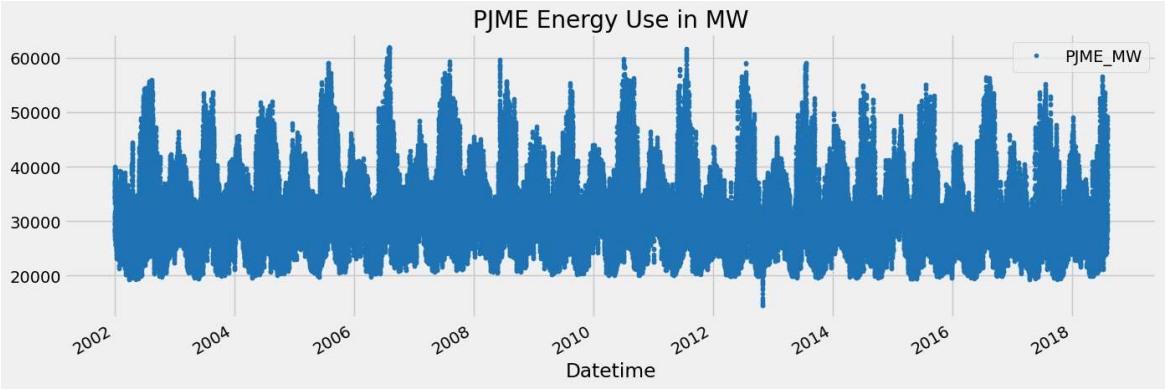
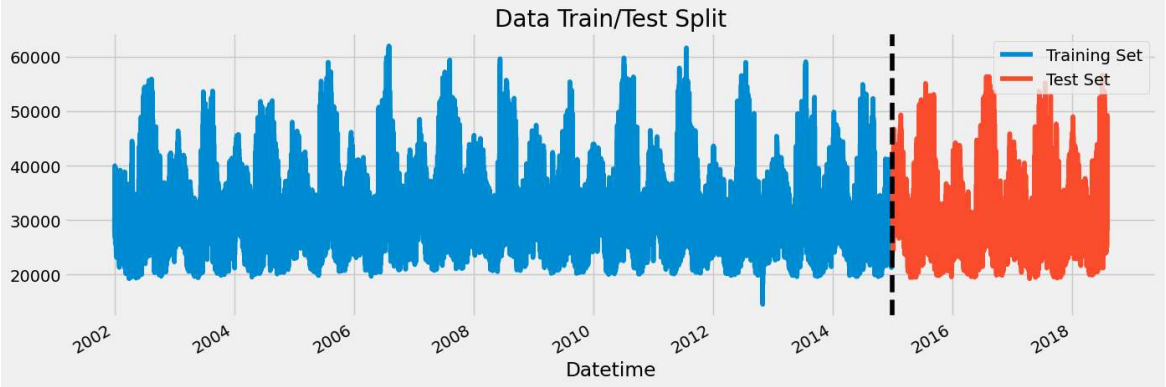
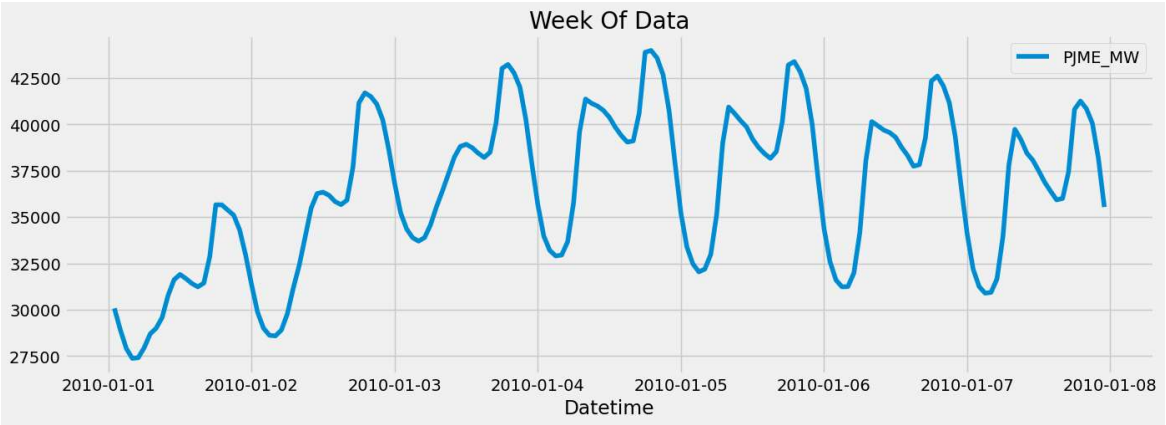
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


import xgboost as xgb
from sklearn.metrics import mean_squared_error
color_pal = sns.color_palette()
plt.style.use('fivethirtyeight')


df = pd.read_csv('PJME_hourly.csv')
df = df.set_index('Datetime')
df.index = pd.to_datetime(df.index)


df.plot(style='.',
        figsize=(15, 5),
        color=color_pal[0],
        title='PJME Energy Use in MW')
plt.show()
```




