In [ ]:

```
from pandas import DataFrame
from pandas import Series
from pandas import concat
from pandas import read_csv
from pandas import datetime
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from math import sqrt
from matplotlib import pyplot
import numpy as np
import pandas as pd
```

```
Using TensorFlow backend.
```

In [ ]:

```
window_size = 48
batch_size_exp = 1
epoch_exp = 10
neurons_exp = 50
predict_values_exp = 6000
lag_exp=48
```

In [ ]:

```
import pandas as pd
data = pd.read_csv('/content/drive/My Drive/T1.csv',index_col="Date/Time")
```

In [ ]:

```
data.to_csv('/content/cleaned_data.csv')
```

In [ ]:

```
dataset = pd.read_csv('/content/cleaned_data.csv', parse_dates=True,index_col="Date/Time")
```

In [ ]:

```
data = dataset.resample('D').sum()
```

```
data.to_csv('/content/cleaned_data.csv')
```

In [ ]:

```
dataset = pd.read_csv('/content/cleaned_data.csv', parse_dates=True,index_col="Date/Time")
```

In [ ]:

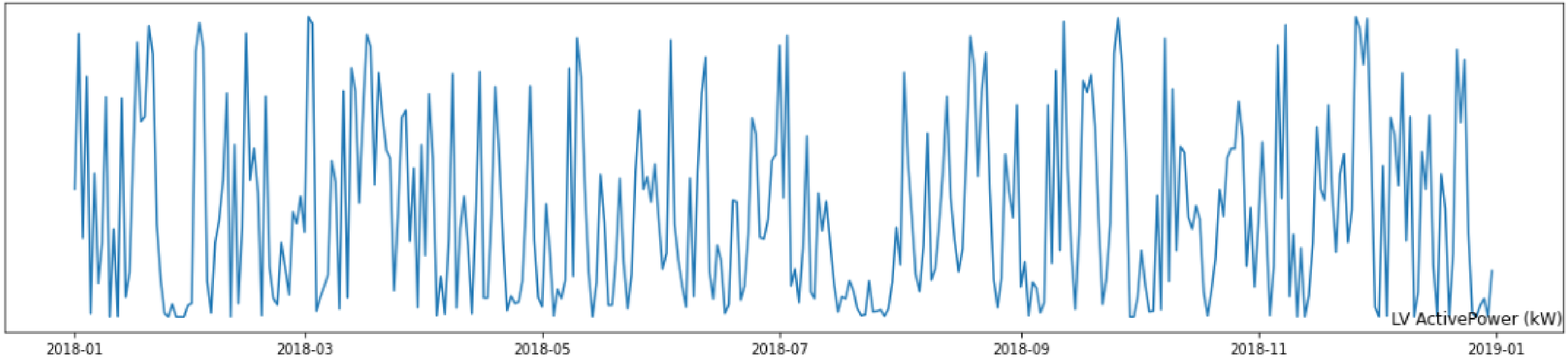
```
data = dataset.resample('D').sum()
```

