

Machine Learning in the Stock Market

...

Akhil Ganesan & Prabhas Adivi



Introduction

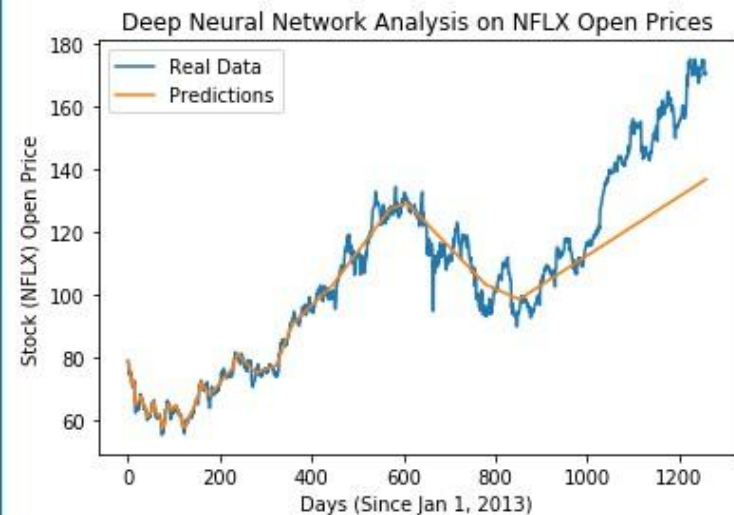


Purpose

- Artificial intelligence is the future
- The stock market is an ideal challenge
- We both enjoy IT, mathematics, and CS

Loss on training data: 2.8493080291748045%

Loss on testing data: 24.676584825552567%



24.67658436896718



Experimental Design & Hypothesis/Goal

Machine Types:

- Statistical Analysis (Lin. Reg.)
- Support Vector Regression (Guided)
- Deep Neural Network (Brain-like)

Testing for Accuracy

Constants: Environment, Return Type

Hypothesis: NN, SA, SVR

Goal: 90%, MAE(x) < 10

```
9985/10000
900 [=====] - 0s 99us/sample - loss: 3.0007 - accuracy: 0.00
9986/10000
900 [=====] - 0s 72us/sample - loss: 2.9528 - accuracy: 0.00
9987/10000
900 [=====] - 0s 68us/sample - loss: 3.0480 - accuracy: 0.00
9988/10000
900 [=====] - 0s 69us/sample - loss: 3.3654 - accuracy: 0.00
9989/10000
900 [=====] - 0s 66us/sample - loss: 3.1164 - accuracy: 0.00
9990/10000
900 [=====] - 0s 70us/sample - loss: 2.9615 - accuracy: 0.00
9991/10000
900 [=====] - 0s 65us/sample - loss: 3.0568 - accuracy: 0.00
9992/10000
900 [=====] - 0s 66us/sample - loss: 3.5866 - accuracy: 0.00
9993/10000
900 [=====] - 0s 65us/sample - loss: 3.1314 - accuracy: 0.00
9994/10000
900 [=====] - 0s 64us/sample - loss: 3.0919 - accuracy: 0.00
9995/10000
900 [=====] - 0s 67us/sample - loss: 3.1652 - accuracy: 0.00
9996/10000
900 [=====] - 0s 65us/sample - loss: 3.1102 - accuracy: 0.00
9997/10000
900 [=====] - 0s 66us/sample - loss: 3.0173 - accuracy: 0.00
9998/10000
900 [=====] - 0s 65us/sample - loss: 3.0682 - accuracy: 0.00
9999/10000
900 [=====] - 0s 66us/sample - loss: 3.0076 - accuracy: 0.00
10000/10000
900 [=====] - 0s 68us/sample - loss: 2.9596 - accuracy: 0.00
```



Materials & Methods



Materials & Procedure

Critical Materials:

- Testing Hardware/Software
- Data Sets

Procedure

- Set up Experiment/Data
- Run training/testing; record results
 - Repeat per machine

