In [159…

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

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

In [160…

```
dataset_train = pd.read_csv('D:\MeezanBankTrain.csv')
training_set = dataset_train.iloc[:, 4:5].values
```

In [161…

```
stock_price_train = dataset_train.iloc[:-120, 4:5].values
```

In [162…

```
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set).reshape(-1,1)
```

In [163…

```
X_train = []
y_train = []
for i in range(120, len(training_set_scaled)):
    X_train.append(training_set_scaled[i-120:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)
```

In [164…

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

In [165…

```
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))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1))
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32, verbose='auto')
```

```
Epoch 1/100
31/31 [==============================] - 15s 207ms/step - loss: 0.0175
Epoch 2/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0053
Epoch 3/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0046
Epoch 4/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0036
Epoch 5/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0040
Epoch 6/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0033
Epoch 7/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0032
Epoch 8/100
```

```
31/31 [==============================] - 6s 206ms/step - loss: 0.0029
Epoch 9/100
31/31 [==============================] - 7s 222ms/step - loss: 0.0029
Epoch 10/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0028
Epoch 11/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0030
Epoch 12/100
31/31 [==============================] - 7s 224ms/step - loss: 0.0028
Epoch 13/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0027
Epoch 14/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0026
Epoch 15/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0026
Epoch 16/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0025
Epoch 17/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0026
Epoch 18/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0022
Epoch 19/100
31/31 [==============================] - 6s 210ms/step - loss: 0.0023
Epoch 20/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0022
Epoch 21/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0021
Epoch 22/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0025
Epoch 23/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0019
Epoch 24/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0022
Epoch 25/100
31/31 [==============================] - 6s 204ms/step - loss: 0.0023
Epoch 26/100
31/31 [==============================] - 6s 204ms/step - loss: 0.0021
Epoch 27/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0019
Epoch 28/100
31/31 [==============================] - 6s 204ms/step - loss: 0.0018
Epoch 29/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0017
Epoch 30/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0020
Epoch 31/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0017
Epoch 32/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0017
Epoch 33/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0018
Epoch 34/100
31/31 [==============================] - 6s 204ms/step - loss: 0.0019
Epoch 35/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0016
Epoch 36/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0015
Epoch 37/100
31/31 [==============================] - 7s 216ms/step - loss: 0.0016
Epoch 38/100
31/31 [==============================] - 7s 219ms/step - loss: 0.0015
Epoch 39/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0016
Epoch 40/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0015
Epoch 41/100
```

```
31/31 [==============================] - 6s 208ms/step - loss: 0.0015
Epoch 42/100
31/31 [==============================] - 7s 211ms/step - loss: 0.0015
Epoch 43/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0016 0s - loss: 0.0
Epoch 44/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0013
Epoch 45/100
31/31 [==============================] - 7s 229ms/step - loss: 0.0014
Epoch 46/100
31/31 [==============================] - 7s 219ms/step - loss: 0.0015
Epoch 47/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0012
Epoch 48/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0014
Epoch 49/100
31/31 [==============================] - 7s 211ms/step - loss: 0.0014
Epoch 50/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0012
Epoch 51/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0013
Epoch 52/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0012
Epoch 53/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0013
Epoch 54/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0013
Epoch 55/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0013
Epoch 56/100
31/31 [==============================] - 7s 219ms/step - loss: 0.0012
Epoch 57/100
31/31 [==============================] - 8s 249ms/step - loss: 0.0012
Epoch 58/100
31/31 [==============================] - 8s 246ms/step - loss: 0.0013
Epoch 59/100
31/31 [==============================] - 7s 222ms/step - loss: 0.0012
Epoch 60/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0012
Epoch 61/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0011
Epoch 62/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0012
Epoch 63/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0011
Epoch 64/100
31/31 [==============================] - 7s 213ms/step - loss: 9.9180e-04
Epoch 65/100
31/31 [==============================] - 7s 210ms/step - loss: 0.0013
Epoch 66/100
31/31 [==============================] - 6s 205ms/step - loss: 0.0012
Epoch 67/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0010
Epoch 68/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0010
Epoch 69/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0012
Epoch 70/100
31/31 [==============================] - 6s 208ms/step - loss: 9.8986e-04
Epoch 71/100
31/31 [==============================] - 6s 205ms/step - loss: 9.4222e-04
Epoch 72/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0011
Epoch 73/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0010
Epoch 74/100
```

```
31/31 [==============================] - 6s 209ms/step - loss: 9.2096e-04
Epoch 75/100
31/31 [==============================] - 7s 212ms/step - loss: 0.0010
Epoch 76/100
31/31 [==============================] - 6s 209ms/step - loss: 8.7655e-04
Epoch 77/100
31/31 [==============================] - 6s 207ms/step - loss: 9.4342e-04
Epoch 78/100
31/31 [==============================] - 7s 210ms/step - loss: 0.0010
Epoch 79/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0011
Epoch 80/100
31/31 [==============================] - 7s 213ms/step - loss: 8.8762e-04
Epoch 81/100
31/31 [==============================] - 7s 215ms/step - loss: 0.0010
Epoch 82/100
31/31 [==============================] - 7s 210ms/step - loss: 8.0324e-04
Epoch 83/100
31/31 [==============================] - 7s 210ms/step - loss: 9.9165e-04
Epoch 84/100
31/31 [==============================] - 7s 211ms/step - loss: 9.8138e-04
Epoch 85/100
31/31 [==============================] - 6s 207ms/step - loss: 9.3371e-04
Epoch 86/100
31/31 [==============================] - 6s 208ms/step - loss: 9.8747e-04
Epoch 87/100
31/31 [==============================] - 6s 207ms/step - loss: 8.4087e-04
Epoch 88/100
31/31 [==============================] - 6s 208ms/step - loss: 8.4971e-04
Epoch 89/100
31/31 [==============================] - 6s 209ms/step - loss: 8.1239e-04
Epoch 90/100
31/31 [==============================] - 7s 212ms/step - loss: 7.9537e-04
Epoch 91/100
31/31 [==============================] - 7s 212ms/step - loss: 8.4832e-04
Epoch 92/100
31/31 [==============================] - 7s 214ms/step - loss: 8.9286e-04
Epoch 93/100
31/31 [==============================] - 7s 212ms/step - loss: 8.2904e-04
Epoch 94/100
31/31 [==============================] - 6s 207ms/step - loss: 9.0221e-04
Epoch 95/100
31/31 [==============================] - 7s 211ms/step - loss: 7.7353e-04
Epoch 96/100
31/31 [==============================] - 7s 210ms/step - loss: 8.1358e-04
Epoch 97/100
31/31 [==============================] - 7s 210ms/step - loss: 8.3357e-04
Epoch 98/100
31/31 [==============================] - 7s 212ms/step - loss: 9.1424e-04
Epoch 99/100
31/31 [==============================] - 7s 212ms/step - loss: 8.0726e-04
Epoch 100/100
31/31 [==============================] - 7s 211ms/step - loss: 7.7565e-04
```

Out[165…

```
<keras.callbacks.History at 0x1b6b3b3a160>
```

In [166…

```
#history = regressor.fit(X_train, y_train, validation_split=0.33, epochs=150, batch_size=
```