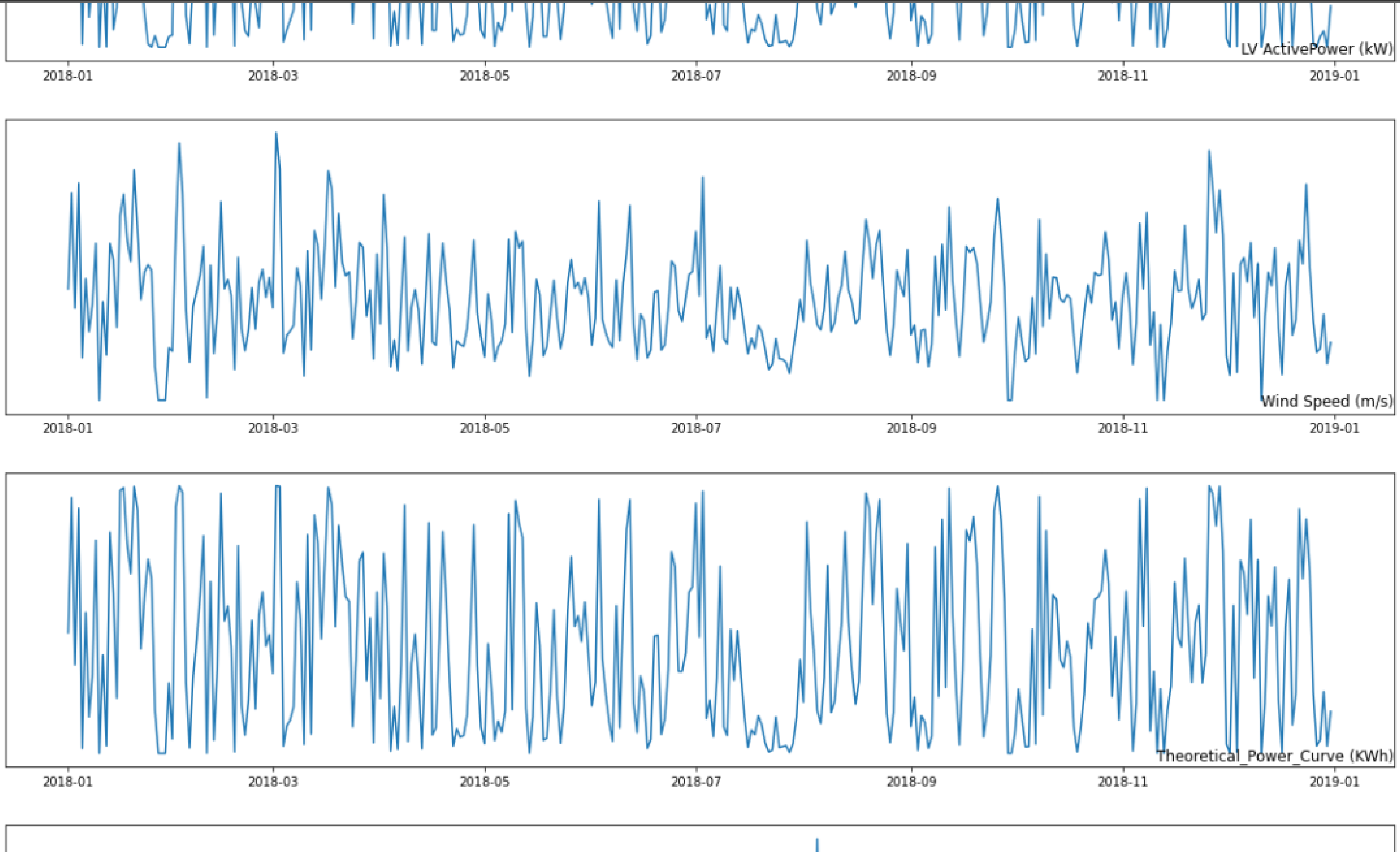
In [ ]:

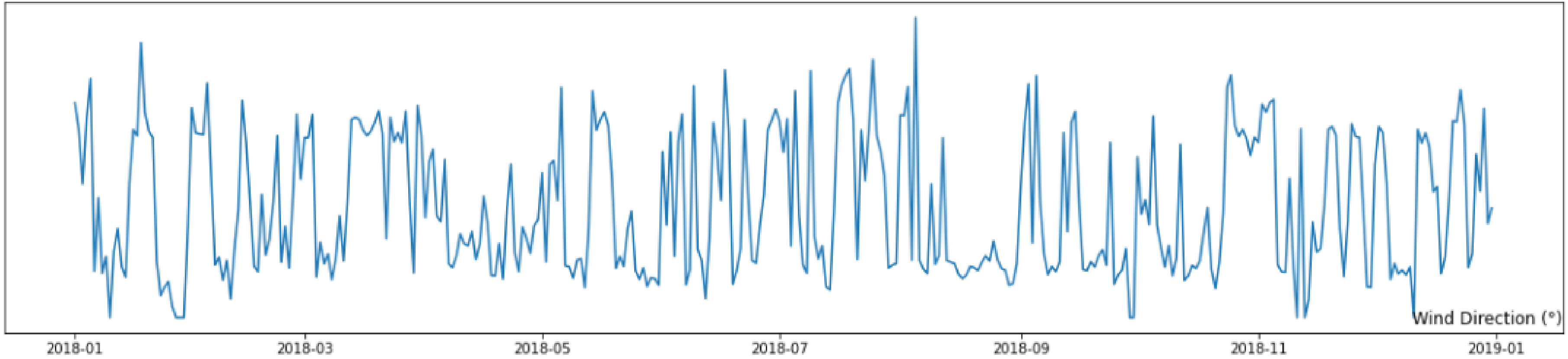
```
from matplotlib import pyplot as plt
```

