Thee Meensuk (tmeensuk), Petros Dawit (pdawit), Dikshyant Rai (drai), Kasemsan Kongsala (kkongsal)

Introduction

In the previous blog posts, we introduced some regression models to predict the price of the bitcoin. We did that because, in the final results, we will use all the models that we think they're useful. On a given day, if many models agrees, we can confidently says that we have some conclusive results (either the price of bitcoin will go up or down). On the other hand, if the models do not agree, we can say that we have some inconclusive results for today, and try again in the future.

In this blog post, we will introduce the machine learning approach to predict the price of bitcoin. We will use the closing price of 7 prior days (whether the price has gone up or down) and feed it to the machine learning algorithm in python (here, we use kt-learn) to predict whether the price of the bitcoin today will go up or down

Data Pre-processing

For the data pre-processing, we used excel to quickly create our csv file. We wanted to do a the machine learning approach. Our csv file contains 10 columns. The first column is the date and the second is the closing price for that date. The third column is called 'Label' and it is comparing the current closing price to the day prior. We will be using this column to compare to our machine learning approach when seeing how accurate it is. The next 7 columns after is 'X Days Prior' where X is a number ranging from 2-8. These columns are filled with 1s and 0s and describe whether or not the price for the current date has gone up, represented by 1, or down, represented by 0 from X days prior. The equation we used in excel is in this format, =IF($B3>$B11,1,0), where the difference in the value of the columns is the date difference and the if statement is true, it returns 1, 0 otherwise. The columns in B(the second column) contain the price so we are comparing the closing price.

Below is an image of part of our csv file data.

Date	Close	Label	2 Days Prior	3 Days Prior	4 Days Prior	5 Days Prior	6 Days Prior	7 Days Prior	8 Days Prior
3/17/17	1087.88	0	0	0	0	0	0	0	0
3/16/17	1172.62	0	0	0	0	0	1	0	1
3/15/17	1257.32	1	1	1	1	1	1	1	1
3/14/17	1245.86	1	1	1	1	1	1	1	0
3/13/17	1242.46	1	1	1	1	1	1	0	0
3/12/17	1226.62	1	1	1	1	0	0	0	0
3/11/17	1176.55	1	0	1	0	0	0	0	0
3/10/17	1116.97	0	0	0	0	0	0	0	0
3/9/17	1190.89	1	0	0	0	0	0	0	0
3/8/17	1150.05	0	0	0	0	0	0	0	0
3/7/17	1233.05	0	0	0	0	0	1	1	1
3/6/17	1278.49	1	1	0	1	1	1	1	1
3/5/17	1273	1	0	1	1	1	1	1	1
3/4/17	1260	0	1	1	1	1	1	1	1
3/3/17	1285.33	1	1	1	1	1	1	1	1
3/2/17	1257.6	1	1	1	1	1	1	1	1
3/1/17	1226.39	1	1	1	1	1	1	1	1
2/28/17	1191.21	0	1	1	1	1	1	1	1
2/27/17	1194.64	1	1	1	1	1	1	1	1
2/26/17	1179.05	1	0	0	1	1	1	1	1
2/25/17	1152.2	0	0	1	1	1	1	1	1
2/24/17	1180.14	0	1	1	1	1	1	1	1
2/23/17	1188.11	1	1	1	1	1	1	1	1
2/22/17	1130.01	1	1	1	1	1	1	1	1
2/21/17	1124.62	1	1	1	1	1	1	1	1
2/20/17	1084	1	1	1	1	1	1	1	1
2/19/17	1051.8	0	0	1	1	1	1	1	1
2/18/17	1056.4	1	1	1	1	1	1	1	1
2/17/17	1055.46	1	1	1	1	1	1	1	1
2/16/17	1032.7	1	1	1	1	1	1	1	0
2/15/17	1011.53	1	1	1	0	1	1	0	0
2/14/17	1008.88	1	1	0	1	1	0	0	0
2/13/17	1000.79	1	0	1	1	0	0	0	0
2/12/17	1000.73	0	1	1	0	0	0	0	0
2/11/17	1012.4	1	1	0	0	0	1	0	0
2/10/17	996.08	1	0	0	0	0	0	0	0
2/9/17	986	0	0	0	0	0	0	0	1

The Machine Learning

When we did the machine learning, we used three models in sk-learn's modules: Naive Bayes, Logistic Regression, and Linear SVM. We found the exact same results for all three models. That is same values for training mean accuracy, cross validation score (mean and standard deviation) and predicted mean accuracy.

It is not surprising that we got the same results for all three models. The most important reason is the close relationship between each column that we have chosen in our machine learning. Bitcoin prices tend to move in trends and not a random walk. This means that if the price movements 2, 3, 4, 5, 6, 7, and 8 days prior to a given day tend to have more 1's than 0's, meaning that the prices increased more than decreased, the price movement of that given day will be likely to be 1, and vice versa. The close relationships between the features and label has made it fairly easy to predict the label given a set of features. Therefore, any sensible model will give tend to give the same label given a set of features.

Naive Bayes assumes that all features are independent from each other. That means the price movements 2 to 8 days prior will have the same contribution to the model's prediction. As a result, the trend is directly related to the number of 1's and 0's in a given set of features, or the price movements 2 to 8 days prior. If we have more 1's, the label is likely 1. If we have more 0's, the label is likely 0.

Logistic Regression gives coefficients to the columns of price movements 2 to 8 days prior. However, in a stable bitcoin marketplace, the trend does not turn back within 8 days, so the price trend can be predicted from the price movement 2 days prior as much as the movement 8 days prior. Of course there will be volatile times where the abrupt turnback of price 2 days prior carries more weight than that 8 days prior. However, these events are rare and do not make much difference to the training data. Therefore, the Logistic Regression naturally converges to the same model as Naive Bayes.

Linear SVM uses simple linear kernels to predict the prices, so it has similar performance as Logistic Regression. It comes naturally that Linear SVM will yield the same result as Logistic Regression and therefore Naive Bayes.

Another reason why all three models produced the same result is the relatively small dataset. Even when the models are likely to produce similar results, what we got here from our machine learning is the exact same results to the 10th decimal place. If we re-run the algorithm with a larger dataset, we might still get similar results but not similarity to this precision.

The Results

Naive Bayes Model:

Training Mean Accuracy	Cross Validation Score		Predicted Mean Accuracy
	Mean	Standard Deviation	
0.630681818182	0.630738208429	0.0487526200048	0.578947368421

Logistic Regression Model:

Training Mean Accuracy	Cross Validation Score		Predicted Mean Accuracy
	Mean	Standard Deviation	
0.630681818182	0.630738208429	0.0487526200048	0.578947368421

SVM Model:

Training Mean Accuracy	Cross Validation Score		Predicted Mean Accuracy
	Mean	Standard Deviation	
0.630681818182	0.630738208429	0.0487526200048	0.578947368421