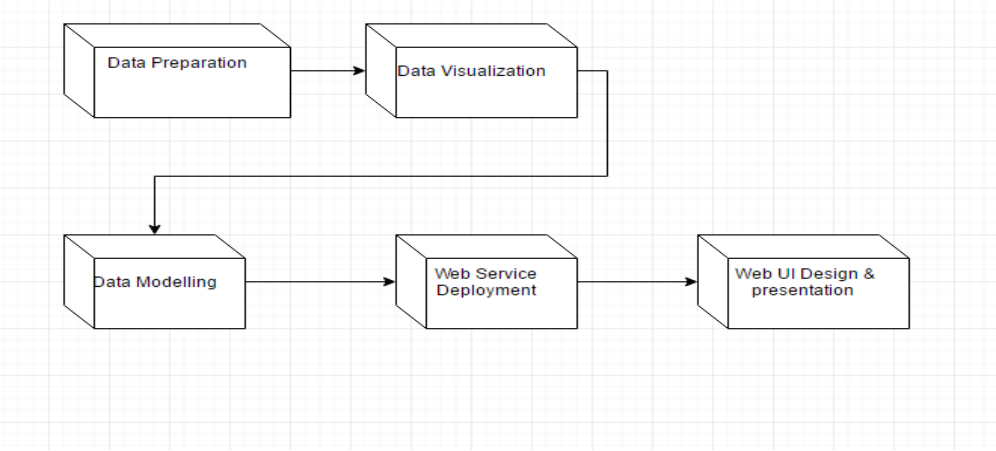
Problem Statement

The city of Boston has different types of building each of them having different kind of power needs. The Boston Power Department would want to monitor and forecast the need of these building to provide the required power to these building in future in an economic way. Also the buildings would want to monitor their power consumption and find out reasons for high consumption so that they can take preventive measures to conserve power. What will we do to help them?

1. Meter Data Monitoring
2. Micro and Macro Analysis of the buildings in Boston
3. Future power demand forecast
4. Decision making on power consumption optimization

All this will be available to the user in form of Website where they can check their usage for the year 2014 which will be displayed in form of intuitive visualizations. They will also be able to forecast their future usage by simply entering their Account number and the date they want to forecast the usage for.

Basic Deployment Pipeline



Data Preparation & Cleansing

Tools Used: Alteryx and Python script was used for preprocessing and cleaning the data.

Steps: (step1-2 are done using python, step 3

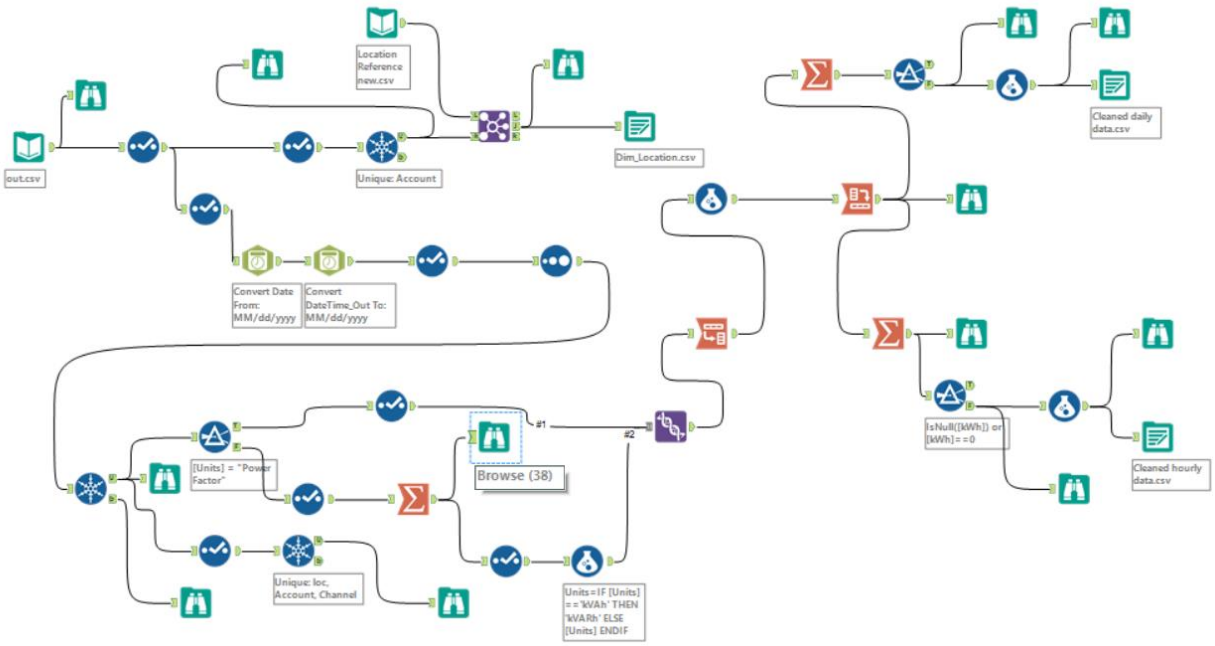
1. Extract site location information from all csv files' name. Location name and location type has been extracted, which will be used to attach to the merged dataset and create location dimension table.
2. Attach location name to each csv file and merge all csv files into one csv file.

In [11]:
```
import glob
import pandas as pd
filenames = glob.glob('.\Desktop\INFO 7390 ADS\Assignment3\CY2014 COB Interval data 1/*.csv')
filenames1 = glob.glob('.\Desktop\INFO 7390 ADS\Assignment3\CY2014 COB Interval data 2/*.csv')
f = filenames+filenames1
frame = pd.DataFrame()
list_ = []

for index in range(len(f)):
    file_ = f[index][63:(f[index].rfind('.csv', 0, len(f[index])))]
    if file_[-2:] == '.2' or file_[-2:] == '.1':
        file_ = file_[0:-2]
    if file_[-5:] == ' 2014' or file_[-5:] == '.2014':
        file_ = file_[0:-5]
    if file_[:7] == 'Agassiz':
        file_ = 'SCH.Agassiz'
    else:
        file_ = file_[4:]
    if file_[-11:] == '.2014.1 (2)' or file_[-11:] == '.2014.2 (2)':
        file_ = file_[:-11]
    if file_[-9:] == '.2014 (2)':
        file_ = file_[:-9]
    if file_ == 'SCH.ArtsJan.Dec2014':
        file_ = 'SCH.Arts'
    if file_[:14] == 'SCH.Blackstone':
        file_ = 'SCH.Blackstone'
    point = file_.find('.', 0, len(file_))
    loctype = file_[:point]
    loc = file_[point+1:]
    df = pd.read_csv(f[index],index_col=False, header=0)
    df = df.rename(columns={'24:00:00': '24:00'})
    for item in df:
        df['loctype']=loctype
        df['loc'] = loc
    #df.fillna('0')
    list_.append(df)

frame = pd.concat(list_)
for item in frame:
    frame.fillna('0')
frame.to_csv('.\Desktop\INFO 7390 ADS\Assignment3\outnew.csv', sep=',')
```

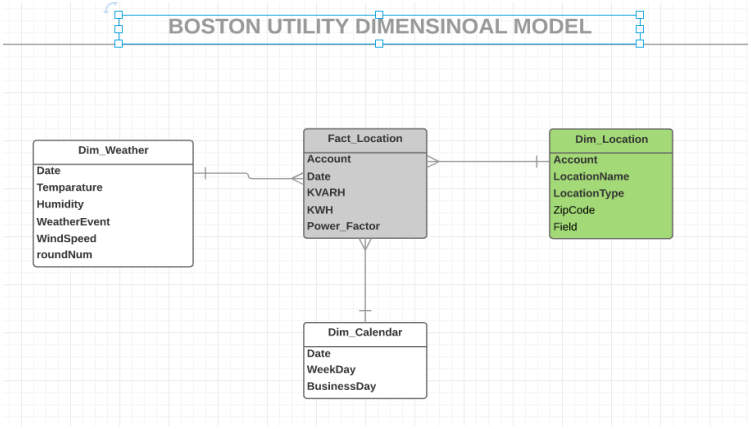
3. Collect external data to build extra dimensions and provide more predictors for modeling purpose. Weather data, geo data and calendar data have been collected from multiple data sources[1].
4. Clean and transform the data from step 2 and step 3 using Alteryx.
    a. Generate location dimensions and integrate account information.
    b. Delete duplicated rows of same account, same date and same unit when multiple rows of data for same unit containing exact same numeric values.

[1] Weather data: https://www.wunderground.com
Calendar: http://www.data.gov/
Location: https://www.google.com/maps

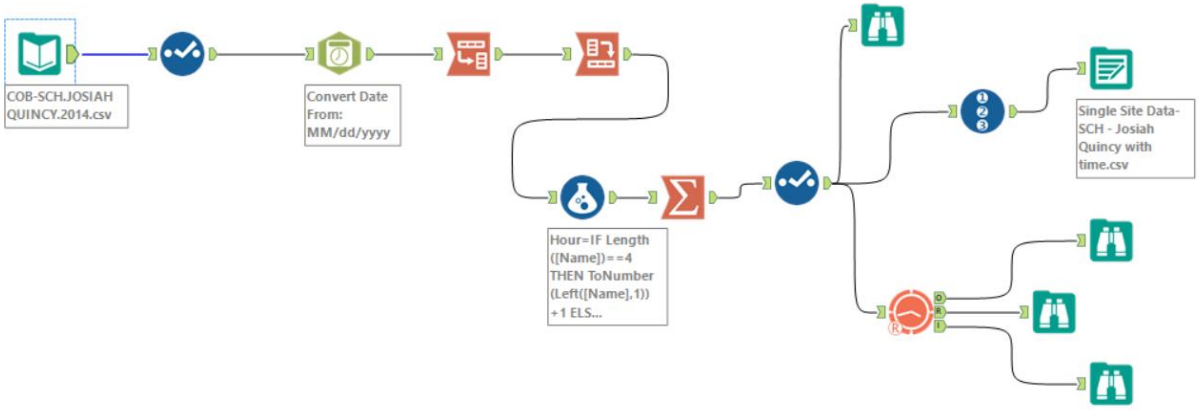
c. Combine values of same account, same date and same unit when multiple rows of data containing multiple channels.
d. Replace missing values with 0 for kVARrh & kWh, replace missing values with 1 for Power_Factor.
e. Transpose all time interval data into two columns (time&value)
f. Transpose multiple units values into three columns. (time&kWh&kVARh&PowerFactor)
g. Summerzise 5min/15min interval data into hourly data & daily data.
h. Export two tables to two different csv files for visualization and modeling purpose.



5. Dimensional model with a star schema was developed through step 1-4, which will save storage when the fact data set grows. It also meets visualization tools schema requirements.

