پیش بینی بازار سهام با استفاده LSTM واعمال اندیکاتورهای میانگین متحرک ساده و نمایی بر روی داده های بازار ازالگوریتم

```
import math
import pandas as pd
from pandas_datareader import data as pdr
import yfinance as yfin
yfin.pdr_override()
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense,LSTM
import matplotlib.pyplot as plt
#import mplfinance as mplf
```

جلسه اول : روش های مختلف برداشت دیتاست از دیتا ریدر و یاهو به صورت آنلاین و پیش پردازش دیتا

```
# Dataset Generating


#df=pd.read_csv('TSLA.csv')
#print(df)
#data= pd.read_csv('stock_data.csv')
data = pdr.get_data_yahoo('SPY', start='2012-01-01', end='2022-12-17')
print(data)
```

```
[**********100%**********]  1 of 1 completed
                  Open        High         Low       Close   Adj Close  \
Date
2012-01-03  127.760002  128.380005  127.430000  127.500000  103.202309
2012-01-04  127.199997  127.809998  126.709999  127.699997  103.364182
2012-01-05  127.010002  128.229996  126.430000  128.039993  103.639366
2012-01-06  128.199997  128.220001  127.290001  127.709999  103.372284
2012-01-09  128.000000  128.179993  127.410004  128.020004  103.623253
...                ...         ...         ...         ...         ...
2022-12-12  394.109985  398.950012  393.410004  398.950012  395.616516
2022-12-13  410.220001  410.489990  399.070007  401.970001  398.611298
2022-12-14  401.609985  405.500000  396.309998  399.399994  396.062744
2022-12-15  394.299988  395.250000  387.890015  389.630005  386.374390
2022-12-16  385.179993  386.579987  381.040009  383.269989  381.812805

                Volume
Date
2012-01-03  193697900
2012-01-04  127186500
2012-01-05  173895000
2012-01-06  148050000
2012-01-09   99530200
...               ...
2012-12-12   75405800
2012-12-13  123782500
2012-12-14  108111300
2012-12-15  117705900
2012-12-16  119858000

[2759 rows x 6 columns]
```

```
df=data
df.shape
```

```
(2759, 6)
```

```
plt.figure(figsize=(16,8))
plt.title('Close Price History')
plt.plot(df['Close'])
plt.xlabel('Date , fontsize=18')
plt.ylabel('Close Price USD ($)', fontsize=18)
plt.show()
```


```
# Show Candlestick Chart
#mplf.plot(df, type='candle', mav=(18, 30, 60), volume=True)
```

```
# creat the 'Close Cloumn'
data=df.filter(['Close'])
print(data)
# convert to numpy
dataset=data.values
print(dataset.shape)
# the number of rows to be trained
training_data_len= math.ceil(len(dataset)*.8)
training_data_len
```

```
                Close
Date
2012-01-03  127.500000
2012-01-04  127.699997
2012-01-05  128.039993
2012-01-06  127.709999
```

```
2012-01-09  128.020004
...                ...
2022-12-12  398.950012
2022-12-13  401.970001
2022-12-14  399.399994
2022-12-15  389.630005
2022-12-16  383.269989

[2759 rows x 1 columns]
(2759, 1)
2208
```

```
# scale the data
scalar=MinMaxScaler(feature_range=(0,1))
scaled_data=scalar.fit_transform(dataset)

print(scaled_data.shape)
```

```
(2759, 1)
```

```
import cv2
from google.colab.patches import cv2_imshow


#img_1=cv2.imread('Project_1.PNG')
#cv2_imshow(img_1)
```

جلسه دوم: آماده سازی داده ها برای فرآیند آموزش و LSTM برای پیش بینی بازار سهام

* ایجاد مدل الگوریتم

ساختار الگوریتم LSTM

```
#img=cv2.imread('LSTM.PNG')
#cv2_imshow(img)


# creat scaled data for training

train_data=scaled_data[0:training_data_len,:]

# split data to x_train , x_test
x_train=[]
y_train=[]

for i in range(60,len(train_data)):
  x_train.append(train_data[i-60:i,0])
  y_train.append(train_data[i,0])




#convert  the x_train , y_train to numpy arrays

x_train,y_train=np.array(x_train),np.array(y_train)


#Reshape the data

x_train=np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))
```

```
x_train.shape
```

```
(2148, 60, 1)
```

```
#Build the LSTM model

model=Sequential()
model.add(LSTM(50, return_sequences=True , input_shape=(x_train.shape[1],1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
```

```
# Compile the Model
model.compile(optimizer='adam',loss='mean_squared_error')
```

```
# Train the Model
model.fit(x_train,y_train,batch_size=1,epochs=1)
```

```
2148/2148 [==============================] - 88s 39ms/step - loss: 7.5903e-04
<keras.callbacks.History at 0x7f9a9e9120e0>
```

```
#img_2=cv2.imread('Project_2.PNG')
#cv2_imshow(img_2)
```

