NATIONAL INSTITUTE OF TECHNOLOGY
TADEPALLIGUDEM,ANDHRA PRADESH
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Short-term electricity load forecasting

PRESENTED BY:

A.PARAMESWARUDU - 520106
BHOOMIKA - 520117
CH.RUPESH KUMNAR - 520118
G.UDAY CHAITANYA - 520131
MANJUSHA - 520148

As energy demand is increasing globally,the energy management system is becoming more important.Electricity consumption is one among thing which is rapidly increasing.As for of

Resource planning like accurate predictions of electricity consumption are necessary which involves determining the amount of electricity that will be needed in a given period of time this resource planing helps and utilises to ensure that they have enough electricity generation capacity to meet the demand. And next by electricity consumption prediction one can be able to reduce the operating costs by energy providers so that they can optimise their generation production,..etc

And It is used to balance the load on power grid so that power outages,risk of blackouts can be reduced and mainly by predicting electricity consumption it helps to notify the people in advance and help to conserve the energy during those periods thus helps in saving of money on electricity bills and reduce their overall electricity consumption everyday.

Thus we came up with a machine learning model for solving these issues and help the people In the above mentioned aspects .

The ML model we used for prediction of electricity demand in an area for a month using DECISION TREE REGRESSION model .

Designing and Deployment

Collection of Data

source: google data sheets

Machine algorithm coding

Collecting train data

Re-scaling the Features

we use the MinMax scaling method.

Building a model

Making Predictions Using the Final Model

we have to scale the test data.

Evaluating the model on test data

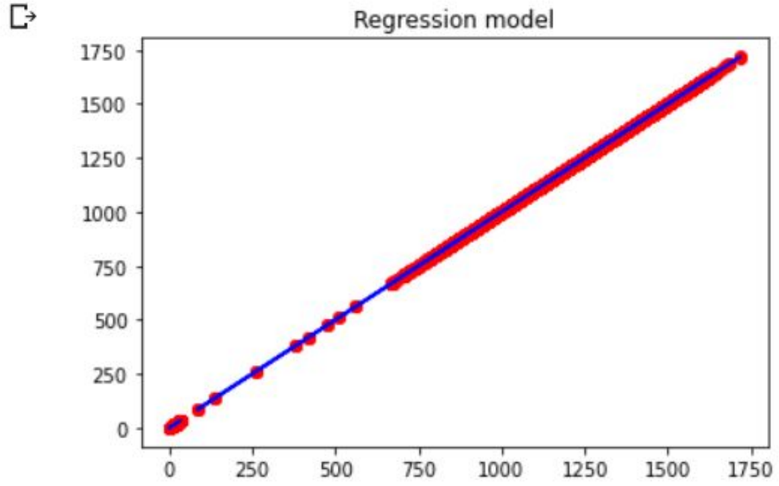
Train data

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	week_X-2	week_X-3	week_X-4	MA_X-4	dayOfWee	weekend	holiday	Holiday_ID	hourOfDay	T2M_toc	DEMAND		
2	1064.749	1204.567	1117.815	1118.154	1	1	0	0	1	26.79153	1161.618		
3	1041.752	1145.534	1082.569	1078.062	1	1	0	0	2	26.70797	1130.464		
4	1021.158	1069.803	1050.125	1040.469	1	1	0	0	3	26.61654	1093.878		
5	990.2549	1047.71	1039.166	1021.726	1	1	0	0	4	26.54229	1083.033		
6	1005.237	1038.584	1039.137	1021.294	1	1	0	0	5	26.46719	1081.271		
7	973.1783	1001.868	1015.55	991.568	1	1	0	0	6	26.55721	1034.659		
8	1031.529	1058.642	1087.745	1058.302	1	1	0	0	7	27.20181	1087.613		
9	1154.605	1198.957	1215.136	1186.062	1	1	0	0	8	28.30975	1222.473		
10	1250.148	1279.342	1304.366	1279.056	1	1	0	0	9	29.49511	1318.157		
11	1340.052	1407.258	1375.544	1367.45	1	1	0	0	10	30.60363	1390.835		
12	1375.7	1386.494	1409.008	1386.027	1	1	0	0	11	31.52697	1422.351		
13	1358.616	1352.078	1400.08	1366.595	1	1	0	0	12	32.16378	1403.727		
14	1354.176	1361.787	1369.549	1347.756	1	1	0	0	13	32.45352	1381.365		
15	1357.193	1367.345	1360.95	1338.173	1	1	0	0	14	32.40798	1378.305		
16	1339.5	1351.295	1341.97	1323.059	1	1	0	0	15	32.03555	1352.117		
17	1297.681	1296.048	1315.645	1291.762	1	1	0	0	16	31.44122	1323.744		
18	1243.472	1248.154	1250.057	1240.555	1	1	0	0	17	30.51122	1273.18		
19	1251.922	1260.34	1255.644	1254.147	1	1	0	0	18	29.33788	1316.245		
20	1313.751	1336.286	1317.837	1316.641	1	1	0	0	19	28.35513	1364.008		
21	1293.673	1292.955	1300.386	1290.894	1	1	0	0	20	27.72732	1343.955		
22	1260.538	1260.059	1288.795	1263.908	1	1	0	0	21	27.33901	1312.52		
23	1208.272	1212.737	1253.106	1213.85	1	1	0	0	22	27.13217	1264.822		
24	1176.861	1142.354	1216.196	1169.831	1	1	0	0	23	26.9739	1212.079		
25	1128.545	1104.462	1161.473	1125.248	2	1	0	0	0	26.85302	1169.301		
26	1095.351	1068.458	1137.631	1092.743	2	1	0	0	1	26.72292	1132.644		
27	1069.897	1042.442	1107.044	1066.252	2	1	0	0	2	26.60699	1100.064		