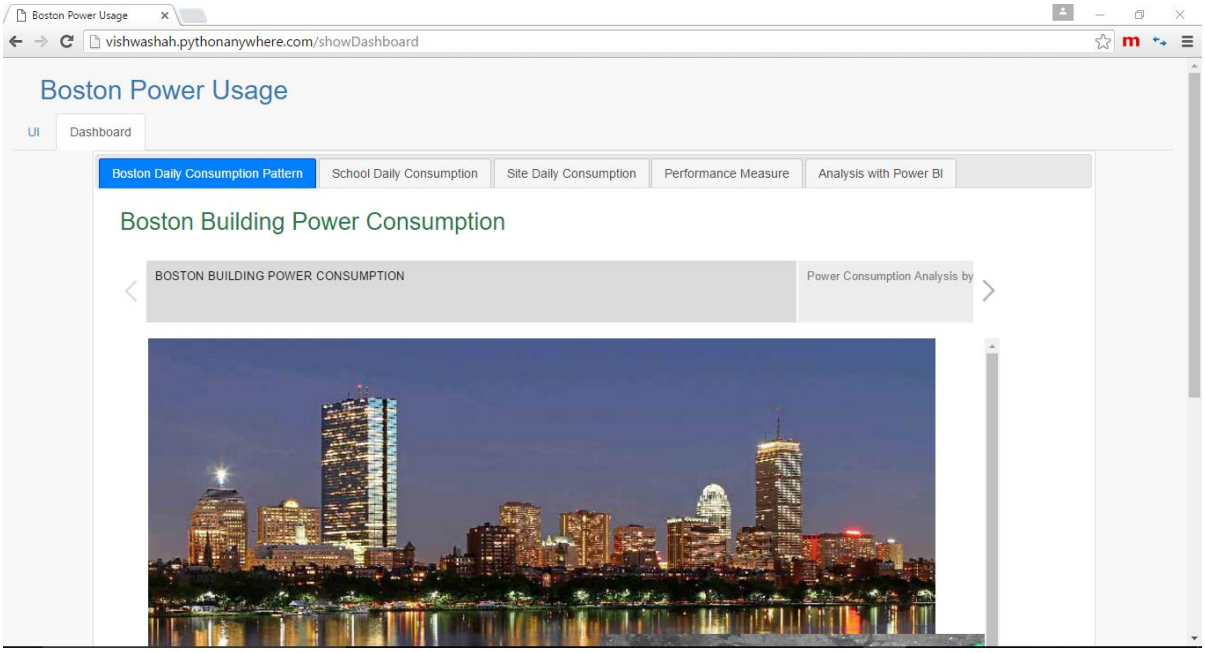


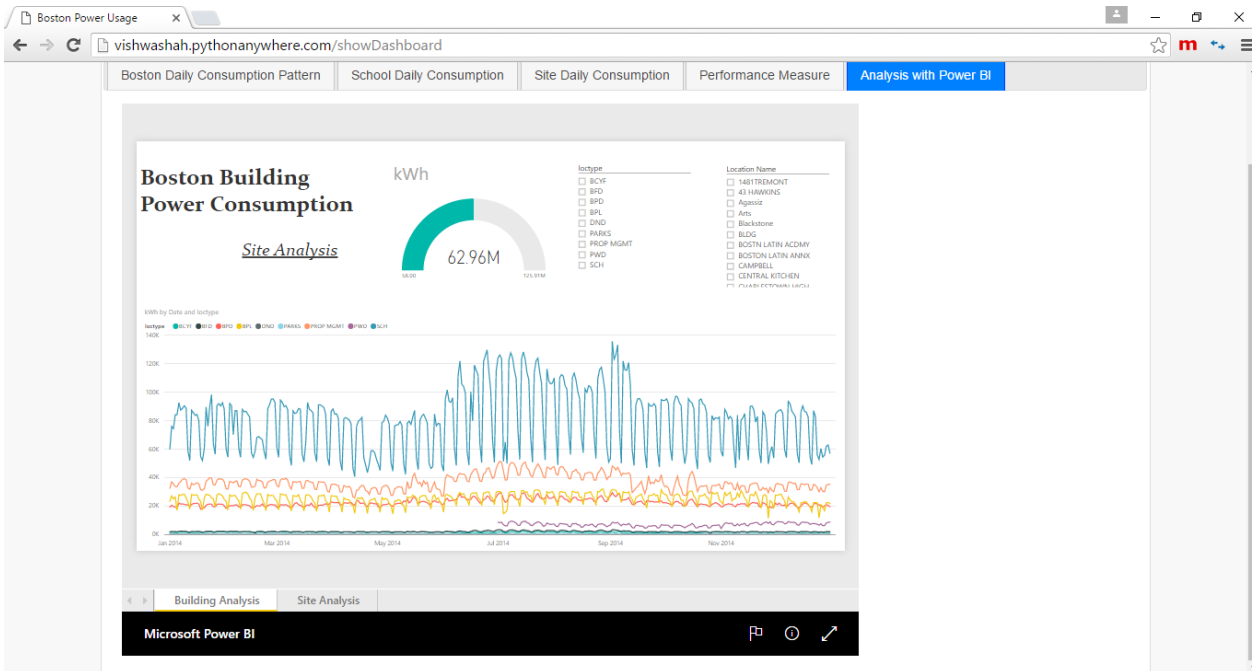
6. In order to explore alternative models, a sample data set has been created with 5 minutes interval data to test time series models. The site we choose was Josiah Quincy public school. Similar functions has been used for data preprocessing and cleaning purposes.



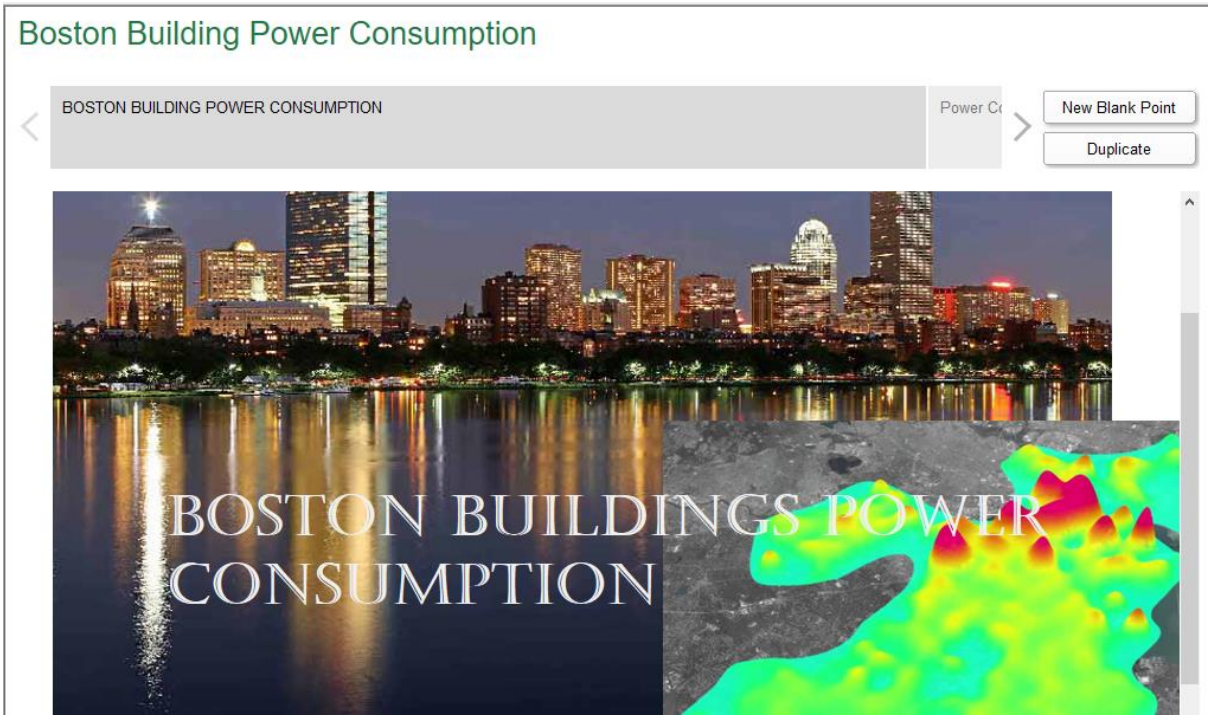
Data Visualization

Tools Used: Tableau and Power BI
Tableau Public was used to publish the visualizations on Website. Power BI Visualizations were published online and embedded on Website using Iframe.

