MANAI MORTADHA _ 3GII/SSE

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
df=pd.read_excel("/content/consumption.xlsx")
df.head()
```

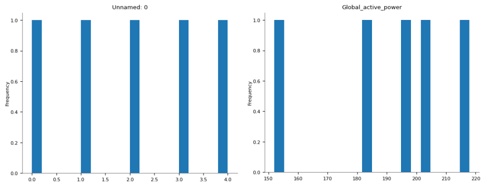
1 to 5 of 5 entries Filter

index	Unnamed: 0	datetime	Global_active_power ▼
1	1	2006-12-16 18:00:00	217.932
2	2	2006-12-16 19:00:00	204.014
3	3	2006-12-16 20:00:00	196.114
4	4	2006-12-16 21:00:00	183.388
0	0	2006-12-16 17:00:00	152.024

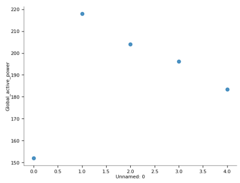
Show 25 per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

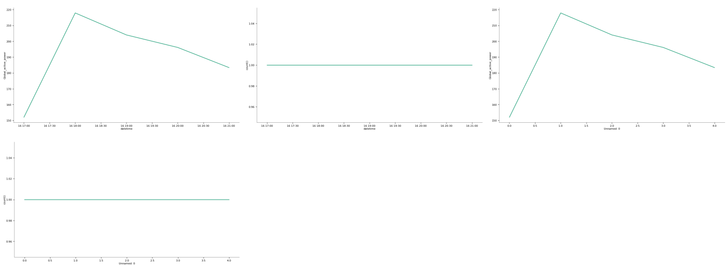
Distributions



2-d distributions



Time series



Values

```
def load_data(data, seq_len, column_index ):
    x = []
    y = []
    for i in range(seq_len, len(data)):
        x.append(data.iloc[i-seq_len : i, 2:3])
        y.append(data.iloc[i, 2:3])
    return np.array(x), np.array(y)
```

```
# Predict 'Global_active_power'
training_size = int(len(df)*0.8)
x, y = load_data(df, 20, 2)

x_train = x[:training_size]
y_train = y[:training_size]
x_test = x[training_size:]
y_test = y[training_size:]

reg = LinearRegression().fit(x_train.reshape(x_train.shape[0], -1), y_train)
y_pred = reg.predict(x_test.reshape(x_test.shape[0], -1))
```

```
x_train
```

```
array([[[152.024],
        [217.932],
        [204.014],
        ...,
        [217.734],
        [148.26 ],
        [114.952]],

       [[217.932],
        [204.014],
        [196.114],
        ...,
        [148.26 ],
        [114.952],
        [ 99.646]],

       [[204.014],
        [196.114],
        [183.388],
        ...,
        [114.952],
        [ 99.646],
        [125.558]],

       ...,

       [[ 52.294],
        [ 18.344],
        [ 21.426],
        ...,
        [ 94.156],
        [176.376],
        [ 95.372]],

       [[ 18.344],
        [ 21.426],
        [ 28.664],
        ...,
        [176.376],
        [ 95.372],
        [131.892]],

       [[ 21.426],
        [ 28.664],
        [ 39.316],
        ...,
        [ 95.372],
        [131.892],
        [129.16 ]]])
```

```
y_train
```

```
array([[99.646],
       [125.558],
       [179.124],
       ...,
       [131.892],
       [129.16],
       [130.652]], dtype=object)
```

```
y_test
```

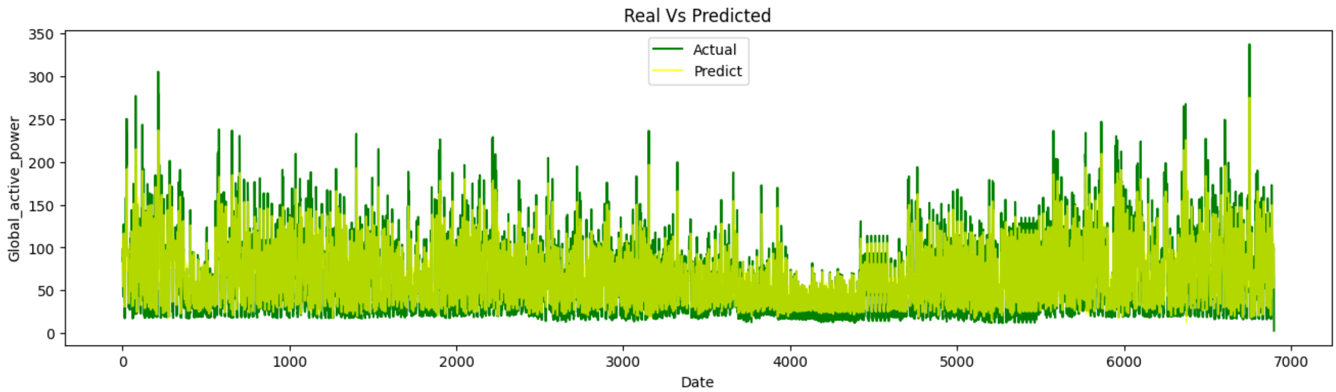
```
array([[98.156],
       [83.74],
       [86.452],
       ...,
       [99.56],
       [69.822],
       [2.804]], dtype=object)
```

```
x_test
```

```
array([[[ 28.664],
        [ 39.316],
        [ 86.182],
        ...,
        [131.892],
        [129.16 ],
        [130.652]],

       [[ 39.316],
        [ 86.182],
        [139.194],
        ...,
```

```
           [129.16 ],
           [130.652],
           [ 98.156]],

          [[ 86.182],
           [139.194],
           [129.596],
           ...,
           [130.652],
           [ 98.156],
           [ 83.74 ]],

          ...,

          [[ 57.42 ],
           [ 17.658],
           [ 16.86 ],
           ...,
           [ 64.076],
           [103.554],
           [ 94.408]],

          [[ 17.658],
           [ 16.86 ],
           [ 16.406],
           ...,
           [103.554],
           [ 94.408],
           [ 99.56 ]],

          [[ 16.86 ],
           [ 16.406],
           [ 17.902],
           ...,
           [ 94.408],
           [ 99.56 ],
           [ 69.822]]]])
```

```
plt.figure(figsize=(16,4))
plt.plot(y_test, color='green',label='Actual')
plt.plot(y_pred, alpha=0.7, color='yellow',label='Predict')
plt.title('Real Vs Predicted')
plt.xlabel('Date')
plt.ylabel('Global_active_power')
plt.legend()
plt.show()
```




```
rmse = np.sqrt(mean_squared_error(y_test, y_pred1)).round(2)
# Exclude zero values
mask = y_test != 0
mape = np.round(np.mean(np.abs(y_test[mask]-y_pred[mask])/y_test[mask])*100,2)

#mape = np.round(np.mean(np.abs(y_test-y_pred)/y_test)*100,2)

print('RMSE:', rmse)
print('MAPE:', mape)
```

```
    RMSE: 32.02
    MAPE: 54.32
```