In [167…

```
# list all data in history
print(history.history.keys())
```

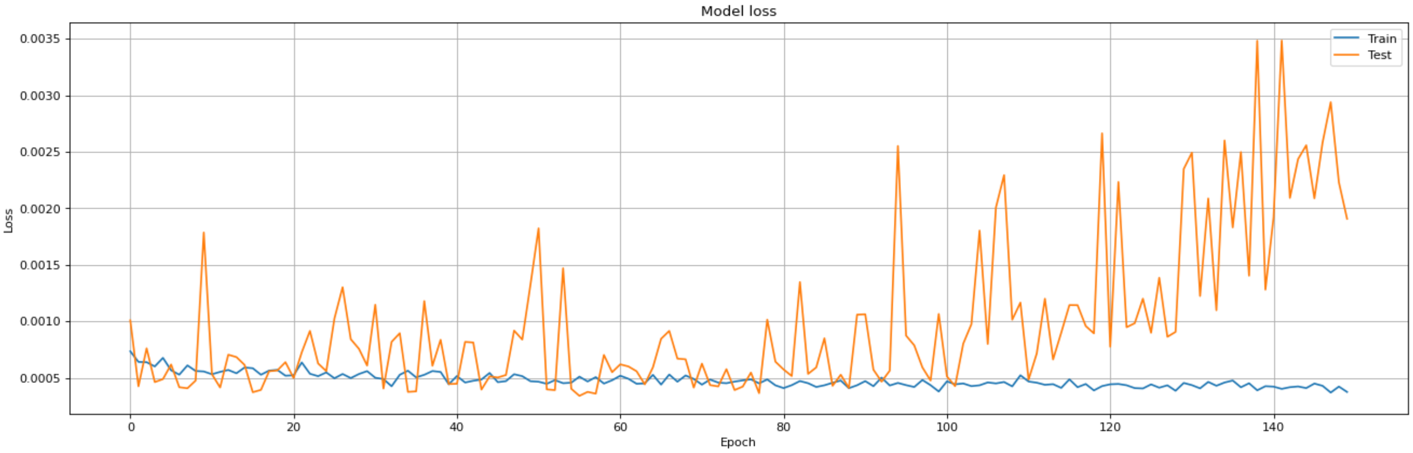
```
dict_keys(['loss', 'val_loss'])
```

In [168…

```
# import matplotlib as plt
# # summarize history for accuracy
# plt.plot(history.history['accuracy'])
# plt.plot(history.history['val_accuracy'])
# plt.title('model accuracy')
# plt.ylabel('accuracy')
# plt.xlabel('epoch')
# plt.legend(['train', 'test'], loc='upper left')
# plt.show()
```

In [169…

```
# summarize history for loss
plt.figure(figsize=(20, 6),dpi=80)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'],loc='upper right')
plt.grid()
plt.show()
```



In [170…

```
dataset_test = pd.read_csv('D:/MeezanBankTest.csv')
```

In [171…

```
dataset_test.replace(' ','',inplace=True)
dataset_test = dataset_test.dropna()
```

In [216…

```
data2 = dataset_test.iloc[:, 4:5].values
total_test = pd.concat((dataset_train['Close'], dataset_test['Close']), axis=0)
```

Out[216…

```
0      67.98
1      67.24
2      69.00
3      65.89
4      66.66
       ...
15    133.71
16    134.32
17    133.15
18    130.19
19    133.89
Name: Close, Length: 1131, dtype: float64
```

In [173…

```
inputs = total_test[len(total_test) - len(dataset_test) - 120:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
```

In [174…

```
X_test = []
for i in range(120, len(inputs)):
    X_test.append(inputs[i-120: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)
```

In [175…

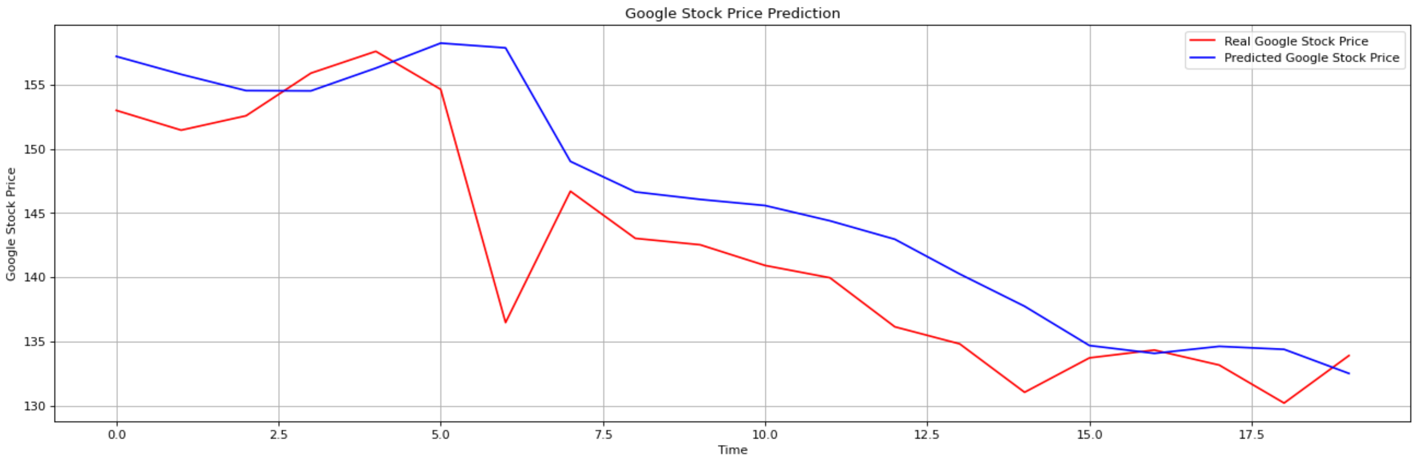
```
predicted_stock_price_train = regressor.predict(X_train)
predicted_stock_price_train = sc.inverse_transform(predicted_stock_price_train)
predicted_stock_price_train.shape
```

Out[175…

```
(991, 1)
```

In [176…

```
import matplotlib.pyplot as plt
plt.figure(figsize=(20, 6),dpi=80)
plt.plot(dataset_test['Close'], 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.grid()
plt.show()
```



In [177…

```
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
testScore =(mean_squared_error(dataset_test['Close'], predicted_stock_price))
print('Test Score: %.2f RMSE' % (testScore))
print('Test Score: %.2f MAE' % mean_absolute_error(dataset_test['Close'], predicted_stock_
print('Test Score: %.2f R2' % r2_score(dataset_test['Close'],predicted_stock_price))
```

```
Test Score: 36.55 RMSE
Test Score: 4.20 MAE
Test Score: 0.54 R2
```

In [178…

```
print(mean_squared_error(stock_price_train, predicted_stock_price_train))
print(r2_score(stock_price_train, predicted_stock_price_train))
```

```
298.18329660692814
-0.9219621300204883
```

In [179…

```
real_data = [inputs[len(inputs) - 120:len(inputs + 1), 0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data , (real_data.shape[0], real_data.shape[1], 1))
```

In [180…

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

```
Prediction : [[132.66466]]
```

