MSc Computing & Information Systems

Masters Project Outline/Specification

Theodore Sanfred Hinanto

Supervisor: Nikos Tzevelekos

Machine learning implementation in trading stock and forex with RESTful API application model

# Introduction & Background

Trading has been around for more than a century as the stock market has been established before 19th century. Today, with the advancement in technology, buying and selling securities are mostly done electronically. People with internet access can easily buy stocks from hundreds of stocks brokers.

Machine learning was firstly emphasized in the discovery of Bayes Theorem in 18th century, however, it has only been several years since deep learning, which is part of the machine learning family, has become more feasible that machine learning has now become an integral part of many applications.

API was created with the purpose of free-for-all environment in late 90’s with the current public APIs has reached the number of thousands. Implementation of API according to RESTful standard has been deemed as simple to use.

# Overview

The system will predict stock and forex in real-time based of 1day timeframe on machine learning and manual algorithm on historical data derived from API. The result then will be visualised in a website format along with the capability of the user to save his configuration, for example ‘favorite’ stock/forex in system database.

# Objective

The main objective is to implement machine learning method to predict the outcome of a stock or forex which is either going up or down in the D1 timeframe.

Other goals in the project:

* Implementing different supervised learning methods along with different random sets of training, validation, and test data
* Besides machine learning, a set of strategies based on several technical trading indicators also will be put in the algorithm
* Use Database system to store user information (e.g. user ‘saved’ stock)
* Visualisation in a form of website (HTML) with the optional goal of using cloud services

# Design and Timeline

The steps to be done and the expected deadline **[DD/M]** will be as follow:

1. Collecting data which is from Alpha Vintage API to get real-time and historical stock and forex data **[17/6]**
2. Clean and pre-process the data using Python in Jupyter Notebook **[17/6]**
3. Split data and apply data analytic using machine learning libraries such as numpy, pandas, scikit-learn, matplotlib, and possibly tensorflow **[20/6]**

The data will be split as training (60-70%), validation (15-20%), and test (15-20%)

1. Try 3 supervised learning methods, each with 2-3 different randomised sets **[27/6]**
2. Linear Regression
3. Extreme Gradient Boosting (XGBoost)
4. Long Short Term Memory (LSTM) or K Nearest Neighbor (KNN)
5. Analyse the effectiveness of the results using root mean square error (RMSE) and mean absolute percentage error (MAPE) **[29/6]**
6. Using python, create 2 manual algorithms to predict the trading outcome that is not based on machine learning (for sample and comparison purposes), for example algorithm based on moving average, relative strength index, etc. **[3/7]**
7. 2 Exponential Moving Average (EMA) strategy

A period of 10 and 20 of EMA, when 10EMA is crossing up the 20EMA, a buy trend. Similarly, when 10EMA is crossing down the 20EMA, a sell trend.

1. Relative Strength Index (RSI) strategy

Using 14 period RSI, when the RSI > 75, it is overbought. Similarly, when the RSI < 25, it is oversold.

1. Draw conclusions of all the results gained so far possibly with visualisation **[5/7]**

Choosing the best supervised learning method result to be used as a result to be deployed in the web app, saving the ML result using pickle or save into RDBMS

1. Create the Flask python app and the website for visualisation of the result using HTML/JS/CSS/Django/Jinja2 to integrate with the main python code **[10/7]**

\*Use jupyter extension to visualise the system

The input ‘date’ value will be the current date when the user accesses the website and the value will be passed into the python code that will be used for step 1-6

Using stocker (<https://github.com/WillKoehrsen/Data-Analysis/tree/master/stocker>) as a stock analysis toolkit to create a simulation of buy/sell strategy

1. Create SQL Database using sqlite as it is built into Python or another alternative RDBMS. **[15/7]**

Main purpose for the use of SQL database is to store user information: username, password, ‘saved’ stock/forex

1. Systems integration of the REST API, Python main code, Database, and web app. **[20/7]**
2. Optimization, testing, and conclusion **[ 22/7]**
3. Draft report **[29/7]**

* Use appmode extension

<https://github.com/oschuett/appmode>

* LSTM github

<https://github.com/thushv89/datacamp_tutorials/blob/master/Reviewed/lstm_stock_market_prediction.ipynb>

<https://blog.usejournal.com/stock-market-prediction-by-recurrent-neural-network-on-lstm-model-56de700bff68>

<https://github.com/sebastianheinz/stockprediction>

<https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/>

* Use external library for technical indicator calculation

<https://github.com/mrjbq7/ta-lib>

* ARIMA

<https://robotwealth.com/fitting-time-series-models-to-the-forex-market-are-arimagarch-predictions-profitable/>

* Github guide

<https://github.com/NGYB/Stocks/blob/master/StockPricePrediction/StockPricePrediction_v2_lin_reg.ipynb>