Stock price trend predictor using machine learning

Aman Danish

1 Domain

According to Efficient Market Hypothesis, it is impossible to achieve consistent risk adjusted returns above the benchmark in the long term. This is based on the assumption that all the underlying information of stock is available to all investors, which is immediately reflect in its price. However, the notion has been challenged since the beginning and with the developments in predictive modelling, has been revisited in recent literature.

In past, Rajshree and Pradipta Dash designed novel decision support sysstems using artificial neural networks to propose set of rules for efficient trading [1]. Jigar Patel and Sahil shah employed Artificial Neural Networks, SVM , Random Forest and Naive Bayes to predict the direction of movement of stocks of Indian capital markets [2]. Jan Ivan used domain knowledge, machine learning and a money management strategy to create substantial profits through simulations on the Oslo Stock Exchange[3]. Lamartine Almeida and Adriano proposed a method of automatic stock trading through technical analysis and nearest neighbor classification[4].

In this project, we will be applying similar machine learning techniques to explore their effects in predicting trends and will try to determine the their efficacy in short-medium timeframe. Also, a comparison of techniques will be performed to evaluate the most accurate approach.

Me, being from financial background was most fascinated about this topic because I was always perplexed in choosing between so many Technical indicators for stock trading, never understanding which of them was most effective. This project gives me a golden opportunity for me to analyze this problem through empirical evidence.

2 Problem Statement

Primary objective: Build a model that takes daily trading data over a certain date range as input, and prints the most likely stock price trend coming within next 10 days. This will be a classification problem for which the predictor variable will be Technical indicators which will be calculated through daily price data. The response variable will be the Price trend in the form of labels {-1 : Bear trend, 0 : No change in trend, 1 : Bullish Trend}

The problem can be formulated as:

$$T_i = f(Momentum_i, Volume_i, Volatility_i)$$

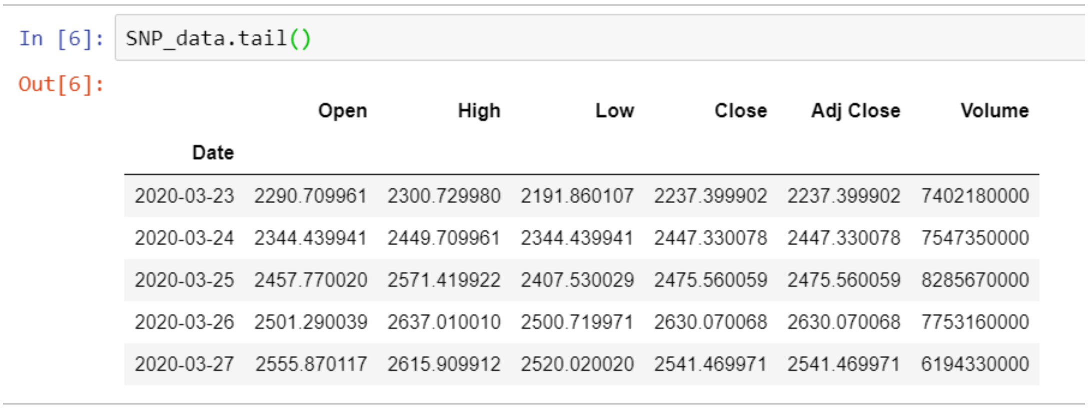
where:
i = a point in time
T = {1,0,-1} Price trends
f = the Classifier function
Momentum,Volume,Volatility are the respective price indicators

Secondary objective: Analyse and determine which factor is most prominent in predicting the price trend for short-medium time period.

3 Input and Dataset

The only data which will be used for this project is last 60-70 years S&P 500 index daily price data. It will be sourced as csv through Federal Reserve Bank's website:'https://fred.stlouisfed.org/series/'. Another possibility to source data is to get access to Quandl API, or API of other websites such as trading economics which require additional subscription.

The input dataset will contain features such as opening price (Open), highest price the stock traded at (High), how many stocks were traded (Volume) and closing price adjusted for stock splits and dividends (Adjusted Close).



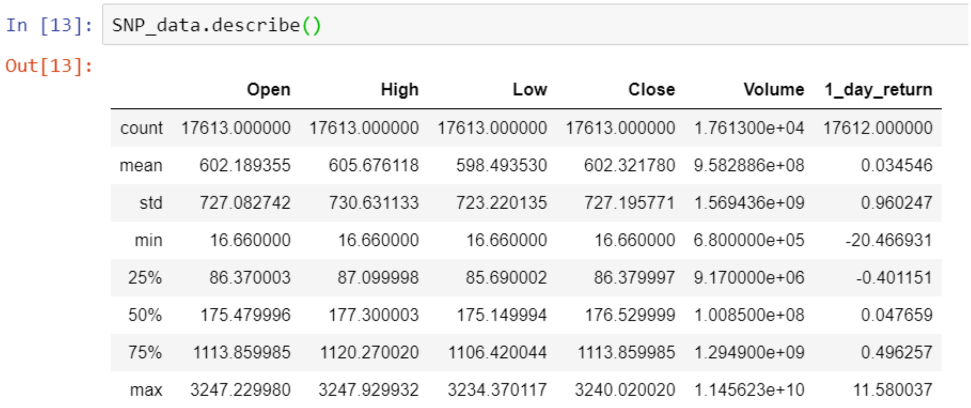
```
In [6]: SNP_data.tail()
Out[6]:
```

Date	Open	High	Low	Close	Adj Close	Volume
2020-03-23	2290.709961	2300.729980	2191.860107	2237.399902	2237.399902	7402180000
2020-03-24	2344.439941	2449.709961	2344.439941	2447.330078	2447.330078	7547350000
2020-03-25	2457.770020	2571.419922	2407.530029	2475.560059	2475.560059	8285670000
2020-03-26	2501.290039	2637.010010	2500.719971	2630.070068	2630.070068	7753160000
2020-03-27	2555.870117	2615.909912	2520.020020	2541.469971	2541.469971	6194330000

