

```
In [2]: get_ipython().system('pip install plotly==3.10.0')

from chart_studio import plotly

Requirement already satisfied: plotly==3.10.0 in c:\users\abada\anaconda3\lib\site-packages (3.10.0)
Requirement already satisfied: retrying==1.3.3 in c:\users\abada\anaconda3\lib\site-packages (from plotly==3.10.0) (1.3.4)
Requirement already satisfied: nbformat>=4.2 in c:\users\abada\anaconda3\lib\site-packages (from plotly==3.10.0) (5.5.0)
Requirement already satisfied: six in c:\users\abada\anaconda3\lib\site-packages (from plotly==3.10.0) (1.16.0)
Requirement already satisfied: decorator>=4.0.6 in c:\users\abada\anaconda3\lib\site-packages (from plotly==3.10.0) (5.1.1)
Requirement already satisfied: pytz in c:\users\abada\anaconda3\lib\site-packages (from plotly==3.10.0) (2022.1)
Requirement already satisfied: requests in c:\users\abada\anaconda3\lib\site-packages (from plotly==3.10.0) (2.28.1)
Requirement already satisfied: jsonschema>=2.6 in c:\users\abada\anaconda3\lib\site-packages (from nbformat>=4.2->plotly==3.10.0) (4.16.0)
Requirement already satisfied: fastjsonschema in c:\users\abada\anaconda3\lib\site-packages (from nbformat>=4.2->plotly==3.10.0) (2.16.2)
Requirement already satisfied: traitlets>=5.1 in c:\users\abada\anaconda3\lib\site-packages (from nbformat>=4.2->plotly==3.10.0) (5.1.1)
Requirement already satisfied: jupyter_core in c:\users\abada\anaconda3\lib\site-packages (from nbformat>=4.2->plotly==3.10.0) (4.11.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\abada\anaconda3\lib\site-packages (from requests->plotly==3.10.0) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\abada\anaconda3\lib\site-packages (from requests->plotly==3.10.0) (1.26.11)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\abada\anaconda3\lib\site-packages (from requests->plotly==3.10.0) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\abada\anaconda3\lib\site-packages (from requests->plotly==3.10.0) (3.3)
Requirement already satisfied: pyrsistent!=0.17.0,!0.17.1,!0.17.2,>=0.14.0 in c:\users\abada\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2->plotly==3.10.0) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\abada\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=4.2->plotly==3.10.0) (21.4.0)
Requirement already satisfied: pywin32>=1.0 in c:\users\abada\anaconda3\lib\site-packages (from jupyter_core->nbformat>=4.2->plotly==3.10.0) (302)
```

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

get_ipython().run_line_magic('matplotlib', 'inline')