Building a model

```
import matplotlib.pyplot as plt
import numpy as np
import csv

X= dataset.iloc[1:,1:]
plt.scatter(X,y,color = 'red')
plt.plot(X,regressor.predict(X),color = 'blue')
plt.title("Regression model")
plt.xlabel("")
plt.ylabel('')
plt.show()
```



Test data

week_X-2	week_X-3	week_X-4	MA_X-4	dayOfWee	weekend	holiday	Holiday_ID	hourOfDay	T2M_toc	DEMAND
962.2865	906.958	970.345	938.0049	1	1	0	0	1	25.3085	954.2018
933.3221	863.5135	912.1755	900.2841	1	1	0	0	2	25.14144	913.866
903.9817	848.4447	900.2688	881.7043	1	1	0	0	3	25.00674	903.3637
900.9995	839.8821	889.9538	876.4588	1	1	0	0	4	24.89971	889.0806
904.3481	847.1073	893.6865	879.1908	1	1	0	0	5	24.82156	910.1472
912.508	848.4718	879.2323	877.0279	1	1	0	0	6	24.83019	922.1737
927.5976	897.8229	932.4876	920.3819	1	1	0	0	7	25.79995	939.9442
1075.349	1021.672	1048.972	1057.195	1	1	0	0	8	26.98031	1077.858
1116.643	1108.944	1167.907	1138.179	1	1	0	0	9	28.03182	1179.66
1121.905	1165.476	1257.507	1189.291	1	1	0	0	10	28.90606	1255.157
1149.28	1179.063	1254.583	1202.8	1	1	0	0	11	29.55517	1253.441
1124.328	1173.09	1216.9	1176.489	1	1	0	0	12	30.03652	1223.612
1100.531	1161.412	1202.156	1153.51	1	1	0	0	13	30.26553	1160.284
1081.435	1133.313	1197.262	1138.098	1	1	0	0	14	30.21115	1124.888
1065.033	1100.61	1169.003	1124.01	1	1	0	0	15	29.73275	1112.419
1056.869	1082.688	1136.705	1061.039	1	1	0	0	16	28.98828	1081.741
1053.786	1051.414	1101.945	929.2451	1	1	0	0	17	27.99349	1064.858
1046.294	1088.869	1107.041	1057.654	1	1	0	0	18	27.02188	1095.57
1063.864	1144.272	1142.155	1107.915	1	1	0	0	19	26.52001	1116.665
1054.031	1102.247	1097.233	1072.618	1	1	0	0	20	26.24984	1094.677
1026.633	1064.151	1074.654	1044.417	1	1	0	0	21	26.01879	1075.208
982.6391	1030.457	1041.324	1012.044	1	1	0	0	22	25.87951	1041.083
928.4173	979.5945	999.634	958.0036	1	1	0	0	23	25.74008	988.5723
891.4272	956.5766	968.0526	929.7912	2	1	0	0	0	25.5949	939.5286
873.3819	924.2795	944.0556	905.4873	2	1	0	0	1	25.44845	904.3712
844.9238	887.4576	928.7193	873.7144	2	1	0	0	2	25.32375	883.9448

ERROR VALUE

```
1  y = test.iloc[:,1:].values
2  print(y)
3  error = ((y-y_pred)**2)/10
4  error
5
```

```
[[1.20456680e+03 1.11781520e+03 1.11815425e+03 ... 1.00000000e+00
  2.67915283e+01 1.16161770e+03]
 [1.14553410e+03 1.08256880e+03 1.07806195e+03 ... 2.00000000e+00
  2.67079712e+01 1.13046350e+03]
 [1.06980260e+03 1.05012520e+03 1.04046907e+03 ... 3.00000000e+00
  2.66165405e+01 1.09387770e+03]
 ...
 [1.25780200e+03 1.33115510e+03 1.26953927e+03 ... 2.20000000e+01
  2.74784485e+01 1.13306770e+03]
 [1.20367490e+03 1.29241170e+03 1.22425613e+03 ... 2.30000000e+01
  2.71874634e+01 1.09822150e+03]
 [1.13616850e+03 1.24473800e+03 1.16728503e+03 ... 0.00000000e+00
  2.69726501e+01 1.07889030e+03]]
array([[1.28832710e-01, 2.73799029e-01, 6.13848594e-01, ...,
        2.14880952e+00, 4.77404737e-06, 7.76179337e-01],
       [4.44737196e+00, 4.85330251e-01, 1.34906242e+00, ...,
        5.95238095e-03, 5.32807433e-03, 9.52258886e-01],
       [7.68436407e-01, 8.81146205e-02, 5.94033027e-02, ...,
        2.38095238e+00, 5.54121108e-03, 2.13657604e-01],
       ...,
       [4.93097336e-02, 3.39758462e+00, 6.27053054e-01, ...,
        5.95238095e-03, 1.08399762e-02, 2.87920441e-01],
       [6.84295736e-02, 2.92512038e-02, 1.60423738e+00, ...,
        2.91666667e-01, 2.31177123e-03, 2.89172672e+00],
       [3.22394287e-01, 9.85531176e+00, 2.69323020e-02, ...,
        2.88095238e+00, 6.90741664e-03, 1.36766664e+00]])
```

THANK YOU