Project Apollo

Final Project - Machine Learning
7/10/2018
Ray Trounday


HD

Is it possible to predict where a stock price is headed?


61.8175
82.230
16.370
39.500
62.748
0.472
0.420
0.1325
1.250
0.340
0.340
0.412
4.300
0.130
0.010
1.53%
0.66%
0.21%
1.50%
2.03%
0.87%
0.65%
0.96%
0.80%
0.17%
22.550
30.400
16.310
39.900
N/A
200
600
3400
40.710
16.380

Steps

1. Import Libraries
2. Read Data
3. Define Feature Variables
4. Define Dependent Variable
5. Split the Data to Train and Test Data Set
6. Create the Linear Regression Model
7. Predict the Price

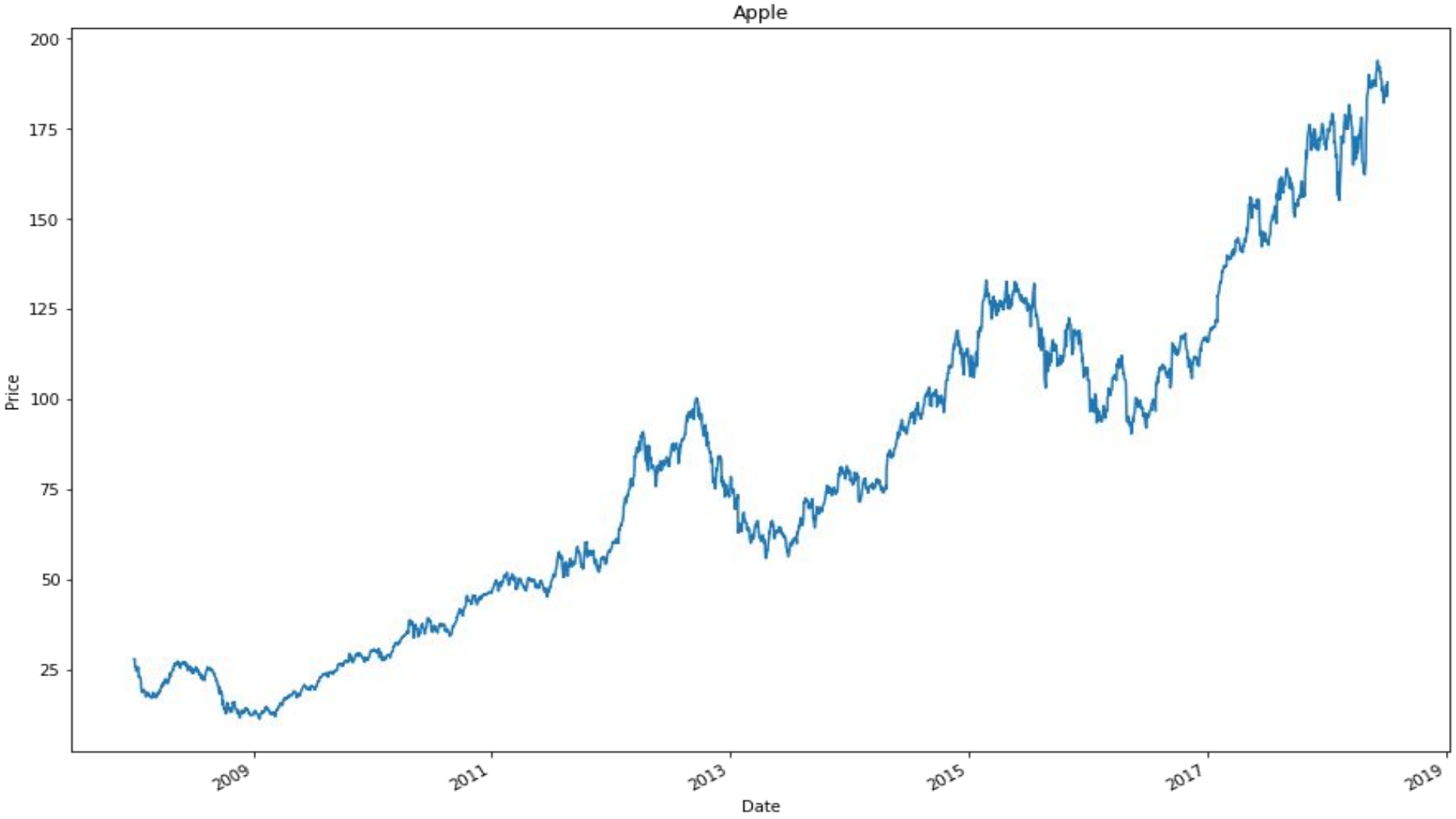
