Course 12-752: Term Project

Project Title: Comparison of the accuracy of linear regression model based on the granularity of data: Scaife Hall

Team Members

1. Rushil Desai (Andrew ID: rushid)
2. Varun Deshpande (Andrew ID: varund)
3. Sakshi Mishra (Andrew ID: sakshimi)

Section 1: Importing general modules

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
import scipy.spatial.distance as dist
%matplotlib inline
```

Section 2: Loading the Power Data

Imported Glob module to accecss all CSV data files in the folder. Load one file at a time and concatenate to a dataframe variable 'frame'

In [3]:
```
import glob
allFiles = glob.glob('4months' + "/*.csv")
frame = pd.DataFrame()
list_ = []
for file_ in allFiles:
    df = pd.read_csv(file_,index_col=None, header=0)
    list_.append(df)
frame = pd.concat(list_)
frame.reset_index(drop=True)
frame = frame.iloc[:,(5,4,2,11)]
```

Section 3: Conditioning, subsetting and grouping power data

Select only those rows within 'frame' which have lights in the Branch Name. Store these rows in a new dataframe called 'power'

In [14]:
```
scheck = lambda d: 'Lights' in d or 'Light' in d or 'Lts' in d
S_ind = list(map(scheck,frame.BranchName))
power = frame[S_ind].reset_index(drop=True)
```

Display variable power: It has 4 column - dates, panel names, branch name and average wattage

In [18]:
```
power
```

Out[18]:

	DateStamp	PanelName	BranchName	AvgWatt
0	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Rm 305, 307	0
1	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Rm 304, 306 & Hall	0
2	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. 312, 313	132
3	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Rm 318	0
4	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. 314, 315	24
5	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Rm 301	0
6	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts Rm 316	0
7	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts Rm 303	0
8	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Women's Rest Rm.	0
9	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Cor. 319	0
10	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lights Stairway	0
11	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lts. Stairway	333
12	2015-01-15 00:00:00	3rd Floor Ladies' Room	Lights 317	0
13	2015-01-15 00:00:00	4th Floor Men's Room	Lts. Rm 4003, 404	0
14	2015-01-15 00:00:00	4th Floor Men's Room	Lts. Rm. 426	0
15	2015-01-15 00:00:00	4th Floor Men's Room	Lts. Rm 402	0
16	2015-01-15 00:00:00	4th Floor Men's Room	Lts. Rm 401	0
17	2015-01-15 00:00:00	4th Floor Men's Room	Lts. Rm 406	0
18	2015-01-15 00:00:00	4th Floor Men's Room	Lts. Rm 408	0
19	2015-01-15 00:00:00	4th Floor Men's Room	Emergency Lts.	0
20	2015-01-15 00:00:00	4th Floor Men's Room	Restroom Lts.	168
21	2015-01-15 00:00:00	4th Floor Men's Room	404 Lts.	0

22	2015-01-15 00:00:00	4th Floor Men's Room	Lights Penthouse	0
23	2015-01-15 00:00:00	4th Floor Men's Room	Lights Penthouse	0
24	2015-01-15 00:00:00	Auditorium 1A	Lecture Hall 400W Metal ? Lights	0
25	2015-01-15 00:00:00	Auditorium 1A	Rear House Lights	0
26	2015-01-15 00:00:00	Auditorium 1A	Lecture Hall 400W Metal ? Lights	0
27	2015-01-15 00:00:00	Auditorium 1A	Lecture Hall 400W Metal ? Lights	0
28	2015-01-15 00:00:00	Auditorium 1B	Projection Rm Lights	0
29	2015-01-15 00:00:00	Auditorium 1B	Lobby Lights	0
...	...	...	...	...
319849	2015-03-31 23:45:00	4th Floor Men's Room	Lts. Rm. 426	0
319850	2015-03-31 23:45:00	4th Floor Men's Room	Lts. Rm 402	0
319851	2015-03-31 23:45:00	4th Floor Men's Room	Lts. Rm 401	0
319852	2015-03-31 23:45:00	4th Floor Men's Room	Lts. Rm 406	0
319853	2015-03-31 23:45:00	4th Floor Men's Room	Lts. Rm 408	0
319854	2015-03-31 23:45:00	4th Floor Men's Room	Emergency Lts.	0
319855	2015-03-31 23:45:00	4th Floor Men's Room	Restroom Lts.	168
319856	2015-03-31 23:45:00	4th Floor Men's Room	404 Lts.	0
319857	2015-03-31 23:45:00	4th Floor Men's Room	Lights Penthouse	0
319858	2015-03-31 23:45:00	4th Floor Men's Room	Lights Penthouse	0
319859	2015-03-31 23:45:00	Auditorium 1A	Lecture Hall 400W Metal ? Lights	0
319860	2015-03-31 23:45:00	Auditorium 1A	Rear House Lights	0
319861	2015-03-31 23:45:00	Auditorium 1A	Lecture Hall 400W Metal ? Lights	0
319862	2015-03-31 23:45:00	Auditorium 1A	Lecture Hall 400W Metal ? Lights	0

319863	2015-03-31 23:45:00	Auditorium 1B	Projection Rm Lights	0
319864	2015-03-31 23:45:00	Auditorium 1B	Lobby Lights	0
319865	2015-03-31 23:45:00	Auditorium 1B	Canopy Lights	0
319866	2015-03-31 23:45:00	Auditorium 1B	Cove Lights Lobby	0
319867	2015-03-31 23:45:00	Auditorium 1B	Light ctrl. 110-114	0
319868	2015-03-31 23:45:00	Dean's Office 2	Lobby Bathroom Lights	552
319869	2015-03-31 23:45:00	Penthouse 277	4th Floor Lighting	0
319870	2015-03-31 23:45:00	Penthouse 277	4th Floor Lights	0
319871	2015-03-31 23:45:00	Penthouse 277	4th Floor Lighting	192
319872	2015-03-31 23:45:00	Penthouse 277	4th Floor Lighting	1092
319873	2015-03-31 23:45:00	Penthouse 277	Penthouse Lights	156
319874	2015-03-31 23:45:00	Penthouse 277	2nd Floor Lighting	0
319875	2015-03-31 23:45:00	Penthouse 277	2nd Floor Lighting	0
319876	2015-03-31 23:45:00	Penthouse 277	2nd Floor Lighting	0
319877	2015-03-31 23:45:00	Penthouse 277	2nd Floor Lighting	840
319878	2015-03-31 23:45:00	Scaife Hall 2F	Elevator Lights	0

319879 rows × 4 columns

Use parser function from dateutil module to convert the timestamps in power from a string to a datetime - Timestamp object

In [16]:
```
from dateutil import parser
power.DateStamp = power.DateStamp.apply(parser.parse)
```

Select 15 minute time interval data and re-store in power

In [17]:
```
intervals = lambda d: (d.time().minute)%15 == 0
indexes_15 = list(map(intervals,power.DateStamp))
power = power[indexes_15].reset_index(drop=True)
```

Use group by function of Pandas dataframe to group the power data based on

1. Datestamp (P_Total)
2. Datestamp & Panel Name (P_Panel)

In [19]:
```
P_Total = power.groupby(['DateStamp'], as_index=False)
P_Panel = power.groupby(['DateStamp','PanelName'],as_index=False)
```

Use .sum() attribute of group object to sum up the Average power based on the group variables defined above

Create grouped data for first & second set of group variables and call it Lighting_Total and Lighting_Panel respectively

In [69]:

```
Lighting_Panel = P_Panel.sum()
Lighting_Panel.columns = ['Timestamp','PanelName','AvgPower']
Lighting_Panel
```

Out[69]:

	Timestamp	PanelName	AvgPower
0	2015-01-15 00:00:00	3rd Floor Ladies' Room	489
1	2015-01-15 00:00:00	4th Floor Men's Room	168
2	2015-01-15 00:00:00	Auditorium 1A	0
3	2015-01-15 00:00:00	Auditorium 1B	0
4	2015-01-15 00:00:00	Dean's Office 2	576
5	2015-01-15 00:00:00	Penthouse 277	2346
6	2015-01-15 00:00:00	Scaife Hall 2F	0
7	2015-01-15 00:15:00	3rd Floor Ladies' Room	408
8	2015-01-15 00:15:00	4th Floor Men's Room	168
9	2015-01-15 00:15:00	Auditorium 1A	0
10	2015-01-15 00:15:00	Auditorium 1B	0
11	2015-01-15 00:15:00	Dean's Office 2	576
12	2015-01-15 00:15:00	Penthouse 277	3069
13	2015-01-15 00:15:00	Scaife Hall 2F	0
14	2015-01-15 00:30:00	3rd Floor Ladies' Room	480
15	2015-01-15 00:30:00	4th Floor Men's Room	168
16	2015-01-15 00:30:00	Auditorium 1A	0
17	2015-01-15 00:30:00	Auditorium 1B	0
18	2015-01-15 00:30:00	Dean's Office 2	576
19	2015-01-15 00:30:00	Penthouse 277	2496
20	2015-01-15 00:30:00	Scaife Hall 2F	0
21	2015-01-15 00:45:00	3rd Floor Ladies' Room	420

22	2015-01-15 00:45:00	4th Floor Men's Room	168
23	2015-01-15 00:45:00	Auditorium 1A	0
24	2015-01-15 00:45:00	Auditorium 1B	0
25	2015-01-15 00:45:00	Dean's Office 2	576
26	2015-01-15 00:45:00	Penthouse 277	2328
27	2015-01-15 00:45:00	Scaife Hall 2F	0
28	2015-01-15 01:00:00	3rd Floor Ladies' Room	583
29	2015-01-15 01:00:00	4th Floor Men's Room	168
...	...	...	...
50970	2015-03-31 22:45:00	Penthouse 277	3888
50971	2015-03-31 22:45:00	Scaife Hall 2F	0
50972	2015-03-31 23:00:00	3rd Floor Ladies' Room	799
50973	2015-03-31 23:00:00	4th Floor Men's Room	168
