Gabriel Schneider de Jesus
Udacity - Machine Learning Engineer Nanodegree Program

Proposal for Using machine learning and technical analysis on stock prices predictions

25th July 2018

Domain Background

Ever since the creation of the capital market, new tools to predict market prices have been developed in order to maximize profits and reduce losses, whether they are computational or not.

One of these tools, which has undergone several modifications over the years due to the technological advance, is the technical analysis, which consists of analyzing the market exclusively and drawing conclusions from these analyzes.

Another important tool that has grown exponentially with the advent of technology is the negotiating and predictive algorithms, which, through a sequence of instructions, interpret the quotations of an action, generating predictions for the future prices of that action, and may even generate orders in a manner automated.

The objective of this project will be to build a predictive algorithm that uses in addition to the quotations, indicators of technical analysis to generate predictions of the stock market, for this we will use models of machine learning, making use of historical data for the training, testing and evaluating process.

A problem statement

The stock prices prediction is a challenge since the emergence of the stock market, over the years, new models of prediction have been created and new technologies have been developed for this.

This project will have as challenge to obtain historical data of a certain stock, to create some technical analysis indicators and, from their quotations use machine learning techniques on them to predict future closing values.

The datasets and inputs

For data collection we will write a python script, which will collect from Yahoo finance, the daily quotations of a stock of the S&P 500. From this data we will calculate our technical analysis indicators, thus obtaining the inputs for our machine learning models.

A solution statement

According to my knowledge obtained through nanodegree programs and some research, a good solution to the problem would be to use recurrent neural networks as a learning model. This project will be programmed in a Jupyter notebook and script files, for a good organization of the project, easy visualization, understanding and reusability.

Initially we will collect the data, then we will create functions in python to compute the indicators of technical analysis, create a historical model of regression (Support Vector Regression) and a modern model for regression (LSTM Neural Networks), being able to compare the obtained results between the two models, verifying how relevant is the improvement when using a deep learning model.

A benchmark model

For this project, we will use a regression model known as Support Vector Regression, since Support Vector Machines can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. With this model, we will be able to perform performance comparisons with our modern approach denoted above in Solution Statement.

A set of evaluation metrics

In order to measure the performance of our regression models, we will use the mean squared error approach known as MSE, witch evaluates the difference between an estimated value and the actual value.

The MSE is based on the mean square value of the error in the range of patterns presented and can be calculated as follows:

$$MSE = \frac{\sum_{t=1}^{n} (target_t - output_t)^2}{n}$$

At where:

n is number of patterns presented

target is target value

output is output value of the model

An outline of the project design

This project will be divided into five main parts described below:

1. Preparation of dataset

- Get the data
- Add technical analysis information to the data
- Normalize the data
- Divide our data set into training and testing
- Visualize our preprocessing results

2. Benchmark Model

- Create an SVR as Benchmark Model
- Train the benchmark model
- Evaluate the model
- Tune the model
- Evaluate the tuned model
- Visualize the results of your predictions graphically

3. LSTM Model

- Create our LSTM Neural Network Model
- Train the model
- Evaluate the LSTM model
- Tune the model
- Evaluate the tuned model
- Visualize the results of your predictions graphically

4. Comparing the results

- Compare and analyze the results obtained between the models
- Create a discussion about improvements that future projects can implement on this project.

5. Report

- Generate a report describing the results obtained during the project construction process

NOTES:

1. The report will not be contained in the jupyter notebook, there will be a pdf document for it.
2. The libraries will be incorporated into the topic that requires their use, for a better understanding of the project.
3. The final project may undergo some minor variations as it has not yet been implemented.

References

https://en.wikipedia.org/wiki/Support_vector_machine

https://cmtassociation.org/kb/history-of-technical-analysis/

https://docs.microsoft.com/es-es/azure/machine-learning/studio/algorithm-choice

REDES NEURAIS ARTIFICIAIS PARA PREDIÇÃO NO MERCADO ACIONÁRIO BRASILEIRO - Fabio Maritan Pereira