Stock Price Prediction Using Machine Learning Algorithm.

By: Kelvin Oyanna

Email: dotkelplus@gmail.com

Twitter: @KelvinOyanna

Call: 08036467038

About Me

- I’m a data scientist with core interest in data analytics technology, Machine Learning and Artificial intelligence.

- I love building tech communities. I’m currently the Co-organizer & team lead for Pydata Lagos.

- I love Fried plantain - Dodo

What is a stock?

Stocks also referred to as shares or equity represents a fractional ownership in a company.

What is the stock market?

The stock market refers to public markets that exist for issuing, buying and selling stocks that trade on a stock exchange or over-the-counter.

Simply put, the stock market is a place where investors can sell or buy ownership of companies in the form of stocks or shares.

What's the fuss about the Stock Market?

Global Market Capitalization = $80 Trillion.

Heading steadily towards $100 Trillion

Nigeria Stock Exchange (NSE)

The Market Capitalization of NSE as at January 2018 stood at N14.18 Trillion.

Why Stock Price Prediction?

First, let me tell you a story…

Me after Investing in Stock

Grab some popcorn

&

Watch my money grow like grass.

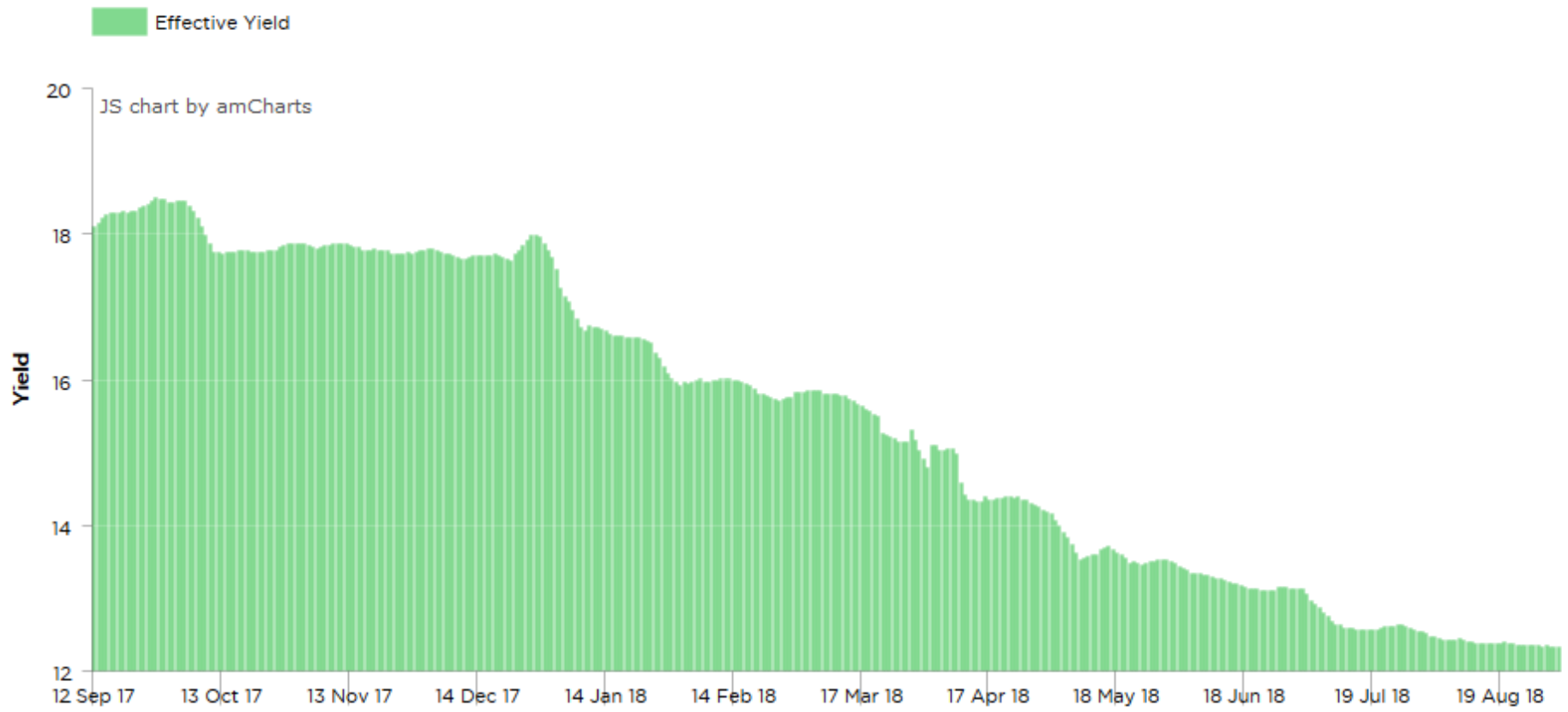
Me after seeing my investment yield in 3 months