In [181…

```
regressor_30 = Sequential()
regressor_30.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1]
regressor_30.add(Dropout(0.2))
regressor_30.add(LSTM(units = 50, return_sequences = True))
regressor_30.add(Dropout(0.2))
regressor_30.add(LSTM(units = 50, return_sequences = True))
regressor_30.add(Dropout(0.2))
regressor_30.add(LSTM(units = 50))
regressor_30.add(Dropout(0.2))
regressor_30.add(Dense(units = 30))
regressor_30.compile(optimizer = 'adam', loss = 'mean_squared_error')
regressor_30.fit(X_train, y_train, epochs = 100, batch_size = 32, verbose='auto')
```

```
Epoch 1/100
31/31 [==============================] - 15s 232ms/step - loss: 0.0449
Epoch 2/100
31/31 [==============================] - 7s 210ms/step - loss: 0.0116
Epoch 3/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0079
Epoch 4/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0070
Epoch 5/100
31/31 [==============================] - 7s 212ms/step - loss: 0.0062
Epoch 6/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0052
Epoch 7/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0048
Epoch 8/100
31/31 [==============================] - 7s 216ms/step - loss: 0.0044
Epoch 9/100
31/31 [==============================] - 7s 212ms/step - loss: 0.0044
Epoch 10/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0039
Epoch 11/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0037
Epoch 12/100
31/31 [==============================] - 7s 209ms/step - loss: 0.0036
Epoch 13/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0035
Epoch 14/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0030
Epoch 15/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0032
Epoch 16/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0030
Epoch 17/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0030
Epoch 18/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0026
Epoch 19/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0026
Epoch 20/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0027
```

```
Epoch 21/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0024
Epoch 22/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0024
Epoch 23/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0024
Epoch 24/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0023
Epoch 25/100
31/31 [==============================] - 7s 220ms/step - loss: 0.0022
Epoch 26/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0021
Epoch 27/100
31/31 [==============================] - 7s 210ms/step - loss: 0.0022
Epoch 28/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0020
Epoch 29/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0020
Epoch 30/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0020
Epoch 31/100
31/31 [==============================] - 7s 218ms/step - loss: 0.0019
Epoch 32/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0018
Epoch 33/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0018
Epoch 34/100
31/31 [==============================] - 6s 209ms/step - loss: 0.0018
Epoch 35/100
31/31 [==============================] - 7s 217ms/step - loss: 0.0016
Epoch 36/100
31/31 [==============================] - 6s 206ms/step - loss: 0.0016
Epoch 37/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0018
Epoch 38/100
31/31 [==============================] - 7s 216ms/step - loss: 0.0017
Epoch 39/100
31/31 [==============================] - 7s 216ms/step - loss: 0.0017
Epoch 40/100
31/31 [==============================] - 7s 223ms/step - loss: 0.0016
Epoch 41/100
31/31 [==============================] - 7s 228ms/step - loss: 0.0016
Epoch 42/100
31/31 [==============================] - 7s 240ms/step - loss: 0.0015
Epoch 43/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0016
Epoch 44/100
31/31 [==============================] - 7s 223ms/step - loss: 0.0016
Epoch 45/100
31/31 [==============================] - 7s 216ms/step - loss: 0.0016
Epoch 46/100
31/31 [==============================] - 7s 210ms/step - loss: 0.0016
Epoch 47/100
31/31 [==============================] - 6s 207ms/step - loss: 0.0013
Epoch 48/100
31/31 [==============================] - 7s 210ms/step - loss: 0.0013
Epoch 49/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0013
Epoch 50/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0012
Epoch 51/100
31/31 [==============================] - 7s 237ms/step - loss: 0.0012
Epoch 52/100
31/31 [==============================] - 8s 246ms/step - loss: 0.0012
Epoch 53/100
31/31 [==============================] - 7s 225ms/step - loss: 0.0013
```

```
Epoch 54/100
31/31 [==============================] - 7s 222ms/step - loss: 0.0013
Epoch 55/100
31/31 [==============================] - 8s 246ms/step - loss: 0.0013
Epoch 56/100
31/31 [==============================] - 7s 241ms/step - loss: 0.0013
Epoch 57/100
31/31 [==============================] - 8s 270ms/step - loss: 0.0013
Epoch 58/100
31/31 [==============================] - 7s 239ms/step - loss: 0.0011
Epoch 59/100
31/31 [==============================] - 7s 236ms/step - loss: 0.0012
Epoch 60/100
31/31 [==============================] - 6s 208ms/step - loss: 0.0011
Epoch 61/100
31/31 [==============================] - 7s 211ms/step - loss: 0.0012
Epoch 62/100
31/31 [==============================] - 7s 214ms/step - loss: 0.0011
Epoch 63/100
31/31 [==============================] - 7s 211ms/step - loss: 0.0011
Epoch 64/100
31/31 [==============================] - 7s 211ms/step - loss: 9.5366e-04
Epoch 65/100
31/31 [==============================] - 7s 222ms/step - loss: 0.0010
Epoch 66/100
31/31 [==============================] - 8s 269ms/step - loss: 9.4429e-04
Epoch 67/100
31/31 [==============================] - 8s 255ms/step - loss: 0.0010
Epoch 68/100
31/31 [==============================] - 7s 229ms/step - loss: 9.5600e-04
Epoch 69/100
31/31 [==============================] - 7s 231ms/step - loss: 0.0011
Epoch 70/100
31/31 [==============================] - 9s 277ms/step - loss: 0.0010
Epoch 71/100
31/31 [==============================] - 9s 280ms/step - loss: 9.3798e-04
Epoch 72/100
31/31 [==============================] - 7s 228ms/step - loss: 0.0010
Epoch 73/100
31/31 [==============================] - 7s 236ms/step - loss: 9.7717e-04
Epoch 74/100
31/31 [==============================] - 8s 248ms/step - loss: 8.8100e-04
Epoch 75/100
31/31 [==============================] - 9s 302ms/step - loss: 9.6999e-04
Epoch 76/100
31/31 [==============================] - 8s 264ms/step - loss: 8.9304e-04
Epoch 77/100
31/31 [==============================] - 9s 291ms/step - loss: 9.8030e-04
Epoch 78/100
31/31 [==============================] - 8s 243ms/step - loss: 9.5120e-04
Epoch 79/100
31/31 [==============================] - 9s 298ms/step - loss: 9.6701e-04
Epoch 80/100
31/31 [==============================] - 7s 230ms/step - loss: 8.5260e-04
Epoch 81/100
31/31 [==============================] - 8s 249ms/step - loss: 9.3023e-04
Epoch 82/100
31/31 [==============================] - 7s 241ms/step - loss: 7.3566e-04
Epoch 83/100
31/31 [==============================] - 8s 253ms/step - loss: 8.6357e-04
Epoch 84/100
31/31 [==============================] - 9s 300ms/step - loss: 7.5422e-04
Epoch 85/100
31/31 [==============================] - 10s 322ms/step - loss: 8.2098e-04
Epoch 86/100
31/31 [==============================] - 8s 274ms/step - loss: 8.4154e-04
```

```
Epoch 87/100
31/31 [==============================] - 8s 261ms/step - loss: 8.5373e-04
Epoch 88/100
31/31 [==============================] - 8s 267ms/step - loss: 9.3525e-04
Epoch 89/100
31/31 [==============================] - 8s 260ms/step - loss: 8.7063e-04
Epoch 90/100
31/31 [==============================] - 8s 266ms/step - loss: 9.0100e-04
Epoch 91/100
31/31 [==============================] - 9s 297ms/step - loss: 7.9490e-04
Epoch 92/100
31/31 [==============================] - 9s 281ms/step - loss: 7.9439e-04
Epoch 93/100
31/31 [==============================] - 8s 260ms/step - loss: 7.7378e-04
Epoch 94/100
31/31 [==============================] - 7s 237ms/step - loss: 7.0395e-04
Epoch 95/100
31/31 [==============================] - 8s 242ms/step - loss: 7.7556e-04
Epoch 96/100
31/31 [==============================] - 7s 227ms/step - loss: 8.3410e-04
Epoch 97/100
31/31 [==============================] - 7s 230ms/step - loss: 7.4512e-04
Epoch 98/100
31/31 [==============================] - 7s 234ms/step - loss: 8.4812e-04
Epoch 99/100
31/31 [==============================] - 7s 228ms/step - loss: 7.7746e-04
Epoch 100/100
31/31 [==============================] - 7s 229ms/step - loss: 7.2956e-04
```

Out[181…

```
<keras.callbacks.History at 0x1b6be3a7640>
```

In [182…

```python
real_data_30 = [inputs[len(inputs) - 120:len(inputs + 1), 0]]
real_data_30 = np.array(real_data_30)
real_data_30 = np.reshape(real_data_30 , (real_data_30.shape[0], real_data_30.shape[1], 1)
```

In [183…

```python
prediction_30 = regressor_30.predict(real_data_30)
prediction_30 = sc.inverse_transform(prediction_30)
print(f"Prediction : {prediction_30.transpose()}")
```

