rohitsinha84 / **capstone**

Branch: master ▾ capstone / README.md Find file Copy path

**rohitsinha84** Update README.md b91a9ab a minute ago

1 contributor

91 lines (48 sloc) 4.75 KB

# Machine Learning Engineer Nanodegree

## Capstone Proposal

Rohit Sinha
November 5th, 2017

## Proposal

### Domain Background

The capstone project is to build a stock price indicator using Python. The investment and trading industry makes use of different tools to predict stock prices. Technical analysis is one of the means to evaluate stock prices using past prices and volume. However, technical analysis makes use of patterns and trends to predict stock prices which can be too simplistic or naive. We propose to build a more sophisticated model which uses daily prices and volume data to create factors to predict the adjusted closing prices.

Historical stock prices for a company can be easily be downloaded using multiple freely available APIs.

Numerous academic research have already been carried out in this field. Some of the relevant papers related to our project is listed below:

1. http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf
2. http://cs229.stanford.edu/proj2015/009_report.pdf
3. http://cs229.stanford.edu/proj2012/ShenJiangZhang-StockMarketForecastingusingMachineLearningAlgorithms.pdf

The papers discussed above make use of conventional machine learning models as well as deep learning to predict stock prices.

I have more than 7 years of experience in investment banking industry working as a researcher. I want to use the capstone project to apply the machine learning techniques learned in the nanodegree and use them in my daily research.

### Problem Statement

Use supervised learning as well as deep learning methods to predict stock prices. We create multiple features using the daily stock prices and volume data to predict the adjusted closing prices. We intend to create separate models for different companies (e.g. GOGGLE, FACEBOOK, TESLA) and check the efficiency of the model using either RMSE or R2-Score.

### Datasets and Inputs

Financial time series data using Quandl. https://www.quandl.com/

The time series data has different features which includes OPEN Price,HIGH Price,LOW Price,LAST Price,CLOSE Price,TOTAL TRADE QUANTITY,TURNOVER. We use the existing features to create new features like weekly return, daily return, 6 month average return etc. The time series of features is used to train a model to predict future out of sample test returns.

We will use daily time series data for at least last 10 years for Google, Facebook and Tesla.

## Solution Statement

The study aims to find a best model to predict the future stock price. We start with feature evaluation which includes feature creation as well as feature selection for our study. We train our model using conventional supervised learning methods like SVM, Decision Tree Regressor, Ensemble methods to predict future price. Finally we use deep learning method to predict the prices and compare our results over the same accuracy metrics.

The timeseries data will be split into train and test data sets using the TimeSeriesSplit module from SKlearn. Training data from past will be used to predict prices over future. For e.g. data from 2010-2016 will be used tp predict prices in 2017. The timeseries data will be checked for autocorrelation and stationarity to pick the best set of features.

We plan to use the LSTM deep learning model to predict the stock price.

## Benchmark Model

The benchmark model will be the supervised model (e.g. SVM, Decision Tree Regressor or ADABoost) which performs best in predicting the stock prices. It will then be compared against deep learning models to find if deep learning models can do a better job at predicting prices.

## Evaluation Metrics

Evaluation metrics used in our study will be RMSE (root mean square) and R2 (R-Sqaure).

## Project Design

1. Download Data using Quandl

   Quandl has an excel and python API.

2. Feature generation

   The time series data is used to get rolling and lag metrics of price and volume.

3. Feature selection

   The features generated using rolling and lag metrics are tested statistically.

   The features are checked for autocorrelation and stationarity and then selected.

4. Create different supervised learning model for a selected stock

   Run SVM, ADABoost and Decision Tree Regressor models

5. Select the best supervised learning model

   Select the best supervised model from different options available

6. Create different deep learning model for a selected stock

   Use LSTM model with different combination of layers

7. Select the best deep learning model

   Select the best performing deep learning model

8. Compare results of conventional supervised learning model and deep learning model

9. Repeat the study for other stocks

   Compare which supervised learning model and deep learning model works best for each stock.