```
train = df.loc[df.index < '01-01-2015']
test = df.loc[df.index >= '01-01-2015']

train = df.loc[df.index < '01-01-2015']
test = df.loc[df.index >= '01-01-2015']


fig, ax = plt.subplots(figsize=(15, 5))
train.plot(ax=ax, label='Training Set', title='Data Train/Test Split')
test.plot(ax=ax, label='Test Set')
ax.axvline('01-01-2015', color='black', ls='--')
ax.legend(['Training Set', 'Test Set'])
plt.show()
```



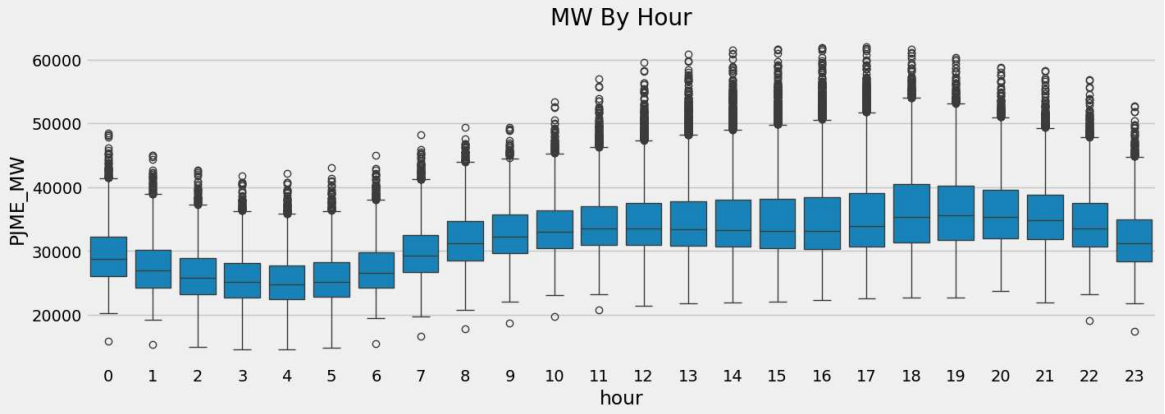

```
fig, ax  = plt.subplots(figsize=(15,5))
sns.lineplot(data=df.loc[(df.index > '2010-01-01') & (df.index < '2010-01-08')])
plt.title('Week Of Data')
plt.show()
```




```
def create_features(df):
    df = df.copy()
    df['hour'] = df.index.hour
    df['dayofweek'] = df.index.dayofweek
    df['quarter'] = df.index.quarter
    df['month'] = df.index.month
    df['year'] = df.index.year
    df['dayofyear'] = df.index.dayofyear
    df['dayofmonth'] = df.index.day
    df['weekofyear'] = df.index.isocalendar().week
    return df

df = create_features(df)
```

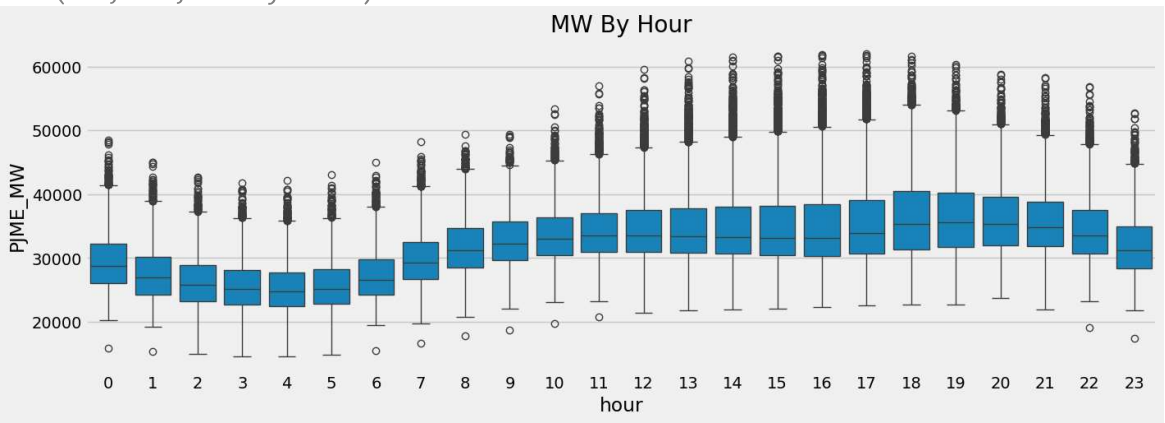
```
fig, ax  = plt.subplots(figsize=(15,5))
sns.boxplot(df,x='hour',y='PJME_MW')
ax.set_title('MW By Hour')
```

