Machine Learning Engineer Nano-degree Capstone Proposal

Akhil Sethia
October 26th, 2017

Domain Background

Finance and Investment

Predicting equity prices in the financial markets is a widely used application of machine learning. Financial models have been historically studied with different statistical methods and simulations. With the inception of machine learning, major advancements have been made to use regression modelling techniques to predict the changing equity prices. Readily available historical price data for equity is available on the internet. Using these price trends and adding relevant other technical indicators to the price information, we can produce an input which is can be used for regression.

Historically, traders and hedge funds created strategies, to find of points of entries in the stock market, post which the stock prices would rise. This is also used to short and sell stocks to make profits. Advanced statistical methods were previously used for the same.

I personally aspire to pursue research in predicting equity prices using machine learning as a career option. Additionally, I have already worked on a project which uses linear regression for the same problem. The results were motivating, but not accurate enough. I invest as a day trader and wish to find better and more systematic investment patterns.

Problem Statement

Prediction of stock prices listed on the Standard & Poor's 500 (S&P 500) benchmark. Taking as input the current market variables like adjusted close and volume, applying transformations to the current variables like finding daily returns, Bollinger bands and RSI index from them and using this to producing acceptable prediction results. Machine learning algorithms, can self learn the underlying trend between each derived input variable and find strategies for predicting future prices.

Datasets and Inputs

A time series dataset, with historical market close prices, volume, high-lows and open-close attribute. We will be considering the S&P500 itself. We will be using this index stock, as it is widely traded by a lot of entities. Because of trades in large volume, the aberrations in stock and unwarranted volatility is low. The outlier trades in the stock are

also low, and hence it makes it easier for our model to detect actual underlying pattern within the prices.

The input for the dataset would be obtained using open sourced data available on Yahoo Finance. The dataset post obtaining will be cleaned. All N.A. values will be dropped from the table, and scaling and pre-processing for all attributes will be performed. To this dataset a higher dimensionality will be added, but introducing derived information in the dataset. These feature will further introduce different dimensions like volatility and price movement channel in the stock. These parameters will be of greater effect in prediction.

The entire duration of testing, training and validation would be performed once when the stock is following a particular momentum and once when the momentum is reversing. This would help us determine whether out model, can predict important reversals, and can be used in real time. This system has sequential data, and the records will be ordered by the date of the trading session. The training and testing data picked will be a contiguous batch from our dataset. There will be little randomisation in choosing testing and training data to maintain sequentiality.

Solution Statement

We would be using a regression model, employing a Recurrent Neural Network with LSTM. Our raw data will first be subject to feature selection and transformation using PCA, ICA or any other feature extraction and selection practise commonly used. The result of this selection will be fed into our model, where we will predict *'Close'* price of a stock after 5 trading session. An activation function of 'Rectified Linear' unit will be used so that we can get regression results.

Benchmark Model

I would be comparing my model with currently in use models for trading. A SVM model trained with the same feature selected model will be used for comparing. Also we will be sing conventionally used models like ARIMA and GARCH comparison. We can also use a multi-layer perceptron model for the same purpose. Amongst these 4 models, 2 models with the highest performance in the specified metrics will be chosen for comparison.

These models will be developed on the same training set as our model, and then their performance on the testing dataset will be recorded. This will be used compare the validity of our model, with the available industry alternatives. Accuracy in price predictions will be used to compare these models.

Evaluation Metrics

Regression models are used here for predicting future prices. Hence, regression metrics will be employed. The two important metrics used will RMS error and R^2 score of the model.

The RMSE metric will be the summation of the mean squared error of the model, which show much do the model predictions differ from the actual predictions. This is an important metric, which will decide our accuracy.

The R^2 score is a model which will quantify, that how much better does our model perform, as compared to simple chance. The higher this score, shows that our model has greater performance.

The Hit Rate will be used to predict the number of times, the value of our model matches with the actual value.

Total Return is an important metric to quantify how well does our model perform. This metric will also be used to make comparisons with between different benchmark models. This will be used as a primary measure for determining the goodness of the model

These 4 measures will be used to determine the validity and the relative performance of the model. When making financial decisions, the accuracy of the model used needs to be extremely high. We would ideally want to always predict the same price with very little deviation. Hence, we would tune our model to output minimum mean squared error, maximum hit ration and R^2 score. Our total return is a very important metric as it would determine success of our model in making money. These are can be instances when our hit ratio is low, because we cannot predict exact prices, but our model gives a high return. A model with a high return is accepted, despite it having lower performances in other metrics. Although, a model, may have performed exceedingly well at one instance, but poorly on other trades. This model may have a high return, but because it has repeatedly not done well, we will discard it. A correct balance between a model providing high return and high reliability needs to be chosen. These metrics will aid this selection.

Project Design

The project has specific segments win which it has been divided. The specific segments are:

1) Raw data collection - I will be collecting data from Yahoo Finance. The OHLCV data will be available to me. Post, collection I would be performing transformations on the data, and finding out various technical indicators and adding them to the raw data. This data combined with the original raw features and the new technical indicators will comprise of the entire dataset.
2) Data pre-processing - The dataset generated in the previous section, will be sent for pre-processing and scaling. Box-cox transformations will be used to transform the feature data and scale it. All ‘na’ values and null rows and columns will be removed from the data, and it will be made ready for being fed into the model.
3) Feature extraction & Feature selection from the data - At this stage, we will run feature selection and feature extraction algorithms and choose the most impactful 7 features from our feature pool.
4) Dividing the data-set - The dataset will then be divided into training and testing set. We will use hold out cross validation on the training set. The training to testing split will be a 0.7-0.3 split and we would leave out 20% of our training set for cross-validation.
5) Training and building the model - At this stage we will be building our RNN model, performing hyper-parameter tuning on it. From various possible combinations possible for our model, we will be choosing the model with the highest validation loss on our validation set.
6) Evaluating the model performance - We would test the model on testing data and see the performance results based on the specific performance parameters.
7) Training and finding performance results of the benchmark model - The benchmark model will be trained and its performance will be evaluated based on the given parameters.
8) Comparing our model with the benchmark model, and assessing its strengths and shortcomings - Here will provide an assessment of the results of our model, and its strong and weak points as compared to the other models. We will see the points and res where our model does better than the other models, and analyse where is performs poorly.
9) Concluding on the usability of the model in comparison with all pre-existing models used in the industry - Based on its performance in various performance metrics and the risk involved in trading stocks, we will conclude whether our model is developed enough to be used in real-time. We will also try to analyse future improvements that can be made to the model, so that it can be effectively used in the financial markets.