50974	2015-03-31 23:00:00	Auditorium 1A	0
50975	2015-03-31 23:00:00	Auditorium 1B	0
50976	2015-03-31 23:00:00	Dean's Office 2	552
50977	2015-03-31 23:00:00	Penthouse 277	3888
50978	2015-03-31 23:00:00	Scaife Hall 2F	0
50979	2015-03-31 23:15:00	3rd Floor Ladies' Room	528
50980	2015-03-31 23:15:00	4th Floor Men's Room	168
50981	2015-03-31 23:15:00	Auditorium 1A	0
50982	2015-03-31 23:15:00	Auditorium 1B	0
50983	2015-03-31 23:15:00	Dean's Office 2	552

50984	2015-03-31 23:15:00	Penthouse 277	3169
50985	2015-03-31 23:15:00	Scaife Hall 2F	0
50986	2015-03-31 23:30:00	3rd Floor Ladies' Room	550
50987	2015-03-31 23:30:00	4th Floor Men's Room	168
50988	2015-03-31 23:30:00	Auditorium 1A	0
50989	2015-03-31 23:30:00	Auditorium 1B	0
50990	2015-03-31 23:30:00	Dean's Office 2	552
50991	2015-03-31 23:30:00	Penthouse 277	2460
50992	2015-03-31 23:30:00	Scaife Hall 2F	0
50993	2015-03-31 23:45:00	3rd Floor Ladies' Room	504
50994	2015-03-31 23:45:00	4th Floor Men's Room	168
50995	2015-03-31 23:45:00	Auditorium 1A	0
50996	2015-03-31 23:45:00	Auditorium 1B	0
50997	2015-03-31 23:45:00	Dean's Office 2	552
50998	2015-03-31 23:45:00	Penthouse 277	2280
50999	2015-03-31 23:45:00	Scaife Hall 2F	0

51000 rows × 3 columns

In [95]:
```
Lighting_Total = P_Total.sum()
Lighting_Total.columns = ['Timestamp','AvgPower']
Lighting_Total
```

Out[95]:

	Timestamp	AvgPower
0	2015-01-15 00:00:00	3579
1	2015-01-15 00:15:00	4221
2	2015-01-15 00:30:00	3720
3	2015-01-15 00:45:00	3492
4	2015-01-15 01:00:00	3655
5	2015-01-15 01:15:00	3576
6	2015-01-15 01:30:00	3730
7	2015-01-15 01:45:00	3378
8	2015-01-15 02:00:00	3469
9	2015-01-15 02:15:00	3951
10	2015-01-15 02:30:00	3288
11	2015-01-15 02:45:00	2752
12	2015-01-15 03:00:00	2929
