

Basic_Machine_Learning_Predicts

September 29, 2021

1 Simple Linear Regression for stock using scikit-learn

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")

import fix_yahoo_finance as yf
yf.pdr_override()
```

```
[2]: stock = 'AAPL'
start = '2016-01-01'
end = '2018-01-01'
data = yf.download(stock, start, end)
data.head()
```

[*****100%*****] 1 of 1 downloaded

```
[2]:
```

	Open	High	Low	Close	Adj Close	\
Date						
2016-01-04	102.610001	105.370003	102.000000	105.349998	100.274513	
2016-01-05	105.750000	105.849998	102.410004	102.709999	97.761681	
2016-01-06	100.559998	102.370003	99.870003	100.699997	95.848511	
2016-01-07	98.680000	100.129997	96.430000	96.449997	91.803276	
2016-01-08	98.550003	99.110001	96.760002	96.959999	92.288696	

	Volume
Date	
2016-01-04	67649400
2016-01-05	55791000
2016-01-06	68457400
2016-01-07	81094400

2016-01-08 70798000

```
[3]: df = data.reset_index()
      df.head()
```

```
[3]:
```

	Date	Open	High	Low	Close	Adj Close	\
0	2016-01-04	102.610001	105.370003	102.000000	105.349998	100.274513	
1	2016-01-05	105.750000	105.849998	102.410004	102.709999	97.761681	
2	2016-01-06	100.559998	102.370003	99.870003	100.699997	95.848511	
3	2016-01-07	98.680000	100.129997	96.430000	96.449997	91.803276	
4	2016-01-08	98.550003	99.110001	96.760002	96.959999	92.288696	

	Volume
0	67649400
1	55791000
2	68457400
3	81094400
4	70798000

```
[4]: X = df.drop(['Date', 'Close'], axis=1, inplace=True)
      y = df[['Adj Close']]
```

```
[5]: df = df.as_matrix()
```

```
[6]: from sklearn.model_selection import train_test_split

      # Split X and y into X_
      X_train, X_test, y_train, y_test = train_test_split(df, y, test_size=0.25,
      ↪random_state=0)
```

```
[7]: from sklearn.linear_model import LinearRegression

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

```
[7]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

```
[8]: intercept = regression_model.intercept_[0]

      print("The intercept for our model is {}".format(intercept))
```

The intercept for our model is -1.2047109976265347e-09

```
[9]: regression_model.score(X_test, y_test)
```

```
[9]: 1.0
```

```
[10]: from sklearn.metrics import mean_squared_error

y_predict = regression_model.predict(X_test)

regression_model_mse = mean_squared_error(y_predict, y_test)

regression_model_mse
```

```
[10]: 2.8264629110010686e-19
```

```
[11]: math.sqrt(regression_model_mse)
```

```
[11]: 5.316448919157475e-10
```

```
[12]: # input the latest Open, High, Low, Close, Volume
# predicts the next day price
regression_model.predict([[167.81, 171.75, 165.19, 166.48, 37232900]])
```

```
[12]: array([[166.48]])
```