* جلسه سوم: تست واعتبار سنجی داده ها و پیش بینی بازار سهام*

```
# Creating the testing dataset

test_data=scaled_data[training_data_len-60:,:]
x_test=[]
y_test=dataset[training_data_len:,:]
print(len(test_data))
for i in range(60,len(test_data)):
  x_test.append(test_data[i-60:i,0])
```

```
611
```

```
#convert x_test to numpy arrays
x_test=np.array(x_test)
```

```
# Reshape x_test

x_test=np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))
```

```
# Prediction
predictions=model.predict(x_test)
predictions=scalar.inverse_transform(predictions)
```

```
18/18 [==============================] - 1s 19ms/step
```

```
# The root mean squared error
rmse=np.sqrt(np.mean(predictions-y_test)**2)
rmse
```

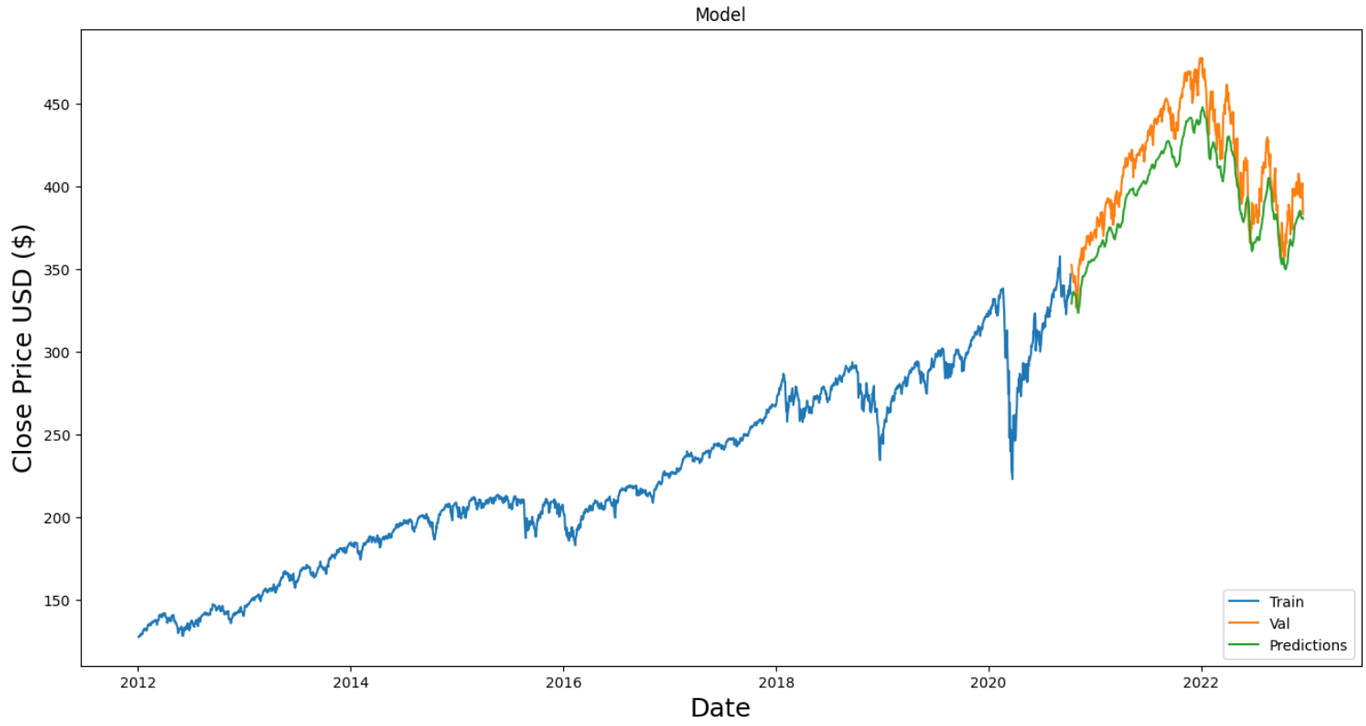
```
19.187286210795712
```

```
#Plot the data

train=data[:training_data_len]
valid=data[training_data_len:]
valid['Predictions']=predictions
plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close Price USD ($)' , fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
plt.show()
```

```
<ipython-input-23-c2c54d2b4ffa>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/ind
  valid['Predictions']=predictions
```




```
# Show the valid and predicted prices
valid
```

