CSE 351 HW2

April 7, 2023

[1]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
import datetime

weather_df = pd.read_csv('/content/drive/MyDrive/weather_data.csv')
energy_df = pd.read_csv('/content/drive/MyDrive/energy_data.csv')
```

[2]:
```
weather_df.head(10)
```

[2]:

	temperature	icon	humidity	visibility	summary	\
0	34.98	partly-cloudy-night	0.64	10.00	Partly Cloudy	
1	16.49	clear-night	0.62	10.00	Clear	
2	14.63	clear-night	0.68	10.00	Clear	
3	13.31	clear-night	0.71	10.00	Clear	
4	13.57	clear-night	0.71	9.93	Clear	
5	20.31	clear-night	0.47	10.00	Clear	
6	19.22	clear-night	0.48	10.00	Clear	
7	18.10	clear-night	0.52	10.00	Clear	
8	17.47	clear-night	0.55	10.00	Clear	
9	15.63	clear-night	0.61	10.00	Clear	

	pressure	windSpeed	cloudCover	time	windBearing	precipIntensity	\
0	1017.69	7.75	0.29	1388534400	279	0.0	
1	1022.76	2.71	0.06	1388538000	195	0.0	
2	1022.32	4.84	0.03	1388541600	222	0.0	
3	1021.64	4.00	0.14	1388545200	209	0.0	
4	1020.73	3.67	0.04	1388548800	217	0.0	
5	1023.25	12.65	0.00	1388552400	283	0.0	
6	1023.21	9.99	0.00	1388556000	283	0.0	
7	1024.25	7.76	0.00	1388559600	255	0.0	
8	1025.02	7.73	0.00	1388563200	245	0.0	
9	1025.16	5.76	0.00	1388566800	220	0.0	

	dewPoint	precipProbability
0	23.89	0.0
1	5.87	0.0

```
2      6.17                0.0
3      5.63                0.0
4      5.87                0.0
5      3.30                0.0
6      2.69                0.0
7      3.42                0.0
8      4.00                0.0
9      4.45                0.0
```

[3]:
```
energy_df.head(10)
```

[3]:
```
            Date & Time  use [kW]  gen [kW]  Grid [kW]   AC [kW]  Furnace [kW]  \
0  2014-01-01 00:00:00  0.304439       0.0   0.304439  0.000058      0.009531
1  2014-01-01 00:30:00  0.656771       0.0   0.656771  0.001534      0.364338
2  2014-01-01 01:00:00  0.612895       0.0   0.612895  0.001847      0.417989
3  2014-01-01 01:30:00  0.683979       0.0   0.683979  0.001744      0.410653
4  2014-01-01 02:00:00  0.197809       0.0   0.197809  0.000030      0.017152
5  2014-01-01 02:30:00  0.397099       0.0   0.397099  0.000442      0.126960
6  2014-01-01 03:00:00  0.590319       0.0   0.590319  0.001858      0.420358
7  2014-01-01 03:30:00  0.538266       0.0   0.538266  0.001071      0.257654
8  2014-01-01 04:00:00  0.189187       0.0   0.189187  0.000056      0.009462
9  2014-01-01 04:30:00  0.653232       0.0   0.653232  0.001663      0.387637

   Cellar Lights [kW]  Washer [kW]  First Floor lights [kW]  \
0            0.005336     0.000126                 0.011175
1            0.005522     0.000043                 0.003514
2            0.005504     0.000044                 0.003528
3            0.005556     0.000059                 0.003499
4            0.005302     0.000119                 0.003694
5            0.005415     0.000054                 0.003627
6            0.005509     0.000043                 0.003562
7            0.005507     0.000020                 0.003582
8            0.005302     0.000141                 0.003708
9            0.005552     0.000033                 0.003532

