Project Proposal for Machine Learning Project

Team Members:

Samman Bikram Thapa

Brittany Miller

Bolu Aiki-Raji

Project Topic:

Predicting if the closing price of stock will rise or fall the next trading day

Project Description:

Introduction:

The stock market is the market in which shares of publicly held companies are issued and traded either through exchanges or over-the-counter markets (Staff, 2015). Many people use the stock market as a way of earning money by betting on or against stocks and indexes. And the million dollar question in the industry is "Will the stock price increase or decrease in future?" If only someone could be sure about the answer to the question, he could make a lot of money buying and selling stocks.

This project is an attempt to try to answer that question to some degree of surety using Machine Learning. We are trying to predict if the close price of stock will rise or fall on the next trading day, so we have approached this as a classification problem.

We will be using Python and Postgresql (subject to change) for calculation and data storage. We will be using the **K-Nearest Neighbour Algorithm** to classify the stocks.

Dataset:

We will use the data provided from www.Quandl.com to get data through their API services. We will be choosing four to six specific stocks to analyze over a period of time. We still have not decided on the stocks to choose for consideration. The initial data will include daily stock open, highest, lowest, and close prices, volume, dividend, and split ratio for each stock underconsideration.

Overview of the project:

We are going to:

1) Get the input for the algorithm includes (IV, independent variables):
    a) Moving average of historical close prices
    b) Trading volume, and open, highest, lowest, and close prices of the present trading day
    c) financial indicators, e.g., DecisionPoint Price Momentum Oscillator (PMO), Money Flow Index (MFI), Percentage Price Oscillator (PPO), and etc
    d) A price movement trend based on local Taylor expansion and spline fitting
2) Perform Data cleaning and Normalizing the input
3) Train the KNN classifier (described below)

4) Predict based on the classification of the model:
    a) the stock price will rise (1)
    b) the stock price will fall (-1)

K-Nearest Neighbour (KNN) classifier:

K-nearest neighbor technique is a machine learning algorithm that is considered as simple to implement (Aha et al. 1991). The kNN algorithm is belongs to the family of instance-based, competitive learning and lazy learning algorithms. Instance-based algorithms are those algorithms that model the problem using data instances (or rows) in order to make predictive decisions. The kNN algorithm is an extreme form of instance-based methods because all training observations are retained as part of the model. Lazy learning refers to the fact that the algorithm does not build a model until the time that a prediction is required. It is lazy because it only does work at the last second. This has the benefit of only including data relevant to the unseen data, called a localized model. A disadvantage is that it can be computationally expensive to repeat the same or similar searches over larger training datasets.

We are using kNN because it is simple to implement, which allows us to add features and fine tune the algorithm, and it does not assume anything about the data, other than a distance measure, which can be calculated consistently between any two instances. As such, it is called non-parametric or non-linear as it does not assume a functional form.

KNN Algorithm overview:

The stock prediction problem can be mapped into a similarity based classification. The historical stock data and the test data is mapped into a set of vectors. Each vector represents N dimension for each stock features. Then, a similarity metric such as Euclidean distance is computed to take a decision. A description of kNN in bullet form is provided below. kNN is considered a lazy learning that does not build a model or function previously, but yields the closest k records of the training data set that have the highest similarity to the test (i.e. query record). Then, a majority vote is performed among the selected k records to determine the class label and then assigned it to the query record.

The equations and calculations that will be applied for predicting next day price includes error estimation, total sum of squared error, average error, cumulative closing price when sorted using predicted values, k-values and training Root Mean Square (RMS) errors.

1. Open the dataset from datasource (CSV or some other format) and split into test/train datasets.
2. Calculate the distance between two stock instances.
    a. using Euclidean distance (subject to change based on adding feature to fine tune)
3. Locate k most similar stock instances:
    a. using Euclidean distance (subject to change based on adding feature to fine tune)
4. Generate a response from a set of stock instances:
    a. allow each neighbor to vote for their class attribute, and take the majority vote as the prediction
5. Summarize the accuracy of predictions:
    a. using Root Mean Square Deviation (RMSD) and Average Estimated Error (AEE)

References:

Staff, I. (2015, January 15). Stock Market. Retrieved February 26, 2017, from http://www.investopedia.com/terms/s/stockmarket.as
Aha, D., Kibler, D.W., Albert, M.K. (1991). Instance-based learning algorithms. Mach Learn, 6, 37–66