Devon Ankar

RECOMMENDATION ALGORITHM

Appendix for DSE 6300 Term Project – 4/26/2019

Goal/Approach

On each given day (or other unit of time), the recommendation algorithm is to make a recommendation to buy, sell, or hold Bitcoin shares based on the Twitter sentiment of that day.

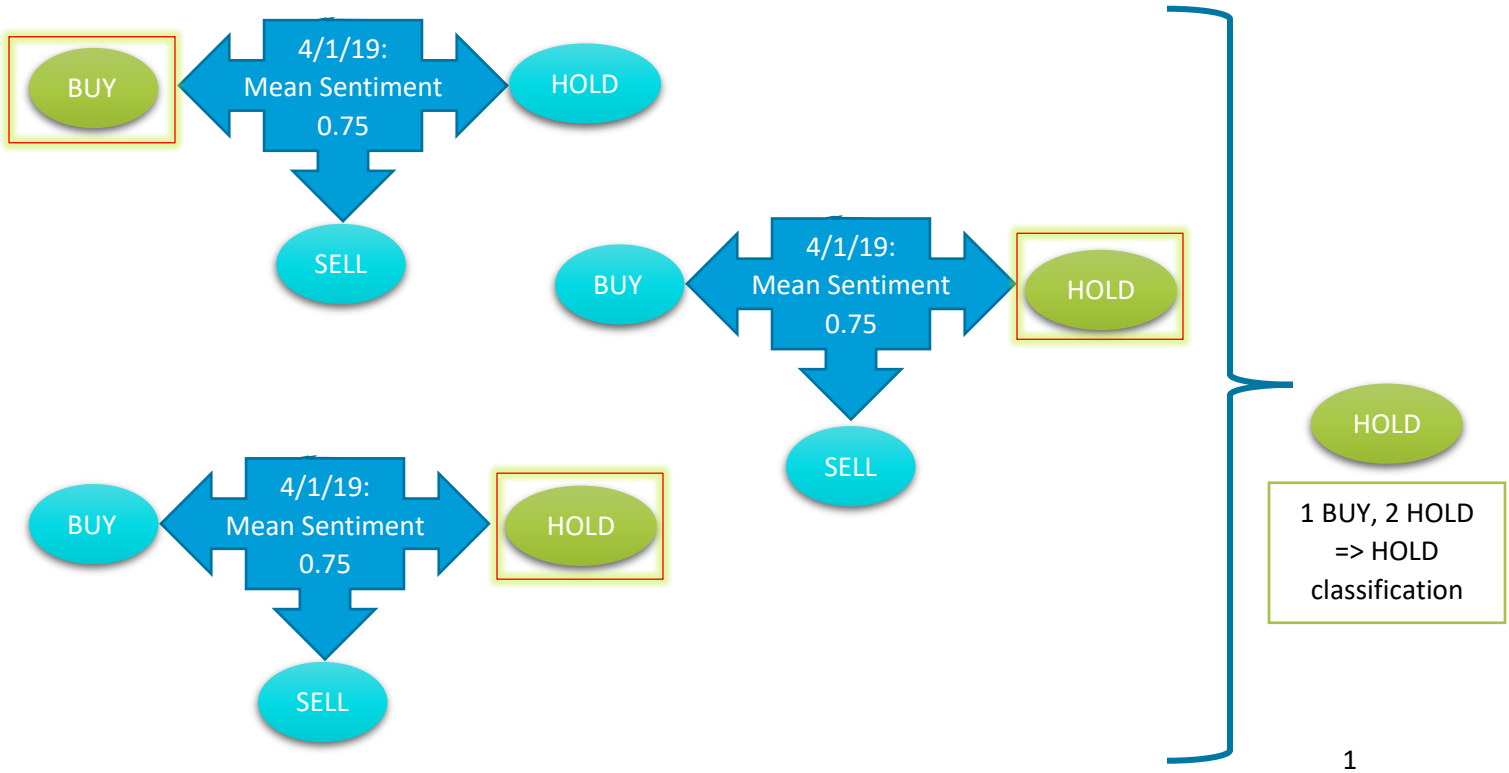
This is to be approached as a classification problem. Each day is to be classified as a buy, sell, or hold based on that day's Twitter sentiment. It is predictive modeling because each day's Twitter sentiment is used to predict the optimal action for that day (buy, sell, or hold) based on past information (training data).

Machine Learning Algorithms for Classification

Below we will discuss three different ML algorithms that can be used for the "BUY", "HOLD", or "SELL" classification. We will test all three algorithms and use the one that has the best performance as measured by highest classification accuracy.

Random Forest

Random forest is a classification and regression algorithm that works by generating a large number of decision trees. The class is determined by which class appears most often, i.e. the mode of the classes. For each day (or other time unit), a random forest algorithm would draw hundreds or thousands of trees like the below to determine the most likely class for that day.



The example day is 4/1/2019, with an average Twitter sentiment of 0.75. Sentiment is assessed on a scale from -1 (negative) to 1 (positive), so 0.75 is a fairly positive sentiment. The random forest algorithm would pick BUY or HOLD in the majority of trees as a result. In this highly condensed example, it picked HOLD twice and BUY only once, so HOLD is the ultimate classification decision.

KNN: *k*-nearest Neighbors

This is considered one of the simplest algorithms that is regularly used for classification. The class of a given point is determined by the *k* closest examples in the training dataset across the entire feature space. An object's class depends on the most common class among its nearest neighbors. For example, if $k = 3$ (*k* should always be an odd number to avoid ties), and two of the nearest neighbors are classified as "HOLD" whereas one is classified as "BUY", then "HOLD" will be the ultimate classification decision of the point.

Logistic Regression

Logistic regression is used when the dependent (Y) variable is categorical in nature, in this case a classification of "BUY", "HOLD", or "SELL", based on the independent (X) variables (features). We need to use a multinomial logistic regression because the Y variable is not binary (e.g. only "BUY" or "SELL"); there are three possible classifications.

If the data is highly imbalanced, for example if there are a very large number or very few days where the training data is labeled with "BUY" or "SELL", we will not use the logistic regression algorithm. Generally speaking, there should be at least 10 data points per feature (in this case, we only have one feature: sentiment). The data is unlikely to be this imbalanced.