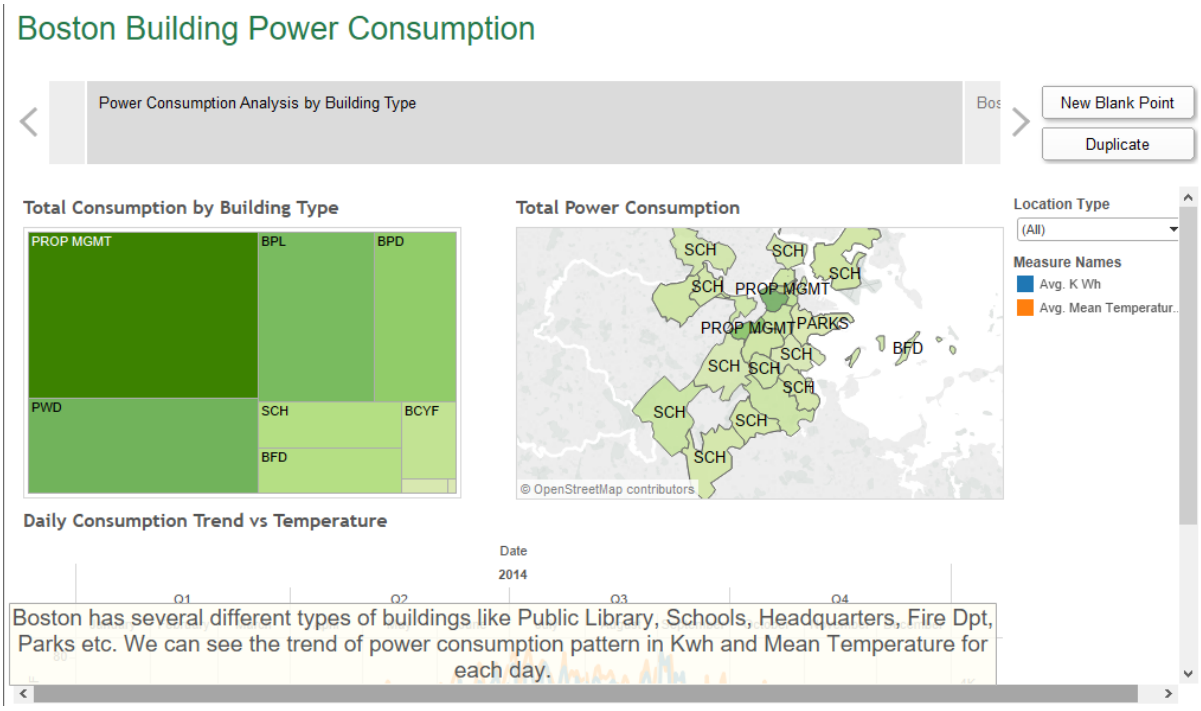




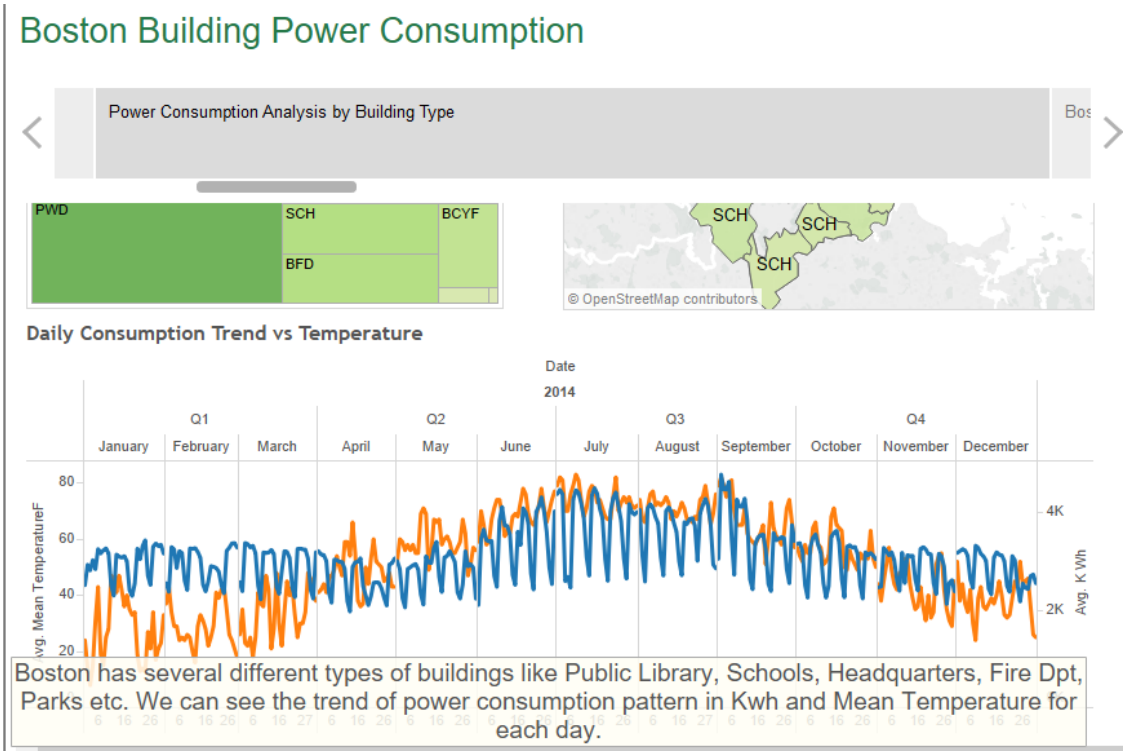
All Tableau Visualizations are in form of Storyboard and explain a story.



The following is the story for all the building in Boston(Macro Analysis)

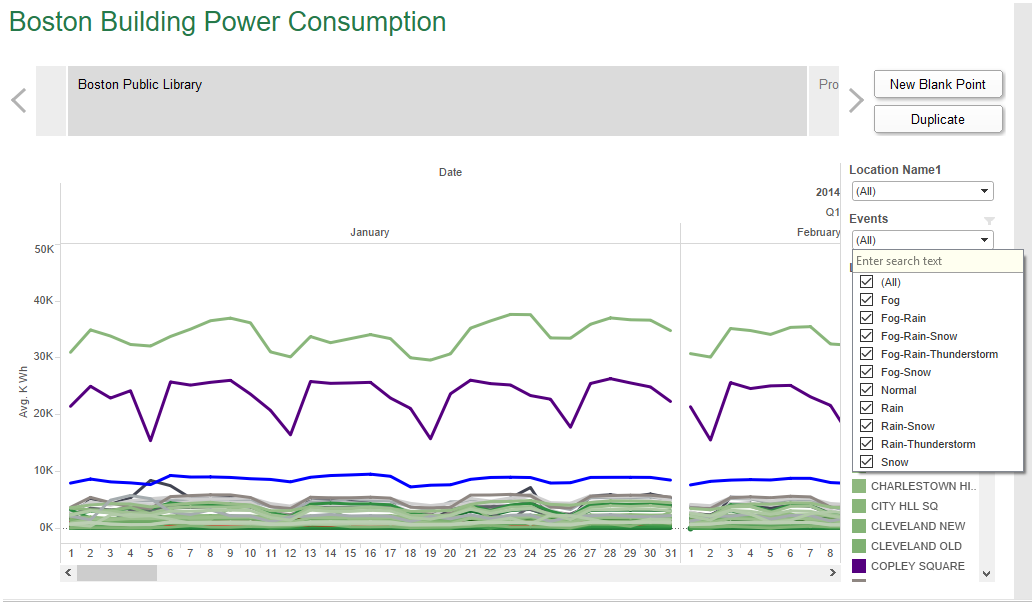


Using Google maps we got in the location variable and plotted it on the map. The map and heat map show which type of location has what power consumption pattern in Kwh. Darker and bigger the area higher is the power consumption. Here you can also filter by

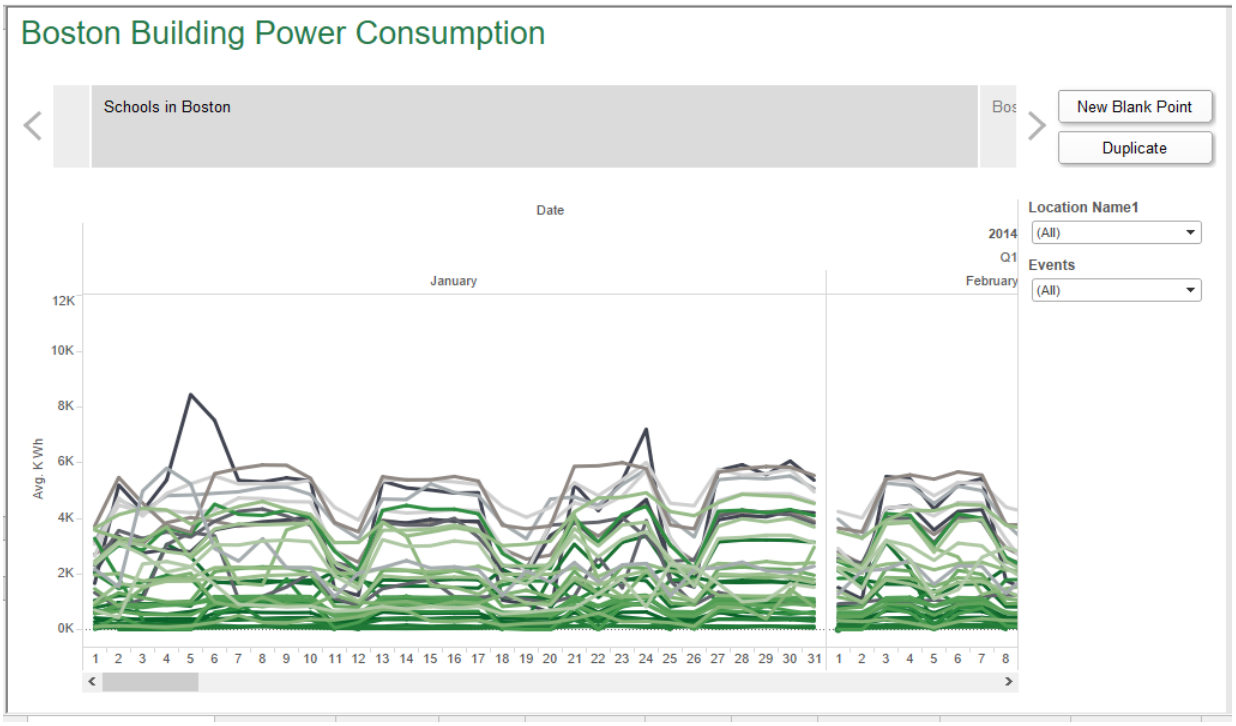


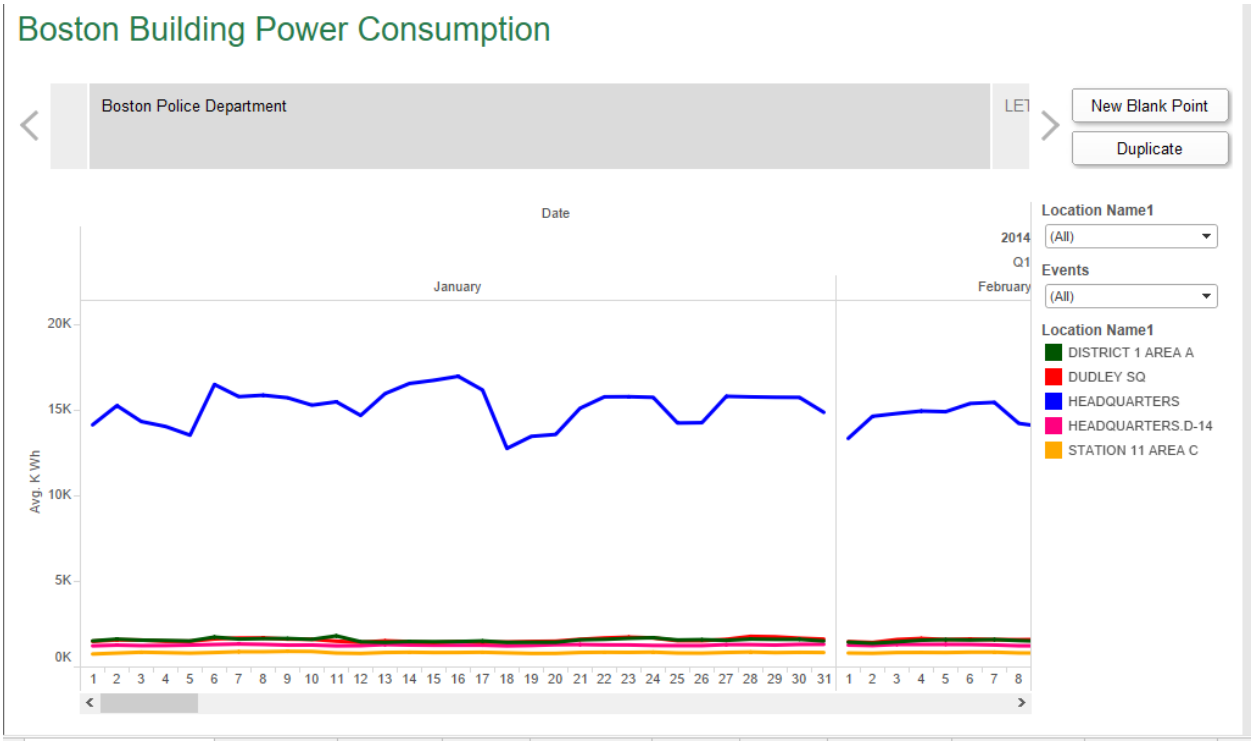
Location type to view only that particular location type consumption.

This time series above shows the total daily avg. consumption in kwh in city of Boston against mean daily temperature. You can filter this too by location type to check the trend of each location.