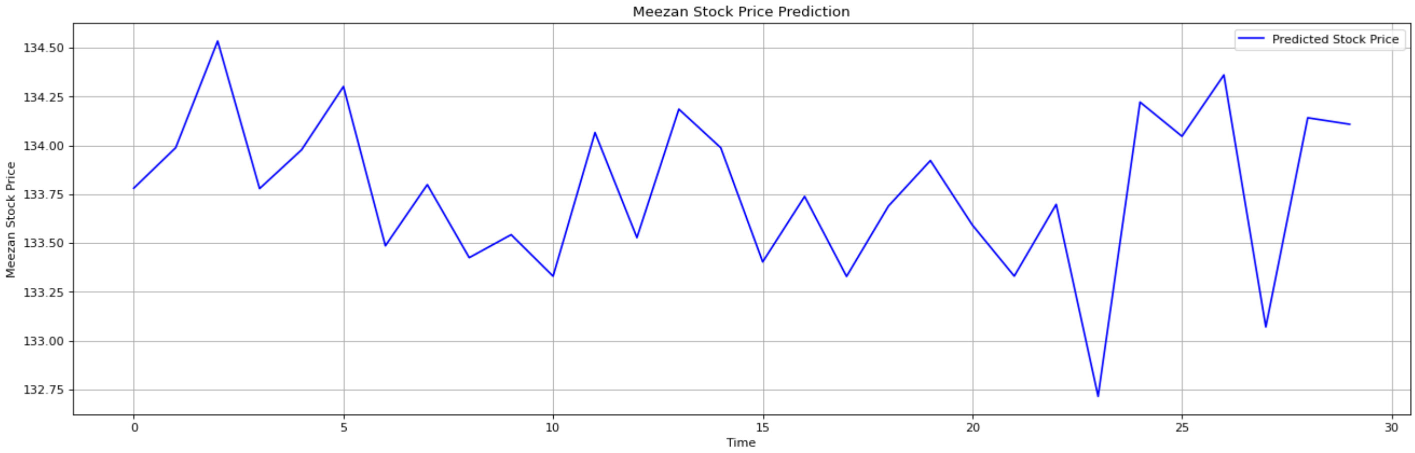
```
Prediction : [[133.78113]
 [133.98868]
 [134.53502]
 [133.7789 ]
 [133.9772 ]
 [134.30203]
 [133.48563]
 [133.799  ]
 [133.4248 ]
 [133.54262]
 [133.32985]
 [134.06601]
 [133.52753]
 [134.1859 ]
 [133.98808]
 [133.40321]
 [133.73857]
 [133.32864]
 [133.68948]
 [133.92258]
 [133.59235]
 [133.33023]
 [133.69797]
 [132.71397]
 [134.22215]
 [134.04697]
 [134.3613 ]
 [133.0693 ]
 [134.14182]
 [134.10864]]
```

In [244…

```
DataFrame = pd.DataFrame(prediction_30)
prediction  = DataFrame.T
Prediction_Total = pd.concat((dataset_test['Close'],prediction), axis=0,sort=True,ignore_i
```

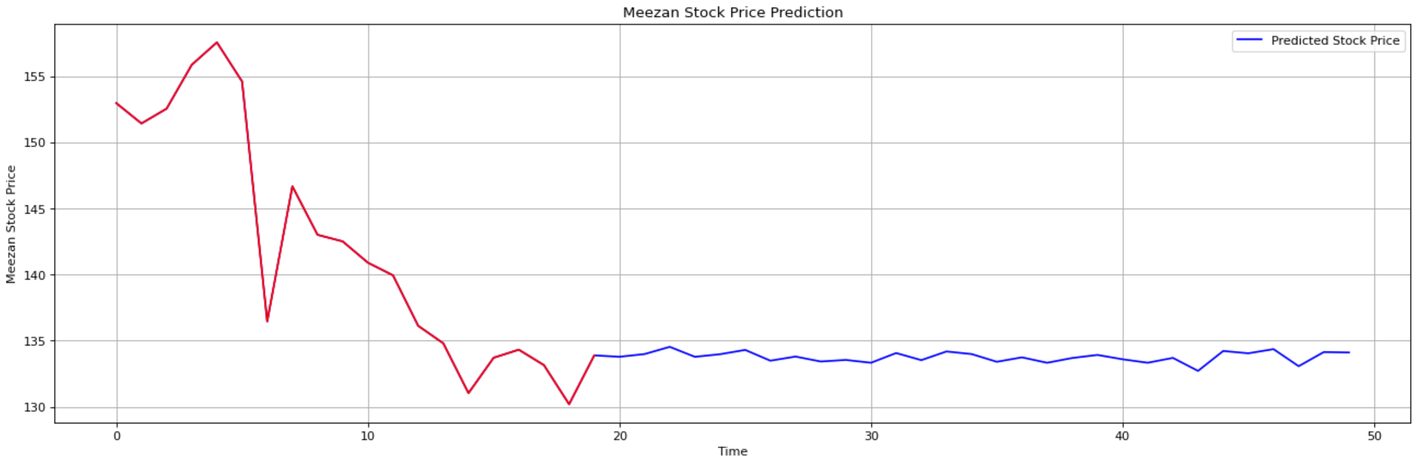
In [251…

```
plt.figure(figsize=(20, 6),dpi=80)
plt.plot(prediction, color = 'blue', label = 'Predicted Stock Price')
plt.title('Meezan Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Meezan Stock Price')
plt.legend()
plt.grid()
plt.show()
```



In [250…

```
plt.figure(figsize=(20, 6),dpi=80)
plt.plot(Prediction_Total, color = 'blue', label = 'Predicted Stock Price')
plt.plot(dataset_test['Close'], color = 'red')
plt.title('Meezan Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Meezan Stock Price')
plt.legend()
plt.grid()
plt.show()
```



In [186…

```
regressor_30.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_20 (LSTM)              (None, 120, 50)           10400

 dropout_20 (Dropout)        (None, 120, 50)           0

 lstm_21 (LSTM)              (None, 120, 50)           20200

 dropout_21 (Dropout)        (None, 120, 50)           0

 lstm_22 (LSTM)              (None, 120, 50)           20200

 dropout_22 (Dropout)        (None, 120, 50)           0

 lstm_23 (LSTM)              (None, 50)                20200

 dropout_23 (Dropout)        (None, 50)                0

 dense_5 (Dense)             (None, 30)                1530

=================================================================
Total params: 72,530
Trainable params: 72,530
Non-trainable params: 0
_________________________________________________________________
```

In [265…

```
import cufflinks as cf
import plotly.graph_objects as go
from plotly.offline import iplot, init_notebook_mode
import matplotlib.pyplot as plt
```

In [266…

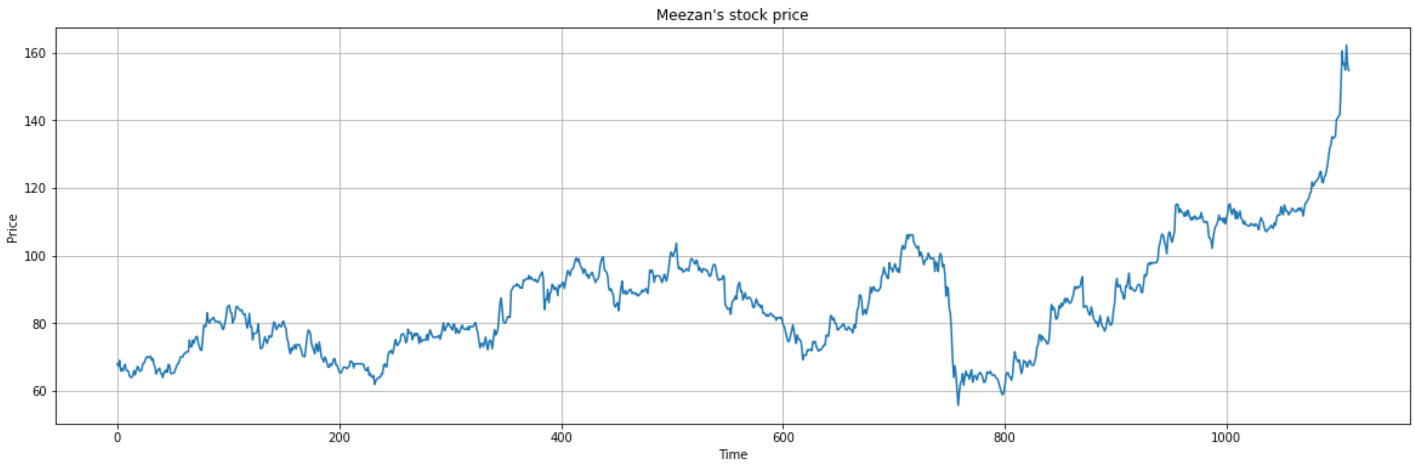
```
cf.go_offline()
init_notebook_mode()
```