13	2015-01-15 03:15:00	3096
14	2015-01-15 03:30:00	2808
15	2015-01-15 03:45:00	2908
16	2015-01-15 04:00:00	3492
17	2015-01-15 04:15:00	3216
18	2015-01-15 04:30:00	3831
19	2015-01-15 04:45:00	3022
20	2015-01-15 05:00:00	4306
21	2015-01-15 05:15:00	3365

22	2015-01-15 05:30:00	3755
23	2015-01-15 05:45:00	2829
24	2015-01-15 06:00:00	5172
25	2015-01-15 06:15:00	3504
26	2015-01-15 06:30:00	3850
27	2015-01-15 06:45:00	3249
28	2015-01-15 07:00:00	3062
29	2015-01-15 07:15:00	2984
...	...	...
7258	2015-03-31 16:30:00	9145
7259	2015-03-31 16:45:00	8167
7260	2015-03-31 17:00:00	8355
7261	2015-03-31 17:15:00	7620
7262	2015-03-31 17:30:00	6646
7263	2015-03-31 17:45:00	6425
7264	2015-03-31 18:00:00	7081
7265	2015-03-31 18:15:00	5563
7266	2015-03-31 18:30:00	6104
7267	2015-03-31 18:45:00	5958
7268	2015-03-31 19:00:00	5831
7269	2015-03-31 19:15:00	5395
7270	2015-03-31 19:30:00	5342
7271	2015-03-31 19:45:00	5400

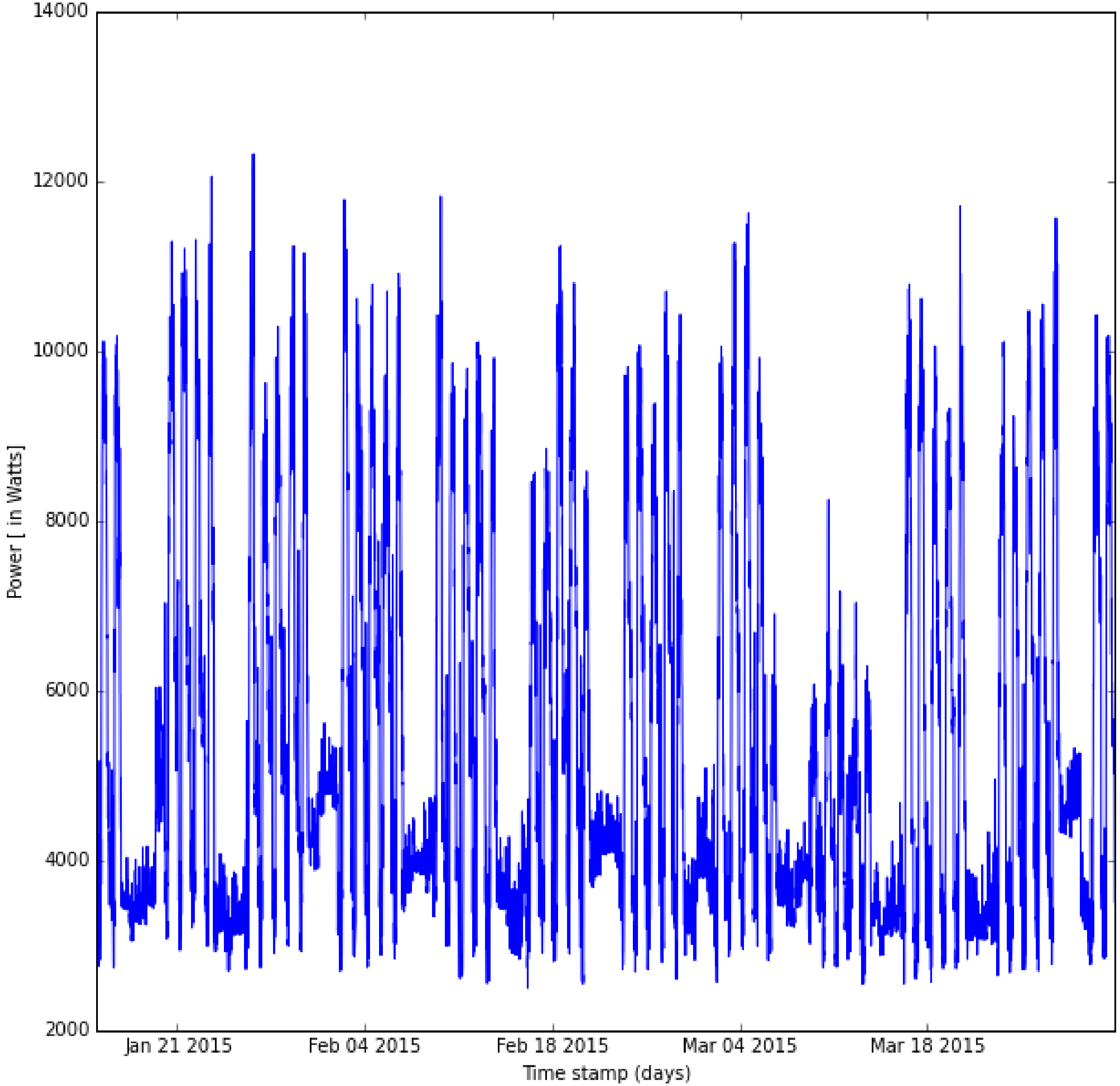
7272	2015-03-31 20:00:00	5664
7273	2015-03-31 20:15:00	5475
7274	2015-03-31 20:30:00	5357
7275	2015-03-31 20:45:00	5436
7276	2015-03-31 21:00:00	5449
7277	2015-03-31 21:15:00	5493
7278	2015-03-31 21:30:00	5556
7279	2015-03-31 21:45:00	5563
7280	2015-03-31 22:00:00	5093
7281	2015-03-31 22:15:00	5029
7282	2015-03-31 22:30:00	5193
7283	2015-03-31 22:45:00	5030
7284	2015-03-31 23:00:00	5407
7285	2015-03-31 23:15:00	4417
7286	2015-03-31 23:30:00	3730
7287	2015-03-31 23:45:00	3504

7288 rows × 2 columns

Plot the graph of Total power over time

In [96]:
```
plt.figure(figsize=(10,10))
plt.plot(Lighting_Total.Timestamp,Lighting_Total.AvgPower)
plt.xlabel('Time stamp (days)')
plt.ylabel('Power [ in Watts]')
```