   Utility Rm + Basement Bath [kW]  Garage outlets [kW]  \
0                         0.003836             0.004836
1                         0.003512             0.004888
2                         0.003484             0.004929
3                         0.003476             0.004911
4                         0.003865             0.004876
5                         0.003749             0.004891
6                         0.003541             0.005007
7                         0.003549             0.004837
8                         0.003837             0.004858
9                         0.003463             0.004878
```

	MBed + KBed outlets [kW]	Dryer + egauge [kW]	\
0	0.002132	0.000009	
1	0.002137	0.000107	
2	0.002052	0.000170	
3	0.002068	0.000121	
4	0.002087	0.000052	
5	0.002133	0.000024	
6	0.002072	0.000197	
7	0.002136	0.000047	
8	0.002086	0.000034	
9	0.002112	0.000118	

	Panel GFI (central vac) [kW]	Home Office (R) [kW]	Dining room (R) [kW]	\
0	0.007159	0.063666	0.004299	
1	0.007221	0.064698	0.003589	
2	0.007197	0.065109	0.003522	
3	0.007236	0.065032	0.003404	
4	0.007133	0.062451	0.003915	
5	0.007187	0.063814	0.003813	
6	0.007208	0.065277	0.003008	
7	0.007198	0.063142	0.004142	
8	0.007107	0.070109	0.004202	
9	0.007203	0.064450	0.003669	

	Microwave (R) [kW]	Fridge (R) [kW]
0	0.004733	0.042589
1	0.004445	0.096008
2	0.004396	0.025928
3	0.004262	0.105472
4	0.004407	0.016798
5	0.004398	0.106717
6	0.004008	0.006479
7	0.004565	0.121088
8	0.004704	0.011131
9	0.004194	0.104737

0.1 Task 1

[4]:
```
# group weather_df by date
weather_df["Date"] = [datetime.datetime.fromtimestamp(time).
 ↪strftime("%Y-%m-%d") for time in weather_df.time]
weather_df_gbdate = weather_df.groupby("Date").mean().reset_index()
```

[5]:
```
weather_df_gbdate
```

[5]:

	Date	temperature	humidity	visibility	pressure	windSpeed	cloudCover	time	windBearing	precipIntensity	dewPoint	precipProbability
0	2014-01-01	20.110833	0.556667	9.970000	1025.395000	6.820417	0.031304	1.388576e+09	252.291667	0.000000	6.362083	0.000000
1	2014-01-02	16.382500	0.784583	3.834583	1023.465833	7.433750	0.354444	1.388662e+09	53.458333	0.002004	10.737083	0.074583
2	2014-01-03	6.256667	0.680833	4.509167	1014.428750	12.828333	0.186364	1.388749e+09	207.333333	0.002029	-2.337500	0.080000
3	2014-01-04	2.711667	0.617083	9.822917	1030.096250	5.248333	0.001667	1.388835e+09	240.166667	0.000000	-8.352083	0.000000
4	2014-01-05	17.654167	0.682083	9.134583	1025.275000	3.417083	0.010952	1.388921e+09	208.958333	0.000033	8.615000	0.000417