In [ ]:

```
fig, ax = plt.subplots(figsize=(20,20))
for i in range(len(data.columns)):
  plt.subplot(len(data.columns), 1, i+1)
  name = data.columns[i]
  plt.plot(data[name])
  plt.title(name, y=0, loc='right')
  plt.yticks([])
plt.show()
fig.tight_layout ()
```




LV ActivePower (kW)
2018-01
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01
Wind Speed (m/s)
2018-01
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01
Theoretical_Power_Curve (KWh)
2018-01
2018-03
2018-05
2018-07
2018-09
2018-11
2019-01



In [ ]:
```
years=['2018']
```

In [ ]:
```
data['2018']
```

Out[ ]:

	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
Date/Time				
2018-01-01	221069.445599	1155.308353	233710.566903	33081.874832
2018-01-02	489997.826721	2153.488136	495800.747854	29051.079056
2018-01-03	136232.521423	954.536103	171240.275532	20612.306661
2018-01-04	415684.403381	2255.877879	474600.579209	30733.485031
2018-01-05	5638.593391	440.559367	9638.282642	36843.185165
...	...	...	...	...
2018-12-27	0.000000	498.415094	14472.857153	25175.384390

In [ ]:

```
data_train = data.loc[:'2018-10-31',:]['Wind Direction (°)']
data_train
```

Out[ ]:

```
Date/Time
2018-01-01    33081.874832
2018-01-02    29051.079056
2018-01-03    20612.306661
2018-01-04    30733.485031
2018-01-05    36843.185165
                  ...
2018-10-27    27909.599655
2018-10-28    29007.160919
2018-10-29    27611.306152
2018-10-30    24990.768890
2018-10-31    27780.833587
Freq: D, Name: Wind Direction (°), Length: 304, dtype: float64
```

In [ ]:

```
data_test = data.loc['2018-11-01':'2018-12-31',:]['Wind Direction (°)']
data_test
```

Out[ ]:

```
Date/Time
2018-11-01    26989.353104
2018-11-02    32912.813599
2018-11-03    31654.088837
2018-11-04    33115.995506
2018-11-05    33630.758800
                  ...
2018-12-27    25175.384390
2018-12-28    19508.626114
2018-12-29    32254.909293
2018-12-30    14497.344503
2018-12-31    16871.772377
Freq: D, Name: Wind Direction (°), Length: 61, dtype: float64
```

In [ ]:

```
data_train.shape
```

Out[ ]: (304,)

In [ ]:
```
data_test.shape
```

Out[ ]: (61,)

In [ ]:
```
data_train.head()
```

Out[ ]:
```
Date/Time
2018-01-01    33081.874832
2018-01-02    29051.079056
2018-01-03    20612.306661
2018-01-04    30733.485031
2018-01-05    36843.185165
Freq: D, Name: Wind Direction (°), dtype: float64
```

In [ ]:
```
data_train = np.array(data_train)
```

In [ ]:
```
X_train, y_train = [],[]

for i in range(3, len(data_train)-3):
  X_train.append(data_train[i-3:i])
  y_train.append(data_train[i:i+3])
```

In [ ]:
```
X_train,y_train=np.array(X_train),np.array(y_train)
```

In [ ]:
```
X_train.shape, y_train.shape
```

Out[ ]: ((298, 3), (298, 3))