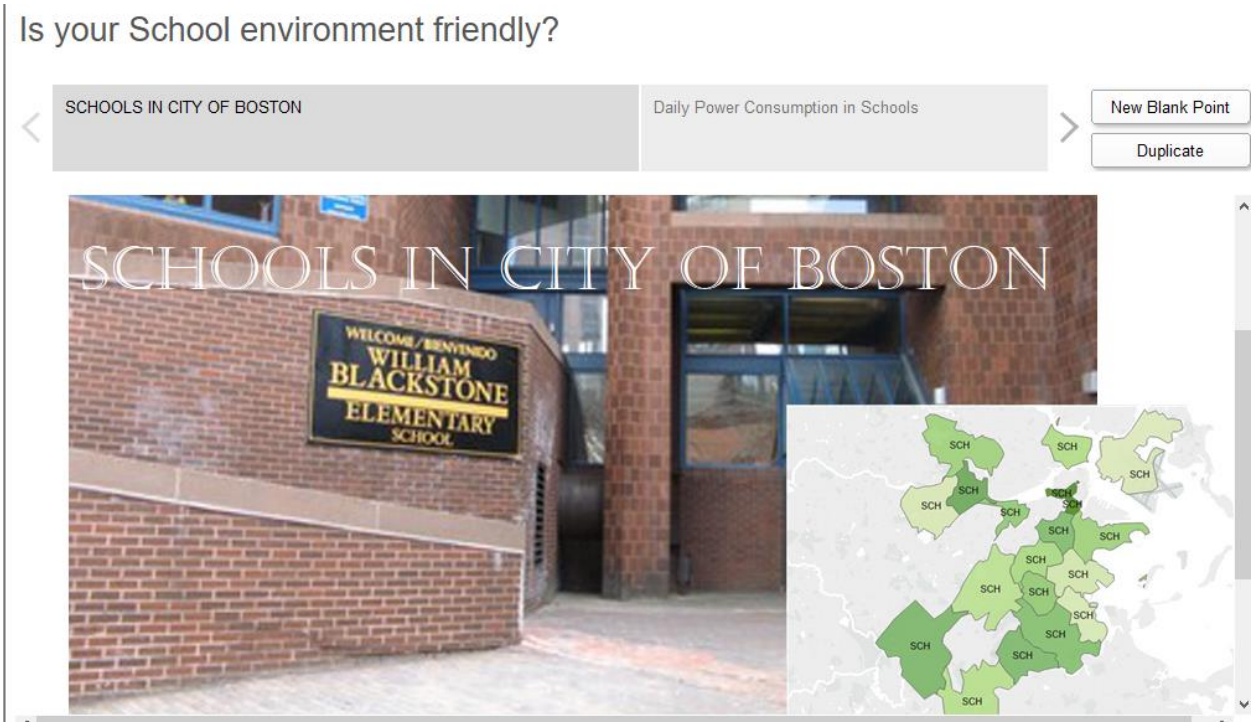


In the above visualization we can see the time series for Boston Public Library buildings. There are many such types of building and you can filter on one or many of them to check the trend of consumption on daily basis. Here we have also added event filter which helps us check the changes in trend with respect to the events like snow, rain etc.
Similar visualizations are prepared for different type of building ie. School, BPD.

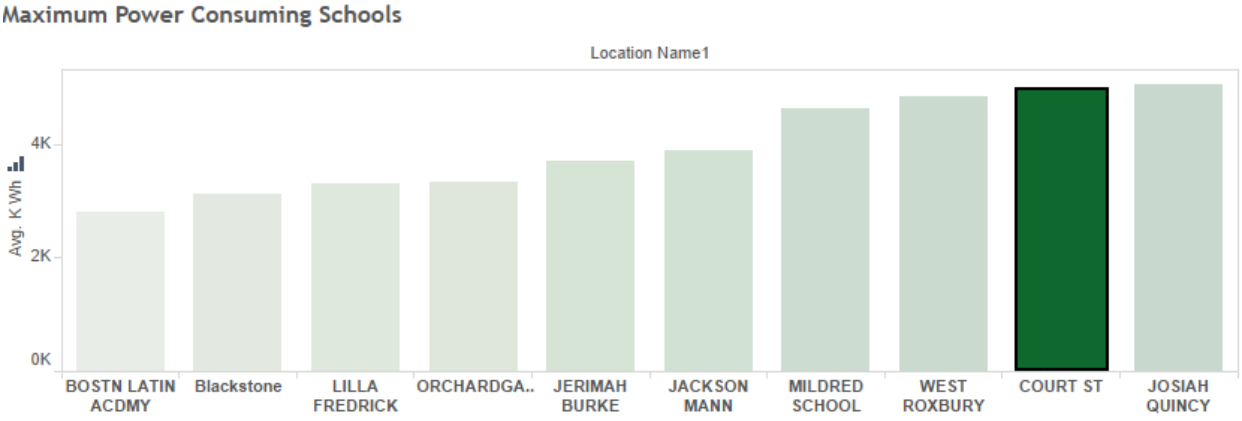




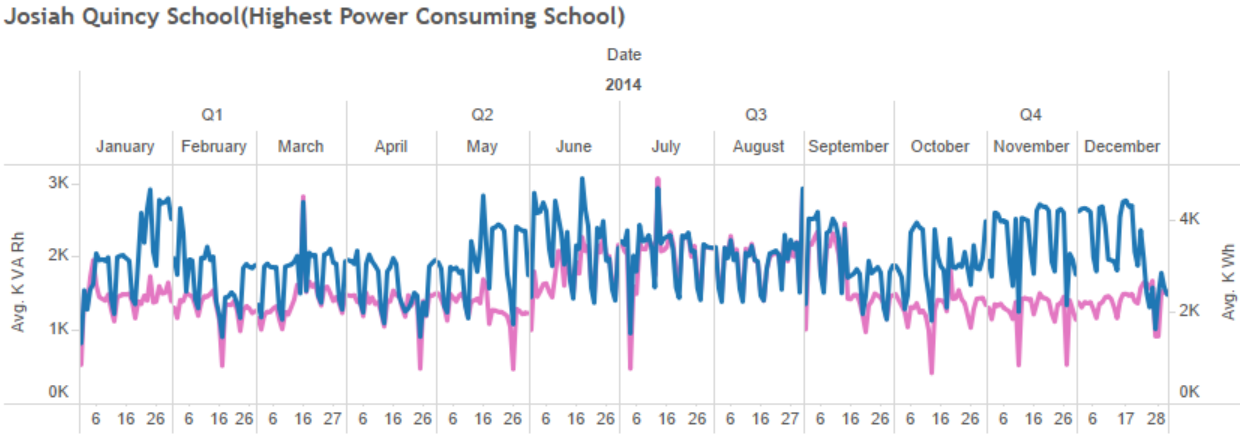
We have many buildings in the type schools and so we picked this type for Micro Analysis



Top 10 power consuming schools



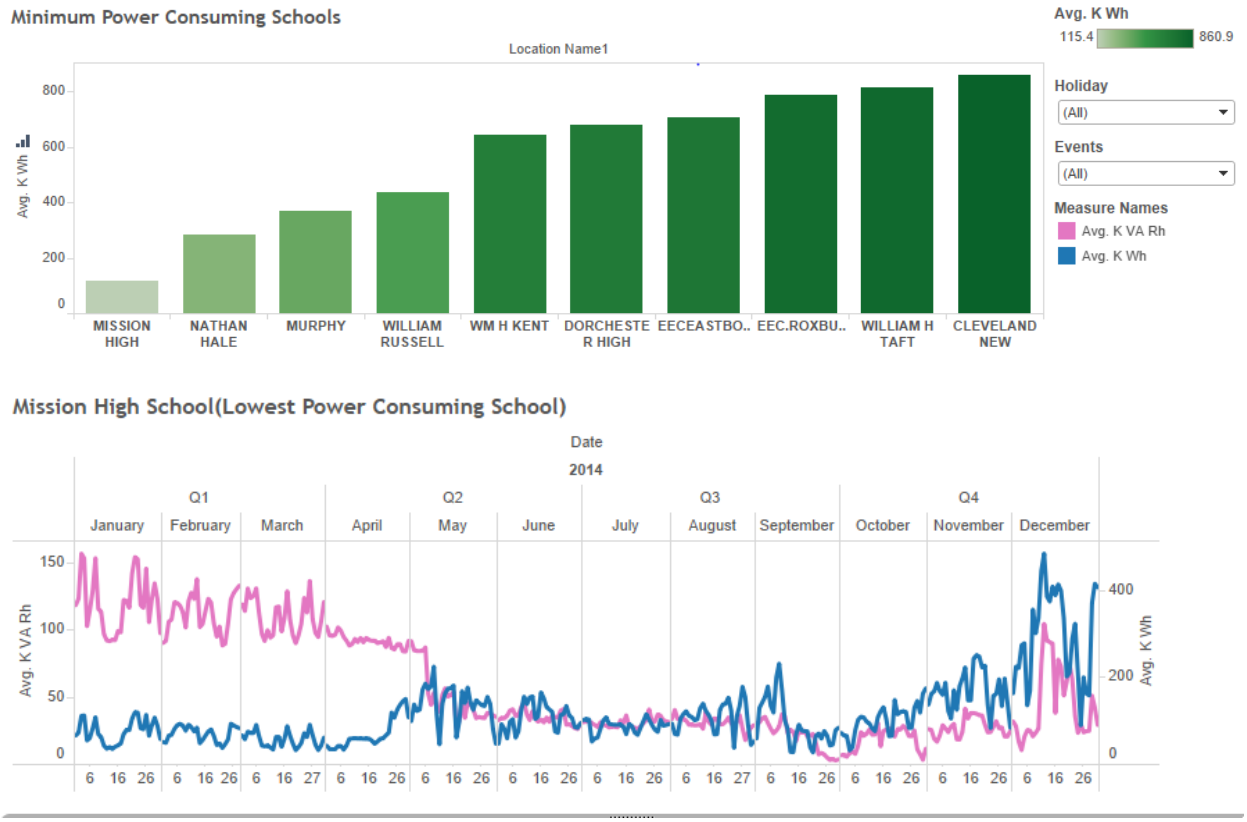
Now we see the time series for the highest power consuming School



We have added filter by Holiday and Events. We can see on filtering on holidays the power consumption is comparatively lower than on non holidays.

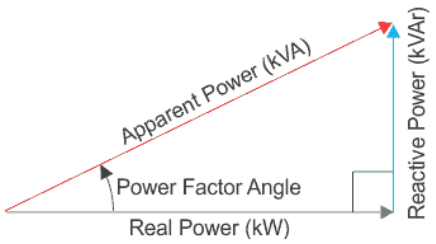
Similar analysis is done for the bottom 10 schools in power consumption too.

