```
globals().clear
import time
import math
import pandas as pd
import numpy as np
from matplotlib import pyplot
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
%matplotlib inline
from datetime import datetime
pd.options.display.max_rows = 5000
pd.options.display.max_columns = 500
```

```
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestRegressor
```

```
# Load dataset
#path = '/content/drive/MyDrive/CapstoneProject/code/02. Machine Learning/data/'
df = pd.read_excel('merged_onehot_test.xlsx')
df.shape
```

```
     (13564, 66)
```

```
t1=df
t1.index=t1['Date']
t1.drop(columns=t1.columns[0:2],
        axis=1,
        inplace=True)
t1.head()
```

	TSLA_close	TSLA_vol_4_ave	TSLA_vwap_4_ave	TSLA_trans_4_ave	nasx_clos
Date					
2020-06-01 10:30:00	176.600	6531560.00	174.371825	29927.25	952
2020-06-01 10:45:00	176.748	4872685.00	175.236475	22062.00	953

```
t1.shape
```

```
(13564, 64)
```

2020-06-

```
train_df = t1.loc['2020-06-01 10:30:00':'2021-12-31 16:00:00']
test_df = t1.loc['2022-01-01 09:30:00':'2022-05-27 16:00:00']
```

```
01        175.400        2607445.00        176.151450        12176.75        95
```

```
import time


start = time.time()
predictions = list()
a=1
count_time=list()

for i in test_df['week_label'].unique():

    st = time.time()
    scale_X = MinMaxScaler()

    test_subset = test_df[test_df['week_label']==i]
    print(train_df.index[0])
    print(train_df.index[-1])
    print(test_subset.index[0])
    print(test_subset.index[-1])
    train_stand = train_df.copy()
    test_stand = test_subset.copy()

    X_train, y_train = train_stand.iloc[:,2:65], train_stand.iloc[:,0]
    X_train = scale_X.fit_transform(X_train)
    rf = RandomForestRegressor(n_estimators = 100, max_depth = 100)
    rf.fit(X_train,y_train)
    X_test, y_test = test_stand.iloc[:,2:65], test_stand.iloc[:,0]
    X_test = scale_X.transform(X_test)
    y_hat=rf.predict(X_test)
    predictions.append(y_hat)
    et = time.time()
    used_time=et-st
    count_time.append(used_time)
    train_df = train_df.append(test_df[test_df['week_label']==i])
    train_df=train_df.drop(train_df[train_df['week_label']==a].index)
    a+=1


    print(train_df.index[0])
    print(train_df.index[-1])
    print('Time taken:'+str(used_time))
    print('---------------------------')

end = time.time()
print("total used time"+str(end-start))
```

```
Time taken:10.073755979537964
---------------------------
2020-09-08 09:30:00
2022-04-08 16:00:00
2022-04-11 09:30:00
2022-04-14 16:00:00
2020-09-14 09:30:00
2022-04-14 16:00:00
Time taken:9.748496055603027
---------------------------
2020-09-14 09:30:00
2022-04-14 16:00:00
2022-04-18 09:30:00
2022-04-22 16:00:00
2020-09-21 09:30:00
2022-04-22 16:00:00
Time taken:9.361459016799927
---------------------------
2020-09-21 09:30:00
2022-04-22 16:00:00
2022-04-25 09:30:00
2022-04-29 16:00:00
2020-09-28 09:30:00
2022-04-29 16:00:00
Time taken:9.398331880569458
---------------------------
2020-09-28 09:30:00
2022-04-29 16:00:00
2022-05-02 09:30:00
2022-05-06 16:00:00
2020-10-05 09:30:00
2022-05-06 16:00:00
Time taken:9.61611270904541
---------------------------
2020-10-05 09:30:00
2022-05-06 16:00:00
2022-05-09 09:30:00
2022-05-13 16:00:00
2020-10-12 09:30:00
2022-05-13 16:00:00
Time taken:9.35092544555664
---------------------------
```

```
2020-10-12 09:30:00

2022-05-13 16:00:00
2022-05-16 09:30:00
2022-05-20 16:00:00
2020-10-19 09:30:00
2022-05-20 16:00:00
Time taken:9.408068180084229
---------------------------
2020-10-19 09:30:00
2022-05-20 16:00:00
2022-05-23 09:30:00
2022-05-27 16:00:00
2020-10-26 09:30:00
2022-05-27 16:00:00
Time taken:10.33316969871521
---------------------------
total used time208.79703855514526
```

```
df_expe = pd.DataFrame(test_df.iloc[:,0])
pred_list= list()
for i in range(len(predictions)):
  pred_list=pred_list+predictions[i].tolist()


df_pred = pd.DataFrame(pred_list,index=test_df.index,columns= ['predict'])

df_Result = pd.concat([df_expe,df_pred],axis=1)

df_Result
```

```
2022-05-26 15:00:00   703.8000   709.123648
2022-05-26 15:15:00   704.0733   709.407120
2022-05-26 15:30:00   708.6400   706.171643
2022-05-26 15:45:00   707.5500   705.924214
2022-05-26 16:00:00   708.2200   705.595794
2022-05-27 09:30:00   735.1000   702.728110
2022-05-27 09:45:00   741.9400   709.437814
2022-05-27 10:00:00   740.2100   723.536763
2022-05-27 10:15:00   747.6900   720.922105
2022-05-27 10:30:00   750.4500   737.922416
2022-05-27 10:45:00   747.0500   743.238414
2022-05-27 11:00:00   751.5034   745.354393
2022-05-27 11:15:00   751.1136   750.779016
2022-05-27 11:30:00   751.4000   749.893828
2022-05-27 11:45:00   753.1350   748.903928
2022-05-27 12:00:00   753.3241   751.485871
2022-05-27 12:15:00   749.5700   754.491417
2022-05-27 12:30:00   750.3192   754.969176
2022-05-27 12:45:00   750.4500   751.604822
2022-05-27 13:00:00   753.1285   751.345382
2022-05-27 13:15:00   752.0998   751.370943
2022-05-27 13:30:00   750.2650   751.848696
2022-05-27 13:45:00   751.5700   751.901624
2022-05-27 14:00:00   751.8050   751.352857
2022-05-27 14:15:00   754.2700   751.491555
2022-05-27 14:30:00   755.1675   751.293295
2022-05-27 14:45:00   755.5150   754.363616
2022-05-27 15:00:00   756.9600   754.611151
2022-05-27 15:15:00   758.7000   757.740453
2022-05-27 15:30:00   757.6500   758.110167
2022-05-27 15:45:00   759.6600   758.770454
2022-05-27 16:00:00   759.5000   757.395853
```