

+ Code + Text

 Google_Stock_Price_Train.csv

```
✓ [10] regressor = Sequential()

regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
```

✓ 1s completed at 21:16


Disk 83.32 GB available



+ Code + Text



▶  drive

▶  sample_data

 Google_Stock_Price_Test.csv

Google_Stock_Price_Train.csv

```
[12] regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)
```

```
Epoch 1/100
38/38 [=====] - 5s 125ms/step - loss: 0.0046
Epoch 2/100
38/38 [=====] - 5s 123ms/step - loss: 0.0045
Epoch 3/100
38/38 [=====] - 6s 162ms/step - loss: 0.0045
Epoch 4/100
38/38 [=====] - 5s 132ms/step - loss: 0.0045
Epoch 5/100
38/38 [=====] - 5s 143ms/step - loss: 0.0041
Epoch 6/100
38/38 [=====] - 6s 145ms/step - loss: 0.0040
Epoch 7/100
38/38 [=====] - 5s 121ms/step - loss: 0.0040
Epoch 8/100
38/38 [=====] - 6s 171ms/step - loss: 0.0042
Epoch 9/100
38/38 [=====] - 5s 121ms/step - loss: 0.0035
```

✓ 1s completed at 21:16

content - Googl...

colab.research.google.com

stock price prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

..

drive

sample_data

Google_Stock_Price_Test.csv

Google_Stock_Price_Train.csv

8m

[12]

38/38 [=====] - 6s 158ms/step - loss: 0.0013
Epoch 100/100
38/38 [=====] - 5s 122ms/step - loss: 0.0013
<keras.src.callbacks.History at 0x7b2482bc1060>

0s

[15]

dataset_test = pd.read_csv('Google_Stock_Price_Test.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values

3s

Getting the predicted stock price of 2017

dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)

inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values

inputs = inputs.reshape(-1,1)

inputs = sc.transform(inputs)

X_test = []

for i in range(60, 80):

X_test.append(inputs[i-60:i, 0])

X_test = np.array(X_test)

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

predicted_stock_price = regressor.predict(X_test)

predicted_stock_price = sc.inverse_transform(predicted_stock_price)

1/1 [=====] - 3s 3s/step

1s

Visualising the results

plt.plot(real_stock_price, color = 'red', label = 'Real Google Stock Price')

plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Google Stock Price')

plt.title('Google Stock Price Prediction')

plt.xlabel('Time')

plt.ylabel('Google Stock Price')

plt.legend()

plt.show()

Google Stock Price Prediction

840

830

Real Google Stock Price

Predicted Google Stock Price

1s

completed at 21:16

```
1/1 [=====] - 3s 3s/step
```

```
✓ 1s # Visualising the results
plt.plot(real_stock_price, color = 'red', label = 'Real Google Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
plt.show()
```