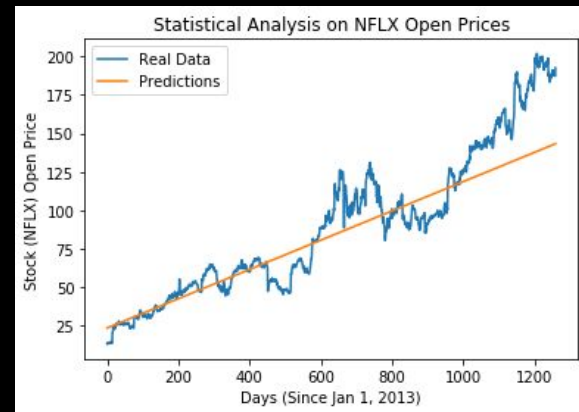
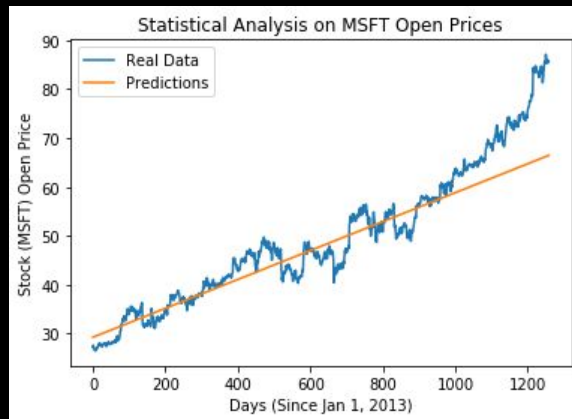
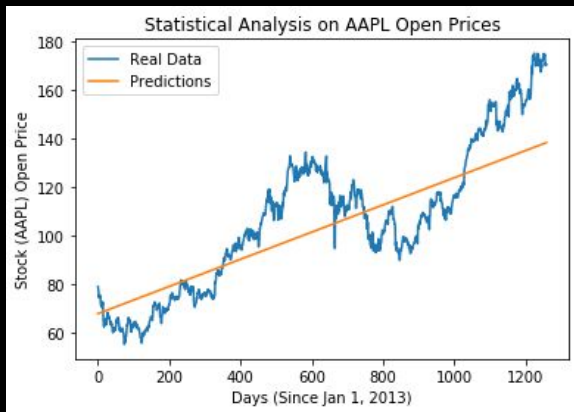
```
25         full_y.append(float(row[1]))
26         i += 1
27
28     train_x = np.reshape(range(0, 1000), (-1, 1))
29     test_x = np.reshape(range(1000, 1259), (-1, 1))
30     train_y = np.reshape(full_y[:1000], (-1, 1))
31     test_y = np.reshape(full_y[1000:], (-1, 1))
32     full_y = np.reshape(full_y, (-1, 1))
33
34     # Building the Model & Learning
35     model = LinearRegression()
36     model.fit(train_x, train_y)
37
38     # Predictions & Results
39     def graph(x, y):
40         plt.plot(x, y, label = "Real Data")
41         z = model.predict(x)
42         plt.plot(x, z, label = "Predictions")
43
44         plt.xlabel('Days (Since Jan 1, 2013)')
45         plt.ylabel('Stock (AAPL) Open Price')
46         plt.title('Statistical Analysis on AAPL Open Prices')
47         plt.legend(loc="upper left")
48         plt.show()
49
50     def mean_absolute_error(y, pred):
51         a = 0
52         b = 0
53         for i in range(len(y)):
54             a += y[i] - pred[i]
55             b += 1
56         return float(a)/b
57
58     graph(full_x, full_y)
59     print(mean_absolute_error(test_y, model.predict(test_x)))
```



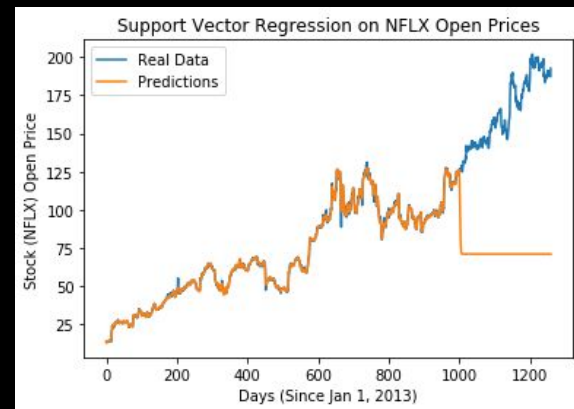
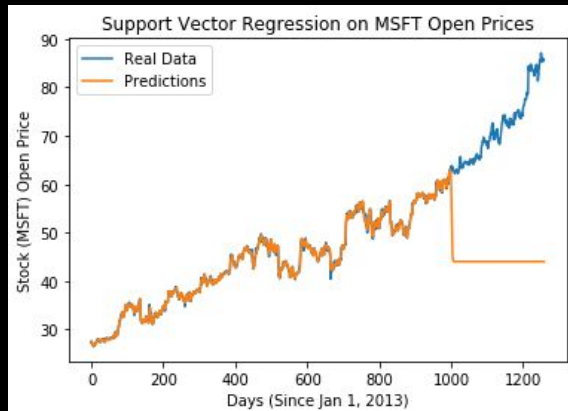
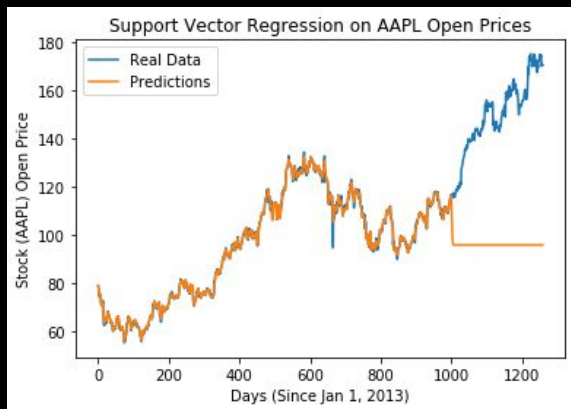
Results & Discussion