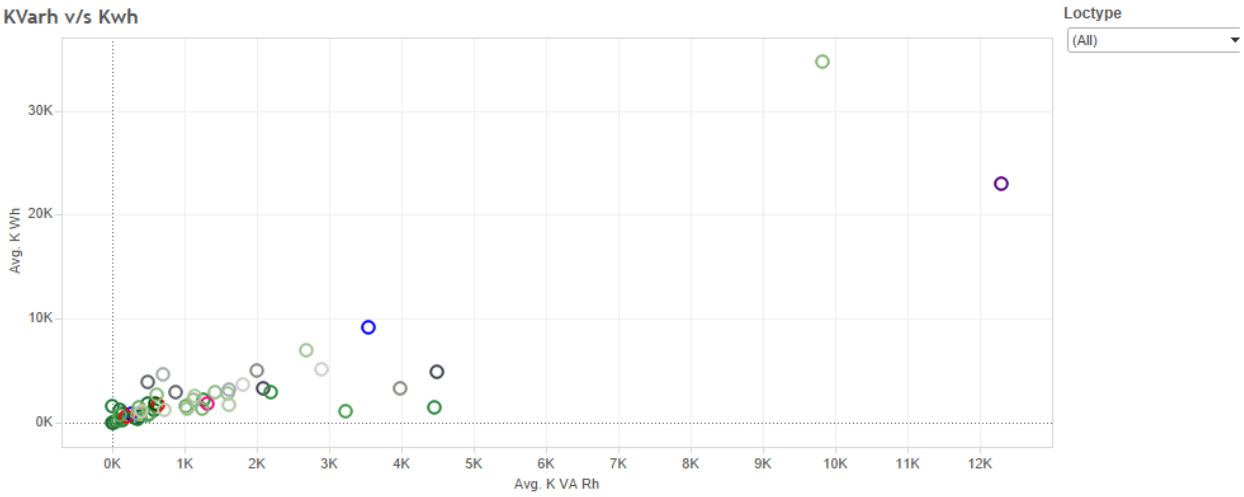
Bottom 10 Power consuming Schools

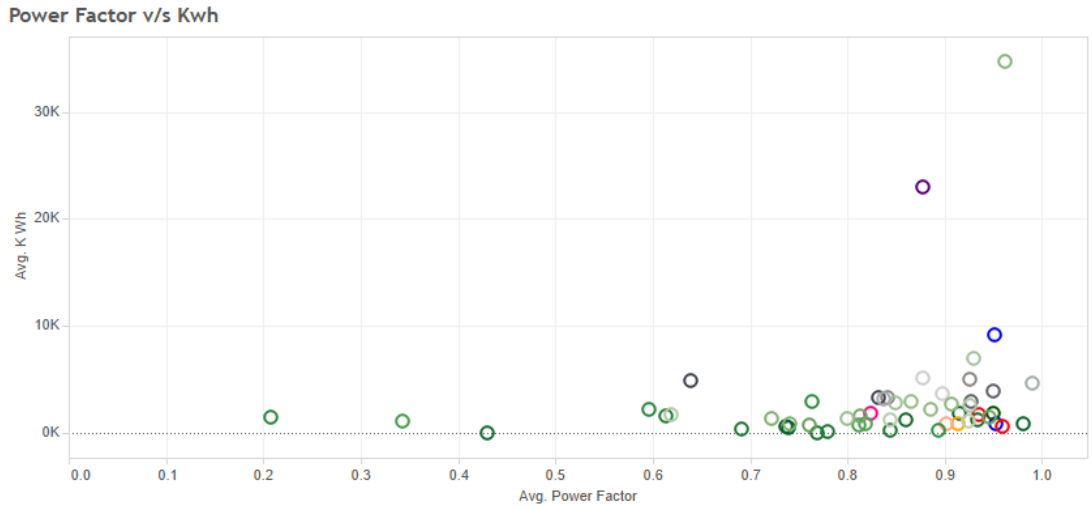


Then we used Power Factor as performance measure and analyse data based on power factor.

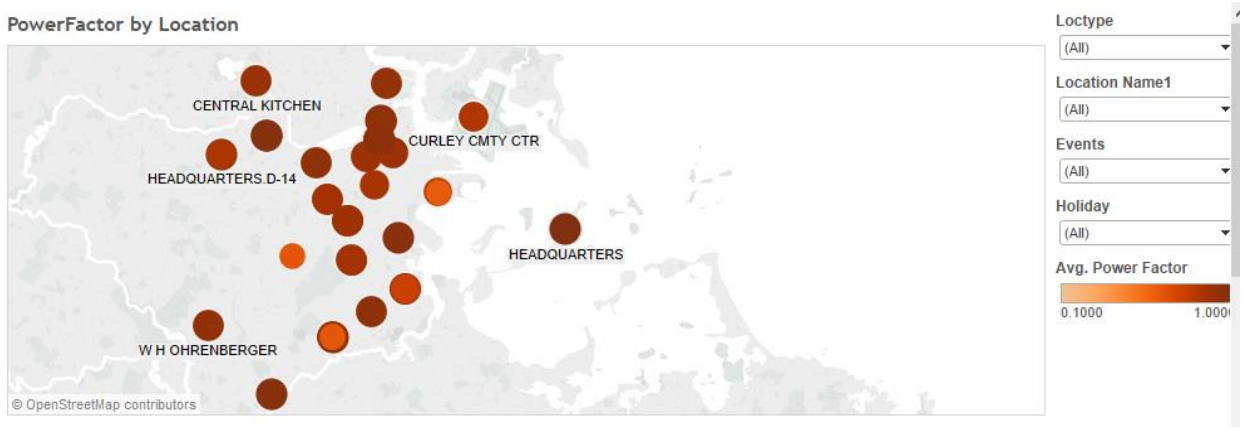
PERFORMANCE MEASUREMENT



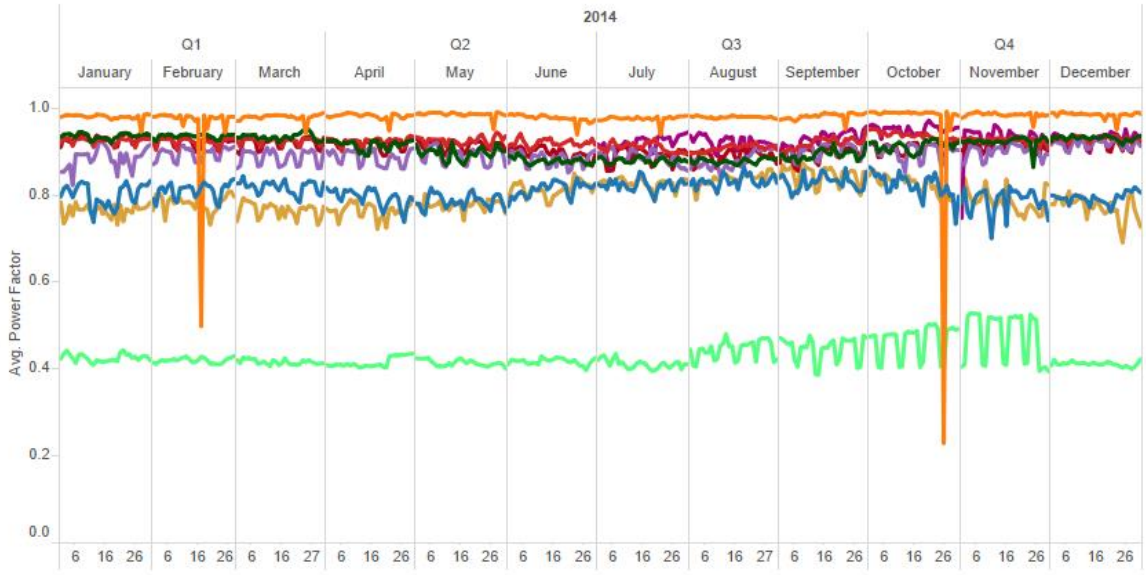




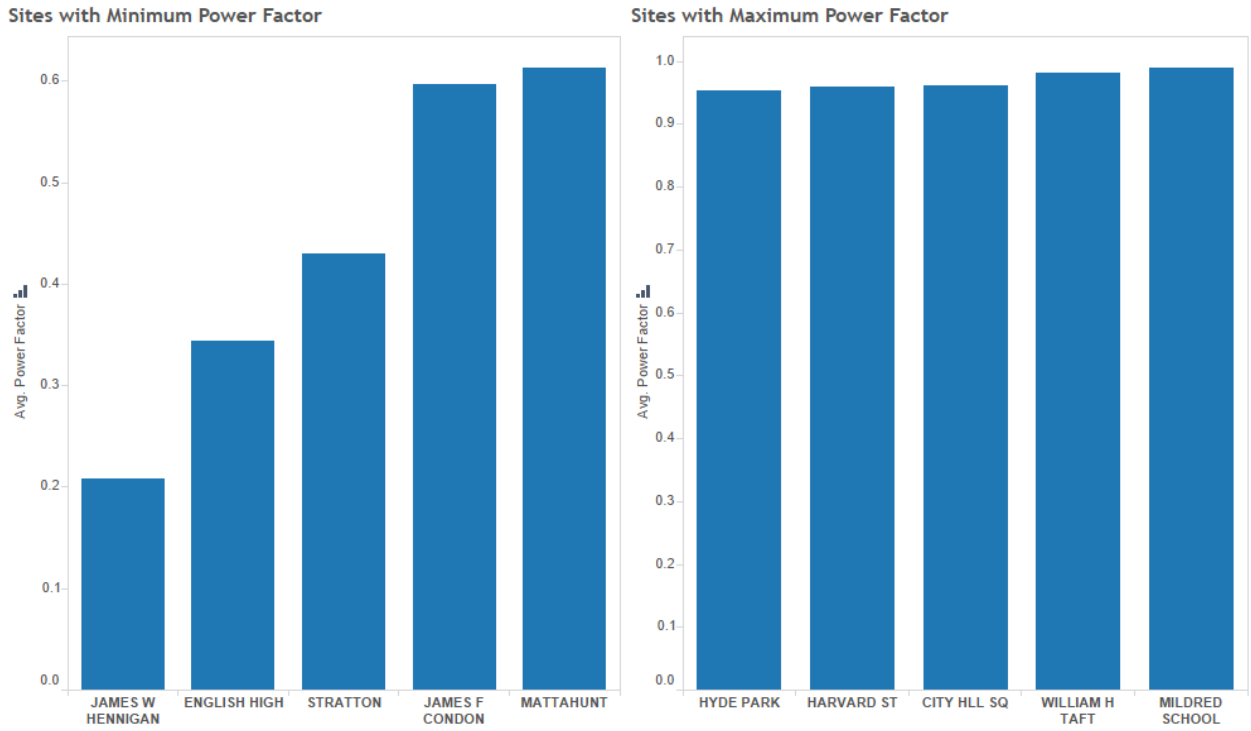
From the above scatter plot we can see the relation between the Kwh and kvarh and kwh and power factor respectively in each location type.