..	…	…	…	…	…	…	…	…	…	…	…	…
360	2014-12-27	35.487083	0.756250	9.246250	1022.081667	3.677083	0.030417	1.419680e+09	243.791667	0.000000	27.992500	0.000000
361	2014-12-28	41.892917	0.763750	9.332917	1013.549167	6.587917	0.245909	1.419766e+09	224.458333	0.003996	34.876250	0.137917
362	2014-12-29	34.728333	0.592083	9.997083	1018.870833	8.129583	0.119167	1.419853e+09	281.833333	0.000000	21.570000	0.000000
363	2014-12-30	24.846667	0.488750	9.998333	1026.102083	7.566667	0.031250	1.419939e+09	312.041667	0.000000	7.772083	0.000000
364	2014-12-31	19.522917	0.552917	9.986250	1025.940833	5.943750	0.117917	1.420025e+09	260.083333	0.000000	5.332500	0.000000

[365 rows x 12 columns]

[6]:
```
# group energy_df by date
energy_df["Date"] = [str(val).split()[0] for val in energy_df["Date & Time"]]
energy_df_gbdate = energy_df.groupby("Date").sum().reset_index()
```

[7]:
```
energy_df_gbdate
```

[7]:

	Date	use [kW]	gen [kW]	Grid [kW]	AC [kW]	Furnace [kW]	\
0	2014-01-01	65.013592	0.0	65.013592	0.042977	8.814319	
1	2014-01-02	32.305336	0.0	32.305336	0.047452	10.830045	
2	2014-01-03	31.164468	0.0	31.164468	0.055865	12.417151	
3	2014-01-04	45.287782	0.0	45.287782	0.048827	11.147332	
4	2014-01-05	36.316643	0.0	36.316643	0.039831	9.301135	
..	…	…	…	…	…	…	
360	2014-12-27	35.046127	0.0	35.046127	0.029014	4.306682	
361	2014-12-28	37.695824	0.0	37.695824	0.044794	4.640888	
362	2014-12-29	28.675929	0.0	28.675929	0.037950	6.143640	
363	2014-12-30	31.514313	0.0	31.514313	0.062139	7.283703	
364	2014-12-31	28.674498	0.0	28.674498	0.060092	8.030594	

	Cellar Lights [kW]	Washer [kW]	First Floor lights [kW]	\
0	1.137579	0.750298	0.567603	
1	0.600321	0.323182	0.506440	
2	0.442453	0.004276	0.507426	
3	0.674477	1.046294	0.515988	
4	0.686189	0.235143	0.519449	
..	…	…	…	
360	0.475110	0.019649	0.825872	
361	0.456018	0.035832	0.874563	
362	0.463515	0.020115	0.552812	
363	0.467821	0.033911	0.764434	
364	0.494868	0.026109	0.559797	

	Utility Rm + Basement Bath [kW]	Garage outlets [kW]	\
0	0.178529	0.261094	
1	0.178024	0.282479	
2	0.176649	0.279159	
3	0.180056	0.344005	
4	0.178556	0.348489	
..	…	…	
360	0.112661	0.255620	
361	0.115673	0.257369	
362	0.112204	0.274396	
363	0.115933	0.238277	
364	0.114914	0.273398	

	MBed + KBed outlets [kW]	Dryer + egauge [kW]	\
0	0.254839	31.938131	
1	0.798316	5.423866	
2	0.746972	0.005554	
3	0.640721	19.994908	
4	0.584570	9.493912	
..	…	…	
360	5.422751	0.005953	

