(click here to open code in Google Colab :- https://colab.research.google.com/drive/1kgqz8Xs6RW6K9JkrWkvu_MzYze0CHaSk?usp=sharing)

(Github :- https://github.com/sandeshdesai07/Stock-price-prediction.git)

Source Code

```
#Import the libraries

import pandas as pd

import xgboost as xgb

import matplotlib.pyplot as plt


#Load the small_dataset

data=pd.read_csv('stock_small_data.csv')


#Show data

data


#Show the data visually

data['Close'].plot(color='g')


#Split the data into training and testing data sets

train_data=data.iloc[:int(.49*len(data)), :]

test_data=data.iloc[int(.49*len(data)):, :]


#Define the features and target variable

features=['Open','Volume']

target='Close'


#Create and train the model

model=xgb.XGBRegressor()

model.fit(train_data[features],train_data[target])
```

```
#Make and show the predictions on the test data
predictions=model.predict(test_data[features])
print(predictions)

#Show the actual values
print('Actual Values: ')
print(test_data[target])

#Show the models accuracy
accuracy=model.score(test_data[features],test_data[target])
print('Accuracy:')
print(accuracy*100,"%")

#Plot the predictions and the close price
plt.plot(data['Close'],label='Close Price',color='g')
plt.plot(test_data[target].index,predictions,label='Predictions',color='r')
plt.legend()
plt.show()


#Load the large_dataset
data=pd.read_csv('stock_large_data.csv')

#Show data
data

#Show the data visually
data['Close'].plot(color='g')

#Split the data into training and testing data sets
```

```
train_data=data.iloc[:int(.99*len(data)), :]
test_data=data.iloc[int(.99*len(data)):, :]

#Define the features and target variable
features=['Open','Volume']
target='Close'

#Create and train the model
model=xgb.XGBRegressor()
model.fit(train_data[features],train_data[target])

#Make and show the predictions on the test data
predictions=model.predict(test_data[features])
print(predictions)

#Show the actual values
print('Actual Values: ')
print(test_data[target])

#Show the models accuracy
accuracy=model.score(test_data[features],test_data[target])
print('Accuracy:')
print(accuracy*100,"%")

#Plot the predictions and the close price
plt.plot(data['Close'],label='Close Price',color='g')
plt.plot(test_data[target].index,predictions,label='Predictions',color='r')
plt.legend()
plt.show()
```