In [ ]:

```
pd.DataFrame(X_train).head()
```

Out[ ]:

	0	1	2
0	33081.874832	29051.079056	20612.306661
1	29051.079056	20612.306661	30733.485031
2	20612.306661	30733.485031	36843.185165
3	30733.485031	36843.185165	7151.613260
4	36843.185165	7151.613260	18498.549602

In [ ]:

```
x_scalar = MinMaxScaler()
X_train = x_scalar.fit_transform(X_train)
```

In [ ]:

```
y_scalar = MinMaxScaler()
y_train = y_scalar.fit_transform(y_train)
```

In [ ]:

```
pd.DataFrame(X_train).head()
```

Out[ ]:

	0	1	2
0	0.715178	0.628038	0.445605
1	0.628038	0.445605	0.664409
2	0.445605	0.664409	0.796491
3	0.664409	0.796491	0.154607
4	0.796491	0.154607	0.399909

In [ ]:

```
X_train.shape
```

Out[ ]: (298, 3)

In [ ]:
```
X_train = X_train.reshape(298, 3, 1)
```

In [ ]:
```
X_train.shape
```

Out[ ]: (298, 3, 1)

In [ ]:
```
reg = Sequential()
reg.add(LSTM(units = 200, activation = 'relu', input_shape=(3,1)))
reg.add(Dense(3))
```