In [267…

```
TICKER = "Meezan"
dataset_train["Close"].plot(title=f"{TICKER}'s stock price",figsize=(20, 6),grid=True,xlab
```

Out[267…

```
<AxesSubplot:title={'center':"Meezan's stock price"}, xlabel='Time', ylabel='Price'>
```

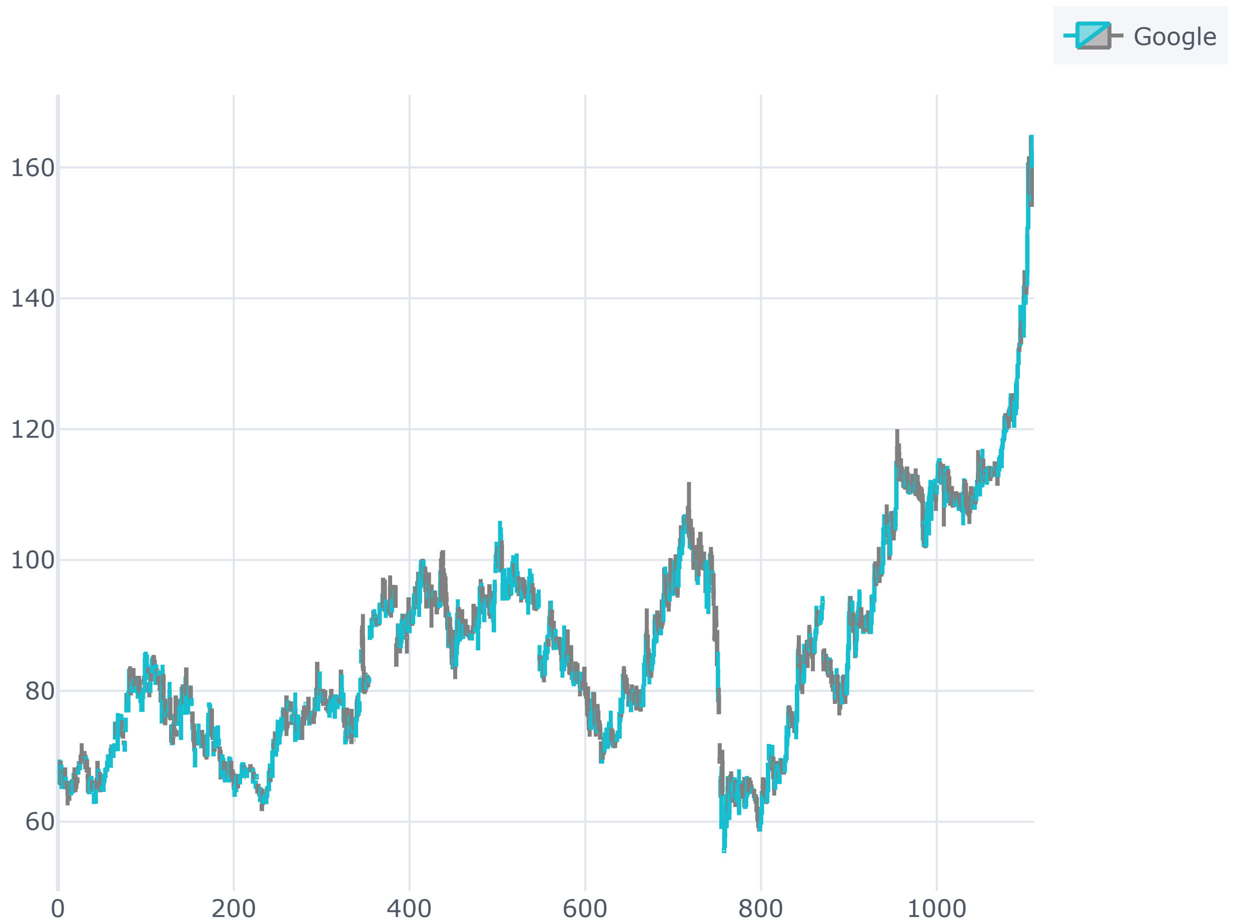


In [268…

```
qf = cf.QuantFig(dataset_train, title="Meezan's stock price in 2021", name='Google')
qf.iplot()
```

Meezan's stock price in 2021

Meezan's stock price in 2021



Export to plot.ly »

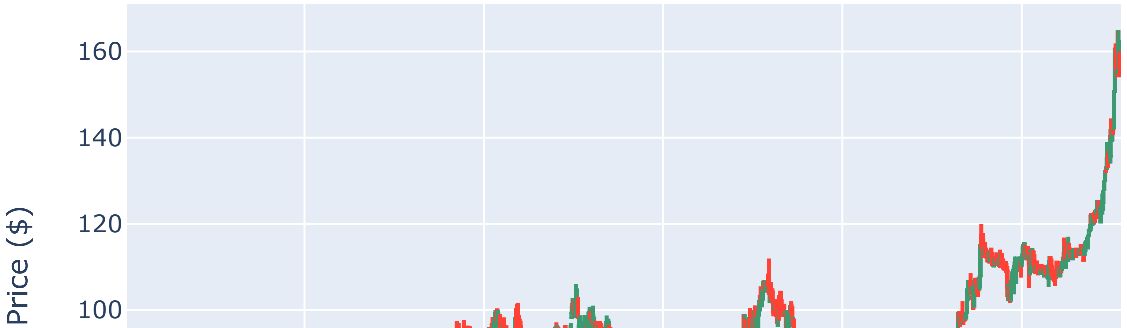
In [269…

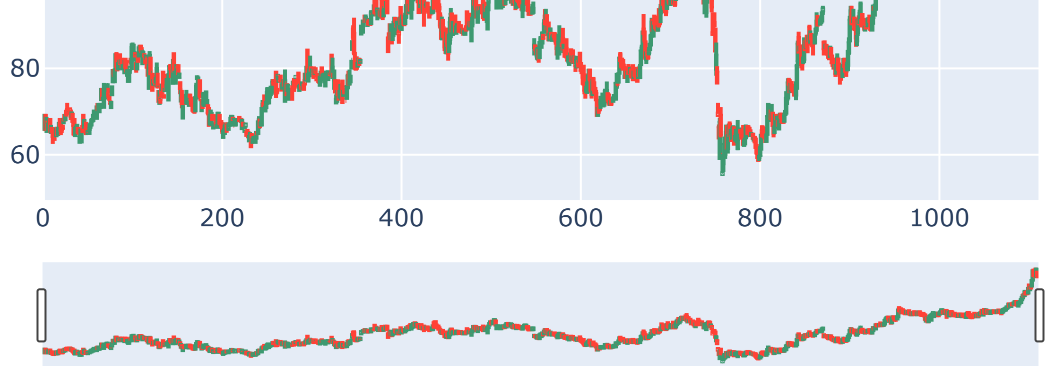
```
fig = go.Figure(data=
    [go.Candlestick(x=dataset_train.index,
                    open=dataset_train["Open"],
                    high=dataset_train["High"],
                    low=dataset_train["Low"],
                    close=dataset_train["Close"])]
)

fig.update_layout(
    title=f"{TICKER}'s adjusted stock price",
    yaxis_title="Price ($)"
)

fig.show()
```