Me seeing my investment yield after 1 year.

Here is what happened:

FUND PERFORMANCE

Money Market Fund | Discovery Fund | Aggressive Growth Fund | Ethical Fund



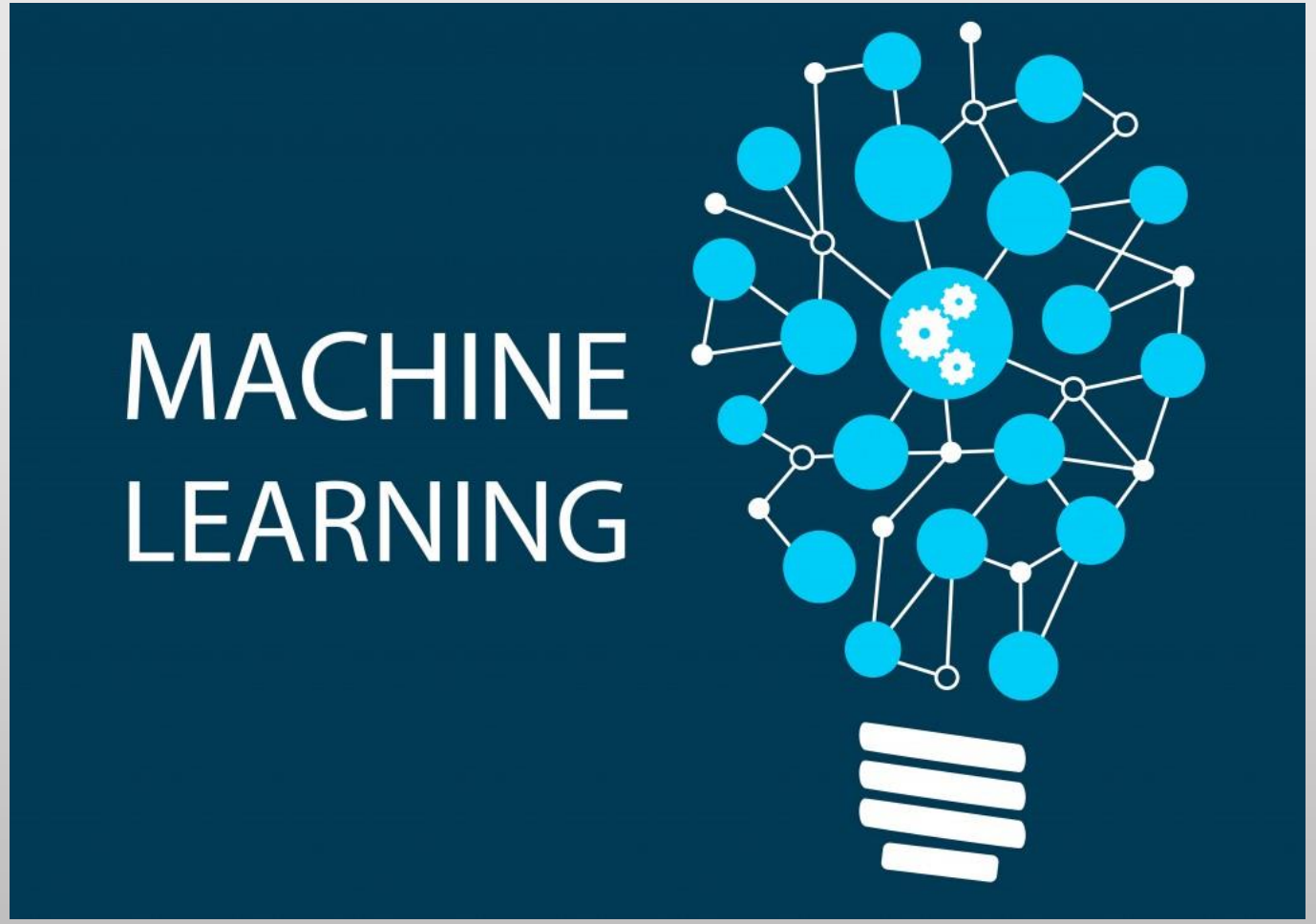
Think About this:

Imagine I understand all the underlying factors that influence the prices of stocks so I can purchase stocks that will yield better returns.