```
361                 11.602281             0.008270
362                  5.951963             0.005461
363                 11.100021             0.008893
364                  7.741381             0.005618

     Panel GFI (central vac) [kW]  Home Office (R) [kW]  Dining room (R) [kW]  \
0                        0.350291              3.272944              0.200970
1                        0.346679              3.475469              0.207041
2                        0.344061              3.615520              0.201975
3                        0.346872              3.700408              0.203913
4                        0.346070              3.699178              0.197897
..                            …                     …                     …
360                      0.015400              0.473471              0.668127
361                      0.018872              0.473571              0.657405
362                      0.015199              0.493595              0.670818
363                      0.020299              0.512197              0.680587
364                      0.017158              0.514595              0.597449

     Microwave (R) [kW]  Fridge (R) [kW]
0              4.997037         4.639598
1              1.534426         3.881399
2              1.667553         3.671391
3              1.029198         3.357907
4              1.619991         4.373730
..                  …                …
360            0.642506         3.839653
361            0.311556         3.510436
362            0.279923         3.702587
363            0.623743         4.555577
364            0.513711         3.118014

[365 rows x 18 columns]
```

[8]:
```
# merge two dataframes
merged_df = pd.merge(weather_df_gbdate, energy_df_gbdate, on='Date')
```

[9]:
```
merged_df
```

[9]:
```
           Date  temperature  humidity  visibility     pressure  windSpeed  \
0    2014-01-01    20.110833  0.556667    9.970000  1025.395000   6.820417
1    2014-01-02    16.382500  0.784583    3.834583  1023.465833   7.433750
2    2014-01-03     6.256667  0.680833    4.509167  1014.428750  12.828333
3    2014-01-04     2.711667  0.617083    9.822917  1030.096250   5.248333
4    2014-01-05    17.654167  0.682083    9.134583  1025.275000   3.417083
..          …           …         …           …           …          …
360  2014-12-27    35.487083  0.756250    9.246250  1022.081667   3.677083
361  2014-12-28    41.892917  0.763750    9.332917  1013.549167   6.587917
```

```
362  2014-12-29    34.728333  0.592083    9.997083  1018.870833   8.129583
363  2014-12-30    24.846667  0.488750    9.998333  1026.102083   7.566667
364  2014-12-31    19.522917  0.552917    9.986250  1025.940833   5.943750

     cloudCover          time  windBearing  precipIntensity  …  \
0      0.031304  1.388576e+09   252.291667         0.000000  …
1      0.354444  1.388662e+09    53.458333         0.002004  …
2      0.186364  1.388749e+09   207.333333         0.002029  …
3      0.001667  1.388835e+09   240.166667         0.000000  …
4      0.010952  1.388921e+09   208.958333         0.000033  …
..          …           …           …               …  …
360    0.030417  1.419680e+09   243.791667         0.000000  …
361    0.245909  1.419766e+09   224.458333         0.003996  …
362    0.119167  1.419853e+09   281.833333         0.000000  …
363    0.031250  1.419939e+09   312.041667         0.000000  …
364    0.117917  1.420025e+09   260.083333         0.000000  …

     First Floor lights [kW]  Utility Rm + Basement Bath [kW]  \
0                   0.567603                         0.178529
1                   0.506440                         0.178024
2                   0.507426                         0.176649
3                   0.515988                         0.180056
4                   0.519449                         0.178556
..                       …                            …
360                 0.825872                         0.112661
361                 0.874563                         0.115673
362                 0.552812                         0.112204
363                 0.764434                         0.115933
364                 0.559797                         0.114914

     Garage outlets [kW]  MBed + KBed outlets [kW]  Dryer + egauge [kW]  \
0               0.261094                  0.254839           31.938131
1               0.282479                  0.798316            5.423866
2               0.279159                  0.746972            0.005554
3               0.344005                  0.640721           19.994908
4               0.348489                  0.584570            9.493912
..                   …                         …                   …
360             0.255620                  5.422751            0.005953
361             0.257369                 11.602281            0.008270
362             0.274396                  5.951963            0.005461
363             0.238277                 11.100021            0.008893
364             0.273398                  7.741381            0.005618

     Panel GFI (central vac) [kW]  Home Office (R) [kW]  Dining room (R) [kW]  \
0                        0.350291              3.272944              0.200970
1                        0.346679              3.475469              0.207041
2                        0.344061              3.615520              0.201975
```

```
3                        0.346872              3.700408              0.203913
4                        0.346070              3.699178              0.197897
..                            …                     …                     …
360                      0.015400              0.473471              0.668127
361                      0.018872              0.473571              0.657405
362                      0.015199              0.493595              0.670818
363                      0.020299              0.512197              0.680587
364                      0.017158              0.514595              0.597449

     Microwave (R) [kW]  Fridge (R) [kW]
0              4.997037         4.639598
1              1.534426         3.881399
2              1.667553         3.671391
3              1.029198         3.357907
4              1.619991         4.373730
..                  …                …
360            0.642506         3.839653
361            0.311556         3.510436
362            0.279923         3.702587
363            0.623743         4.555577
364            0.513711         3.118014