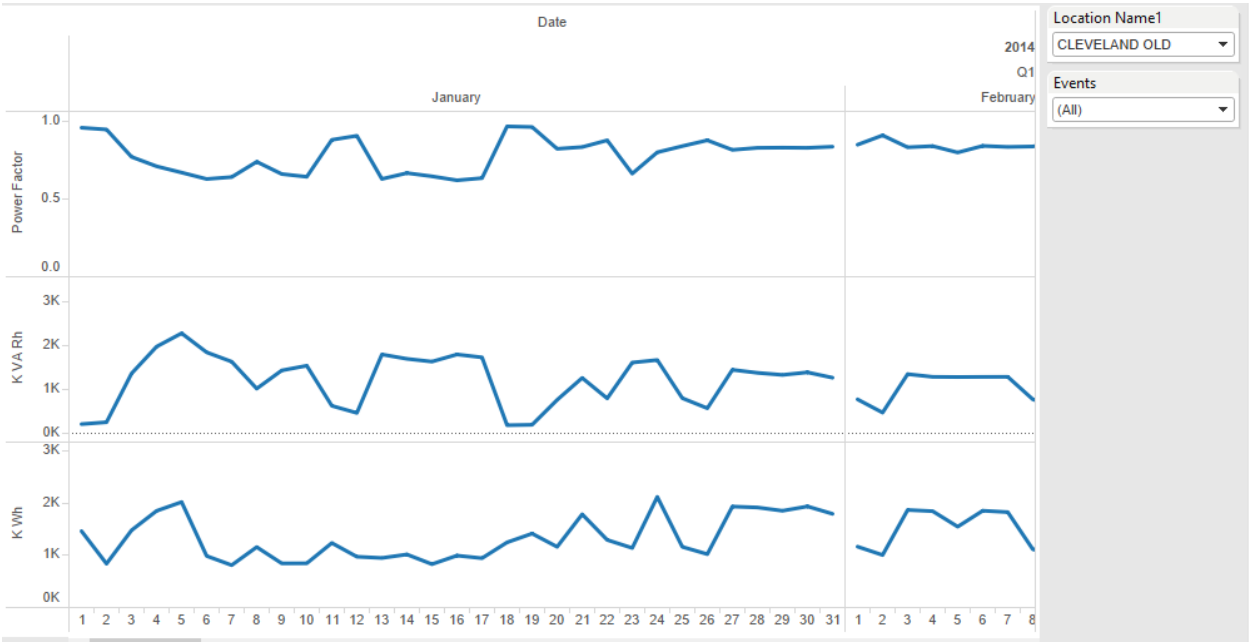
The above visualization shows us the power factor of each location on the map.



The above visualization is the power factor trend by location type. We can drill down to location name. We can filter by various events, holidays and the location name and check the trend in power factor.



The above visualization shows the top and bottom 5 buildings by power factor.



The above visualization helps the user pick their building and see the time series of consumption over the period of year 2014.

Power BI Analysis(Similar to Tableau)

While working with Power BI there were few challenges we faced:

1. Making maps only with help of zip-codes isn't possible. Whereas in Tableau the zip code generates the Latitude and longitude so you can create maps with help of zip codes in Tableau.
2. Publishing your work in Power BI with your organizational account is not possible, so we made use of personal account to publish it and add it to the Website.

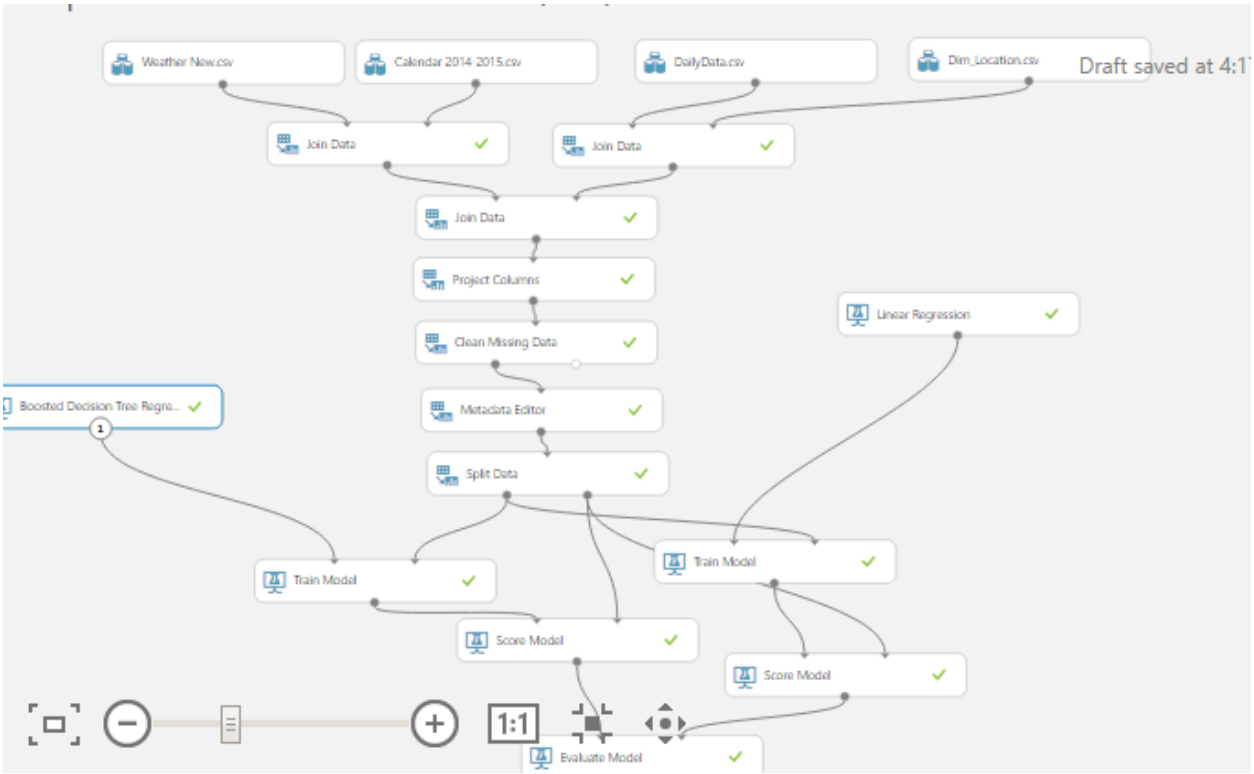
Data Modelling

Tools Used: Microsoft Azure ML Studio

Building the Model

Steps:

1. Join all the different csv with Weather, Location, Calendar and Daily Data using the Join Data component of Azure.
2. Joining these sources will create some duplicate rows so using project column component exclude these columns from the input.
3. Clean Missing Data: There are null values in Events which are replaced by Normal.
4. Split the data into 60% Training and 40% Testing.
5. Apply the 2 algorithms (Boosted Decision Tree and Linear Regression)
6. Train the model based on the algorithm
7. Score the model
8. Evaluate the model



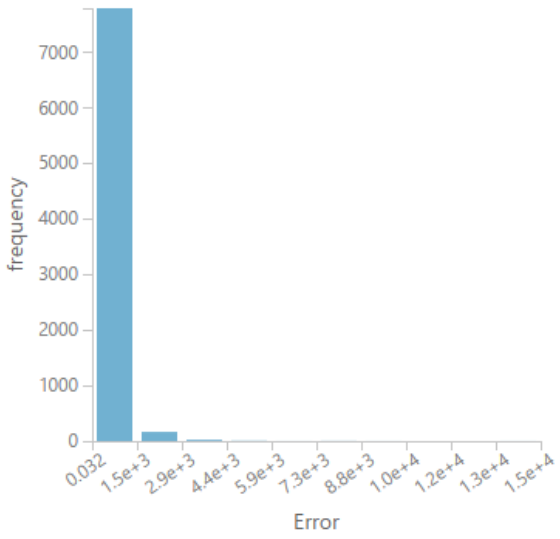
Algorithms used in the model:

- Boosted Decision Tree Regression
- Linear Regression

On evaluating the model we understand that the results in Boosted Decision Tree are better.

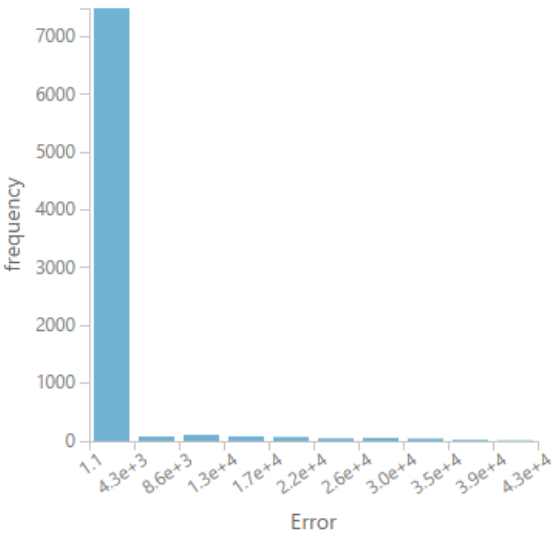
Metrics

Mean Absolute Error	329.914533
Root Mean Squared Error	591.228184
Relative Absolute Error	0.11476
Relative Squared Error	0.010877
Coefficient of Determination	0.989123



Metrics

Mean Absolute Error	2878.2292
Root Mean Squared Error	5592.83459
Relative Absolute Error	1.001187
Relative Squared Error	0.973295
Coefficient of Determination	0.026705



Alternative modeling method exploration

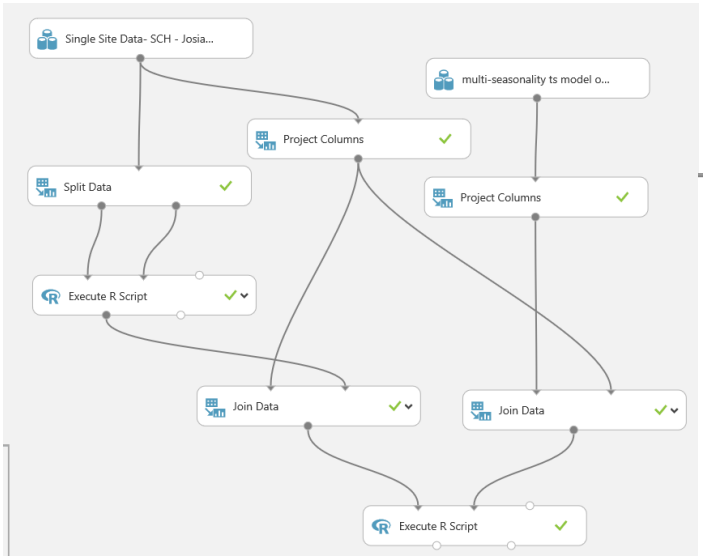
Tools: Microsoft Azure Machine Learning Studio, R Studio.

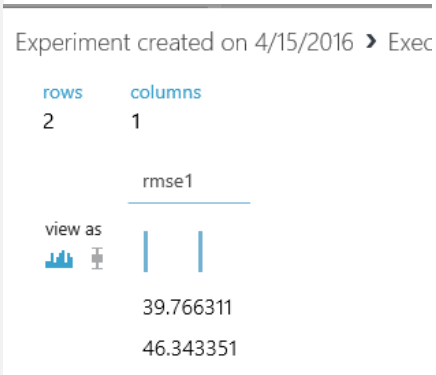
Time series models have also been used as alternative options to predict single site' power usage.

