```
from google.colab import files
uploaded= files.upload()
```

Choose Files Coca-Cola_…tory (1).csv

Coca-Cola_stock_history (1).csv(text/csv) - 1233831 bytes, last modified: 10/14/2025 - 100% done

```
Saving Coca-Cola_stock_history (1).csv to Coca-Cola_stock_history (1).csv
```

```
import pandas as pd
cc=pd.read_csv("Coca-Cola_stock_history (1).csv")
cc.head()
```

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	1962-01-02	0.050016	0.051378	0.050016	0.050016	806400	0.0	0
1	1962-01-03	0.049273	0.049273	0.048159	0.048902	1574400	0.0	0
2	1962-01-04	0.049026	0.049645	0.049026	0.049273	844800	0.0	0
3	1962-01-05	0.049273	0.049892	0.048035	0.048159	1420800	0.0	0
4	1962-01-08	0.047787	0.047787	0.046735	0.047664	2035200	0.0	0

Next steps: Generate code with cc | New interactive sheet

```
lstm_model = Sequential()
lstm_model.add(LSTM(50, return_sequences=True, input_shape=(0, 1)))
lstm_model.add(LSTM(50, return_sequences=False))
lstm_model.add(Dense(25))
lstm_model.add(Dense(1))
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `in
  super().__init__(**kwargs)
```

```
data = cc.filter(["Close"])
dataset = data.values
```

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

```
# Splitting the data into training and testing sets
train_size = int(len(dataset) * 0.8)
train_data = scaled_data[0:train_size, :]
test_data = scaled_data[train_size - 60:, :]

# Preparing the data for LSTM model
def create_dataset(data, look_back=60):
    X, y = [], []
    for i in range(len(data) - look_back):
        X.append(data[i:(i + look_back), 0])
        y.append(data[i + look_back, 0])
    return np.array(X), np.array(y)

look_back = 60
X_train, y_train = create_dataset(train_data, look_back)
X_test, y_test = create_dataset(test_data, look_back)
```

```
lstm_model.compile(optimizer='adam', loss='mean_squared_error')
lstm_model.fit(X_train, y_train, batch_size=1, epochs=1)
```

```
12188/12188 ━━━━━━━━━━━━━━━━━━━━ 301s 24ms/step - loss: 1.3347e-04
<keras.src.callbacks.history.History at 0x7f7cfbf3c380>
```

```
lstm_model.compile(optimizer='adam', loss='mean_squared_error')
lstm_model.fit(X_train, y_train, batch_size=1, epochs=1)
```

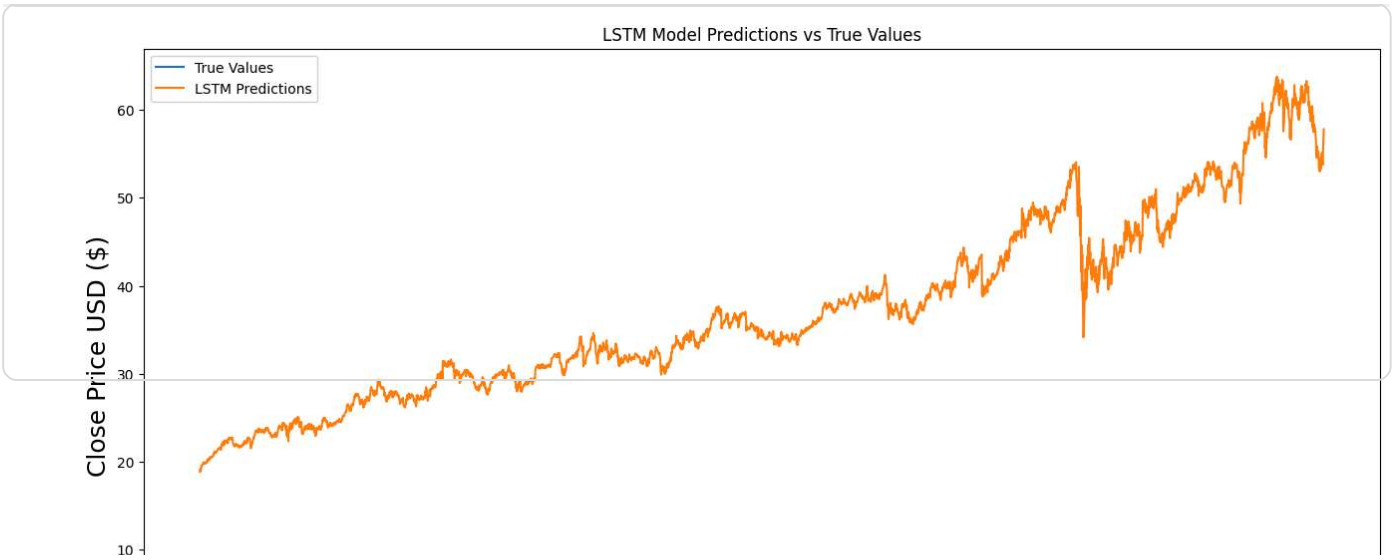
```
12188/12188 ━━━━━━━━━━━━━━━━━━━━ 295s 24ms/step - loss: 3.0791e-05
<keras.src.callbacks.history.History at 0x7f7cfc31bd40>
```

```
lstm_predictions = lstm_model.predict(X_test)
lstm_predictions = scaler.inverse_transform(lstm_predictions)

rmse = np.sqrt(np.mean(((lstm_predictions - y_test) ** 2)))
print('LSTM Model RMSE:', rmse)
```

```
96/96 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step
LSTM Model RMSE: 37.947840251746456
```

```
# Visualizing the Results for LSTM
plt.figure(figsize=(16,8))
plt.title('LSTM Model Predictions vs True Values')
plt.plot(y_test, label='True Values')
plt.plot(lstm_predictions, label='LSTM Predictions')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)
plt.legend()
plt.show()
```


LSTM Model Predictions vs True Values
True Values
LSTM Predictions
Close Price USD ($)
60
50
40
30
20
10