Strategy Learner

Abhishek Chatrath
Computer Science
University of Georgia

Introduction

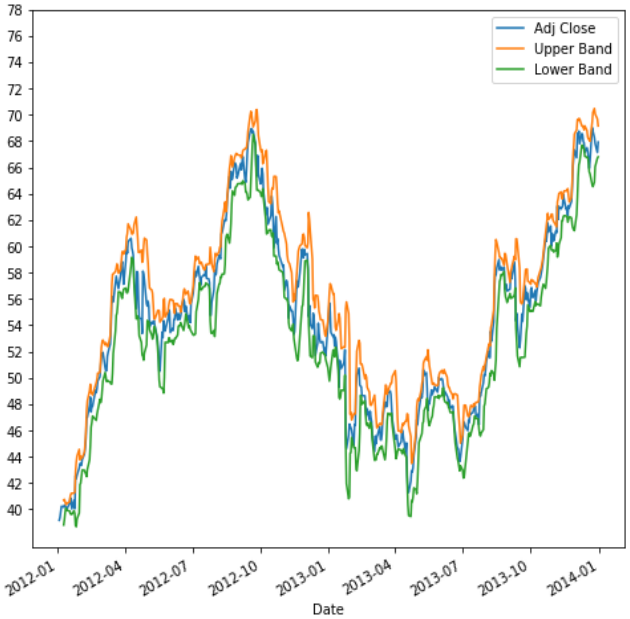
Here, strategy Learner is a machine learning model that trains on a strategy to provide a prediction on what moves to make while trading in the stock market. Our strategy learner presented here uses a basic stock trading strategy to determine whether to SHORT, go LONG or CASH out a stock . The learner uses classification methods in machine learning to train and predict the labels based on the strategy up to 87% accurately.

Dataset

We are using the stock data for IBM from 01-01-2012 to 12-31-2013 from Yahoo Finance for the Learner.

Features

1. **Bollinger Bands** : A Bollinger Band is a technical analysis tool defined by a set of lines plotted two standard deviations (positively and negatively) away from a simple moving average (SMA) of the security's price but can be adjusted to user preferences. Bollinger Bands were developed and copyrighted by famous technical trader John Bollinger.



Bollinger Bands for IBM (01-01-2012 to 12-31-2013)

When using Bollinger Bands, designate the upper and lower bands as price targets. If the price deflects off the lower band and crosses above the 20-day average (the middle line), the upper band comes to represent the upper price target. In a strong uptrend, prices usually fluctuate between the upper band and the 20-day moving average. When that happens, a crossing below the 20-day moving average warns of a trend reversal to the downside.

2. **Momentum** : In terms of stock trading, momentum refers to the phenomenon where the stock which is increasing will keep increasing or a stock decreasing in value will keep on decreasing. Momentum traders focus on acceleration in a stock's price or in a company's earnings or revenues. These traders then take on a long or short position in the stock, in the hopes that the momentum will continue in the same direction. In this way, momentum traders are akin to trend traders, although they tend to rely primarily on short-term movements rather than on fundamental particulars of companies. Momentum trading can be difficult to accomplish successfully, making it a branch of trading that is typically reserved for experienced investors.

3. **Simple Moving Average :** A simple moving average (SMA) is an arithmetic moving average calculated by adding recent closing prices and then dividing that by the number of time periods in the calculation average. A simple, or arithmetic, moving average that is calculated by adding the closing price of the security for a number of time periods and then dividing this total by that same number of periods. Short-term averages respond quickly to changes in the price of the underlying, while long-term averages are slow to react.

LABELS

The classifications are:

- +1: LONG
- 0: CASH
- -1: SHORT

The labels are calculated based on the below algorithm:

```
ret = (price[t+N]/price[t]) - 1.0
if ret > YBUY:
    Y[t] = +1 # LONG
else if ret < YSELL:
    Y[t] = -1 # SHORT
else:
    Y[t] = 0 # CASH
```

YBUY – Threshold for when to go long = 0.6856

YSELL – Threshold for when to short. = - 0.5935

We selected N as 5. For YBUY we selected, mid value between mean N-day return value and Maximum N-day return value, For YSELL we selected, mid value between mean N-day return value and Minimum N-day return value.

Training and Testing Data

The final dataset was divided into Train and Test data in the ratio of 80:20.

Classification Models

1. **Random Forest**

   Random forests, also known as random decision forests, are a popular ensemble method that can be used to build predictive models for both classification and regression problems. Ensemble methods use multiple learning models to gain better predictive results — in the case of a random forest, the model creates an entire forest of random uncorrelated decision trees to arrive at the best possible answer.

   A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement if bootstrap=True (default).

   For our model, we set the number of trees to 100 and minimum sample leaves to 7.

   Accuracy : 0.85149

2. **Logistic Regression**

   Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest.

   Logistic regression is not a regression algorithm but a probabilistic classification model. For our case, we selected all default values and set the multi_class option to ‘auto’.

   Accuracy : 0. 85149

3. **KNN Classifier**

   KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

   For our purpose we selected number of neighbors to be 7.

   Accuracy : 0.87129

4. Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

For our model, we selected kernel as linear and the C-value as default.

Accuracy : 0.85149

5. Naïve Bayes

Naïve Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network.

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. The goal of any probabilistic classifier is, with features x_0 through x_n and classes c_0 through c_k, to determine the probability of the features occurring in each class, and to return the most likely class.

Accuracy : 0.65347

Experiments

To see if we can improve the accuracies, we tried changing the thresholds YBUY and YSELL. For our experiment, we chose two different sets of values, one set was closer to the maximum N-day return and minimum N-day return, and the other set was closer to the mean N-day return.

1) The first value set for YBUY and YSELL was chosen as below

YBUY = ((mean(N_dya_return) + max(N_day_return))/2 + max(N_day_return))/2
YSELL = ((mean(N_dya_return) + min(N_day_return))/2 + min(N_day_return))/2

2) The second value set for YBUY and YSELL was chosen as below

YBUY = ((mean(N_dya_return) + max(N_day_return))/2 + mean(N_day_return))/2
YSELL = ((mean(N_dya_return) + min(N_day_return))/2 + mean(N_day_return))/2

Results

The results of the experiments are shown below:

Mean = 0.00622951555547203						
Max = 0.13089780361690084						
Min = -0.12493044397265274						
		Accuracy				
Y_Buy	**Y_Sell**	**Random Forest**	**Logistic Regression**	**KNN**	**SVM**	**Naïve Bayes**
0.06856	-0.05935	0.85149	0.85149	0.87129	0.85149	0.65347
0.0374	-0.02656	0.55446	0.57426	0.59406	0.56436	0.24752
0.08415	-0.07575	0.9505	0.9505	0.93069	0.9505	0.74257

Conclusion

So, we saw that our strategy learner was able to learn the simple strategy and achieve some excellent results. The threshold when closer to minimum and maximum N-day-return, produce a better result than other chosen values in our experiment.

Future Work

The next step is to use a more complex strategy to train the model and build an orderbook to check against the market to see if our model can earn us some money.

Ethical Consideration

This project can be extended for advance strategic learning for stock trading and a place to start financial analysis and should not be used for any other purpose.

References

1) https://www.investopedia.com/terms/b/bollingerbands.asp
2) https://www.investopedia.com/articles/technical/102201.asp
3) https://www.bollingerbands.com/bollinger-bands
4) https://www.investopedia.com/terms/s/sma.asp
5) https://www.investopedia.com/university/introduction-stock-trader-types/momentum-traders.asp
6) https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
7) https://www.datascience.com/resources/notebooks/random-forest-intro
8) https://acadgild.com/blog/logistic-regression-multiclass-classification
9) https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
10) https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
11) https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72
12) https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7
13) https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
14) https://scikit-learn.org/stable/modules/naive_bayes.html
15) https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54