```
Text(0.5, 1.0, 'MW By Hour')
```




```
fig, ax  = plt.subplots(figsize=(15,5))
sns.boxplot(df,x='hour',y='PJME_MW')
ax.set_title('MW By Hour')
```

```
Text(0.5, 1.0, 'MW By Hour')
```




```
train = create_features(train)
test = create_features(test)
```

```
FEATURES = ['hour','dayofweek','month','quarter','dayofyear','year','dayofmonth','weekofyear']
TARGET= 'PJME_MW'
```

```
X_train = train[FEATURES]
y_train = train[TARGET]

X_test = test[FEATURES]
y_test = test[TARGET]
```

```
reg = xgb.XGBRegressor(n_estimators = 1000,early_stopping_rounds=50,learning_rate = 0.01)
reg.fit(X_train,y_train,eval_set=[(X_train,y_train),(X_test,y_test)])
```

```
[372]   validation_0-rmse:2870.31200    validation_1-rmse:3738.42736
[373]   validation_0-rmse:2868.81207    validation_1-rmse:3738.36724
[374]   validation_0-rmse:2868.16869    validation_1-rmse:3738.15687
[375]   validation_0-rmse:2866.03174    validation_1-rmse:3738.41079
[376]   validation_0-rmse:2865.36819    validation_1-rmse:3738.26432
[377]   validation_0-rmse:2863.24986    validation_1-rmse:3738.34085
[378]   validation_0-rmse:2861.76744    validation_1-rmse:3738.55812
[379]   validation_0-rmse:2861.14563    validation_1-rmse:3738.28292
[380]   validation_0-rmse:2859.08678    validation_1-rmse:3738.33655
[381]   validation_0-rmse:2858.43448    validation_1-rmse:3738.16474
[382]   validation_0-rmse:2856.40101    validation_1-rmse:3738.51241
[383]   validation_0-rmse:2854.95434    validation_1-rmse:3738.87711
[384]   validation_0-rmse:2854.38715    validation_1-rmse:3738.66543
[385]   validation_0-rmse:2852.41636    validation_1-rmse:3738.81413
[386]   validation_0-rmse:2851.82719    validation_1-rmse:3738.52508
[387]   validation_0-rmse:2849.03133    validation_1-rmse:3738.91568
[388]   validation_0-rmse:2847.62612    validation_1-rmse:3739.31607
[389]   validation_0-rmse:2847.04543    validation_1-rmse:3739.06852
[390]   validation_0-rmse:2845.04718    validation_1-rmse:3739.47185
[391]   validation_0-rmse:2844.50698    validation_1-rmse:3739.25359
[392]   validation_0-rmse:2842.21483    validation_1-rmse:3739.62302
[393]   validation_0-rmse:2840.85553    validation_1-rmse:3739.96833
[394]   validation_0-rmse:2840.24593    validation_1-rmse:3739.83430
[395]   validation_0-rmse:2838.30211    validation_1-rmse:3740.47916
[396]   validation_0-rmse:2837.69509    validation_1-rmse:3740.24160
[397]   validation_0-rmse:2834.21322    validation_1-rmse:3741.54565
[398]   validation_0-rmse:2832.31367    validation_1-rmse:3742.00943
[399]   validation_0-rmse:2830.89344    validation_1-rmse:3742.43845
[400]   validation_0-rmse:2829.91995    validation_1-rmse:3742.53191
[401]   validation_0-rmse:2826.52117    validation_1-rmse:3743.86795
[402]   validation_0-rmse:2825.12237    validation_1-rmse:3744.43689
[403]   validation_0-rmse:2823.59232    validation_1-rmse:3744.89431
[404]   validation_0-rmse:2823.02015    validation_1-rmse:3744.61571
[405]   validation_0-rmse:2821.09646    validation_1-rmse:3745.11380
[406]   validation_0-rmse:2819.76167    validation_1-rmse:3745.04665
[407]   validation_0-rmse:2819.19292    validation_1-rmse:3744.91443
[408]   validation_0-rmse:2816.95261    validation_1-rmse:3745.40713
[409]   validation_0-rmse:2815.48183    validation_1-rmse:3745.71765
[410]   validation_0-rmse:2814.00750    validation_1-rmse:3745.82338
[411]   validation_0-rmse:2812.16009    validation_1-rmse:3746.33463
[412]   validation_0-rmse:2811.60781    validation_1-rmse:3746.12998
[413]   validation_0-rmse:2808.27209    validation_1-rmse:3746.33315
[414]   validation_0-rmse:2806.83027    validation_1-rmse:3746.34193
[415]   validation_0-rmse:2806.24000    validation_1-rmse:3746.13283
```

▾ XGBRegressor

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=50,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.01, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=1000, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
```