Algorithms used in the model:

- ARIMA
- TBATS

```
library("forecast")
file <- "C:/Users/vanwu/Desktop/INFO 7390 ADS/Assignment3/Single Site Data- SCH - Josiah Quincy with t
school <- read.csv(file, header = TRUE)
View(school)
tsdata <- school["kWh"]

##Modeling use first 10 month data.
tsdata1 <- head(tsdata, 7920)
tsdata2 <- tail(school, 840)
tsdata1 <- as.numeric(tsdata1$kWh)
View(tsdata2)
taylor <- msts(tsdata1, seasonal.periods=c(24,168))
View(taylor)
taylor.fit <- tbats(taylor)
plot(forecast(taylor.fit))

rmse()
##FORECAST THE rest of the data.

forr <- forecast(taylor.fit,h=840)
output <- data.frame(time = tsdata2$Time, forecast = forr$mean)
write.csv(output, file = "C:/Users/vanwu/Desktop/INFO 7390 ADS/Assignment3/multi-seasonality ts model
plot(forr)
```





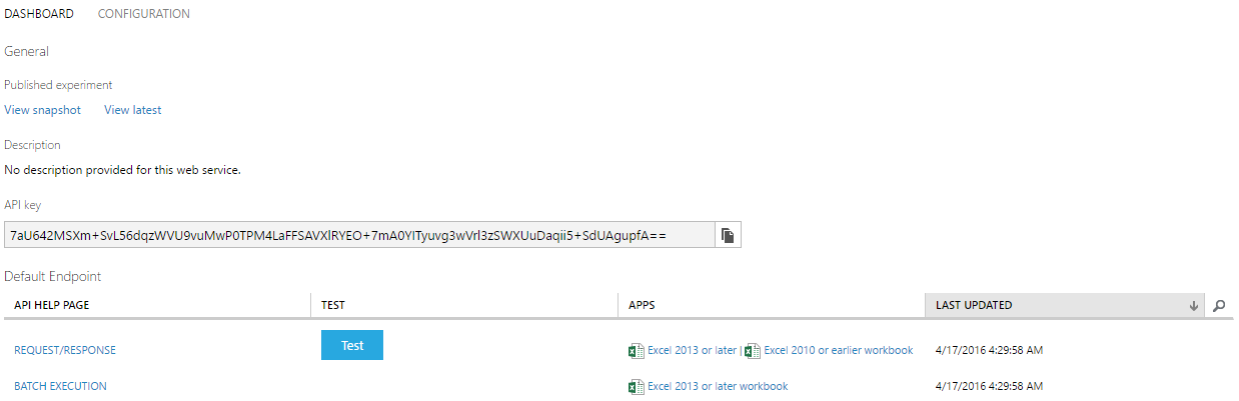
Results:

ARIMA (Autoregressive Integrated Mean Average) with a weekly seasonality has a lower RMSE value, which indicates that ARIMA gives a better performance.

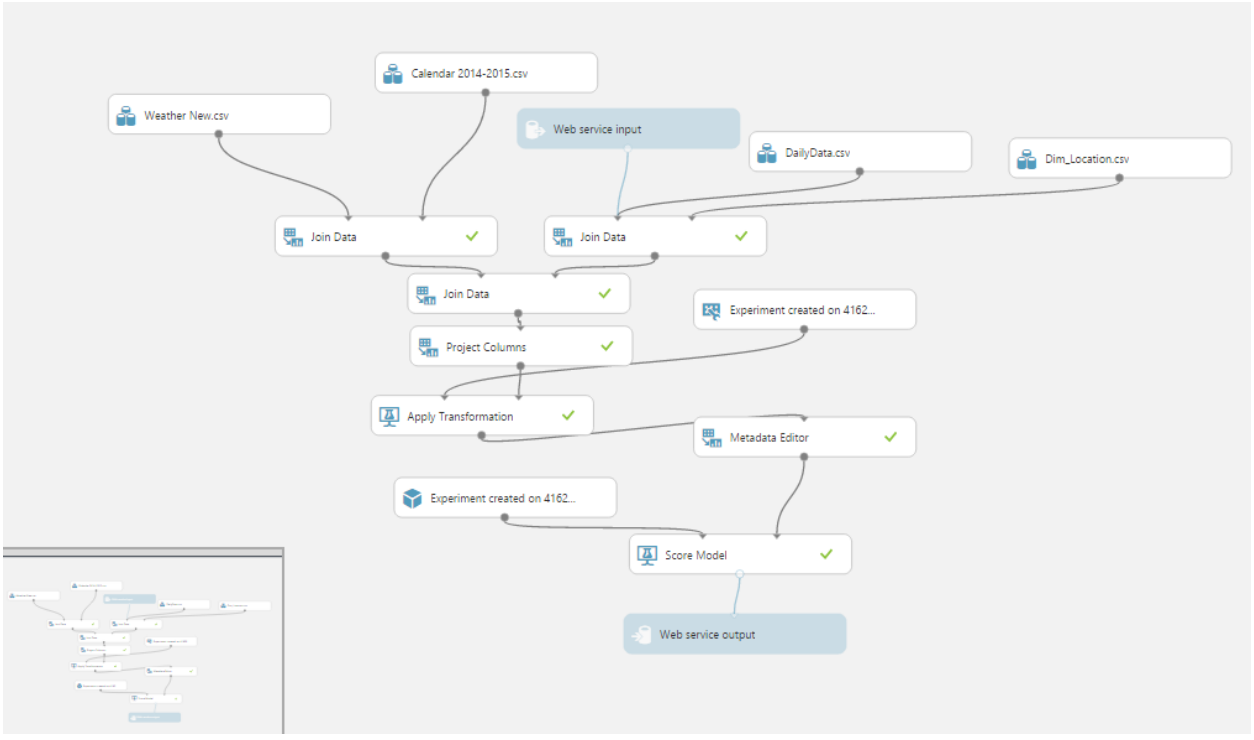
Deploying the Web Service

Steps:

1. The user inputs the Account Number and Date in the period of 2015 as we have our calendar data of year 2014-2015 and hence with the help of these two fields we predict the scored Kwh of a particular site. So as shown in the figure below the web service input is pointed to the data having Account and Date as input



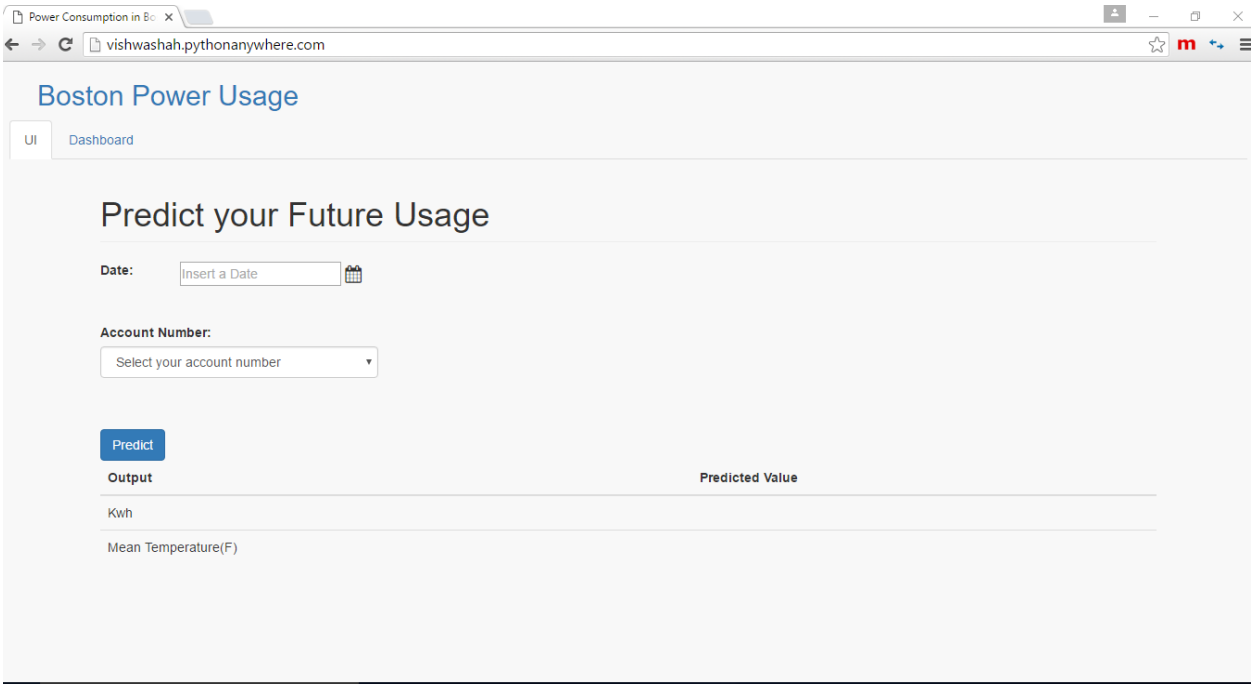
fields.

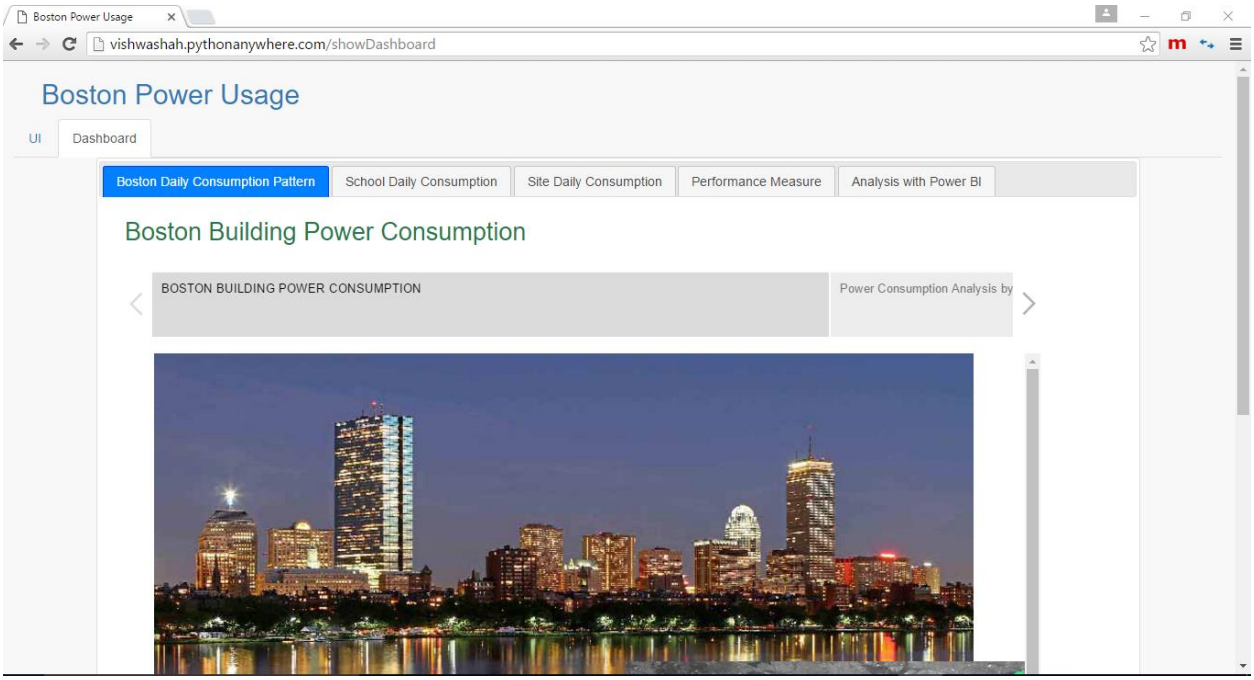


2. The Web service is deployed using REST API in the python server and in the frontend the data is consumed using JQuery AJAX function. The Web Pages are redirected using Flask Framework which is a python framework used for request dispatching in the web applications

3. The technologies used for the FrontEnd are: Bootstrap, HTML5, CSS, Javascript, JQuery, Font-Awesome.

- The images below shows the website. PythonAnywhere was used for hosting the website.