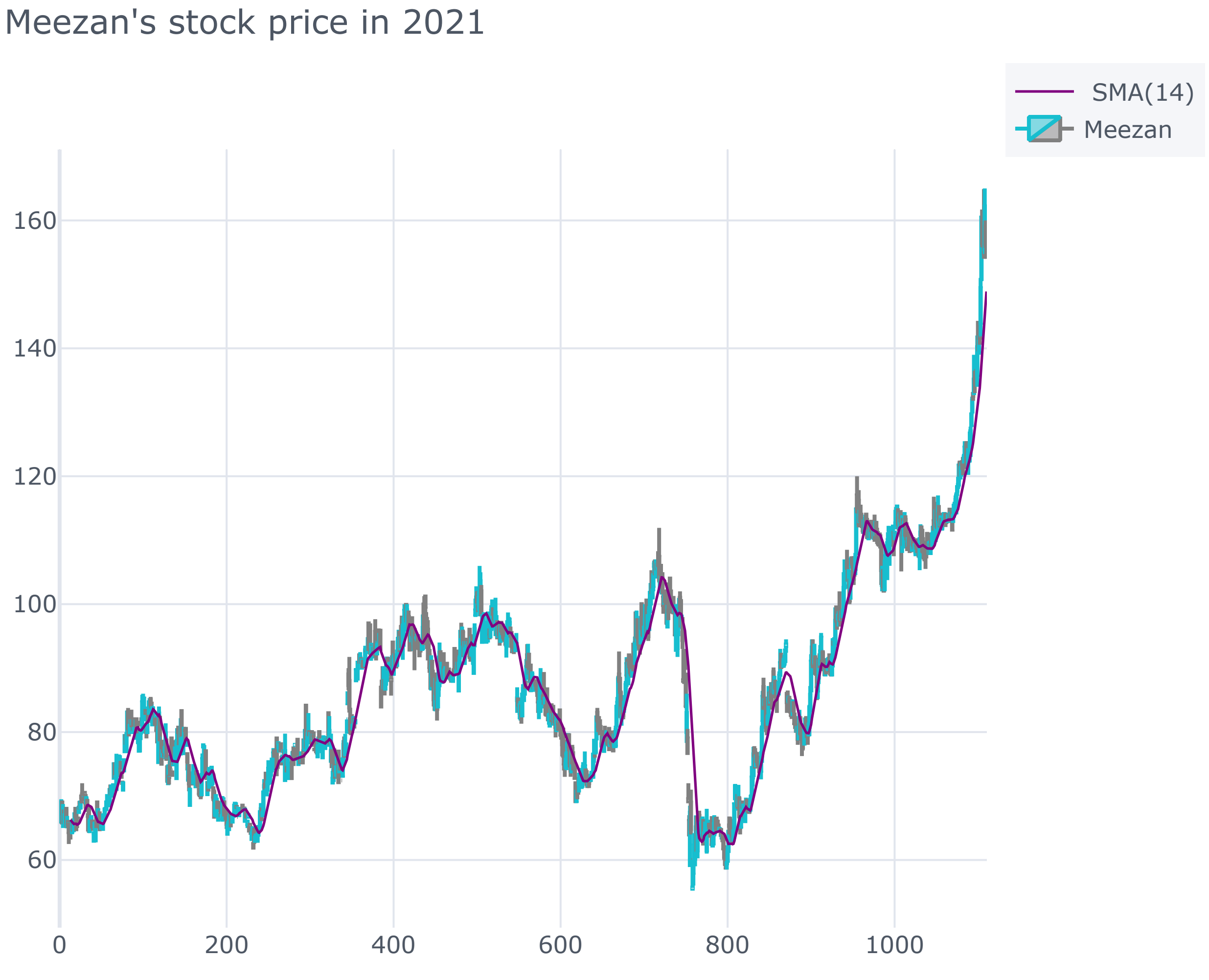
Meezan's adjusted stock price





In [270…

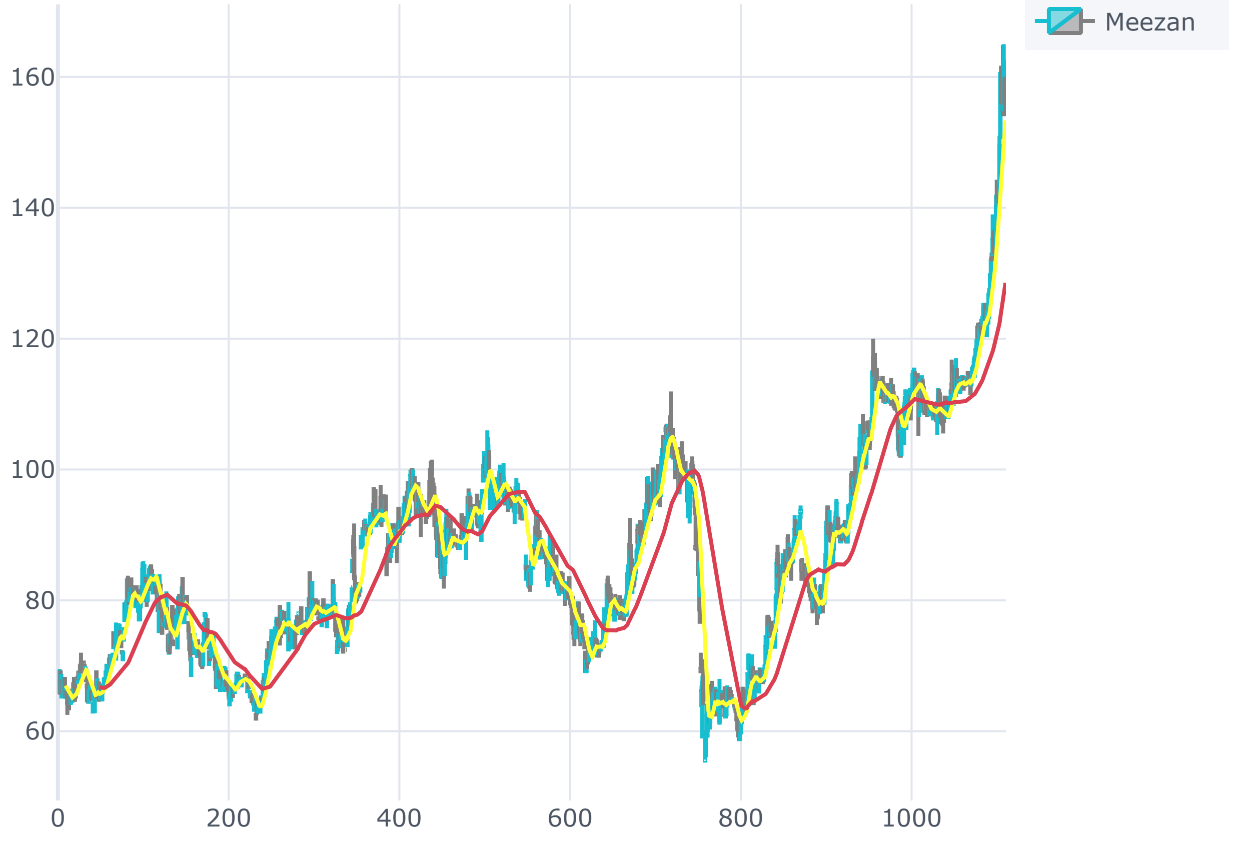
```
qf = cf.QuantFig(dataset_train, title="Meezan's stock price in 2021", name='Meezan')
qf.add_sma(periods=14, column='Close', color='purple')
qf.iplot()
```



Export to plot.ly »

