Domain Background

Stock market, Forex, cryptocurency trading, etc are some of the options where people can invest money into in order to generate some profit. No matter which option from listed above is chosen, the main idea of the a long-term or a short-term investment remains the same: buy an financial instrument at low price and sell at higher price. Although the main trading principle seems to be quite simple, there are a lot of people who not only didn't make any profit during trading but lost a lot of money. There are many reasons that could lead to such a sad outcome. Making profitable trades in the stock or in any market is a very difficult task because there a huge number of factors that influence the direction of the price such as economic condition of the company(bringing to the market new products or services will strength the company's stock price), political situation inside the company(new CEO can rebuild/ruin the reputation of the company, a corporate reorganization), investors/customers sentiment can be positive or negative(that also affect the price), even economical or political decisions of the government can impact the performance of the company(overseas sells) and many other factors. As a result, the stock markets are subjected to a very quick price change generating random fluctuation and high volatility of the price turning profitable investments into a challenging task. Despite stock's market complexity, hedge funds, investment companies and individuals input a huge amount of time, money and efforts in order to understand the market behavior for placing profitable investments. A big number of fundamental indicators(price-to-earnings ratio, earnings per share, to name a few) which describe economical and financial condition of a company, were invented. I addition to that, a wide range of technical indicators (Relative Strength Index, Moving Average Convergence Divergence to name a few) were invented for better understanding of dynamic of the stock price. Also, with the rise of the social media, it's sentiment analysis plays an important role in making investments based on people's thoughts and feelings about certain company. Gathering, processing and analyzing such huge amount of data seems impossible even for a big group of trained and skilled people. This is where computers come into play. For the last 20 years, computing power and data storage made a huge step forward allowing to gather, store, process and analyze enormous amount of data at a high speed and at a low cost. Cloud technologies provides an easy access to scalable clusters of computers which are affordable for companies and individual researchers. Along with computing power, a big breakthrough was made in the field of machine learning and artificial intelligence. Self-driving cars, computer vision, voice recognition, text translation, medicine(robots-surgeons, disease detection), recommendation systems are few fields where machine learning shines. The investment field was not left aside. Now-days, computers not just placing buy/sell orders. Algorithms are scanning thousands of stocks to find the most promising ones in terms of investment, they perform pattern recognition by scanning historical data, they can predict predict future price or market direction[1,2], they can select the most important indicators out of hundreds available, they can perform trading in automated way (algotrading) without emotions! As a result, companies and individuals take advantage of machine learning capabilities to

builds semi-automated or fully automated trading systems to generate profit for customers or to get a financial freedom.

References:

1. R. Choudhry, K. Garg. A Hybrid Machine Learning System for Stock Market Forecasting. *World Academy of Science, Engineering and Technology,* 15, 2008.

2. L. Khaidem et al. Predicting the direction of stock market prices using random forest. *Applied Mathematical Finance,* 3, 2016.

Problem Statement

The goal of this project is to predict the stock price movement such as up or down for a short-term investments(3-15 days) using machine learning. To tackle this problem two approaches will be taken:

1. Prediction of the stock price movement based on Linear Regression. Open price will serve as input data and Close price(for the same day) is a target which will be predicted. But making a decision about price movement(up or down) based on predicted Close price as continuous output may lead to unfortunate outcome. For example, if at time t the Close price for a stock is $605 and for t+1 is $608 but if the predicted Close prices for t is $602 and for t+1 is $601, then there is a problem with a trend direction although the prediction error is small. Real trend is up($605 < $608) but the predicted trend is down($602 > $601). So, no matter how small the error is in Linear regression, the continuous value is not desired format to succeed in the trend prediction. A way to get this around is to convert the continuous data into categorical one. Regression problem is converted into a classification task. If Price(t+1) > Price(t) then it is an uptrend(label 1), if Price(t+1) < Price(t) then it is a downtrend(label -1). The same procedure applies to the predicted prices. Having both sets of labels an accuracy score can be calculated taking into account the number of matched and mismatched labels in the sets.

2. Prediction of the stock price movement based on Classification. In contrast to the first approach, the classification algorithm will use multiple input features and predict the price trend(based on Close price) for the next day. If Price(t+1) > Price(t) then it is an uptrend(label 1), if Price(t+1) < Price(t) then it is a downtrend(label -1). The input features will be calculated based on Open, Close, High, Low prices, on their ratios, to be precise. In addition some technical indicators will be used as input features too.