In [ ]:
```
reg.compile(loss='mse', optimizer='adam')
```

In [ ]:
```
reg.fit(X_train, y_train, epochs = 100)
```

```
Epoch 1/100
298/298 [==============================] - 1s 2ms/step - loss: 0.1677
Epoch 2/100
298/298 [==============================] - 0s 368us/step - loss: 0.1229
Epoch 3/100
298/298 [==============================] - 0s 363us/step - loss: 0.0717
Epoch 4/100
298/298 [==============================] - 0s 377us/step - loss: 0.0580
Epoch 5/100
298/298 [==============================] - 0s 360us/step - loss: 0.0529
Epoch 6/100
298/298 [==============================] - 0s 374us/step - loss: 0.0519
Epoch 7/100
298/298 [==============================] - 0s 348us/step - loss: 0.0509
Epoch 8/100
298/298 [==============================] - 0s 397us/step - loss: 0.0504
Epoch 9/100
298/298 [==============================] - 0s 357us/step - loss: 0.0502
```

```
Epoch 12/100
298/298 [==============================] - 0s 381us/step - loss: 0.0496
Epoch 13/100
298/298 [==============================] - 0s 364us/step - loss: 0.0494
Epoch 14/100
298/298 [==============================] - 0s 363us/step - loss: 0.0498
Epoch 15/100
298/298 [==============================] - 0s 371us/step - loss: 0.0493
Epoch 16/100
298/298 [==============================] - 0s 371us/step - loss: 0.0490
Epoch 17/100
298/298 [==============================] - 0s 352us/step - loss: 0.0488
Epoch 18/100
298/298 [==============================] - 0s 362us/step - loss: 0.0487
Epoch 19/100
298/298 [==============================] - 0s 354us/step - loss: 0.0486
Epoch 20/100
298/298 [==============================] - 0s 397us/step - loss: 0.0483
Epoch 21/100
298/298 [==============================] - 0s 363us/step - loss: 0.0487
Epoch 22/100
298/298 [==============================] - 0s 356us/step - loss: 0.0485
Epoch 23/100
298/298 [==============================] - 0s 332us/step - loss: 0.0485
Epoch 24/100
298/298 [==============================] - 0s 339us/step - loss: 0.0484
Epoch 25/100
298/298 [==============================] - 0s 382us/step - loss: 0.0481
Epoch 26/100
298/298 [==============================] - 0s 355us/step - loss: 0.0478
Epoch 27/100
298/298 [==============================] - 0s 351us/step - loss: 0.0478
Epoch 28/100
298/298 [==============================] - 0s 340us/step - loss: 0.0480
Epoch 29/100
298/298 [==============================] - 0s 373us/step - loss: 0.0476
Epoch 30/100
298/298 [==============================] - 0s 341us/step - loss: 0.0476
Epoch 31/100
298/298 [==============================] - 0s 352us/step - loss: 0.0473
```

```
298/298 [==============================] - 0s 334us/step - loss: 0.0470
Epoch 37/100
298/298 [==============================] - 0s 347us/step - loss: 0.0472
Epoch 38/100
298/298 [==============================] - 0s 373us/step - loss: 0.0468
Epoch 39/100
298/298 [==============================] - 0s 368us/step - loss: 0.0469
Epoch 40/100
298/298 [==============================] - 0s 334us/step - loss: 0.0466
Epoch 41/100
298/298 [==============================] - 0s 327us/step - loss: 0.0466
Epoch 42/100
298/298 [==============================] - 0s 362us/step - loss: 0.0465
Epoch 43/100
298/298 [==============================] - 0s 363us/step - loss: 0.0468
Epoch 44/100
298/298 [==============================] - 0s 351us/step - loss: 0.0465
Epoch 45/100
298/298 [==============================] - 0s 360us/step - loss: 0.0464
Epoch 46/100
298/298 [==============================] - 0s 371us/step - loss: 0.0462
Epoch 47/100
298/298 [==============================] - 0s 350us/step - loss: 0.0462
Epoch 48/100
298/298 [==============================] - 0s 361us/step - loss: 0.0462
Epoch 49/100
298/298 [==============================] - 0s 345us/step - loss: 0.0462
Epoch 50/100
298/298 [==============================] - 0s 352us/step - loss: 0.0464
Epoch 51/100
298/298 [==============================] - 0s 339us/step - loss: 0.0459
Epoch 52/100
298/298 [==============================] - 0s 350us/step - loss: 0.0462
Epoch 53/100