[365 rows x 29 columns]
```

0.2 Task 2

[10]:
```
merged_df.columns
```

[10]:
```
Index(['Date', 'temperature', 'humidity', 'visibility', 'pressure',
       'windSpeed', 'cloudCover', 'time', 'windBearing', 'precipIntensity',
       'dewPoint', 'precipProbability', 'use [kW]', 'gen [kW]', 'Grid [kW]',
       'AC [kW]', 'Furnace [kW]', 'Cellar Lights [kW]', 'Washer [kW]',
       'First Floor lights [kW]', 'Utility Rm + Basement Bath [kW]',
       'Garage outlets [kW]', 'MBed + KBed outlets [kW]',
       'Dryer + egauge [kW]', 'Panel GFI (central vac) [kW]',
       'Home Office (R) [kW]', 'Dining room (R) [kW]', 'Microwave (R) [kW]',
       'Fridge (R) [kW]'],
      dtype='object')
```

[11]:
```
dataset = pd.DataFrame()
for col in merged_df.columns[:13]:
    dataset[col] = merged_df[col]
```

[12]:
```
dataset
```

[12]:
```
           Date  temperature  humidity  visibility     pressure  windSpeed  \
0    2014-01-01    20.110833  0.556667    9.970000  1025.395000   6.820417
```

```
1    2014-01-02    16.382500  0.784583    3.834583  1023.465833   7.433750
2    2014-01-03     6.256667  0.680833    4.509167  1014.428750  12.828333
3    2014-01-04     2.711667  0.617083    9.822917  1030.096250   5.248333
4    2014-01-05    17.654167  0.682083    9.134583  1025.275000   3.417083
..          …          …       …          …          …        …
360  2014-12-27    35.487083  0.756250    9.246250  1022.081667   3.677083
361  2014-12-28    41.892917  0.763750    9.332917  1013.549167   6.587917
362  2014-12-29    34.728333  0.592083    9.997083  1018.870833   8.129583
363  2014-12-30    24.846667  0.488750    9.998333  1026.102083   7.566667
364  2014-12-31    19.522917  0.552917    9.986250  1025.940833   5.943750

     cloudCover          time  windBearing  precipIntensity   dewPoint  \
0      0.031304  1.388576e+09   252.291667         0.000000   6.362083
1      0.354444  1.388662e+09    53.458333         0.002004  10.737083
2      0.186364  1.388749e+09   207.333333         0.002029  -2.337500
3      0.001667  1.388835e+09   240.166667         0.000000  -8.352083
4      0.010952  1.388921e+09   208.958333         0.000033   8.615000
..          …           …          …                …        …
360    0.030417  1.419680e+09   243.791667         0.000000  27.992500
361    0.245909  1.419766e+09   224.458333         0.003996  34.876250
362    0.119167  1.419853e+09   281.833333         0.000000  21.570000
363    0.031250  1.419939e+09   312.041667         0.000000   7.772083
364    0.117917  1.420025e+09   260.083333         0.000000   5.332500

     precipProbability   use [kW]
0             0.000000  65.013592
1             0.074583  32.305336
2             0.080000  31.164468
3             0.000000  45.287782
4             0.000417  36.316643
..                 …        …
360           0.000000  35.046127
361           0.137917  37.695824
362           0.000000  28.675929
363           0.000000  31.514313
364           0.000000  28.674498

[365 rows x 13 columns]
```

[13]:
```
train_df = dataset[:len(dataset) - 31].drop("Date",axis=1).drop("time",axis=1)
test_df = dataset[len(dataset) - 31:].drop("Date",axis=1).drop("time",axis=1)
```

0.3 Task 3

[14]:
```
train_x = train_df.drop("use [kW]", 1)
train_y = train_df["use [kW]"]
test_x = test_df.drop("use [kW]", 1)
test_y = test_df["use [kW]"]