Step 1

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn

from sklearn.linear_model import LinearRegression

import fix_yahoo_finance as yf
```

Step 2 - Read Data

```
Df = yf.download('AAPL', '2008-01-01', '2018-07-08')
```

Date	Close
2018-06-29	185.110001
2018-07-02	187.179993
2018-07-03	183.919998
2018-07-05	185.399994
2018-07-06	187.970001


Apple
200
175
150
125
100
75
50
25
Price
2009
2011
2013
2015
2017
2019
Date

Step 3 - Define Feature Variables

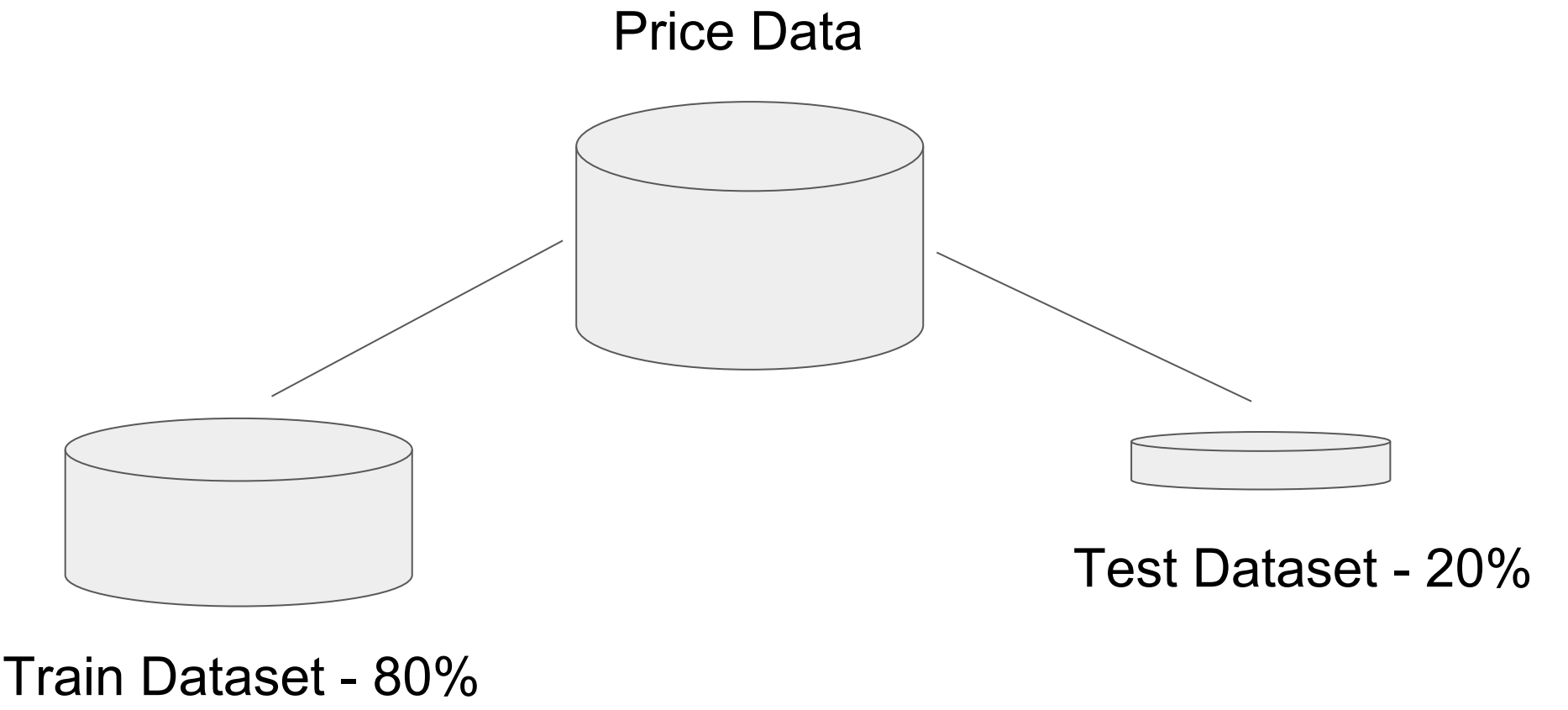
```
Df['SMA_3'] =
Df['Close'].shift(1).rolling(window=3).mean()