Integration of Front end and REST API

Technologies used: Flask, Python Server, Javascript

- Using Python Flask framework for redirecting the web pages

```python
from flask import Flask, render_template, request, jsonify
import urllib2
# If you are using Python 3+, import urllib instead of urllib2
import json
app = Flask(__name__)

@app.route("/")
def index():
    return render_template("blank.html")

@app.route('/showDashboard')
def showSignUp():
    return render_template('index.html')


@app.route("/ml", methods=['POST'])
def ml():
    paramVal1 = request.form["param1"]
    paramVal2 = request.form["param2"]

```

- Using REST API in the python server

```python
    data =  {

        "Inputs": {

                "input1":
                {
                    "ColumnNames": ["Account", "Date", "kVARh", "kWh", "Power_Factor"],
                    "Values": [ [ paramVal1, paramVal2, "0", "0", "0" ],]
                },        },
            "GlobalParameters": {
    }
    }

    body = str.encode(json.dumps(data))

    print(body)

    url = 'https://ussouthcentral.services.azureml.net/workspaces/7e24fa8f60d04a8bb0b4290c9a22645e/services/e9dff4ca464c4a318e0b0a2a08ad0330/execute?api-versi
    api_key = '7aU642MSXm+SvL56dqzWVU9vuMwP0TPM4LaFFSAVX1RYEO+7mA0YITyuvg3wVrl3zSWXUuDaqii5+SdUAgupfA==' # Replace this with the API key for the web service
    headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

    req = urllib2.Request(url, body, headers)

    try:
        response = urllib2.urlopen(req)

```

- In the Frontend calling the API using JQuery AJAX function

```python
    url = 'https://ussouthcentral.services.azureml.net/workspaces/7e24fa8f60d04a8bb0b4290c9a22645e/services/e9dff4ca464c4a318e0b0a2a08ad0330/execute?api-vers
    api_key = '7aU642MSXm+SvL56dqzWVU9vuMwP0TPM4LaFFSAVX1RYEO+7mA0YITyuvg3wVrl3zSWXUuDaqii5+SdUAgupfA==' # Replace this with the API key for the web service
    headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

    req = urllib2.Request(url, body, headers)

    try:
        response = urllib2.urlopen(req)

        # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
        # req = urllib.request.Request(url, body, headers)
        # response = urllib.request.urlopen(req)

        result = response.read()
        print(result)
        return result
    except urllib2.HTTPError, error:
        print("The request failed with status code: " + str(error.code))

        # Print the headers - they include the requert ID and the timestamp, which are useful for debugging the failure
        print(error.info())

        print(json.loads(error.read()))
        return render_template("blank.html")

if __name__ == "__main__":
    app.run(debug=True)
```

```
            </div>
        <!-- /.container-fluid -->

<script   src="https://code.jquery.com/jquery-2.2.3.min.js"   integrity="sha256-a23g1Nt4dtEYOj7bR+vTu7+T8VP13humZFBJNIYoEJo="   crossorigin="anonymous">
<script type="text/javascript">
  $(document).ready(function() {
    $("#btn").click(function() {0
      var param1Val = $('#account').val(),
        param2Val = $('#datepicker').val()

      if (param1Val == '' && param2Val == '') {
        alert("Please fill out the form");
      }

      $.post('/ml', {
        param1: param1Val,
        param2: param2Val
      }).done(function(responseData) {
        // make use of the retruned data!
          var dataObj = JSON.parse(responseData);
          var result = dataObj.Results.output1.value.Values[0][28];
          var result1 = dataObj.Results.output1.value.Values[0][1];
          $('#result').html(result);
          $("#result1").html(result1);
      });
    });
  });
</script>
    <!-- /#page-wrapper -->
```

- The Front-End code snippet - Blank.html

```
</head>

<body>

    <div class="container-fluid">
        <p class="navbar-text"><a href="/" style="text-decoration:none;"><h2>Boston Power Usage</h2></a></p>
        <ul class="nav nav-tabs">
            <li role="presentation" class="active"><a href="/">UI</a></li>
            <li role="presentation"><a href="/showDashboard">Dashboard</a></li>
        </ul>
    </div>

    <!-- Page Content -->

    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <h1 class="page-header">Predict your Future Usage</h1>
            </div>
            <!-- /.col-lg-12 -->
        </div>

        <div class="row">
            <div class="input-group date">
                <div class="col-md-4">
                    <label>Date:</label>   
                </div>
                <div class="col-md-8"></div>
                    <input type="text" id="datepicker" placeholder="Insert a Date" required> <i class="fa fa-calendar fa-lg" aria-hidden="true">
                </div>

        <div class="form-group">
            <label> Account Number:</label>
            <select class="form-control" style="width:300px;" id="account">
                <option value="select">Select your account number</option>
                <option value="26429741007">26429741007</option>
                <option value="26429921005">26429921005</option>
                <option value="26430021001">26430021001</option>
                <option value="26430841010">26430841010</option>
                <option value="26431351001">26431351001</option>
                <option value="26432071004">26432071004</option>
                <option value="26432141005">26432141005</option>
                <option value="26432391006">26432391006</option>
                <option value="26432771009">26432771009</option>
                <option value="26432831019">26432831019</option>
                <option value="26433791006">26433791006</option>
                <option value="26433891004">26433891004</option>
                <option value="26434211012">26434211012</option>
                <option value="26434521006">26434521006</option>
                <option value="26434801002">26434801002</option>
                <option value="26434861006">26434861006</option>
                <option value="26434871005">26434871005</option>
                <option value="26435331009">26435331009</option>
                <option value="26435791004">26435791004</option>
                <option value="26435881003">26435881003</option>
                <option value="26436151018">26436151018</option>
                <option value="26436351006">26436351006</option>
                <option value="26436491000">26436491000</option>
                <option value="26436731017">26436731017</option>
```

```
                    <option value="29184080017">29184080017</option>
                </select>
            </div><br/><br/>
            <button id="btn" class="btn btn-primary">Predict</button>
            <table class="table">
                <thead>
                    <tr>
                        <th>Output</th>
                        <th>Predicted Value</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>Kwh</td>
                        <td id="result"></td>
                    </tr>
                    <tr>
                        <td>Mean Temperature(F)</td>
                        <td id="result1"></td>
                    </tr>
                </tbody>
            </table>
            <!--<div id="result">

            </div>-->
            </div>
        <!-- /.container-fluid -->
```

```
        <!-- /#page-wrapper -->

    </div>
    <!-- /#wrapper -->
    <!-- jQuery -->
    <script   src="https://code.jquery.com/jquery-2.2.3.min.js"   integrity="sha256-a23g1Nt4dtEYOj7bR+vTu7+T8VP13humZFBJNIYoEJo="   crossorigin="anonymous"><

    <!-- Bootstrap Core JavaScript -->
    <script src="/static/js/bootstrap.min.js"></script>

    <!-- Custom Theme JavaScript -->
    <script src="/static/js/sb-admin-2.js"></script>
        <!-- Load jQuery and bootstrap datepicker scripts -->

        <script src="/static/js/bootstrap-datepicker.js"></script>
        <script type="text/javascript">
            // When the document is ready
            $(document).ready(function() {
                $( "#datepicker" ).datepicker({
                    dateFormat: 'mm/dd/yyyy'
                });

            });
        </script>
</body>

</html>
```

- Index.html

```
        <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
    <script>
        $(document).ready(function() {
            $( "#tabs" ).tabs();
        });
    </script>

</head>

<body>

        <!-- Navigation -->
        <div class="container-fluid">
            <p class="navbar-text"><a href="/" style="text-decoration:none;"><h2>Boston Power Usage</h2></a></p>
            <ul class="nav nav-tabs">
                <li role="presentation"><a href="/">UI</a></li>
                <li role="presentation" class="active"><a href="/showDashboard">Dashboard</a></li>
            </ul>
        </div>

        <div class="container">
            <div id="tabs">
            <ul>
                <li><a href="#tabs-1">Boston Daily Consumption Pattern</a></li>
                <li><a href="#tabs-2">School Daily Consumption</a></li>
                <li><a href="#tabs-3">Site Daily Consumption</a></li>
                <li><a href="#tabs-4">Performance Measure</a></li>
                <li><a href="#tabs-5">Analysis with Power BI</a></li>
            </ul>
```

```
        <div class="container">
            <div id="tabs">
            <ul>
                <li><a href="#tabs-1">Boston Daily Consumption Pattern</a></li>
                <li><a href="#tabs-2">School Daily Consumption</a></li>
                <li><a href="#tabs-3">Site Daily Consumption</a></li>
                <li><a href="#tabs-4">Performance Measure</a></li>
                <li><a href="#tabs-5">Analysis with Power BI</a></li>
            </ul>
            <div id="tabs-1">
                <script type='text/javascript' src='https://public.tableau.com/javascripts/api/viz_v1.js'></script><div class='tableauPlaceholder' style='widt
            </div>
            <div id="tabs-2">
                <script type='text/javascript' src='https://public.tableau.com/javascripts/api/viz_v1.js'></script><div class='tableauPlaceholder' style='widt
            </div>
            <div id="tabs-3">
                <script type='text/javascript' src='https://public.tableau.com/javascripts/api/viz_v1.js'></script><div class='tableauPlaceholder' style='widt
            </div>
            <div id="tabs-4">
                <script type='text/javascript' src='https://public.tableau.com/javascripts/api/viz_v1.js'></script><div class='tableauPlaceholder' style='widt
            </div>
            <div id="tabs-5">
                <iframe width="800" height="600" src="https://app.powerbi.com/view?r=eyJrIjoiYjQwMDhlYzctOWFhNS00ZDlhLTk1MjktYzdhN2ZkZGMxZDgyIiwidCI6IjZhYmZjN
            </div>
        </div>
            </div>


        <!-- /#page-wrapper -->
```