In [271…

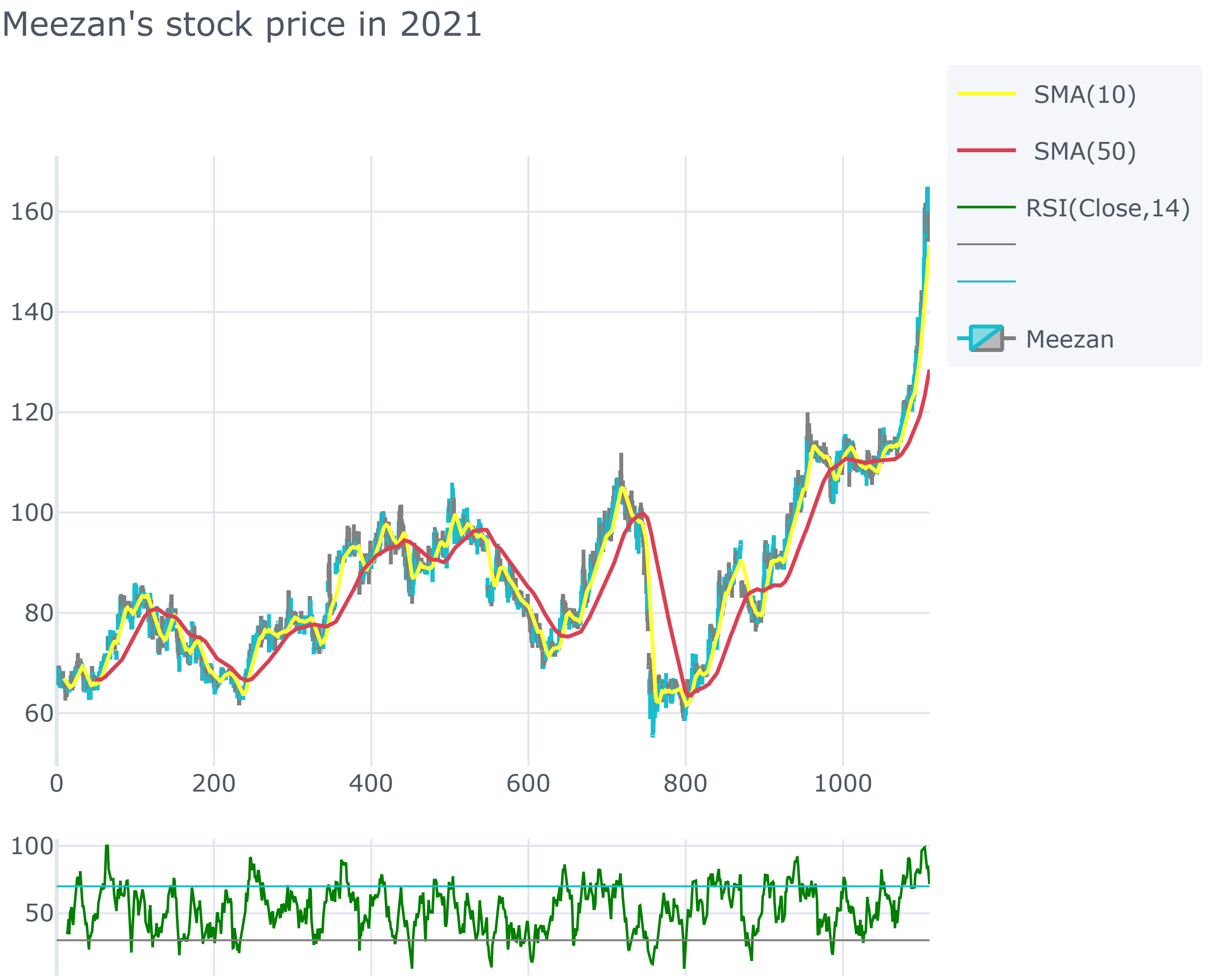
```
qf = cf.QuantFig(dataset_train, title="Meezan's stock price in 2021", name='Meezan')
qf.add_sma([10, 50], width=2, color=['yellow', 'red'])
qf.iplot()
```



Export to plot.ly »

In [272…

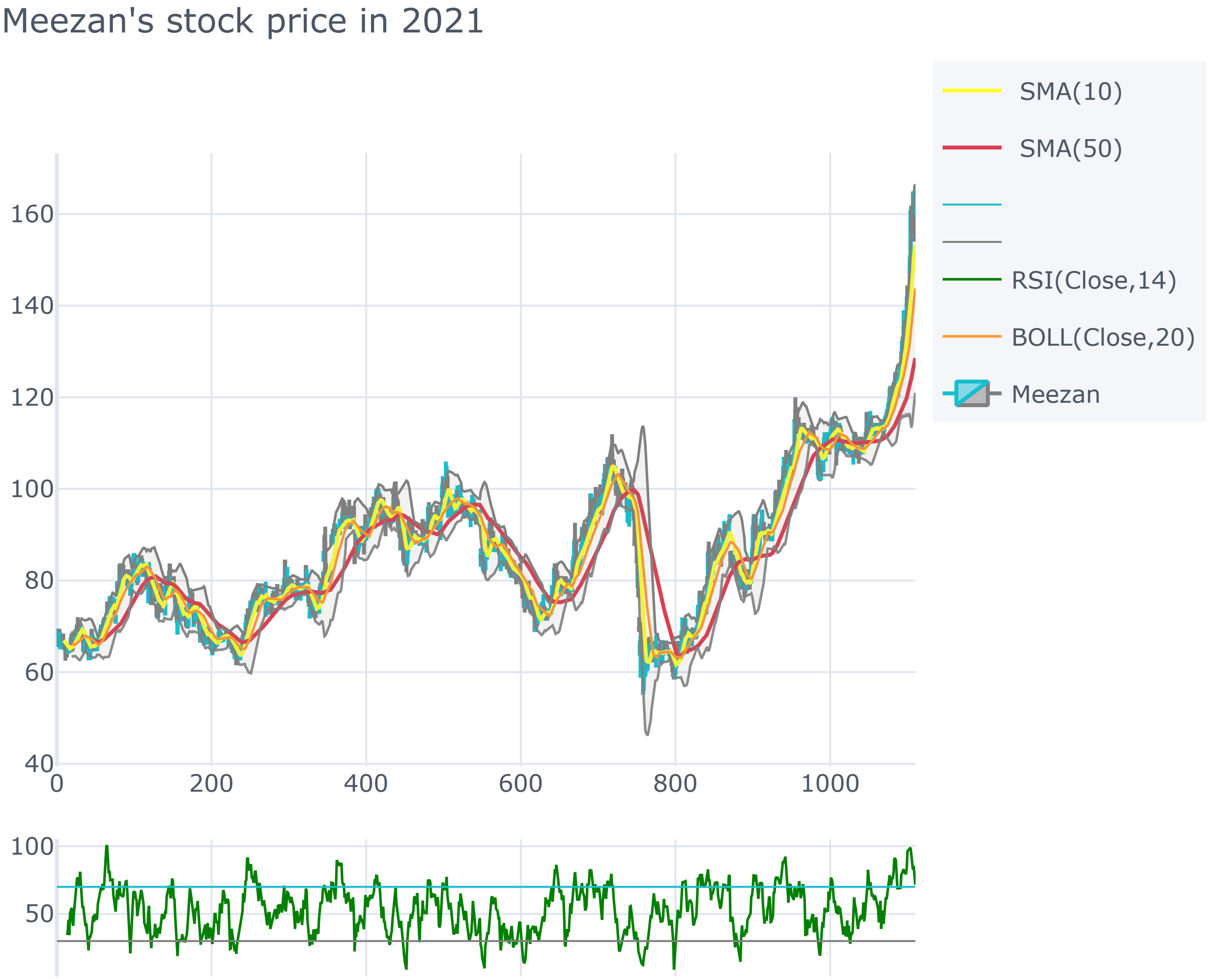
```
qf.add_rsi(periods=14, color='green')
qf.iplot()
```



Export to plot.ly »