Df['SMA_9'] =
Df['Close'].shift(1).rolling(window=9).mean()

Df= Df.dropna()

X = Df[['SMA_3','SMA_9']]
```

Step 4 - Define Dependent Variables

```
y = Df['Close']

y.head()
```

Step 5 - Split Data

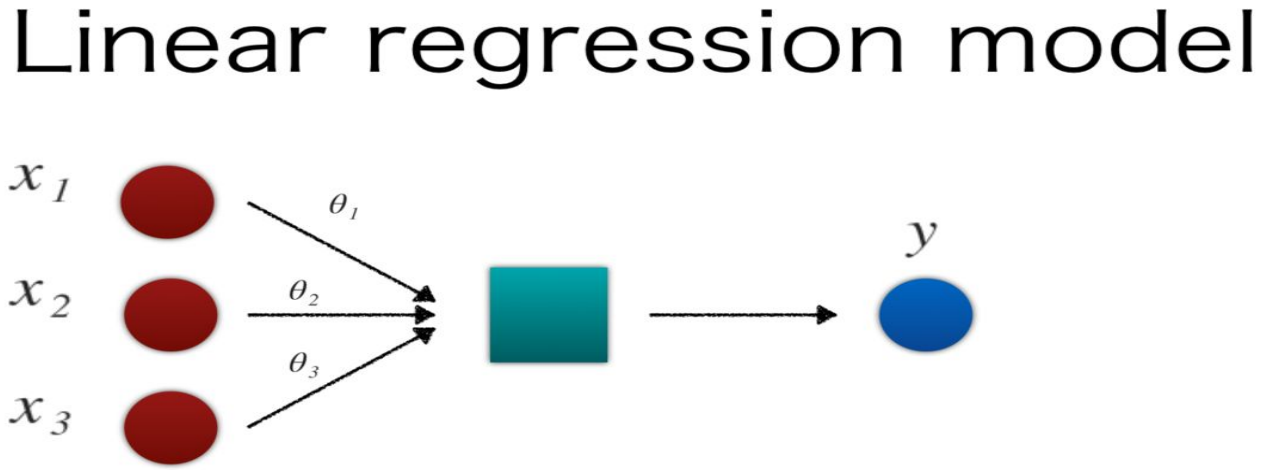
```
t=.8

t= int(t*len(Df))

X_train = X[:t]

y_train = y[:t]

X_test = X[t:]