298/298 [==============================] - 0s 343us/step - loss: 0.0458
Epoch 54/100
298/298 [==============================] - 0s 338us/step - loss: 0.0459
Epoch 55/100
298/298 [==============================] - 0s 339us/step - loss: 0.0462
Epoch 56/100
298/298 [==============================] - 0s 377us/step - loss: 0.0463
```

```
Epoch 56/100
298/298 [==============================] - 0s 377us/step - loss: 0.0463
Epoch 57/100
298/298 [==============================] - 0s 337us/step - loss: 0.0465
Epoch 58/100
298/298 [==============================] - 0s 370us/step - loss: 0.0458
Epoch 59/100
298/298 [==============================] - 0s 396us/step - loss: 0.0457
Epoch 60/100
298/298 [==============================] - 0s 354us/step - loss: 0.0460
Epoch 61/100
298/298 [==============================] - 0s 348us/step - loss: 0.0458
Epoch 62/100
298/298 [==============================] - 0s 351us/step - loss: 0.0460
Epoch 63/100
298/298 [==============================] - 0s 345us/step - loss: 0.0460
Epoch 64/100
298/298 [==============================] - 0s 342us/step - loss: 0.0461
Epoch 65/100
298/298 [==============================] - 0s 374us/step - loss: 0.0457
Epoch 66/100
298/298 [==============================] - 0s 345us/step - loss: 0.0455
Epoch 67/100
298/298 [==============================] - 0s 391us/step - loss: 0.0455
Epoch 68/100
298/298 [==============================] - 0s 348us/step - loss: 0.0458
Epoch 69/100
298/298 [==============================] - 0s 362us/step - loss: 0.0457
Epoch 70/100
298/298 [==============================] - 0s 335us/step - loss: 0.0454
Epoch 71/100
298/298 [==============================] - 0s 342us/step - loss: 0.0456
Epoch 72/100
298/298 [==============================] - 0s 342us/step - loss: 0.0461
Epoch 73/100
298/298 [==============================] - 0s 357us/step - loss: 0.0455
Epoch 74/100
298/298 [==============================] - 0s 335us/step - loss: 0.0456
Epoch 75/100
298/298 [==============================] - 0s 331us/step - loss: 0.0456
```

```
Epoch 76/100
298/298 [==============================] - 0s 344us/step - loss: 0.0454
Epoch 77/100
298/298 [==============================] - 0s 370us/step - loss: 0.0454
Epoch 78/100
298/298 [==============================] - 0s 364us/step - loss: 0.0458
Epoch 79/100
298/298 [==============================] - 0s 327us/step - loss: 0.0454
Epoch 80/100
298/298 [==============================] - 0s 380us/step - loss: 0.0454
Epoch 81/100
298/298 [==============================] - 0s 352us/step - loss: 0.0453
Epoch 82/100
298/298 [==============================] - 0s 344us/step - loss: 0.0452
Epoch 83/100
298/298 [==============================] - 0s 338us/step - loss: 0.0458
Epoch 84/100
298/298 [==============================] - 0s 350us/step - loss: 0.0453
Epoch 85/100
298/298 [==============================] - 0s 344us/step - loss: 0.0455
Epoch 86/100
298/298 [==============================] - 0s 352us/step - loss: 0.0458
Epoch 87/100
298/298 [==============================] - 0s 372us/step - loss: 0.0452
Epoch 88/100
298/298 [==============================] - 0s 363us/step - loss: 0.0455
Epoch 89/100
298/298 [==============================] - 0s 340us/step - loss: 0.0452
Epoch 90/100
298/298 [==============================] - 0s 331us/step - loss: 0.0452
Epoch 91/100
298/298 [==============================] - 0s 359us/step - loss: 0.0452
Epoch 92/100
298/298 [==============================] - 0s 356us/step - loss: 0.0451
Epoch 93/100
298/298 [==============================] - 0s 344us/step - loss: 0.0456
Epoch 94/100
298/298 [==============================] - 0s 341us/step - loss: 0.0452
Epoch 95/100
298/298 [==============================] - 0s 368us/step - loss: 0.0453
```

```
Epoch 97/100
298/298 [==============================] - 0s 358us/step - loss: 0.0453
Epoch 98/100
298/298 [==============================] - 0s 348us/step - loss: 0.0452
Epoch 99/100
298/298 [==============================] - 0s 372us/step - loss: 0.0452
Epoch 100/100
298/298 [==============================] - 0s 352us/step - loss: 0.0455
```