Imagine I can build an automated trading system that can understand these factors, accurately predict future prices of stocks and tell me which stock is safe/profitable to buy.

Your guess is as good as mine:

Machine Learning to the rescue!



“Machine Learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world.”

- Nvidia

Let’s dive into implementing a Machine Learning Algorithm to predict stock prices…

The Goal:

Our goal is to build a Machine Learning Model to predict stock prices.

Outcome:

The outcome of our work will be predicted prices & a measure of the percentage of accuracy of our model in predicting the prices.

jupyter Stock Price Prediction Last Checkpoint: Last Friday at 11:28 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Code

Importing Libraries

In [136]:

```
import pandas as pd
import requests
import numpy as np
from bs4 import BeautifulSoup
```

Create an empty dataframe to store our data

In [137]:

```
data = pd.DataFrame()
pd.options.mode.chained_assignment = None #Prevent error Warning of a copy in memory
```

jupyter Stock Price Prediction Last Checkpoint: Last Friday at 11:28 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Code

Build a simple Web Scraper to import Stock Market data from Ngtradeonline.com

In [138]:

```
count = 0
while count < 20    :
    res = requests.get("http://www.ngtradeonline.com/Home/PriceHistory?page="+str(count)+"&s
    soup = BeautifulSoup(res.content,'lxml') #Use Beautiful soup to prettify our data
    table = soup.find_all('table')[0] #Find all the html tag named 'table'
    html_data = pd.read_html(str(table))[0] #Read the content of found html tables
    data = data.append(html_data) # Append to the empty Dataframe above
    print("page "+str(count)+" Done.") #Keep track of each web page read
    count = count + 1
data.to_csv('GTB_data.csv') # Export the data to a CSV file
```

```
page 0 Done.
page 1 Done.
page 2 Done.
page 3 Done.
page 4 Done.
page 5 Done.
page 6 Done.
page 7 Done.
page 8 Done.
```

Let's look at the Data

```
In [172]:  1 data.head(20)
```

Out[172]:

	Symbol	Low	Open	Price	Volume	High	Change	Date
0	GUARANTY	31.95	34.50	32.50	0	33.35	-2.00	9/11/2018
1	GUARANTY	34.50	35.00	34.50	0	35.00	-0.50	9/10/2018
2	GUARANTY	34.40	34.95	35.00	0	35.30	0.05	9/7/2018
3	GUARANTY	34.95	36.65	34.95	0	36.50	-1.70	9/6/2018
4	GUARANTY	36.65	37.05	36.65	0	36.85	-0.40	9/5/2018
5	GUARANTY	36.50	36.50	37.05	0	37.45	0.55	9/4/2018
6	GUARANTY	36.05	36.00	36.50	0	36.75	0.50	9/3/2018
7	GUARANTY	36.00	38.00	36.00	0	37.85	-2.00	8/31/2018
8	GUARANTY	37.90	39.05	38.00	0	38.00	-1.05	8/30/2018
9	GUARANTY	38.65	39.00	39.05	0	39.05	0.05	8/29/2018
0	GUARANTY	31.95	34.50	32.50	0	33.35	-2.00	9/11/2018
1	GUARANTY	34.50	35.00	34.50	0	35.00	-0.50	9/10/2018

Data transformation

In [139]:

```
#Create a column to hold the percentage spread of the closing price
data['HL_PCT'] = (data['High']- data['Low'])/data['Price'] *100.0

#Create a column to hold the Percentage of Price Change
data['PCT_change'] = (data['Price'] - data['Open']) / data['Open'] * 100.0
```

Generate a new set of data from the transformation above

In [140]:
```
new_data = data[['Price', 'HL_PCT', 'PCT_change', 'Volume']]
```

In [174]:
```
new_data.head(10)
```

Out[174]:

	Price	HL_PCT	PCT_change	Volume	label
0	32.50	4.307692	-5.797101	0	35.00
1	34.50	1.449275	-1.428571	0	34.95
2	35.00	2.571429	0.143062	0	36.65
3	34.95	4.434907	-4.638472	0	37.05
4	36.65	0.545703	-1.079622	0	36.50
5	37.05	2.564103	1.506849	0	36.00
6	36.50	1.917808	1.388889	0	38.00
7	36.00	5.138889	-5.263158	0	39.05
8	38.00	0.263158	-2.688860	0	32.50
9	39.05	1.024328	0.128205	0	34.50

Import the Machine Learning Algorithms from Scikit Learn library

In [142]:
```
from sklearn import preprocessing, cross_validation, svm
from sklearn.linear_model import LinearRegression
import math
```

Feature Engineering - Forcasting the stock prices

In [144]:
```
forcast_value = int(math.ceil(0.01 * len(new_data))) # Deduce the number of times to forcast
# Add a new column that a shifting of the price values based on the size of data
new_data['label'] = new_data['Price'].shift(-forcast_value)
```

In [145]:
```
new_data.head()
```

Out[145]:

	Price	HL_PCT	PCT_change	Volume	label
0	32.50	4.307692	-5.797101	0	35.00
1	34.50	1.449275	-1.428571	0	34.95
2	35.00	2.571429	0.143062	0	36.65
3	34.95	4.434907	-4.638472	0	37.05
4	36.65	0.545703	-1.079622	0	36.50

In [146]:
```
new_data.dropna(inplace= True)
```

Creating the features and the label for training the model

In [147]:
```
x = np.array(new_data.drop(['Price'], axis = 1))
y = np.array(new_data['Price'])
```

Pre-processing: Scaling the data to normalize it's values

In [148]:
```
#Pre-processing
x = preprocessing.scale(x)
```

In [149]:
```
# Splitting the data into training and test set
x_train, x_test, y_train, y_test = cross_validation.train_test_split(x, y, test_size=0.2)
```

Building the Machine Learning Model

Using Linear Regression Machine Learning Algorithm

In [150]:
```
linear_model = LinearRegression()
linear_model.fit(x_train, y_train)
```

Out[150]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

Evaluating the Accuracy of the model

In [151]:
```
accuracy = linear_model.score(x_test, y_test)
accuracy = accuracy * 100
print('Percentage Accuracy = ',np.round(accuracy),'%')
```

```
Percentage Accuracy =  94.0 %
```

Using Support Vector Regression (SVR) Machine Learning Algorithm

In [189]:
```
clf = svm.SVR() #Creating an instance of SVR model
clf.fit(x_train, y_train)   # Fitting the model on the data
```

Out[189]:
```
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='auto',
  kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

Evaluating the Accuracy of the model

In [190]:
```
accuracy = clf.score(x_test, y_test)
accuracy = accuracy * 100
print('Percentage Accuracy = ',np.round(accuracy),'%')
```