y_test = y[t:]
```


Price Data
Train Dataset - 80%
Test Dataset - 20%

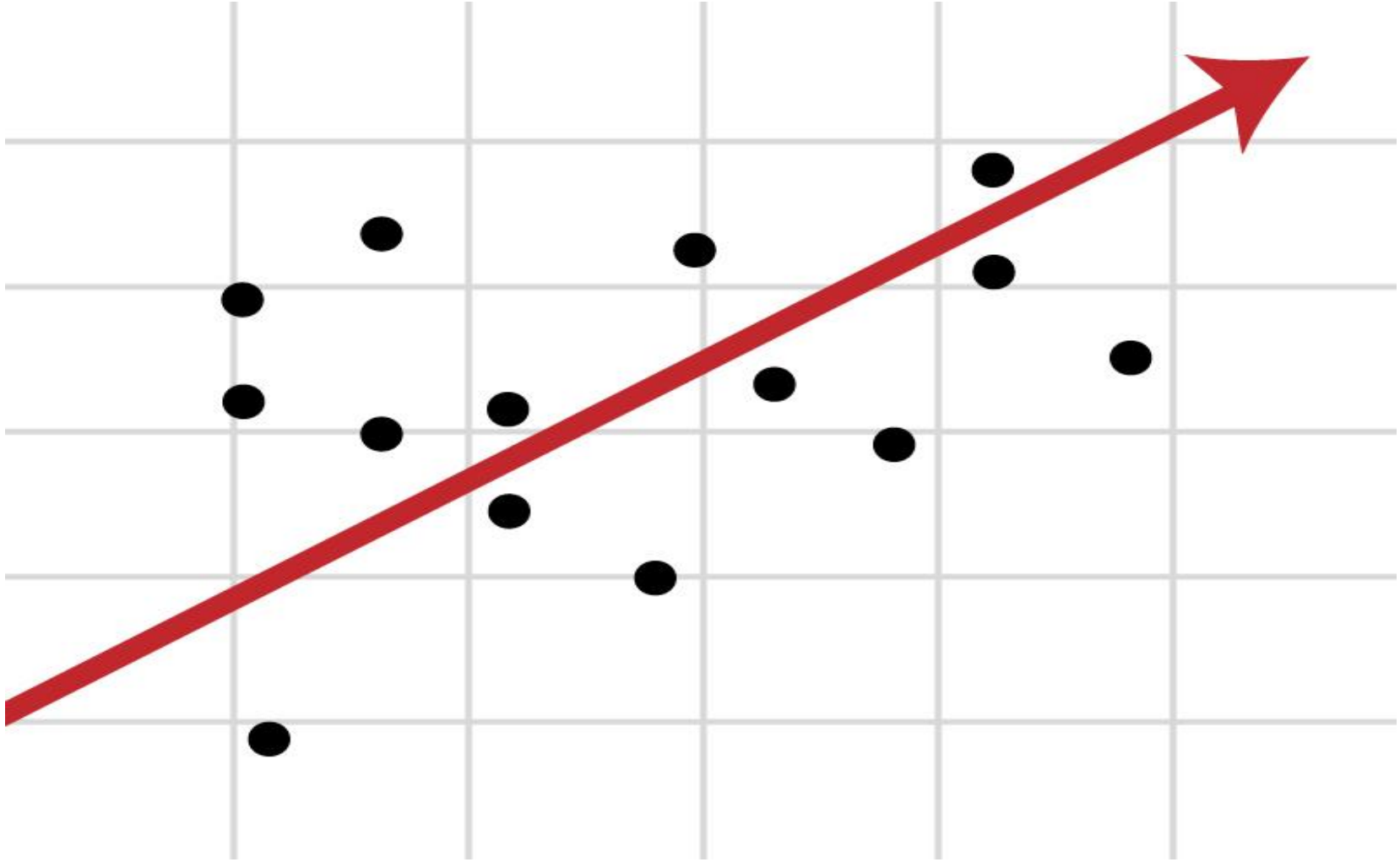
Step 6 - Create Model