```
Out[96]: <matplotlib.text.Text at 0xf0a598c1d0>
```



Section 4: Linear regression and analysis

Define function - DesignMatrix which takes only timestamp as an argument and outputs a stack of indentity matrix with number of rows equal to the numer of timestamps and number of columns equal to the number of 15 minute intervals in a week

In [22]:
```
import math
def DesignMatrix(timestamps):
    tslen = len(timestamps)
    ind = 672
    num = math.ceil(tslen/ind)
    sing = np.identity(ind)

    Dmat = np.tile(sing,(num,1))[0:tslen,:]

    return Dmat
```

Find the Design matrix for the Total lighting consumption

In [97]:
```
DMX = DesignMatrix(Lighting_Total.Timestamp)
```

Define a function 'Beta_hat' which will take a design matrix and a power vector as arguments and outputs the Beta hat values as defined by the function inverse(Xt *X)* Xt * Y where Xt is the transpose of the design matrix and Y is the power vector

In [24]:
```
def beta_hat(X,Y):
    B = np.dot(np.dot(np.linalg.inv(np.dot(X.T,X)),X.T),Y)
    return B
```

Finding Beta hat for Total lighting consumption and calculating predicted power. Here the data set used for training and testing the regression model is the same.

In [98]:
```
Act_power = Lighting_Total.AvgPower
B_Lighting = beta_hat(DMX,Actual_power)
Pred_power = np.dot(DMX,B_Lighting)
```

Defining function Cal_Rsqr which takes arguments Actual power and Predicted power and then calculates & returns the R squared value

In [99]:
```
def Cal_Rsqr(Actual_power,Predict_power):
    Power_mean = np.mean(Actual_power)
    Numer = Actual_power - Predict_power
    Denom = Actual_power - Power_mean
    R_sqr = 1- (np.dot(Numer.T,Numer)/np.dot(Denom.T,Denom))
    return R_sqr
```

Call function Cal_Rsqr for the total lighting consumption