import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import plot

#for offline plotting
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

```
In [6]: tesla = pd.read_csv('supermarket_sales.csv')

tesla.head()

tesla.shape

tesla.info

tesla['Date'] = pd.to_datetime(tesla['Date'])

print(f'Dataframe contains stock prices between {tesla.Date.min()} {tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max() - tesla.Date.min()).days} days')

tesla.describe()

tesla[['Unit price', 'Quantity', 'Tax 5%', 'gross margin percentage', 'Total']].plot(kind='box')

layout = go.Layout(
    title='Stock Prices of Supermarket',
    xaxis=dict(
        title='Date',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='red'
        )
    ),
    yaxis=dict(
        title='Price',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='red'
        )
    )
)

tesla_data = [{'x':tesla['Date'], 'y':tesla['Unit price']}]
plot =go.Figure(data=tesla_data, layout=layout)

iplot(plot)

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import r2_score

X = np.array(tesla.index).reshape(-1,1)
Y = tesla['Unit price']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=10 )

scaler = StandardScaler().fit(X_train)

from sklearn.linear_model import LinearRegression

lm = LinearRegression()
lm.fit(X_train, Y_train)

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

trace0 = go.Scatter(
    x = X_train.T[0],
    y = Y_train,
    mode = 'markers',
    name = 'Actual'
)
trace1 = go.Scatter(
    x = X_train.T[0],
    y = lm.predict(X_train).T,
    mode = 'lines',
    name = 'Predicted'
)
tesla_data = [trace0,trace1]
layout.xaxis.title.text = 'Day'
plot2 = go.Figure(data=tesla_data, layout=layout)

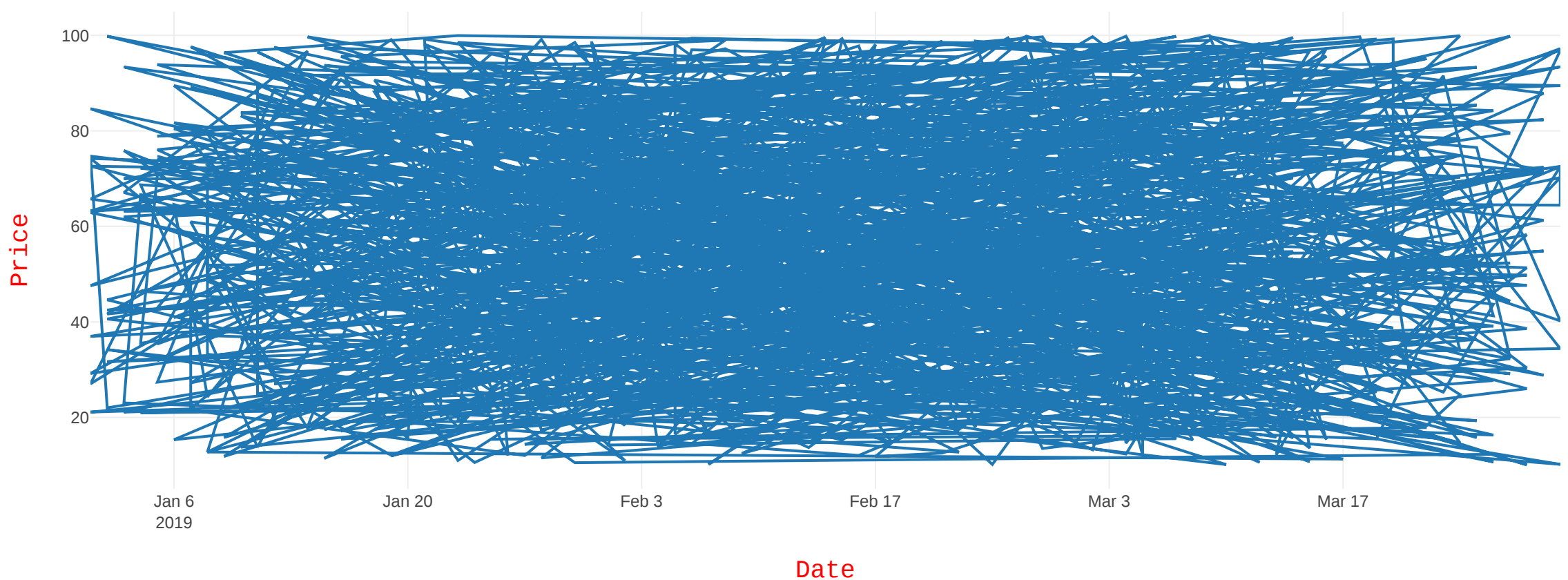
iplot(plot2)

scores = f'''
{'Metric'.ljust(10)}{'Train'.center(20)}{'Test'.center(20)}
{'r2_score'.ljust(10)}{r2_score(Y_train, lm.predict(X_train))}\t{r2_score(Y_test, lm.predict(X_test))}
{'MSE'.ljust(10)}{mse(Y_train, lm.predict(X_train))}\t{mse(Y_test, lm.predict(X_test))}
'''
print(scores)

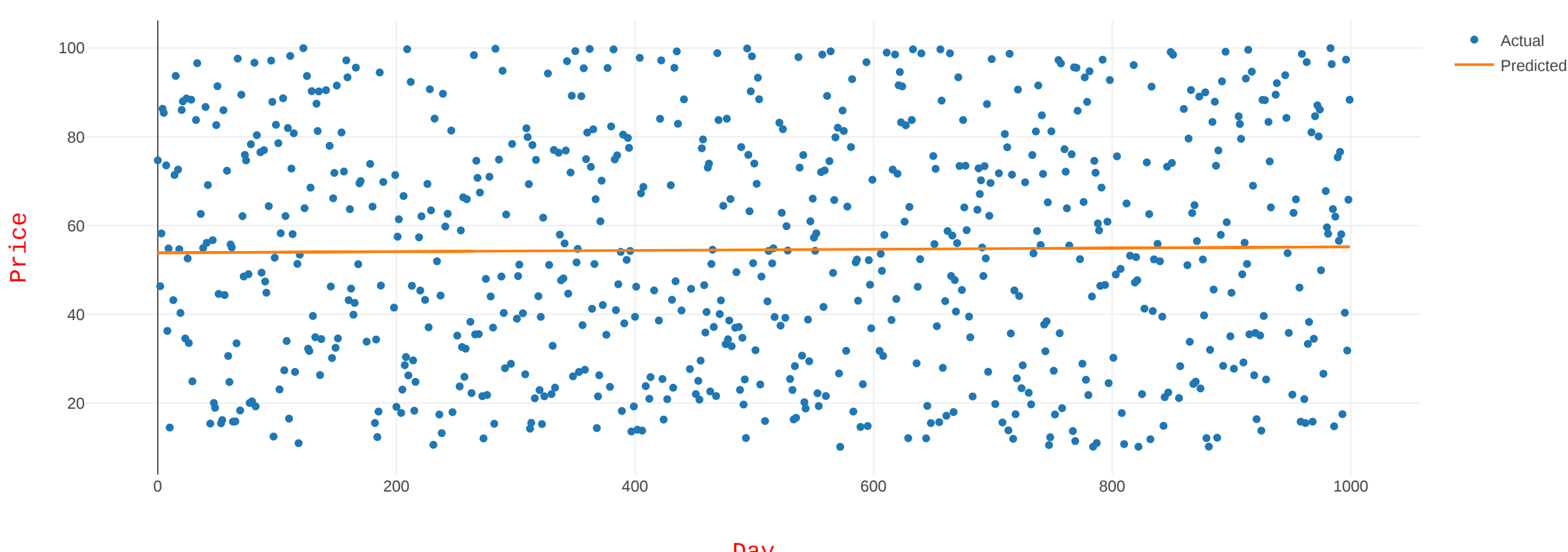
Dataframe contains stock prices between 2019-01-01 00:00:00 2019-03-30 00:00:00
Total days = 88 days
```