Out[ ]:

In [ ]:

```
data_test = np.array(data_test)
```

In [ ]:

```
X_test, y_test = [], []

for i in range(3, len(data_test)-3):
  X_test.append(data_test[i-3:i])
  y_test.append(data_test[i:i+3])
```

In [ ]:

```
X_test, y_test=np.array(X_test), np.array(y_test)
```

In [ ]:

```
X_test = x_scalar.transform(X_test)
y_test = y_scalar.transform(y_test)
```

In [ ]:

```
X_test.shape
```

Out[ ]: (55, 3)

In [ ]:

```
X_test = X_test.reshape(55, 3, 1)
```

In [ ]:

```
y_pred = reg.predict(X_test)
```

In [ ]:
```
y_pred = reg.predict(X_test)
```

In [ ]:
```
y_pred = y_scalar.inverse_transform(y_pred)
```

In [ ]:
```
y_true = y_scalar.inverse_transform(y_test)
```

In [ ]:
```
y_true
```

Out[ ]:
```
array([[33115.99550629, 33630.75879967,  8131.71406555],
       [33630.75879967,  8131.71406555,  7063.8345356 ],
       [ 8131.71406555,  7063.8345356 ,  6995.28148079],
       [ 7063.8345356 ,  6995.28148079, 21468.86600792],
       [ 6995.28148079, 21468.86600792,  8157.80950069],
       [21468.86600792,  8157.80950069,     0.        ],
       [ 8157.80950069,     0.        , 29116.72072411],
       [    0.        , 29116.72072411,     0.        ],
       [29116.72072411,     0.        ,  2793.34759331],
       [    0.        ,  2793.34759331, 14737.10392174],
       [ 2793.34759331, 14737.10392174, 10112.13514081],
       [14737.10392174, 10112.13514081, 10613.06708145],
       [10112.13514081, 10613.06708145, 17370.42786407],
       [10613.06708145, 17370.42786407, 29016.88545227],
       [17370.42786407, 29016.88545227, 29461.56741333],
       [29016.88545227, 29461.56741333, 28115.14625549],
       [29461.56741333, 28115.14625549, 13706.64304161],
       [28115.14625549, 13706.64304161,  6312.51927948],
       [13706.64304161,  6312.51927948, 14784.38091278],
       [ 6312.51927948, 14784.38091278, 29879.50546265],
       [14784.38091278, 29879.50546265, 27907.6651001 ],
       [29879.50546265, 27907.6651001 , 27763.02044678],
       [27907.6651001 , 27763.02044678, 17034.9897837 ],
       [27763.02044678, 17034.9897837 ,  4724.49790764],
       [17034.9897837 ,  4724.49790764,  4672.6817627 ],
       [ 4724.49790764,  4672.6817627 , 23593.9849329 ],
```

```
       [ 6713.11179161,  7321.37919235,  6543.93325424],
       [ 7321.37919235,  6543.93325424,  7789.27488327],
       [ 6543.93325424,  7789.27488327,     0.        ],
       [ 7789.27488327,     0.        , 28994.74107689],
       [    0.        , 28994.74107689, 26877.00087738],
       [28994.74107689, 26877.00087738, 28466.81945801],
       [26877.00087738, 28466.81945801, 26306.70462036],
       [28466.81945801, 26306.70462036, 19395.79733944],
       [26306.70462036, 19395.79733944, 20196.71740341],
       [19395.79733944, 20196.71740341,  6770.31429863],
       [20196.71740341,  6770.31429863,  9318.74159431],
       [ 6770.31429863,  9318.74159431, 18168.07575607],
       [ 9318.74159431, 18168.07575607, 30330.28541565],
       [18168.07575607, 30330.28541565, 30188.12014771],
       [30330.28541565, 30188.12014771, 35097.61787415],
       [30188.12014771, 35097.61787415, 29784.41993713],
       [35097.61787415, 29784.41993713,  7745.90272427],
       [29784.41993713,  7745.90272427,  9866.29644394],
       [ 7745.90272427,  9866.29644394, 25175.38438982],
       [ 9866.29644394, 25175.38438982, 19508.62611389],
       [25175.38438982, 19508.62611389, 32254.90929317],
       [19508.62611389, 32254.90929317, 14497.3445034 ]])
```

In [ ]:

```
def evaluate_model(y_true, y_predicted):
  scores = []
  #Calculate scores for each day
  for i in range(y_true.shape[1]):
    mse = mean_squared_error(y_true[:, i], y_predicted[:, i])
    rmse = np.sqrt(mse)
    scores.append(rmse)

    #calculate score for whole prediction
    total_score = 0
    for row in range(y_true.shape[0]):
      for col in range(y_predicted.shape[1]):
        total_score = total_score + (y_true[row, col] - y_predicted[row, col])**2
    total_score = np.sqrt(total_score/(y_true.shape[0]*y_predicted.shape[1]))
    return total_score, scores
```

```python
  scores = []
  #Calculate scores for each day
  for i in range(y_true.shape[1]):
    mse = mean_squared_error(y_true[:, i], y_predicted[:, i])
    rmse = np.sqrt(mse)
    scores.append(rmse)

    #calculate score for whole prediction
    total_score = 0
    for row in range(y_true.shape[0]):
      for col in range(y_predicted.shape[1]):
        total_score = total_score + (y_true[row, col] - y_predicted[row, col])**2
    total_score = np.sqrt(total_score/(y_true.shape[0]*y_predicted.shape[1]))
    return total_score, scores
```

In [ ]:

```python
evaluate_model(y_true, y_pred)
```

Out[ ]: (10735.520841171958, [9883.835615418826])

In [ ]:

```python
np.std(y_true[0])
```

Out[ ]: 11900.88970862046

In [ ]: