Machine Learning Engineer Nanodegree

Capstone Proposal

Eoin Cunning Febuary 6th, 2019

I propse basing my capstone project around a Kaggle competition, New York Stock Exchange. This is my first experience on Kaggle and hope to appy some of domain knowledge as a developer in finance with technical knowlege from ml course.

Details available at: https://www.kaggle.com/dgawlik/nyse

Domain Background

The project is provides finicial data for NYSE listed stocks. And one of the challenges laid out is to attempt "One day ahead prediction: Rolling Linear Regression, ARIMA, Neural Networks, LSTM".

Traditionally stock trading has been performed by humans but with all the unseen un-quantifible decsions that go into human decision making the idea here is to replace that human element with machine learning. Can a machine given historical tick data make predictions on futures movement and therefore recommend cetrain stocks to buy. Wall steet has been attempting this same problem in various forms with various degrees of sucsess for years now. We know this problem is solveable as there are multiple papers written on the subject from using SVMs to model the Indian Stock Market[1] to using LSTM to predict stock price movement [2].

I have selected this problem as it gives me the opportunity to join a Kaggle competition, interact with other machine learning practicioners, and learning from them and also emparting some of my knowledge by writing public available code with Kernels.

Problem Statement

The Problem here is to take historical time series tick data. Engieer it into a form where we associate features being previous prices and predict the next day value of the same prices. This can be a regression or classification problem as we can attempt to predict the exact value of the next day price (regression) or more simply we could try and output a trade signal that would be true or false depending on what we predict the behaviour to be. (classification)

Datasets and Inputs

The source of the data we will be using for this problem is a Kaggle dataset. From Kaggles description of the data.

Dataset consists of following files:

- prices.csv: raw, as-is daily prices. Most of data spans from 2010 to the end 2016, for companies new on stock market date range is shorter. There have been approx. 140 stock splits in that time, this set doesn't account for that.
- prices-split-adjusted.csv: same as prices, but there have been added adjustments for splits. securities.csv: general description of each company with division on sectors
- fundamentals.csv: metrics extracted from annual SEC 10K fillings (2012-2016), should be enough to derive most of popular fundamental indicators.

We have 851264 records of prices and asjusted prices and 505 different companies with descriptions. This price data acts as both our inputs and outs as it is both the prices are both the historical data we need ot feed into our algorithm but also the futures values we are trying to predict for the previous timeseries data.

What will be key is to arragne to data to try and associate historical data with future data one potential method could be to use TimeSeriesSplit. As demonstrated in this code snippet on the data: ```

Load the Pandas libraries with alias 'pd'

... import pandas as pd

read in the data parse date values

... data = pd.read_csv("prices-split-adjusted.csv", parse_dates=['date'])

search for GOOG symbol data using date as the index

... data = data[data['symbol'] == "GOOG"]

from sklearn.model_selection import TimeSeriesSplit

get the first set of prices as a array

... prices=data['close'][:10].values

set splits to 5

... tcsv = TimeSeriesSplit(n_splits=5) for train_index,test_index in tcsv.split(prices): ... X_train, X_test = prices[train_index], prices[test_index] ... print("TRAIN:",X_train, "TEST:", X_test) ... TRAIN: [312.20530836 310.83045863 302.99481256 295.94124207 299.88646989] TEST: [299.433161] TRAIN: [312.20530836 310.83045863 302.99481256 295.94124207 299.88646989 299.433161] TEST: [294.1380165] TRAIN: [312.20530836 310.83045863 302.99481256 295.94124207 299.88646989 299.433161 294.1380165] TEST: [292.44932364] TRAIN: [312.20530836 310.83045863 302.99481256 295.94124207 299.88646989 299.433161 294.1380165 292.44932364] TEST: [293.82417336] TRAIN: [312.20530836 310.83045863 302.99481256 295.94124207 299.88646989 299.433161 294.1380165 292.44932364 293.82417336] TEST: [288.9175486] ``` Here we can see we have arranged the data such that there is a test value associated with all the previous historical values.

Solution Statement

A solution to his problem would been a trained model that given historical tick data could either sucseffully predict and up and down price movent in the given stock. Or predict the next day value to within a decided margin of error. To be considered a solution to this problem it must out perform a benchmark model of our choosing.

Benchmark Model

If we impose the constraints that prices must move in ticks and must move up or down. Then theoricatilly the very worst we model we ccould come up with would randomly guess up or down based on previous price movement. This lends itself well to a Random Forest Regressor so that is what we will use as the benchmark to compare our trained and refined model.

Evaluation Metrics

The evaluation metric we have selected for this problem to quantify its performance is the R-square and root-mean-squared-error. As we will be atempting to calculate the exact price of the stack and have an exact value to compare this to we can easily calculate the square root of the average of sqaure errors. Formula:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2}$$

F1_score Formula:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

We could also use a f-score to evaluate our model if all we considered was if we predicted correctly the correct direction of the price movement.

Project Design

Firstly we need ot analyse the data and pull out what information we feel is relavent. Answer question such as are the high and low relavent or only the Close price. we can do this through data exploration. And while doing so look for outliers in the data and consider if any of these should be excluded from training our model.

Once we have done this analysis and potentially filtered our data set we will need to do some feature engineering to get ther data into a state to feed into our chossen model/models. Here is where we will consider splitting our data into testing and trainging data and cross validation.

Next we will need to select a model that is sutible for this problem using the sklearn documention[4] as a guide to consider our various options regards algorithmns.

Considering first that we are trying to predict a quantiy and not a category with our goal to correctly predict as close to the correct next price as possible we can first consider this as regression problem. We will not have as much as 100K samples of data so our first candidate for a mode could be a SGD regressor. Based on the literature surround this problem a SVM could be reasonable approach. We may suggest trying to traing multiple models and then combining them with a method such a boosting or bagging.

Once we have a trained model we run both it and our benchmark algorithm against our test data. Present and preform analysis the results and try to draw a conclusion of whether we have sucsefully solved the problem we have layed out.

References

[1] Support Vector Machines for Prediction of Futures Prices in Indian Stock Market - Shom Prasad Das, Sudarsan Padhy

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.969&rep=rep1&type=pdf

[2] Predicting Stock Prices Using LSTM - Murtaza Roondiwala, Harshal Patel, Shraddha Varma

https://www.ijsr.net/archive/v6i4/ART20172755.pdf

[4] https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html