Financial Forecasting For Stock Market Data

Abhishek Raval
College of Computer and Information Sciences, Northeastern University
P.O. Box 02120
Boston, USA
raval.a@husky.neu.edu

Ishan H Patel
College of Computer and Information Sciences, Northeastern University
P.O. Box 02120
Boston, USA
i@patelishan.com

Ishan L Patel
College of Computer and Information Science, Northeastern University
P.O. Box 02120
Boston, USA
i@ishanpatel.in

ABSTRACT

Forecasting whether the price of stock will increase or decrease is a difficult task. We try to predict whether the price of stock will outperform or underperform based on the fluctuations of features in dataset till previous day.

CCS CONCEPTS

• **Machine Learning → Classification & Prediction**; *Redundancy*; Error Analysis •**Artificial Intelligence →** Random Forest, Support Vector Machine, Neural network Algorithms.

ADDITIONAL KEYWORDS AND PHRASES

Stock Prediction, Artificial Intelligence

ACM Reference format:
Abhishek Raval, Ishan Patel, Ishan Patel, 2017. SIG Proceedings Paper in word Format.

1 INTRODUCTION

Forecasting whether the stock of certain product will escalate or plummet down is a tough task. Using Classic prediction methods isn't a solution as there are lots of factors to be considered upon and each factor having it's own complexity. In financial markets, genetic algorithms are most commonly used to find the combination values of parameters in trading rules. [1]Earlier we decided to use Dow Jones dataset of UC Irvine ML Repository on which genetic algorithm was already used. We had to do lot of pre-processing on that data, and to avoid wastage of time we decided to go with free data provided by Yahoo Stocks. [2]Our Dataset Contained 35 features. We selected the best 20 features out of the 35 features with the help of f_classif function of skicit for classification.

2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

2.1 Dataset Features

Our Dataset had following features: DE Ratio, Trailing P/E, Price/Sales, Price/Book, Profit Margin, Operating Margin, Return on Assets, Return on Equity, Revenue Per Share, Market Cap, Enterprise Value, Forward P/E,PEG Ratio, Enterprise Value/Revenue, Enterprise Value/ EBITDA, Revenue, Gross Profit, EBITDA, Net Income Avl to Common ,Diluted EPS, Earnings Growth, Revenue Growth, Total Cash, Total Cash Per Share, Total Debt, Current Ratio, Book Value Per Share, Cash Flow, Beta, Held by Insiders, Held by Institutions, Shares Short (as of, Short Ratio, Short % of Float, Shares Short (prior).

2.2 Data Pre-Processing

```
X = np.array(data_df[all_features].values)
X = preprocessing.scale(X)
```

Where,

All_features is list of all the features of dataset as mentioned above.

np is Numpy library imported as variable np.

2.2 Random Forest code

We use RandomForestClassifier for performing Random Forest classification on given dataset, where the best result is obtained at 1001 branches and at max depth of 50. And impurity is kept as none. The entire code is available in randomforest.py

```
Rf = RandomForestClassifier
(n_estimators =1001 ,max_depth=50 ,bootstrap = True, oob_score= False, max_leaf_nodes=None, min_impurity_decrease = 0.0, min_impurity_split = None)
```

Building RandomForest Model in randomforest.model file.

```
filename = "randomforest.model"
joblib.dump(rf, filename)
infos.append([rf, "Random Forest"])
```

2.3 Artificial Neural Networks code

We use MLPClassifier for performing Artificial Neural Networks on given dataset, where the best result is obtained by keeping maximum hidden layer as 4000 and stopping the function at 1001th iteration.

```
ne= MLPClassifier(hidden_layer_sizes=4000,max_iter=1001)
```

Saving the Neural Networks Model in neuralnetwork.model file.

```
filename = "neural network.model"
joblib.dump(ne, filename)
infos.append([ne, "Neural Network"])
```

2.4 Support Vector Machine code

We use svm.LinearSVC for performing Support Vector Machine on given dataset, where the best result is obtained by keeping maximum iteration as 5000.

```
svm = svm.LinearSVC(penalty='l2', loss='squared_hinge' ,random_state=None, max_iter=5000).
```

Saving the model for Support Vector Machine for our dataset in SupportVectorMachine.model file.

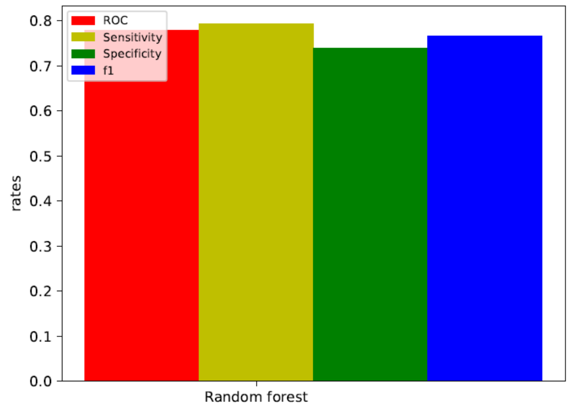
```
filename = "SupportVectorMachine.model"
```

```
joblib.dump(svm, filename)
infos.append([svm, "SupportVectorMachine"])
```

3 RESULTS AND DISCUSSION

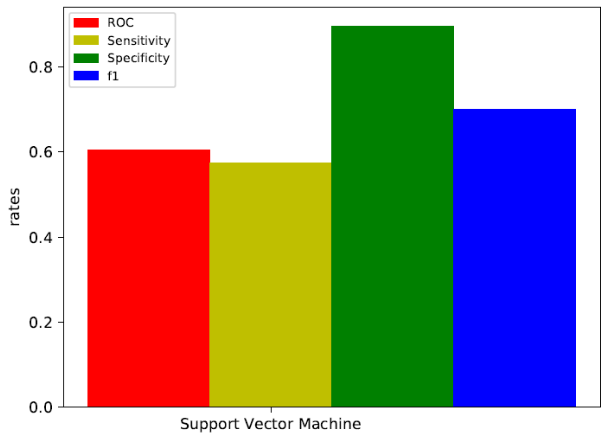
3.1 Performance of Random Forest

Accuracy: 0.758389261745 = 75.838%
Precision: 0.785714285714 = 78.57%
Recall: 0.785714285714 = 78.57%
F1: 0.785714285714 = 78.57%



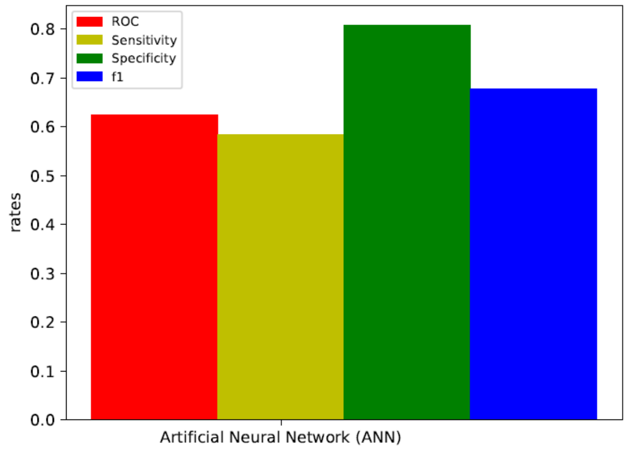
3.2 Performance of SVM

Accuracy:0.624161073826 = 62.41%
Precision:0.627272727273 = 62.72%
Recall:0.821428571429 = 82.14%
F1:0.711340206186s. = 71.13%



3.3 Performance of Neural Networks

Accuracy:0.624161073826 = 62.41%
Precision:0.584158415842 = 58.41%
Recall:0.808219178082 = 80.82%
F1:0.67816091954 = 67.81%



4 CONCLUSIONS

In summary, after performing classification and prediction using three algorithms, we found Random forest to be best performing algorithm for given dataset which was deterministic and kept changing. So we can conclude that Deterministic Datasets where changes are evitable and frequent, Random Forest predicts with higher accuracy.

ACKNOWLEDGMENTS

This work was inspired from work by Michael Scott Brown, Michael J. Pelosi, and Henry Dirska described in Dynamic-Radius Species-Conserving Genetic Algorithm for the Financial Forecasting of Dow Jones Index. And Yahoo Finance for providing the stockmarket dataset.

REFERENCES

[1] Dynamic-Radius Species-Conserving Genetic Algorithm for the Financial Forecasting of Dow Jones Index Stocks Michael Scott Brown, Michael J. Pelosi, and Henry Dirska.
[2] https://finance.yahoo.com/most-active
[3] http://scikit-learn.org/stable/modules/tree.html
[4] http://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.htm
[5] http://scikit-learn.org/stable/modules/svm.html
[6] https://matplotlib.org/
[7] https://scikit-neuralnetwork.readthedocs.io/en/latest/
[8] http://scikit-learn.org/stable/modules/neural_networks_supervised.html