```
Percentage Accuracy =  88.0 %
```

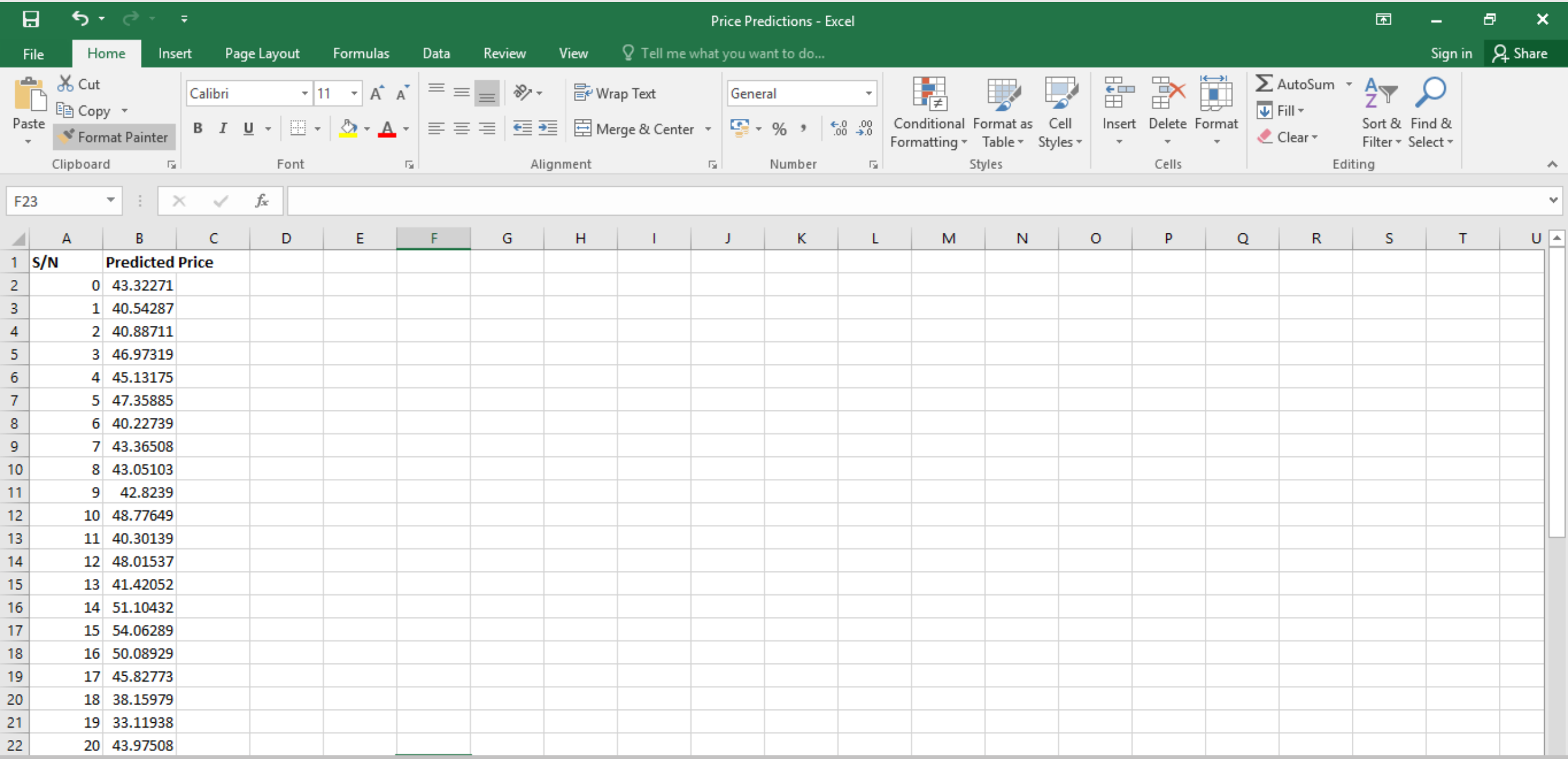
Using multiple kernels for the SVR

In [192]:

```
for k in ['linear','poly','rbf','sigmoid']:
    clf = svm.SVR(kernel=k)
    clf.fit(x_train, y_train)
    confidence = clf.score(x_test, y_test)
    print('Percentage Accuracy = ', k, np.round(accuracy), '%')

```

```
Percentage Accuracy =  linear 88.0 %
Percentage Accuracy =  poly 88.0 %
Percentage Accuracy =  rbf 88.0 %
Percentage Accuracy =  sigmoid 88.0 %
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	**S/N**	**Predicted Price**																			
2	0	43.32271																			
3	1	40.54287																			
4	2	40.88711																			
5	3	46.97319																			
6	4	45.13175																			
7	5	47.35885																			
8	6	40.22739																			
9	7	43.36508																			
10	8	43.05103																			
11	9	42.8239																			
12	10	48.77649																			
13	11	40.30139																			
14	12	48.01537																			
15	13	41.42052																			
16	14	51.10432																			
17	15	54.06289																			
18	16	50.08929																			
19	17	45.82773																			
20	18	38.15979																			
21	19	33.11938																			
22	20	43.97508																			

Conclusion

This model can be deployed as a web service on an automated trading platform.

Link to Datasets & Code on GitHub:

https://github.com/KelvinOyanna/Pycon-Nigeria-2018-Resource.git

Got Questions?

Thank you!