In [100]:
```
Cal_Rsqr(Act_power,Pred_power)
```

Out[100]:
```
0.78928765887872021
```

Section 5: Predicting for Lighting consumption for Scaife building using train and test datasets

Define separate train and test datasets. Here we have used alternate weeeks for train and the remaining alternate for test

Used the isocalendar function to extract the week number in order to segregate the data

In [101]:
```
W_check = lambda d : d.isocalendar()[1]%2 == 1
W_indices = list(map(W_check, Lighting_Total.Timestamp))
Train_Lighting = Lighting_Total[W_indices]
Test_Lighting = Lighting_Total[np.invert(W_indices)]
```

Removing the first 4 days of data to allow the train and test datasets to start at the same 15 minute time interval of the week

In [102]:
```
Train_Lighting = Train_Lighting.iloc[384:,:]
```

Generating design matrices for train and test dataset by calling fucntion DesignMatrix

In [103]:
```
TrainDMX = DesignMatrix(Train_Lighting.Timestamp)
TestDMX = DesignMatrix(Test_Lighting.Timestamp)
```

Calculating Beta hat for train data set

In [104]:
```
LBs = beta_hat(TrainDMX, Train_Lighting.AvgPower)
```

Estimate predicted power using beta hat and test matrix. Calculate R square value

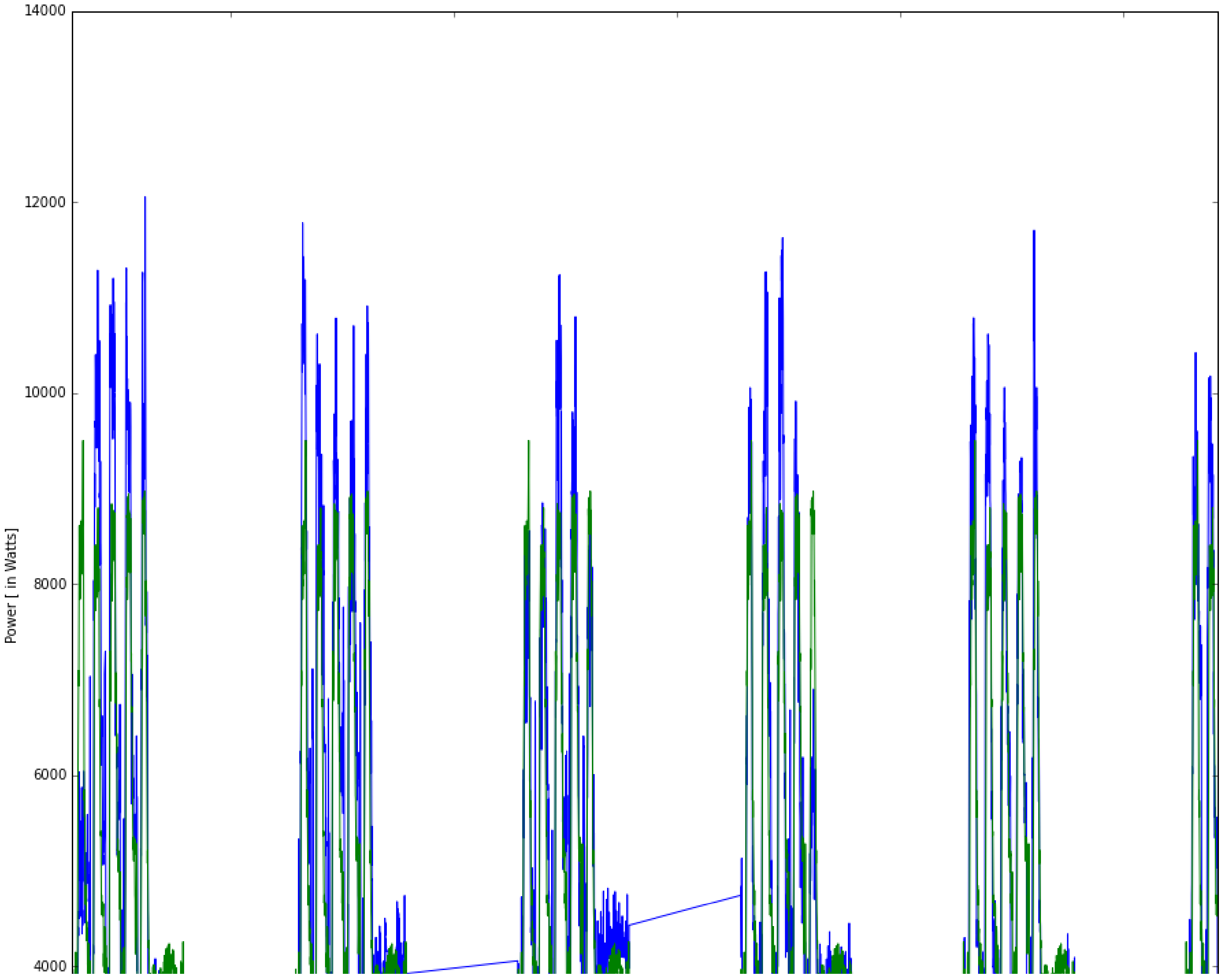
In [105]:
```
Lighting_predpower = np.dot(TestDMX,LBs)
Lighting_actpower = Test_Lighting.AvgPower
Cal_Rsqr(Lighting_actpower,Lighting_predpower)
```

Out[105]: 0.77797630351169911

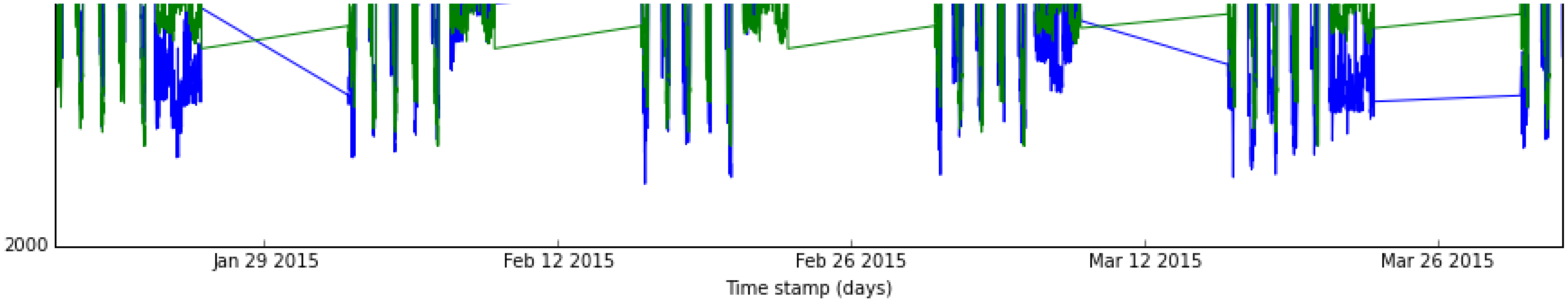
Plot graph of Actual power versus Predicted power with a common time axis

In [107]:
```
plt.figure(figsize=(15,15))
plt.plot(Test_Lighting.Timestamp,Lighting_actpower,Test_Lighting.Timestamp,Lighting_predpower)
plt.xlabel('Time stamp (days)')
plt.ylabel('Power [ in Watts]')
```