```
## Create Linear regression Model

linear = LinearRegression().fit(X_train,y_train)
```

What is a Linear Regression Model?

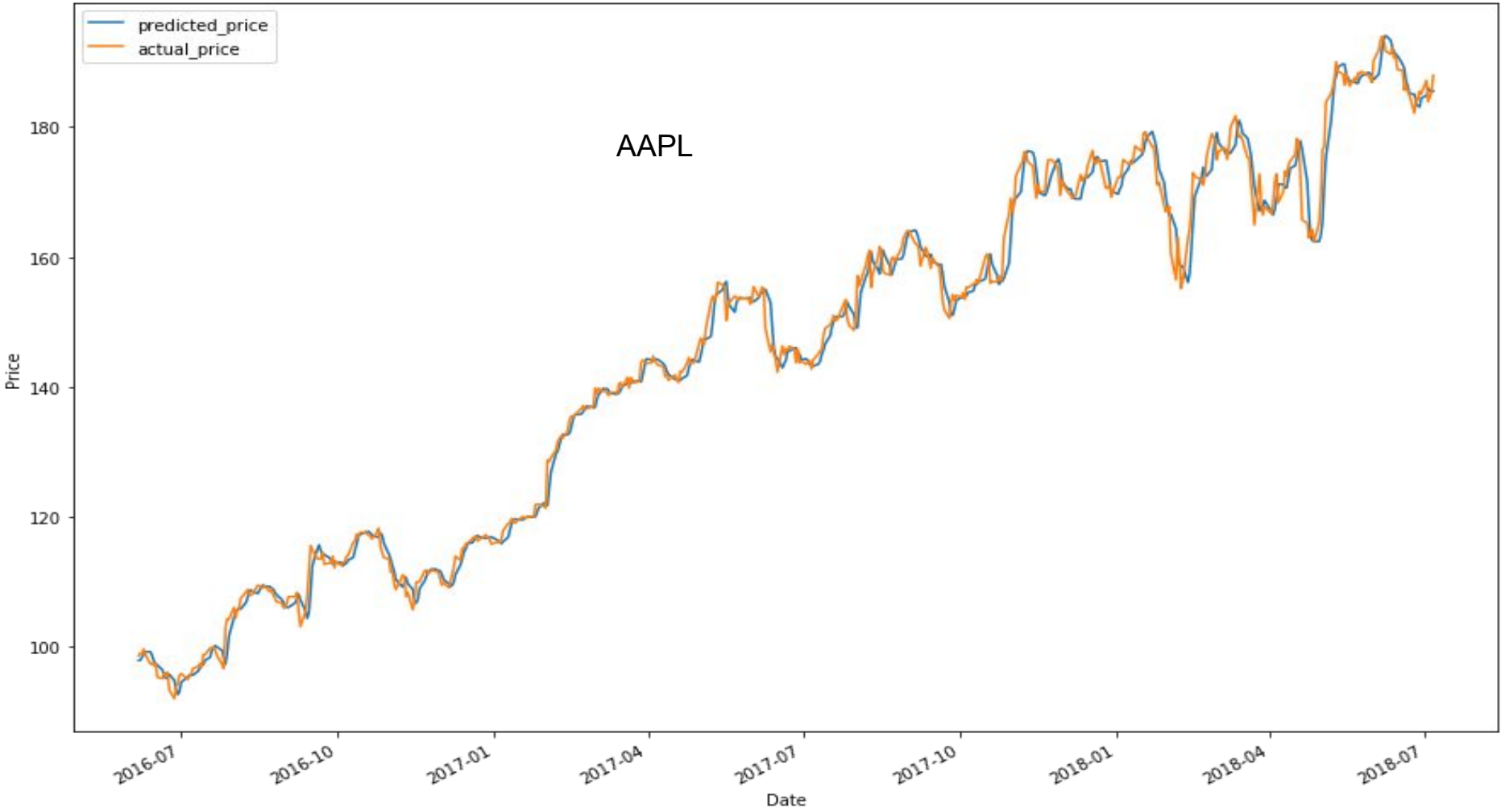


$$y=\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_3$$



Step 7

```
p_price = linear.predict(X_test)

p_price = pd.DataFrame(p_price,index=y_test.index,columns = ['price'])

p_price.plot(figsize=(15,10))

y_test.plot()

plt.legend(['predicted_price','actual_price'])

plt.ylabel("Price")

plt.show()
```


predicted_price
actual_price
AAPL
Price
180
160
140
120
100
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07
Date

Fit Score - AAPL

