Name: Jehan Patel

College: Vellore Institute of Technology, Bhopal

Year: Year 2 Semester 4

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pandas_datareader as web
import datetime as dt

from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM

#Load Data

############################
company = 'GOOG'
############################
```

```python

start = dt.datetime(2012,1,1)
end = dt.datetime(2021,10,7)

data = web.DataReader(company, 'yahoo', start,end)

scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(data['Close'].values.reshape(-1,1))

prediction_days = 60

x_train = []
y_train = []
```

```
```

```
for x in range(prediction_days, len(scaled_data)):
    x_train.append(scaled_data[x-prediction_days:x,0])
    y_train.append(scaled_data[x,0])

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


model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape = (x_train.shape[1],1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

```

model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train,y_train,epochs=25, batch_size=32)

#Model Accuracy

test_start = dt.datetime(2020,1,1)
test_end = dt.datetime.now()

test_data = web.DataReader(company, 'yahoo', test_start, test_end)
actual_prices = test_data['Close'].values

total_dataset = pd.concat((data['Close'], test_data['Close']), axis=0)

```

```
model_inputs = total_dataset[len(total_dataset)-len(test_data) - prediction_days:].values;
model_inputs = model_inputs.reshape(-1,1)
model_inputs = scaler.transform(model_inputs)


x_test = []

for x in range(prediction_days, len(model_inputs)):
    x_test.append(model_inputs[x-prediction_days:x,0])

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

predicted_prices = model.predict(x_test)
predicted_prices = scaler.inverse_transform(predicted_prices)
```

```
predicted_prices = scaler.inverse_transform(predicted_prices)

plt.plot(actual_prices,color="black", label=f"Actual {company} Price")
plt.plot(predicted_prices,color="green", label=f"Predicted {company} Price")
plt.title(f"{company} Share Price")
plt.xlabel('Time')
plt.ylabel(f'{company} Share Price')
plt.legend()
plt.show()

#predict next day

real_data=[model_inputs[len(model_inputs) + 1 - prediction_days:len(model_inputs+1),0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data,(real_data.shape[0], real_data.shape[1],1))

prediction = model.predict(real_data)
prediction = scaler.inverse_transform(prediction)
print(f"Prediction: {prediction}")
```

Github Link:

JehanPatel/
stocksmachinelearning

A Machine Learning Model which predicts moving averages using data available on Yahoo.

1 Contributor 0 Issues 1 Star 0 Forks