Predict the stock price from the Apple stock

In this Notebook we want to predict the future opening price of the Apple NSDAQ stock.

First let's import our neccessaray libarys:

In [209]:

```
import numpy as np
import pandas as pd
from sklearn import model_selection
from sklearn import linear_model
import math
```

Loading the dataset

Now let's load the Apple stock dataset.

In [210]:

```
names = ['Date','Open','High','Low','Close','Adj Close','Volume']
stock_values = pd.read_csv('AAPL.csv',names=names,engine='python')
stock_values.drop(0, inplace=True)
```

Now let's just create a DataFrame to convert the objects to strings and floats:

In [211]:

```
stock = pd.DataFrame({'1Date': stock_values['Date'].astype('string'),
                      '2Open': stock_values['Open'].astype('float'),
                      '3High': stock_values['High'].astype('float'),
                      '4Low': stock_values['Low'].astype('float'),
                      '5Close': stock_values['Close'].astype('float')}
)
stock.head()
```

Out[211]:

	1Date	2Open	3High	4Low	5Close
1	2017-04-19	141.880005	142.000000	140.449997	140.679993
2	2017-04-20	141.220001	142.919998	141.160004	142.440002
3	2017-04-21	142.440002	142.679993	141.850006	142.270004
4	2017-04-24	143.500000	143.949997	143.179993	143.639999
5	2017-04-25	143.910004	144.899994	143.869995	144.529999

Splitting the dataset into training and testing data

With the sklearn model selection we will now split the data into training and testing data:

In [212]:

```
forecast_col = '5Close'
stock.fillna(-99999, inplace=True)
forecast_out = int(math.ceil(0.01*len(stock)))
stock['6Label'] = stock[forecast_col].shift(-forecast_out)
stock.dropna(inplace=True)
stock.head()
```

Out[212]:

	1Date	2Open	3High	4Low	5Close	6Label
1	2017-04-19	141.880005	142.000000	140.449997	140.679993	143.639999
2	2017-04-20	141.220001	142.919998	141.160004	142.440002	144.529999
3	2017-04-21	142.440002	142.679993	141.850006	142.270004	143.679993
4	2017-04-24	143.500000	143.949997	143.179993	143.639999	143.789993
5	2017-04-25	143.910004	144.899994	143.869995	144.529999	143.649994

In [213]:

```
X = stock[['2Open','3High', '4Low', '5Close']]
Y = stock['6Label']
X_lately = X[-forecast_out:]

test_size = 0.20
seed = 7

features_train, features_test, labels_train, labels_test = model_selec
tion.train_test_split(X, Y, test_size=test_size, random_state=seed)
```

Defining the classifier

Now let's get to the final step of predicting stoc prices. Let'S define the classifier:

In [214]:

```
clf = linear_model.LinearRegression()
clf.fit(features_train,labels_train)
accuarcy = clf.score(features_test,labels_test)
prediction = clf.predict(X_lately)
print("Accuarcy: " + str(accuarcy))
```

```
Accuarcy: 0.853620276289
```

In [215]:

```
print("Here are the next three days stock prices:")
print(prediction)
```

```
Here are the next three days stock prices:
[ 173.71773801  174.18778137  175.09761291]
```