LR = LinearRegression() ## setting up linear regression model
LR = LR.fit(train_x,train_y) ## training the model

prediction = LR.predict(test_x) ## testing the model
```

```
<ipython-input-14-33cf3576784b>:1: FutureWarning: In a future version of pandas
all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  train_x = train_df.drop("use [kW]", 1)
<ipython-input-14-33cf3576784b>:3: FutureWarning: In a future version of pandas
all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  test_x = test_df.drop("use [kW]", 1)
```

[15]:
```
prediction
```

[15]:
```
array([30.43457312, 31.6556055 , 18.30638052, 31.43589939, 23.81815813,
       21.34638909, 22.95965138, 24.90248154, 20.0580725 , 18.53616071,
       19.5070998 , 21.95908842, 25.62554619, 24.56240186, 27.90688889,
       17.04274926, 23.64611421, 26.08520137, 25.60049498, 25.38361334,
       15.0220098 , 13.78488489, 14.2039299 , 16.89665035, 30.40174451,
       34.00289194, 26.72679948, 27.75019123, 30.47762069, 29.75450607,
       25.71173412])
```

[16]:
```
from sklearn.metrics import mean_squared_error
rmse = mean_squared_error(test_y, prediction, squared=False)
print("root mean squared error: ", rmse)
```

```
root mean squared error:  8.740566311137954
```

[17]:
```
# generate a csv file
pd.DataFrame({"Date":dataset[len(dataset) - 31:]["Date"], "Predicted Value":
 ↪prediction}).to_csv("task3.csv", index=False)
```

0.4 Task 4

[18]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score

temparature_label = [1 if temperature >= 35 else 0 for temperature in␣
 ↪weather_df_gbdate["temperature"]]
```

[19]:
```
logistic_regression_data = weather_df_gbdate.copy()
logistic_regression_data["temperature"] = temparature_label
train_dataset = logistic_regression_data[:len(logistic_regression_data)-31].
 ↪drop(["Date", "time"],axis=1)
test_dataset = logistic_regression_data[len(logistic_regression_data)-31:].
 ↪drop(["Date", "time"],axis=1)

train_x = train_dataset.drop("temperature", 1)
train_y = train_dataset["temperature"]
test_x = test_dataset.drop("temperature", 1)
test_y = test_dataset["temperature"]
```

```
<ipython-input-19-26a38683c9ca>:6: FutureWarning: In a future version of pandas
all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  train_x = train_dataset.drop("temperature", 1)
<ipython-input-19-26a38683c9ca>:8: FutureWarning: In a future version of pandas
all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  test_x = test_dataset.drop("temperature", 1)
```

[20]:
```
model = LogisticRegression(max_iter=10000)
model.fit(train_x,train_y)
prediction = model.predict(test_x)

score = f1_score(test_y, prediction)
print("Fl score:", score)
```

```
Fl score: 0.6486486486486487
```

[21]:
```
# generate a csv file
pd.DataFrame({"Date": weather_df_gbdate[len(weather_df_gbdate) - 31:]["Date"],␣
 ↪"Classification":prediction}).to_csv("task4.csv", index=False)
```

0.5 Task 5

[22]:
```
date_format = "%Y-%m-%d %H:%M:%S"
energy_df["Duration"] = \
['Day' if (19*60*60) > datetime.datetime.strptime(time, date_format).
 ↪timestamp()%(60*60*24) >= 6*60*60 \
 else 'Night' \
 for time in energy_df["Date & Time"]]
```

[23]:
```
energy_df[["Date & Time", "Duration"]].head(40)
```

[23]:

	Date & Time	Duration
0	2014-01-01 00:00:00	Night

```
1   2014-01-01 00:30:00    Night
2   2014-01-01 01:00:00    Night
3   2014-01-01 01:30:00    Night
4   2014-01-01 02:00:00    Night
5   2014-01-01 02:30:00    Night
6   2014-01-01 03:00:00    Night
7   2014-01-01 03:30:00    Night
8   2014-01-01 04:00:00    Night
9   2014-01-01 04:30:00    Night
10  2014-01-01 05:00:00    Night
11  2014-01-01 05:30:00    Night
12  2014-01-01 06:00:00      Day
13  2014-01-01 06:30:00      Day
14  2014-01-01 07:00:00      Day
15  2014-01-01 07:30:00      Day
16  2014-01-01 08:00:00      Day
17  2014-01-01 08:30:00      Day
18  2014-01-01 09:00:00      Day
19  2014-01-01 09:30:00      Day
20  2014-01-01 10:00:00      Day
21  2014-01-01 10:30:00      Day
22  2014-01-01 11:00:00      Day
23  2014-01-01 11:30:00      Day
24  2014-01-01 12:00:00      Day
25  2014-01-01 12:30:00      Day
26  2014-01-01 13:00:00      Day
27  2014-01-01 13:30:00      Day
28  2014-01-01 14:00:00      Day
29  2014-01-01 14:30:00      Day
30  2014-01-01 15:00:00      Day
31  2014-01-01 15:30:00      Day
32  2014-01-01 16:00:00      Day
33  2014-01-01 16:30:00      Day
34  2014-01-01 17:00:00      Day
35  2014-01-01 17:30:00      Day
36  2014-01-01 18:00:00      Day
37  2014-01-01 18:30:00      Day
38  2014-01-01 19:00:00    Night
39  2014-01-01 19:30:00    Night
```

[24]:
```
energy_df_gbduration = energy_df.groupby("Duration").mean()
```

[25]:
```
energy_df_gbduration
```

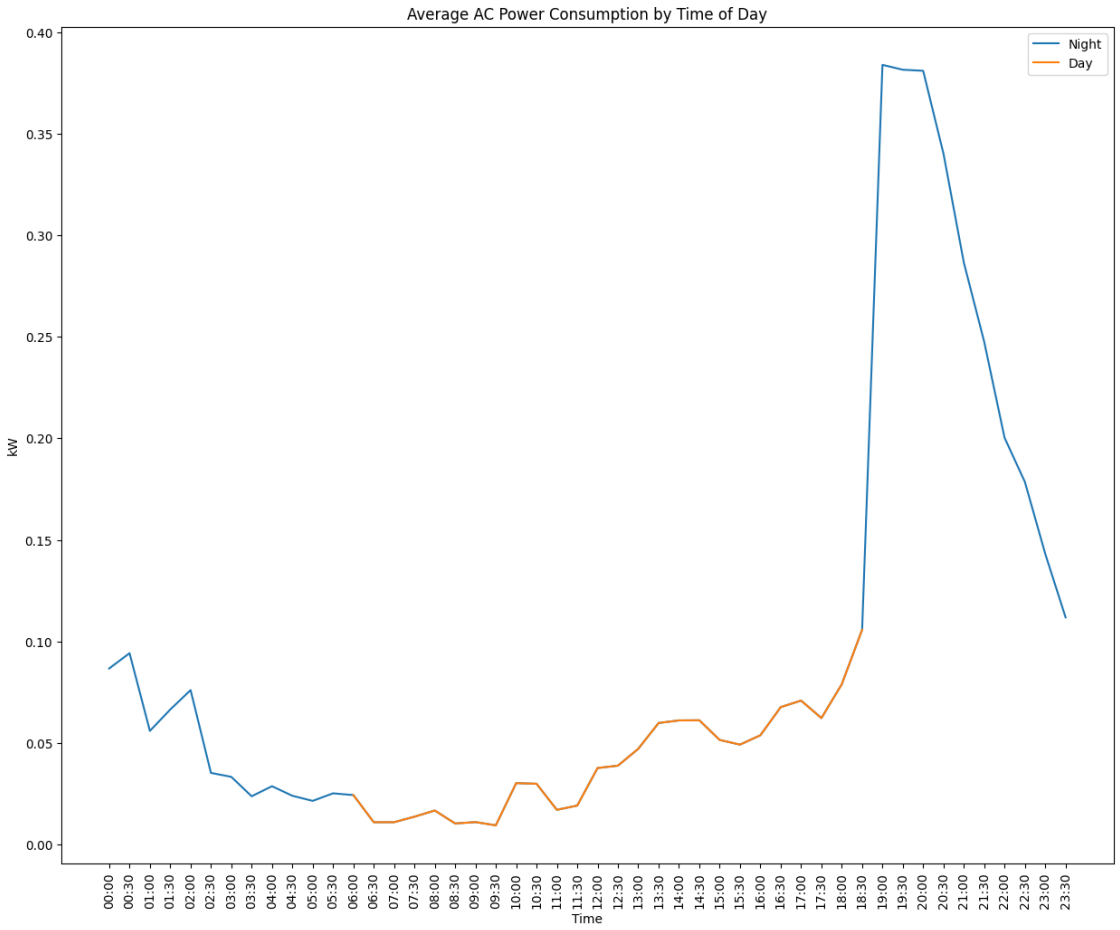
[25]:
```
          use [kW]  gen [kW]  Grid [kW]   AC [kW]  Furnace [kW]  \
Duration
Day       0.658983       0.0   0.658983  0.040263      0.081088
```

```
Night     0.667540       0.0   0.667540  0.146596      0.091561

          Cellar Lights [kW]  Washer [kW]  First Floor lights [kW]  \
Duration
Day                 0.012796     0.004598                 0.013155
Night               0.008955     0.001258                 0.019039

          Utility Rm + Basement Bath [kW]  Garage outlets [kW]  \
Duration
Day                                0.005019             0.006425
Night                              0.005207             0.005386

          MBed + KBed outlets [kW]  Dryer + egauge [kW]  \
Duration
Day                       0.036135             0.103460
Night                     0.057717             0.028489

          Panel GFI (central vac) [kW]  Home Office (R) [kW]  \
Duration
Day                           0.005166              0.054084
Night                         0.004815              0.053245

          Dining room (R) [kW]  Microwave (R) [kW]  Fridge (R) [kW]
Duration
Day                   0.003953            0.021373         0.076055
Night                 0.004461            0.007986         0.070615
```

