Experiment No. 4: Google Stock Price prediction using RNN

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler


data = pd.read_csv(‘GOOG. csv', date_parser =True)


data_training = data[data[‘Date']<’2019-01-01'].copy()
data_test = data[data['Date']>='2019-01-01’].copy()


data_training = data_training.drop([' Date', 'Adj Close'], axis = 1)


scaler = MinMaxScaler()
data_training = scaler.fit_transform(data _training)
data_training


# create RNN with 60 timesteps, ie. look 60 previous time steps
X_ train =[]
y_train =[]


for i in range (60, data_training.shape [0]):
X_train.append(data_training[i-60:1])
y_train.append(data training[i, 0])


X_train, y_train = np.array(X_train), np.array(y _train)


X_train.shape
```

OUT:
(3557, 60, 5)

#Building LSTM

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout

regressior = Sequential()

regressioradd (LSTM (units = 60, ‘activation = 'relu', return_sequences = 'True’,
input_shape = (X_train.shape[1], 5)
regressior.add (Dropout(0.2))

regressior.add (LSTM (units = 60, activation ='relu', return_sequences = True))
regressioradd (Dropout(0.2))

regresslor.add(LSTM (units = 80, activation='relu', return_sequences=True))
regressior.add(Dropout(0.2))

regressior.add(LSTM(units =120, activation = 'relu))
regressior.add(Dropout(0.2))

regressior.add(Dense (units = 1))

regressior.compile (optimizer='adam’, loss = 'mean squared_error’)

regressior.fit(X_train, y_train, epochs=50, batch_size=32)
```

OUT:

```
Epoch 1/50
3557/3557 [=====]-16s 5ms/sample - loss: 0.0137
Epoch 2/50
3557/3557 [====]-12s 3ms/sample - loss: 0.0022
Epoch 3/50
```

3557/3557 [======]12s 3ms/sample - loss: 0.0018
Epoch 4/50
3557/3557 [========]- 12s 3ms/sample - loss: 0.0016
Epoch 5/50
3557/3557 [=====]- 12s 3ms/sample - loss: 0.0016

Epoch 45/50
3557/3557 [=====]-13s 4ms/sample -loss: 6.5112e-04
Epoch 46/50
3557/3557 [=====] -13s 4ms/sample - loss: 6.0908e-04
Epoch 47/50
3557/3557 [===]- 15s 4ms/sample - loss: 6.663 2 e-04
Epoch 48/50
3557/3557 [=======]-15s 4ms/sample - loss: 6.9701e-04
Epoch 49/50
3557/3557 [=====]-16s 4ms/sample - loss: 6.2277 e-04

Epoch 50/50
3557/3557 [=======]-16s 4ms/sample - loss: 6.457 1e-04
<tensorflow.python.keras.callbacks. History at 0x230c796F940>

#Testing

past_60_days = data training.tail(60)
df= past_60_days.append (data_tes, tignore_index = True)

df= df_drop(['Date',’Adj Close'], axis = 1)

inputs=scaler.transform (df)
X_test = []
y_test = []
for i in range (60,inputs.shape[0]):
X_test.append (inputs[i-60:i])
y test.append (inputs[i, 0])
X_test, y_test = np.array (X_test), np.array(y_test)
y_pred = regressior.predict(X_test)

scale = 1/8.18605127e-04

```
y_pred = y_pred*scale
y_test = y_test*scale

# Visualising the results


plt.figure(figsize=(14,5))

pltplot(y_test, color = 'red', label = 'Real Google Stock Price')

plt.plot(y pred, color = ‘blue', label = 'Predicted Google Stock Price')

plt.title('Google Stock Price Prediction')

plt.xlabel("Time')

plt.ylabel ('Google Stock Price')

plt.legend ()

plt.show0
```