Figure 1: Snapshot of input data

Preliminary analysis will be done by adding an additional feature as '1_day_return' which will be daily price % change

Basic statistics of data are as follows:



```
In [13]: SNP_data.describe()
Out[13]:
```

	Open	High	Low	Close	Volume	1_day_return
count	17613.000000	17613.000000	17613.000000	17613.000000	1.761300e+04	17612.000000
mean	602.189355	605.676118	598.493530	602.321780	9.582886e+08	0.034546
std	727.082742	730.631133	723.220135	727.195771	1.569436e+09	0.960247
min	16.660000	16.660000	16.660000	16.660000	6.800000e+05	-20.466931
25%	86.370003	87.099998	85.690002	86.379997	9.170000e+06	-0.401151
50%	175.479996	177.300003	175.149994	176.529999	1.008500e+08	0.047659
75%	1113.859985	1120.270020	1106.420044	1113.859985	1.294900e+09	0.496257
max	3247.229980	3247.929932	3234.370117	3240.020020	1.145623e+10	11.580037

Figure 2: Statistics of features

We can notice from the plot that there are 2 outliers of -20% and 11% during the 2008 financial crisis. These will not be removed in order to retain the realism of data. Also, our features of concern will be dependant on technical indicators and price trend reversal, so the magnitude of returns won't affect our models much. Also, to ensure the outliers do not affect the model, we will also employ regularisation in our classification algorithms.

Though all the calculations will be performed on data for S&P 500 Index, the scope can be extended to other financial product classes or indices/stocks.

4 Solution

We will be developing a system which will determine the upcoming trend based on its technical indicators and print the signal to go long or short. The calculations of its technical indicators will be performed on the system through open source libraries, amongst which most popular in python is TA-lib.

Also, if possible, we will try to determine the threshold values for the technical indicators which lead to most accurate predictions.

5 Benchmark Model

Usually, benchmarking of a sophisticated model is carried out with a model which is primitive and widely used. For classification problems, Logistic regression is the most widely used model for predictive modelling. A hypertuned Logistic Regression model will be used for benchmarking our classifier models.

6 Evaluation Metrics

For Classification problems, there are a broad variety of metrics employed in in assessing the accuracy of a model. Different metrics cater to the problems and inefficiencies of different data.

The most widely used metrics for above purposes are f1 score and accuracy. However, for multi-classification problems with highly imbalanced classes, we would customize our evaluation metric which provides the accuracy of positive classes. We will be more interested in how accurately the model classifies the long or short signals rather than assigning all of them as neutral. To achieve this, we will be using micro f1 score and micro precision, recall as metrics as they will focus more on the less frequent positive labels.

7 Project Design

The project will be divided into two parts:

1. Determine the best model on the basis of smaller size dataset (say 30 years)
2. After finalising the model, train it on larger dataset say (60 years) and determine feature properties which lead to splitting

Following work flow will be followed to analyse the data:

Data Retrieval

Using QuandlAPI or downloaded data, the data will be read into the workbook with parsed dates.

Clean and Explore

Remove NA rows, make line, histogram plot of returns and examine their behavior. Determine if the return trends are correlated with existing features.

Prepare and Transform

Feature engineering will be performed through library TA-lib for technical indicators. Following indicators will be explored to analyse price trend behaviors:

1. RSI
2. SMA 5, 10, 20
3. Parabolic SAR
4. Bollinger Bands

Develop or Choose a model

The problem will be converted from a time series problem to a cross section classification problem. This will be facilitated through features engineered from Technical analysis indicators. For classification problem, following models will be explored and evaluated:

KNN Regression;Random Forest;XGBoost;SVM;Decision Tree Classifier;Logistic Regression (Benchmark)

Train and Tune Model

Data splitting will be performed using the sklearn's test-train split method with stratify parameter set to true. This will ensure the infrequent test labels will evenly get distributed in test and training sets. For hyperparameter tuning, sklearn's Grid search and cross validation will be employed.

Training will be performed on model keeping in mind that there will be highly imbalanced label sets. We will need to use functionalities of the classifiers to deal with this problem.

Evaluate, and finalize a model

The models will be evaluated on the basis of accuracy, micro precision, micro recall and micro f1 score of positive labels. The best model will be trained on a larger dataset and used to generate final results

8 References

1. ”A hybrid stock trading framework integrating technical analysis with machine learning techniques”, Rajashree Dash, Pradipta Kishore Dash

2. ”Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques” Jigar Patel, Sahil Shah, Priyank Thakkar , K Kotecha

3. ”Predicting Stock Prices Using Technical Analysis and Machine Learning” Jan Ivar Larsen”

4. ”A method for automatic stock trading combining technical analysis and nearest neighbor classification” Lamartine Almeida Teixeira a, Adriano Lorena Inácio de Oliveira