0.5.1 The chart below depicts that the majority of individuals tend to use AC during the night, with the highest usage occurring at 7pm, which could be due to a number of factors such as people returning home from work or school and wanting to cool down their living spaces before settling in for the evening.

[26]:
```
import matplotlib.pyplot as plt
import matplotlib as mpl

mpl.rcParams['figure.figsize'] = (15, 12)  # set the default figure size

energy_df["time"] = [str(i).split()[-1][:-3] for i in energy_df["Date & Time"]]
energy_df_gbtime = energy_df.groupby("time").mean().reset_index()
plt.plot(energy_df_gbtime["time"], energy_df_gbtime["AC [kW]"], label =␣
 ↪"Night");
plt.plot(energy_df_gbtime["time"][12:38], energy_df_gbtime["AC [kW]"][12:38],␣
 ↪label = "Day");

plt.xticks(rotation=90)

plt.xlabel("Time")
```

```
plt.ylabel("kW")
plt.title("Average AC Power Consumption by Time of Day")

plt.legend()
plt.show()
```



###Contrary to the AC usage pattern, the majority of people tend to use the microwave during the day, with peak hours observed around breakfast and dinner time. It's possible that the peak in usage around breakfast time is related to people heating up their morning coffee or breakfast foods, while the peak around dinner time is related to meal preparation and reheating leftovers.

[27]:
```
plt.bar(energy_df_gbtime["time"], energy_df_gbtime["Microwave (R) [kW]"], label␣
↪= "Night");
plt.bar(energy_df_gbtime["time"][12:38], energy_df_gbtime["Microwave (R)␣
↪[kW]"][12:38], label = "Day");

plt.xticks(rotation=90)
```

```
plt.xlabel("Time")
plt.ylabel("kW")
plt.title("Average Microwave Power Consumption by Time of Day")

plt.legend()
plt.show()
```