Machine Learning Engineer Nanodegree

Capstone Proposal

Saurabh Kumar
22nd Dec 2017

Proposal

Domain Background

A Stock market is a place where humans and computers buy and sell shares of companies. Shares are small pieces of a company. There is a lot of money involved. Mathematicians and statisticians have been always interested in Stock Market given the amount of money involved. To predict the market behavior has always allured many. People try to find patterns in the behavior of stock market. However, given the huge amount of data and buy/sell decisions carried out every day makes it almost impossible to analyze the stock market manually.

Investment firms, hedge funds and even individuals have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process. In the recent few years, the emergence of organized data, high computational power and machine learning algorithms have made it a bit easier to predict the behavior of stock market

Problem Statement

The problem statement for me here is that, is it possible to apply regression machine learning techniques and predict the market trend. Very honestly, just plotting the prediction of market movement and verifying how close I am to actual.

Datasets and Inputs

I would be using historical data from S&P. The Standard & Poor's 500, often abbreviated as the S&P 500, or just the S&P, is an American stock market index based on the market capitalizations of 500 large companies having common stock listed on the NYSE or NASDAQ.

I have downloaded the data using a link [1].

I found this link from a stack exchange discussion [2].

Inputs in the dataset

Date – Date of the day

Open – Opening market price of the day

High – Highest market price of the day

Low – Lowest market price of the day

Close – Closing market price of the day

Volume – Number of transaction on the given day

Solution Statement

My solution statement is to use supervised machine learning [5] techniques and predict the highest market price of the day.

For this I would be using following Machine Learning Models but this list is not exhaustive

a) **Gradientboost [6]**
b) **Liner Regression [7]**
c) **KNeighboursRegressor[8]**

Benchmark Model

The problem statement has featured in Kaggle competition and also there many existing approaches for the given problem statement. State of the Art here will be my approach using different regression models. The benchmark model would be to compete with Kaggle scoreboard and see how the model performs, the higher the better. I want to keep my approach simple and using one model as Benchmark to compare other model that I use as far as the resources available for computation allow us to proceed.

Evaluation Metrics

I would be using using evaluation metrics for regression like 'explained_variance' and 'r2' score.

Explained Variance: [3]

If $\hat{y}$ is the estimated target output, y the corresponding (correct) target output, and Var is Variance, the square of the standard deviation, then the explained variance is estimated as follow:

$$\texttt{explained\_variance}(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

The best possible score is 1.0, lower values are worse.

R2 Score:[4]

The `r2_score` function computes R², the coefficient of determination. It provides a measure of how well future samples are likely to be predicted by the model. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the score R² estimated over n_{samples} is defined as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \bar{y})^2}$$

where $\bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i$.

Project Design

Libraries:

Scikit-learn

Workflow:

o **Data Collection:** This step involves collection of labelled data for the training and testing process.

o **Pre-processing:** It is an important step in the data mining process. The phrase “garbage in, garbage out” is particularly applicable to data mining and machine learning projects. It is an umbrella term that covers an array of operations to get data into a form more appropriate. Analysing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis. Before performing sentiment analysis of movie review data, we strip out any html tags, white spaces, expand abbreviations and split the reviews into lists of the words they contain.

o **Model Training**: Training a model involves providing the algorithm training data to learn from. In this case, architecture of such an algorithm/model (based on neural networks) must be defined. In the earlier stages, the model will have a simple structure which with trials will evolve to perform better in each iteration. The training data must contain the target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target and it outputs a model that captures these patterns.

o **Model Validation**: The trained model is evaluated with a testing data set. The testing data set is a separate portion of the same data set from which the training set is derived. The main purpose of using the testing data set is to test the generalization ability of a trained model.

Refrences

1. https://stooq.com/q/d/l/?s=^spx&i=d

2. https://quant.stackexchange.com/questions/26078/how-can-one-query-the-google-finance-api-for-dow-jones-and-sp-500-values/26080

3. http://scikit-learn.org/stable/modules/model_evaluation.html#explained-variance-score

4. http://scikit-learn.org/stable/modules/model_evaluation.html#r2-score

5. https://en.wikipedia.org/wiki/Supervised_learning

6. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

7. http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

8. http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html