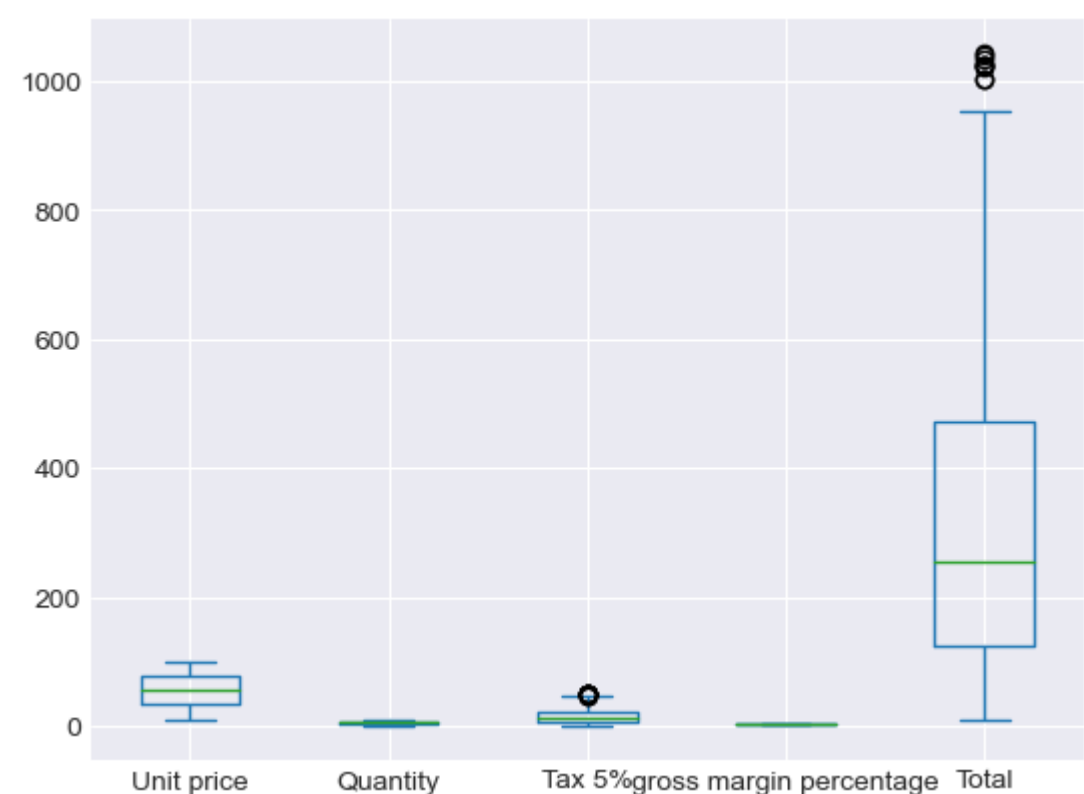
Stock Prices of Supermarket



Stock Prices of Supermarket



Metric	Train	Test
r2_score	0.0002136797308759375	-0.023120849431798485
MSE	703.7128823899028	700.7059069705898



```
In [ ]:
```