```
Out[107]: <matplotlib.text.Text at 0xf0ef33b6d8>
```



14000
12000
10000
8000
6000
4000
Power [in Watts]



Section 6: Predicting for Lighting consumption for different panels of Scaife building using train and test datasets

The logic for calculating R square value for each of the different panel consumptions in Scaife building is the same as given above.

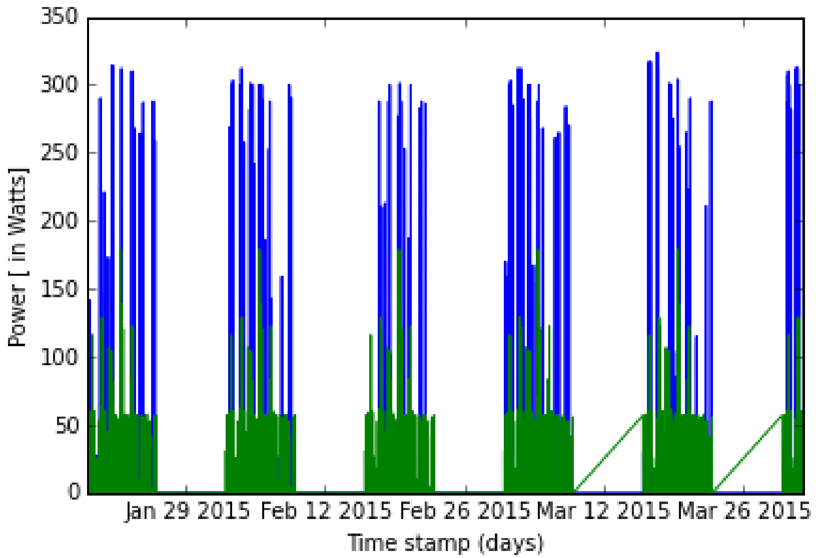
A loop has been created to access the summed up Average power of each panel and the fucntions above have been called in order to calculate R square for each of the panels.

Provision has been given in the end of the loop to plot predicted and actual power of individual panels.

In [123]:

```
count = 0
for name in (Lighting_Panel.PanelName):

    Data = Lighting_Panel[Lighting_Panel.PanelName == name]
    count = count + 1
    W_check = lambda d : d.isocalendar()[1]%2 == 1
    W_indices = list(map(W_check, Data.Timestamp))

    Train_dat = Data[W_indices].iloc[384:,:]
    Test_dat = Data[np.invert(W_indices)]

    TrainDMX = DesignMatrix(Train_dat.Timestamp)
    TestDMX = DesignMatrix(Test_dat.Timestamp)


    LB = beta_hat(TrainDMX,Train_dat.AvgPower)
    Lighting_actpower = Train_dat.AvgPower
    Lighting_predpower = np.dot(TrainDMX,LB)

    R_train_panel = Cal_Rsqr(Lighting_actpower,Lighting_predpower)
    print ('R square value for prediction on train data for panel ' + name + ' is ' + str(R_train_panel))

    Lighting_actpower = Test_dat.AvgPower
    Lighting_predpower = np.dot(TestDMX,LB)

    R_test_panel = Cal_Rsqr(Lighting_actpower,Lighting_predpower)
    print ('R square value for prediction on test data for panel ' + name + ' is ' + str(R_test_panel))

    if (count == 7):
        plt.plot(Test_dat.Timestamp,Lighting_actpower,Test_dat.Timestamp,Lighting_predpower)
        plt.xlabel('Time stamp (days)')
        plt.ylabel('Power [ in Watts]')
        break;
```

```
R square value for prediction on train data for panel 3rd Floor Ladies' Room is 0.276159461651
R square value for prediction on test data for panel 3rd Floor Ladies' Room is 0.000227954885169
R square value for prediction on train data for panel 4th Floor Men's Room is 0.510846667236
R square value for prediction on test data for panel 4th Floor Men's Room is 0.225872287018
R square value for prediction on train data for panel Auditorium 1A is nan
R square value for prediction on test data for panel Auditorium 1A is nan
R square value for prediction on train data for panel Auditorium 1B is nan
R square value for prediction on test data for panel Auditorium 1B is nan
R square value for prediction on train data for panel Dean's Office 2 is 0.370703728658
R square value for prediction on test data for panel Dean's Office 2 is 0.0758048717159
R square value for prediction on train data for panel Penthouse 277 is 0.752271429387
R square value for prediction on test data for panel Penthouse 277 is 0.75773265909
R square value for prediction on train data for panel Scaife Hall 2F is 0.232472329371
R square value for prediction on test data for panel Scaife Hall 2F is -0.183165982575
```



End of code