```
r2 = linear.score(X[t:],y[t:])*100
```

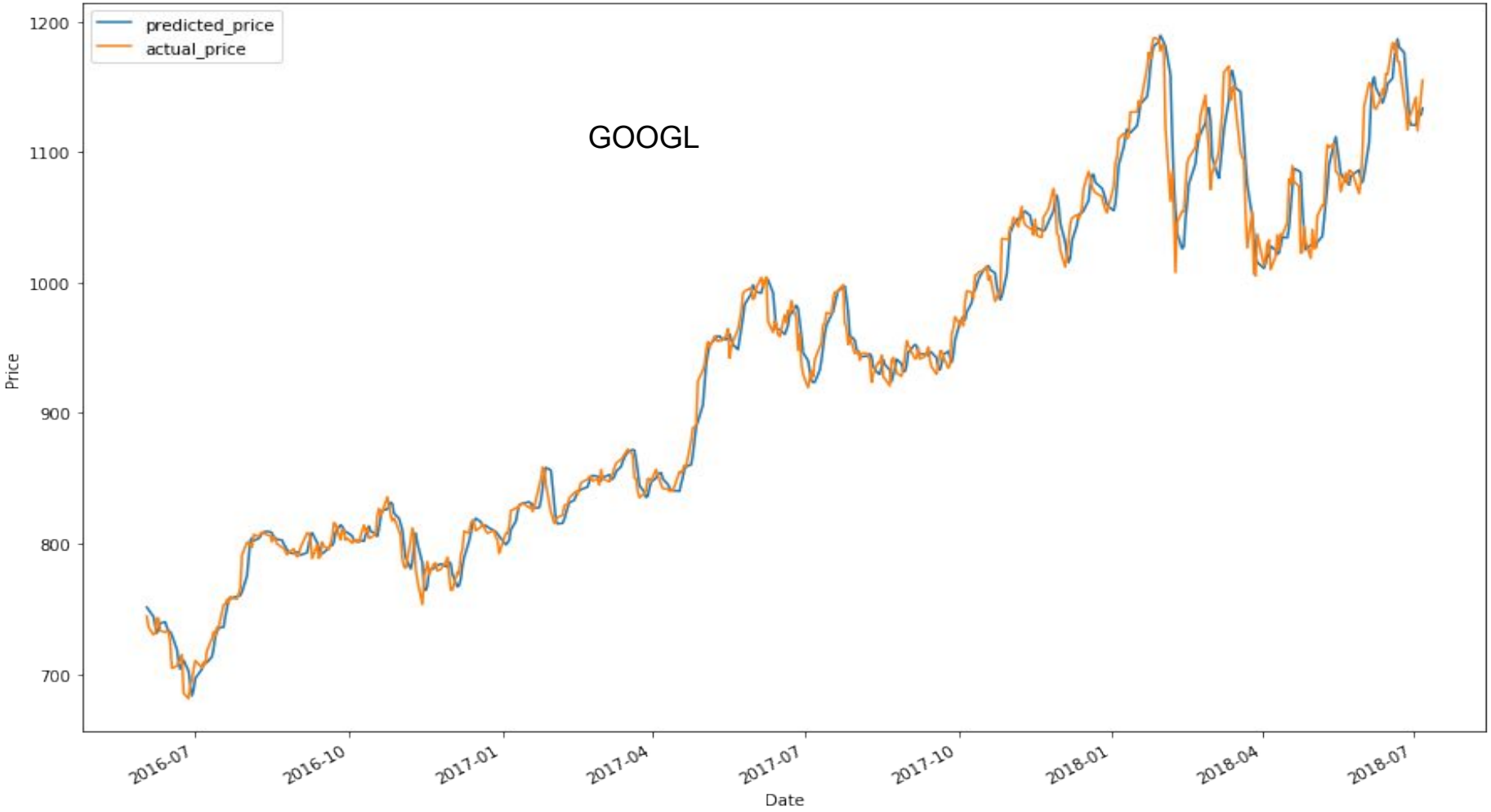
99.31%


predicted_price
actual_price
AMZN
Price
Date
120
110
100
90
80
70
60
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07

Fit Score - AMZN

r2 = linear.score(X[t:],y[t:])*100

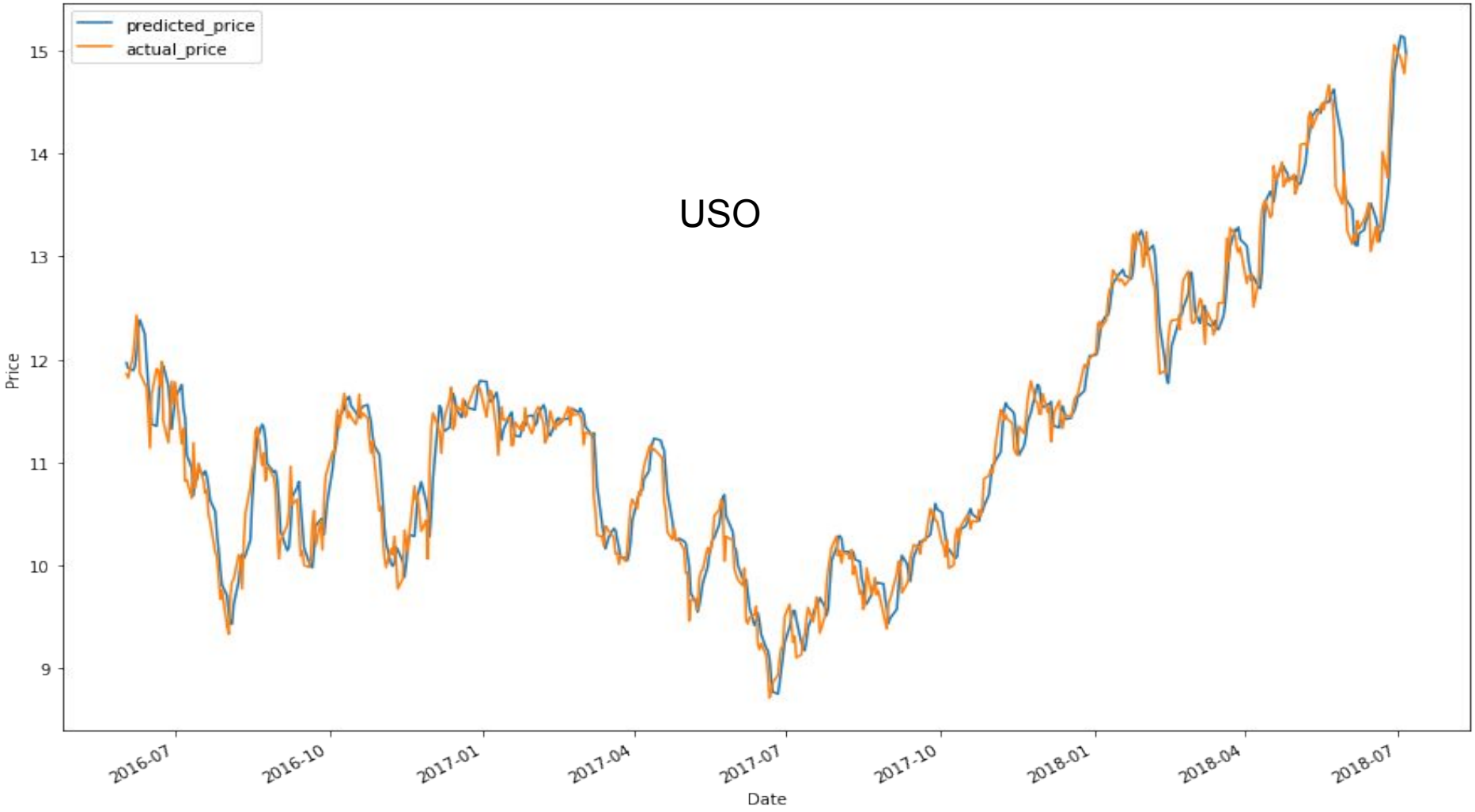
99.54%


predicted_price
actual_price
GOOGL
1200
1100
1000
900
800
700
Price
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07
Date

Fit Score - GOOGL

```
r2 = linear.score(X[t:],y[t:])*100
```

98.65%


predicted_price
actual_price
USO
Price
15
14
13
12
11
10
9
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07
Date

Fit Score - USO

```
r2 = linear.score(X[t:],y[t:])*100
```

96.84%

Future considerations

1. Incorporate other feature variables such as other technical indicators or sentiment analysis
2. Other regression models Polynomial
3. Leverage analysis to arrive at Trading Strategies

Challenges

1. Access to Historical data
2. Trial and Error
3. Overfitting?