Capstone Project Proposal: Stock Price Predictor

Domain Background

As the knowledge and techniques surrounding machine learning increase, the interest in applying this knowledge to stock data for making predictions is growing as well. The stock market is a composition of buyers and sellers of stocks, which are units for representing partial ownership of a company. These stocks have a specified price which can vary each day and time, and it is affected by unpredictable factors such as politics, social trends, the environment, and company-related events.

Stock data is information that represents the movement of stock prices for given companies (or market indexes such as S&P 500) for each day that the stock market operates. It usually has 7 main data fields per day:

- Date: the date of the stock data for that day
- Open: the price of the first stock transaction made after market opens
- High: the highest price of the stock
- Low: the lowest price of the stock
- Close: the price of the first stock transaction made before market closes
- Volume: the number of stocks traded that day
- Adjusted Close: the closing price of a stock after considering corporate actions

Based on this data and derived data, financial analysts can make predictions for the direction the stock prices will take and, therefore, make decisions on buying and selling stocks with the lowest possible risk.

In stock market theory, it is said that the market follows the Efficient Markets Hypothesis, which states that the market "follows a random walk and can be unpredictable based on historical data" (Madge, 2015). This holds true for stock predictions in the short term, however, it is possible to find patterns in stock data in long periods of time, which in turn means that there is a degree of predictableness in the stock market.

I picked this topic because of the many possibilities of training machine learning algorithms with stock market data and its many indexes. I believe it is possible for an algorithm to make better predictions than a human mind because of the way it can find patterns in data more efficiently.

Problem Statement

For stock traders, market predictions are like a compass pointing treasure; money can be made from them. Therefore, development of models that can take in stock data and return predictions is a thriving force in the artificial intelligence field. More specifically, knowing whether the price of a stock will rise (or fall) is profitable information. Given the fact that the stock market is not 100% unpredictable, the possibility for profiting from its historical data exists. Therefore, analyzing stock data with artificial intelligence is a possible solution for this problem, and a model's performance can be measured based on the accuracy of the prediction against the real-world data.

Datasets and Inputs

The datasets used for this project are the historical stock data from all companies that are listed in the NASDAQ-100 index, plus the historical data of the index itself. This choice of datasets was derived from Madge's report for maintaining coherence for later benchmark comparisons. Given a ticker symbol, a retrieved dataset for that company's stocks follows this structure:

		Adjusted Close	**close**	**high**	**low**	**open**	**volume**
symbol	**date**						
GOOG	**2014-10-01 00:00:00+00:00**	568.27	568.27	577.5799	567.0100	576.01	1445027
	2014-10-02 00:00:00+00:00	570.08	570.08	571.9100	563.3200	567.31	1175307
	2014-10-03 00:00:00+00:00	575.28	575.28	577.2250	572.5000	573.05	1138636

Each row in this table represents stock data for a given date.

This information is retrieved through an API from https://www.tiingo.com using a free account. The NASDAQ-100 index information follows the same structure and it can be retrieved freely from https://www.investing.com/indices/nq-100-historical-data. Since Tiingo limits data retrieval by up to 5 years before today for free accounts, this timespan will be used for all retrieved data even though it is not the same timespan used by Madge.

Different ranges of periods will be used to create different input sets as proposed by Madge. The input features to be extracted and explored for model training are the following:

- Price Momentum Oscillator = TC – PPC
  - TC: today's close
  - PPC: previous period's close
- Relative Strength Index = 100 – [100/(1 + RS)]
  - RS: average of x days up-closes divided by average of x days down-closes
- Money Flow Index = 100 *(100/(1 + MR))
  - MR = (PositiveMF / NegativeMF)
  - MF = TP * Volume
  - TP: average of high, low, and close prices for a given period. If the current Typical Price is greater than the previous period's, it is considered Positive Money Flow.
- Exponential Moving Average = [α * TC] + [(1 – α) * YEMA]

    - TC: today's close
    - YEMA: yesterday's exponential moving average
    - α: smoothing factor which is 2/(n+1) where n is the number of days in the period.
- Stochastic Oscillator = [(CP - LP) / (HP - LP)]*100
    - CP: closing price
    - LP: lowest low price in the period
    - HP: highest high price in the period
- Moving Average Convergence/Divergence = (12-day EMA) – (26-day EMA)

The moving Average Convergence/Divergence is the only calculated feature that does not depend on the selected period. These features were proposed by Abdul Salam, Emary, and Zawbaa (2018) in their conference paper for their financial forecasting model.

Solution Statement

An artificial neural network (ANN) will be implemented and trained on the mentioned datasets. This ANN will be trained with different datasets, which will be obtained by applying different combinations of periods for both the stock data and the index data, as well as the time range for the prediction output. The whole dataset will contain stock data for a specific group of companies split in two groups: the training set which will be the data from 2014 to the end of 2017, and the testing set which is stock data from 2018 to 2019. The trained ANN will yield a result that tells whether the stock's price will go up or down in its set prediction period.

Benchmark Model

The model to be used to compare this project's model will be the one implemented by Saahil Madge in their report referenced here. Since I will use data from the same industrial sector and the combination of periods for the different datasets will be the same as well, the results that my model obtains will be coherently measurable against Madge's model's results.

Evaluation Metrics

The model will be evaluated by its accuracy, that is, the percentage of times that the model predicted a price movement correctly on the testing dataset. Since there will be various models trained on different data, various accuracies will be obtained. These accuracies will be compared to the accuracy of the respective benchmark model for each combination of periods in the datasets.

Project Design

The project will be implemented using Python, specifically using Pytorch for defining the feedforward model. The workflow to follow is:

1. Data retrieval and cleaning

   - The data from the NASDAQ-100 index and all other companies will be downloaded from the specified sources and converted to Pandas DataFrames. Each row will be identified by its date.
   - The data will be cleaned by removing records with missing information. Additionally, index and stock data will be trimmed so that they only contain records with the same dates.

2. Feature calculation
   - For each date, the input features are calculated using all different combinations of time periods n = (5, 10, 20, 90, 270 days previous) for both index and stock data.
   - For each obtained dataset their labels are assigned. I'm using different timespans for predicted values m = (1, 5, 10, 20, 90, 270 days after), so each data record's label corresponds to either 1 for a price increase or 0 for decrease after m days. By now there will be 5*5 different datasets, each of which are assigned a label for each of the 6 different m's.
   - Every dataset is then separated into training and testing sets.
3. Model definition and training
   - A unique feedforward model architecture will be defined. The model will receive the calculated input features, pass them through one or more hidden layers, and finally yield a single value (0, 1).
   - The model will be trained separately on every dataset (or as many as possible due to the amount of data).
4. Results and Benchmarking
   - Once the models are trained, the test datasets will be fed to their respective models and accuracy will be calculated.
   - Finally, these accuracies will be compared to the accuracies obtained by Madge.

References

1. Madge, S. (2015) *Predicting Stock Price Direction using Support Vector Machines.* Independent Work Report Spring 2015. Computer Science Department of Princeton University. Available at: https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
2. Abdul Salam, M., Emary, E., Zawbaa, H. (2018). *A Hybrid Moth-Flame Optimization and Extreme Learning Machine Model for Financial Forecasting.* Available at: file:///C:/Users/Juan%20Javier/Downloads/3-AHybridMoth-FlameOptimizationandExtreme.pdf