In [273…

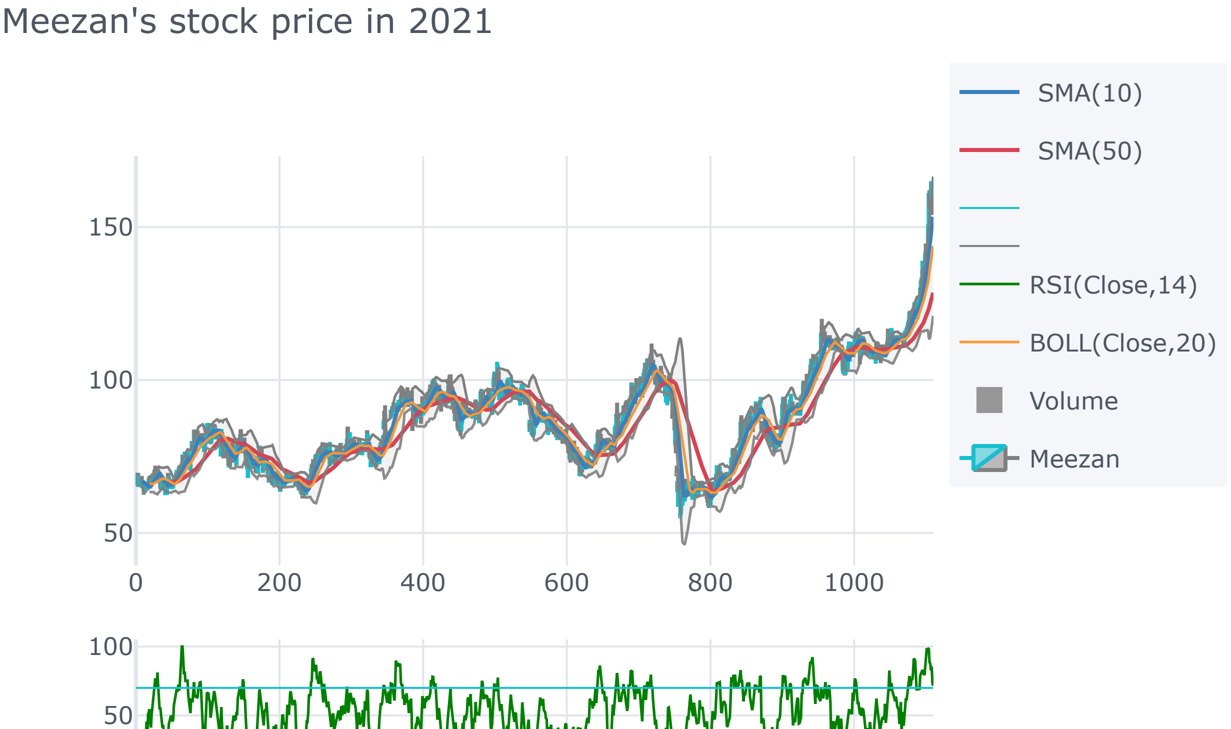
```
qf.add_bollinger_bands(periods=20, boll_std=2 ,colors=['orange','grey'], fill=True)
qf.iplot()
```

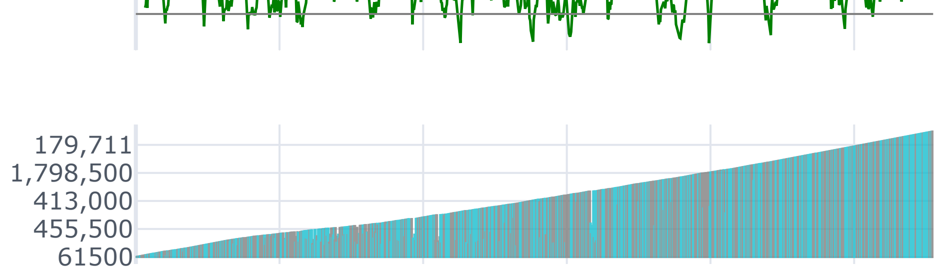


Export to plot.ly »

In [196…

```
qf.add_volume()
qf.iplot()
```

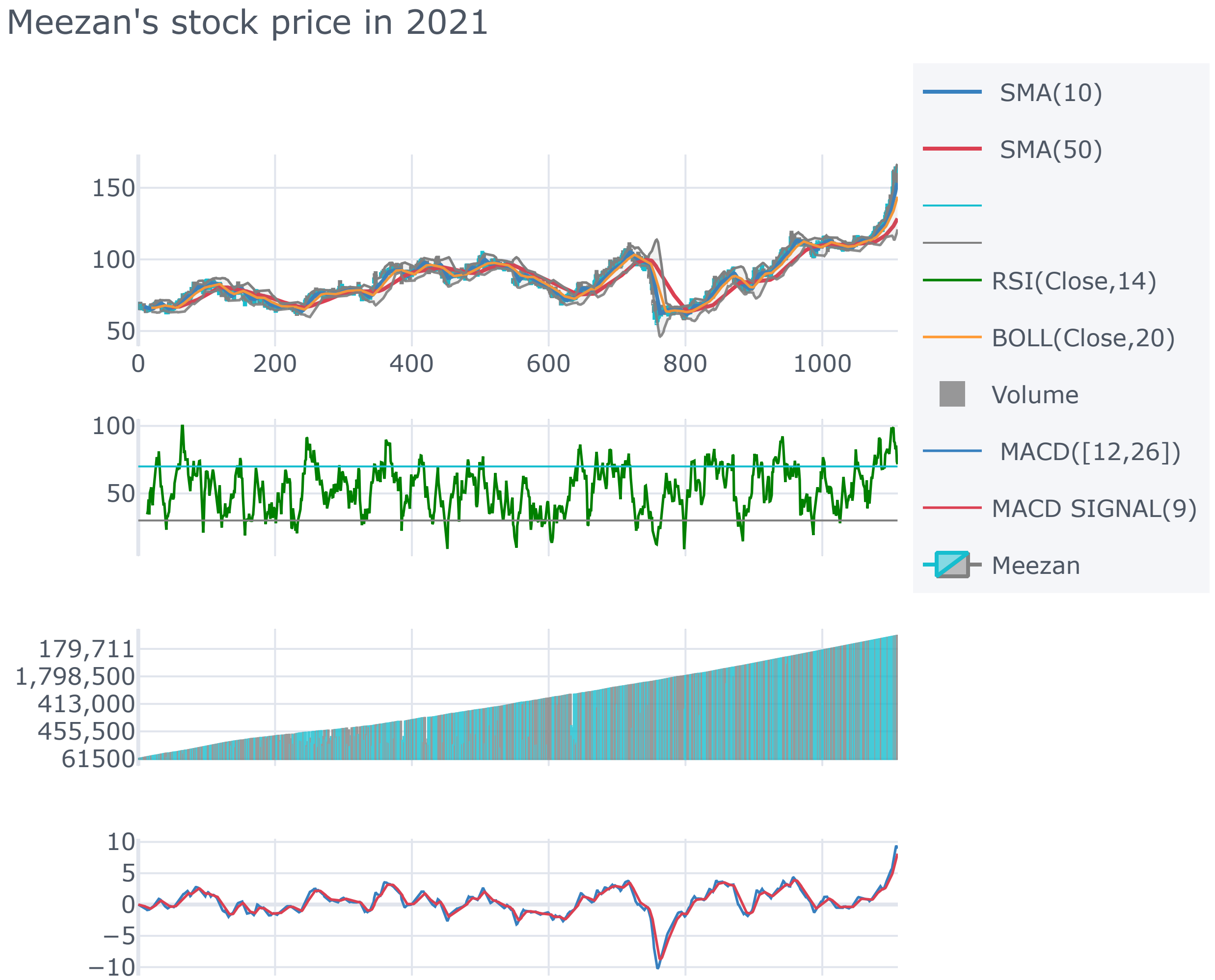




Export to plot.ly »

In [197…

```
qf.add_macd()
qf.iplot()
```



Export to plot.ly »

In [2]:

```
!jupyter nbconvert --allow-chromium-download
```

```
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
```

```
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and include the e
rror message in the cell output (the default behaviour is to abort conversion). This flag
is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with default base
name 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=no
tebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False --NbConvertApp.export_format=no
tebook --FilesWriter.build_directory= --ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True --TemplateExporter.exclud
e_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True --TemplateExporter.exclu
de_input=True --TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is found on the system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This is flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
```

```
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf', 'pytho
n', 'rst', 'script', 'slides', 'webpdf']
            or a dotted object name that represents the import path for an
            ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                  to output to the directory of each notebook. To recover
                                  previous default behaviour (outputting to the current
                                  working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-slidesho
w)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb --to html
            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'noteboo
k', 'pdf', 'python', 'rst', 'script', 'slides', 'webpdf'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX includes
            'base', 'article' and 'report'.  HTML includes 'basic', 'lab' and
            'classic'. You can specify the flavor of the format used.

            > jupyter nbconvert --to html --template lab mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```

```
In [ ]:
```