After getting the performance metrics for both models trained on the same training and testing dataset, a conclusion can be made on which model predicts the price trend the best.

Datasets and Inputs

Financial data (open, close, high, low prices) will be automatically acquired from Yahoo Finance by a build-in Python functionality. Both machine learning models will use a single stock (“Amazon”) but a user can choose any stock available from Yahoo

Finance. A time period for a training dataset 2010-2015 but a user can choose any time period for training. The predictions will be made within 3-15 days right after the last date in the training set. Some technical indicators will be used as inputs for a classification. All indicators will be calculated with TA-Lib library. The following indicators will be included:

EMA – exponential moving average

SMA – simple moving average

BBANDS – Bollinger Bands

RSI – relative strength index

MACD – moving average convergence divergence

In both cases the prediction of the price trend will be based on the Close price since the price direction is the same for Adjusted Close price(it was tested on several stocks).

Solution Statement

As was mentioned in the Problem Statement section, there are two approaches how to predict the stock price trend. One of them is to use Open price as input variable for a linear regression model to predict values of the Close price. In this particular case, continuous data is not suitable for the trend prediction, so the predicted Close price and the real Close price will be converted into labels depending on the difference between the current price and the next price. If the difference is positive then labels is to set to 1 but if difference is negative - label is set to -1. After converting both sets of prices into labels, an accuracy score is calculated. Another approach is different. It is a classification based on multiple input variables, which include ratios of Open, Close, High, Low prices as well as some technical indicators. The target variable is a label(either 1 or -1) calculated based on the Close price described in the first approach. The accuracy score is calculated in the same manner. Having in hands both scores, it is an easy task to determine a better model(approach). Since the stock price is a time series data, a certain caution should be used when splitting the data into a training and testing datasets. Shuffling prices may lead to undesired result so the data will be splitted preserving the sequential order.

Benchmark Model

There are a couple of options on how to create a benchmark model for the classifiers. One of them is a “majority vote” model which assigns class labels for prediction by taking into account a majority class found in the training set. Another option is to use one of the classifiers used for prediction as a benchmark model.

Evaluation Metrics

Both approaches belong to a classification problem. One of the frequently used metrics for classification is accuracy score which measures a fraction of the classifier's predictions that are correct. The accuracy score can be calculated by the following

formula:

$$\textbf{Accuracy} = \frac{\textbf{True positive + True negative}}{\textbf{Total dataset}}$$

In some cases, when dataset is unbalanced, accuracy is not a good choice as a metric but for this project is it suitable since many stocks have almost 50/50 ups and downs.

Project Design

Work on this project will consists of the following steps:

First approach:

1. Get stock prices(Amazon) from Yahoo Finance and select Open and Close prices. Open price is input feature(x) and Close price is a target variable(y). Split both columns into training and testing datasets(x_train, y_train, x_test, y_test) but preserving the sequence.
2. From sklearn import a regression model. Fit the model with the training datasets(x_train, y_train) and run prediction on x_test to get y_pred. Optional: get R2 score and plot true vs predicted Close prices.
3. Convert continuous values in y_test and y_pred into labels 1 and -1 by subtracting the current price from the next price(Price(t+1) - Price(t)). If the difference is positive - label is 1, if negative - label is -1.
4. Get the accuracy score(built-in function in sklearn). Compare to benchmark model or wait with comparison when second model is done.

Second approach:

1. Get stock prices(Amazon) from Yahoo Finance. Use Open, Close, High, Low prices to generate input features. Calculate some technical indicators and add those values to the input features. Do not forget to drop NaN which were generated during calculations. Convert continuous Close values into labels(see approach 1). Since Close(t+1) is predicted variable, shift Close prices by 1. Split the inputs and target into training and testing datasets preserving the sequence(see approach 1).
2. Select several classifiers(Logistic regression, SVC, Random Forest Classifier) and train the classifiers on the training datasets. Make predictions on x_test.
3. Get an accuracy score for each classifier and compare it among them and then compare with the accuracy score from the first approach.
4. Draw a conclusion about performance of approaches to predict the price trend.