Date	Close	Predictions
2020-10-12	352.429993	329.096436
2020-10-13	350.130005	331.380524
2020-10-14	347.929993	333.396759
2020-10-15	347.500000	334.808411
2020-10-16	347.290009	335.711884
...	...	...
2022-12-12	398.950012	380.912354
2022-12-13	401.970001	380.505341
2022-12-14	399.399994	380.821136
2022-12-15	389.630005	381.153564

```
tesla = pdr.get_data_yahoo('SPY', start='2012-01-01', end='2022-12-17')

# Create a new data frame
new_df=tesla.filter(['Close'])

# Get the last 60 day Closing price values and convert the dataframe to an array

last_60_days=new_df[-60:].values

# Scale the data to be values between 0 and 1

last_60_days_scaled=scalar.transform(last_60_days)

# Create an empty list

X_test=[]

X_test.append(last_60_days_scaled)

# Convert the numpy array

X_test=np.array(X_test)

# Reshape the data
X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))

pred_price=model.predict(X_test)

# Get the predicted scalar price
pred_price=scalar.inverse_transform(pred_price)
print(pred_price)
```

```
[*********************100%***********************]  1 of 1 completed
1/1 [==============================] - 0s 28ms/step
[[378.5027]]
```

```
tesla_2= pdr.get_data_yahoo('SPY', start='2022-12-08', end='2022-12-18')
print(tesla_2['Close'])
```

```
[*********************100%***********************]  1 of 1 completed
Date
2022-12-08    396.239990
2022-12-09    393.279999
2022-12-12    398.950012
2022-12-13    401.970001
```

```
2022-12-14    399.399994
2022-12-15    389.630005
2022-12-16    383.269989
Name: Close, dtype: float64
```

```
#img_3=cv2.imread('Project_3.PNG')
#cv2_imshow(img_3)
```

جلسه چهارم: اعمال اندیکاتور میانگین متحرک و میانگین نمایی روی داده های بازار سهام* *

فرمول اندیکاتور میانگین متحرک ساده

```
#img_1=cv2.imread('SMA.PNG')
#cv2_imshow(img_1)
```

فرمول اندیکاتور میانگین متحرک نمایی

```
#img_2=cv2.imread('EMA.PNG')
#cv2_imshow(img_2)
```

```
# Create a function for calculation Moving Average(MA)
def MA(data,period=30,column='Close'):
  return data[column].rolling(window=period).mean()
```

```
# Create a function for calculation Exponential Moving Average(EMA)
def EMA(data,period=30,column='Close'):
  return data[column].ewm(span=period).mean()
```

```
# Create two new columns for storing the 20 day and 50 day MA
df['MA20']=MA(df,20)
df['MA50']=MA(df,50)
```

```
# Create two new columns for storing the 20 day and 50 day MA
df['EMA20']=EMA(df,20)
df['EMA50']=EMA(df,50)
```

```
# Get the buy and sell signals

df['Signal']=np.where(df['MA20']>df['MA50'],1,0)
df['Position']=df['Signal'].diff()

df['Buy']=np.where(df['Position']==1, df['Close'] , np.NAN)
df['Sell']=np.where(df['Position']==-1, df['Close'], np.NAN)
```

```
# Get the buy and sell signals

df['Signal']=np.where(df['EMA20']>df['EMA50'],1,0)
df['Position']=df['Signal'].diff()
```

```
df['Buy']=np.where(df['Position']==1, df['Close'] , np.NAN)
df['Sell']=np.where(df['Position']==-1, df['Close'], np.NAN)



#Show the Close price with the MA ,WMA,EMA and Buy and Sell signals
plt.figure(figsize=(16,8))
plt.title('Close Price History Buy and & Sell Signals' , fontsize=18)
plt.plot(df['Close'], alpha=0.5 , label='Close')
plt.plot(df['MA20'], alpha=0.5 , label='MA20')
plt.plot(df['MA50'], alpha=0.5 , label='MA50')
plt.plot(df['EMA20'], alpha=0.5 , label='EMA20')
plt.plot(df['EMA50'], alpha=0.5 , label='EMA50')


plt.scatter(df.index,df['Buy'], alpha=1, label='Buy Signal', marker='^',color='green')

plt.scatter(df.index,df['Sell'], alpha=1, label='Sell Signal', marker='^',color='red')
plt.xlabel('Date',fontsize=18)
plt.ylabel('Close Price',fontsize=18)
plt.show()
```


```
#img_4=cv2.imread('Project_4.PNG')
#cv2_imshow(img_4)
```

Colab paid products - Cancel contracts here

0s completed at 12:02 PM