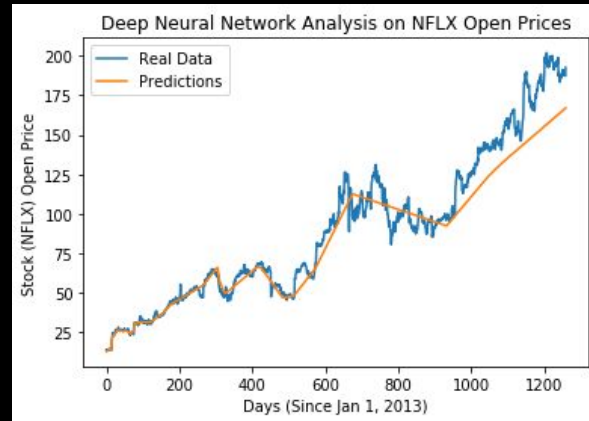
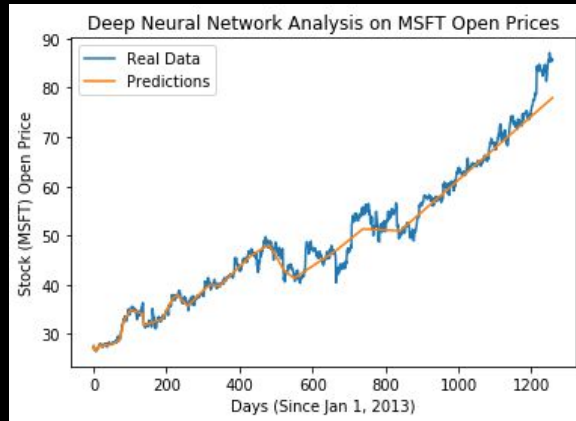
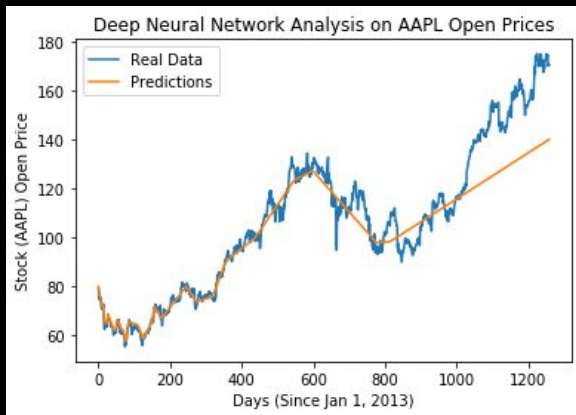
Statistical Analysis



Support Vector Regression



Deep Neural Network

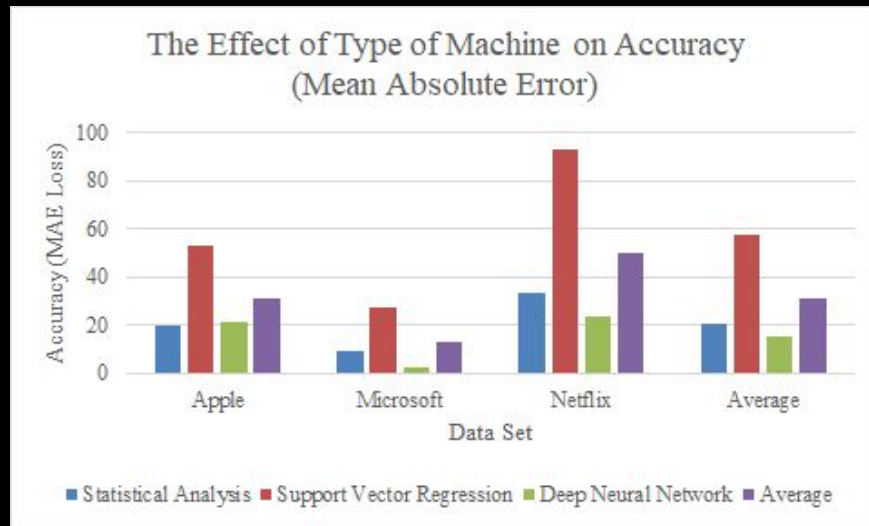


Conclusions, Limitations, & Future Action

1. Deep Neural Network (16.1)
2. Statistical Analysis (20.6)
3. Support Vector Regression (57.8)

Conclusions:

- Toughness of the data sets
- SVR expected values
- SA is best at AAPL
- MSFT NN performance



Science Fair Materials [Public Release]

Google Drive Folder: <https://tinyurl.com/2020SFmat>

(Contains Timeline, Research Paper, Presentation, and Machines/Datasets)

Research Paper (contained within folder): <https://tinyurl.com/2020SFpaper>

PowerPoint Presentation (contained within folder): <https://tinyurl.com/2020SFpre>

Testing Materials (contained within folder): <https://tinyurl.com/2020SFmach>

