In [1]:

```
import quandl
import pandas as pd
import numpy as np
```

In [2]:

```
df = quandl.get("WIKI/FB")
```

In [3]:

```
df.head()
```

Out[3]:

	Open	High	Low	Close	Volume	Ex-Dividend	Split Ratio	Adj. Open	Adj. High	Adj. Low	Adj. Close
Date											
2012-05-18	42.05	45.00	38.00	38.2318	573576400.0	0.0	1.0	42.05	45.00	38.00	38.2318
2012-05-21	36.53	36.66	33.00	34.0300	168192700.0	0.0	1.0	36.53	36.66	33.00	34.0300
2012-05-22	32.61	33.59	30.94	31.0000	101786600.0	0.0	1.0	32.61	33.59	30.94	31.0000
2012-05-23	31.37	32.50	31.36	32.0000	73600000.0	0.0	1.0	31.37	32.50	31.36	32.0000
2012-05-24	32.95	33.21	31.77	33.0300	50237200.0	0.0	1.0	32.95	33.21	31.77	33.0300

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
Open           0
High           0
Low            0
Close          0
Volume         0
Ex-Dividend    0
Split Ratio    0
Adj. Open      0
Adj. High      0
Adj. Low       0
Adj. Close     0
Adj. Volume    0
dtype: int64
```

In [5]:

```
df = df[['Adj. Close']]
```

In [6]:

```
df.head()
```

Out[6]:

```
            Adj. Close
      Date
2012-05-18     38.2318
2012-05-21     34.0300
2012-05-22     31.0000
2012-05-23     32.0000
2012-05-24     33.0300
```

In [11]:

```
forecast = 30
```

In [12]:

```
df['Prediction'] = df[['Adj. Close']].shift(-forecast)
```

In [13]:

```
df.tail()
```

Out[13]:

```
            Adj. Close   Prediction
      Date
2018-03-21      169.39          NaN
2018-03-22      164.89          NaN
2018-03-23      159.39          NaN
2018-03-26      160.06          NaN
2018-03-27      152.19          NaN
```

In [33]:

```
X = np.array(df.drop(['Prediction'], axis = 1))
X = X[:-forecast]
print(X)
```

```
[[ 38.2318]
 [ 34.03  ]
 [ 31.    ]
 ...
 [171.5499]
 [175.98  ]
 [176.41  ]]
```

In [34]:

```
y = np.array(df['Prediction'])
y = y[:-forecast]
print(y)
```

```
[ 30.771   31.2     31.47   ... 159.39   160.06   152.19 ]
```

In [16]:

```
from sklearn.model_selection import train_test_split
```

In [17]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [18]:

```
from sklearn.linear_model import LinearRegression
```

In [19]:

```
model = LinearRegression()
```

In [20]:

```
model.fit(X_train, y_train)
```

Out[20]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fal
se)
```

In [21]:

```
model.score(X_train, y_train)
```

Out[21]:

```
0.9799397219389175
```

In [22]:

```
from sklearn.metrics import classification_report
```

In [23]:

```
predict = model.predict(X_test)
```

In [24]:

```
print(classification_report(predict, y_test))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-24-e6e856185e7c> in <module>
----> 1 print(classification_report(predict, y_test))

~\Anaconda3\lib\site-packages\sklearn\metrics\classification.py in classific
ation_report(y_true, y_pred, labels, target_names, sample_weight, digits, ou
tput_dict)
   1850     """
   1851 
-> 1852     y_type, y_true, y_pred = _check_targets(y_true, y_pred)
   1853 
   1854     labels_given = True

~\Anaconda3\lib\site-packages\sklearn\metrics\classification.py in _check_ta
rgets(y_true, y_pred)
     86     # No metrics support "multiclass-multioutput" format
     87     if (y_type not in ["binary", "multiclass", "multilabel-indicato
r"]):
---> 88         raise ValueError("{0} is not supported".format(y_type))
     89 
     90     if y_type in ["binary", "multiclass"]:

ValueError: continuous is not supported
```

In [25]:

```
score = model.score(X_test, y_test)
print("score: ", score)
```

```
score:  0.9821570952330557
```

In [26]:

```
from sklearn import svm
```

In [30]:

```
clf = svm.SVR(kernel = 'rbf', C = 1e3, gamma = 0.1)
```

In [31]:

```
clf.fit(X_train, y_train)
```

Out[31]:

```
SVR(C=1000.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma=0.1,
    kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

In [32]:

```
score1 = clf.score(X_test, y_test)
print("score1: ", score1)
```

```
score1:  0.9771320911240383
```

In [ ]: