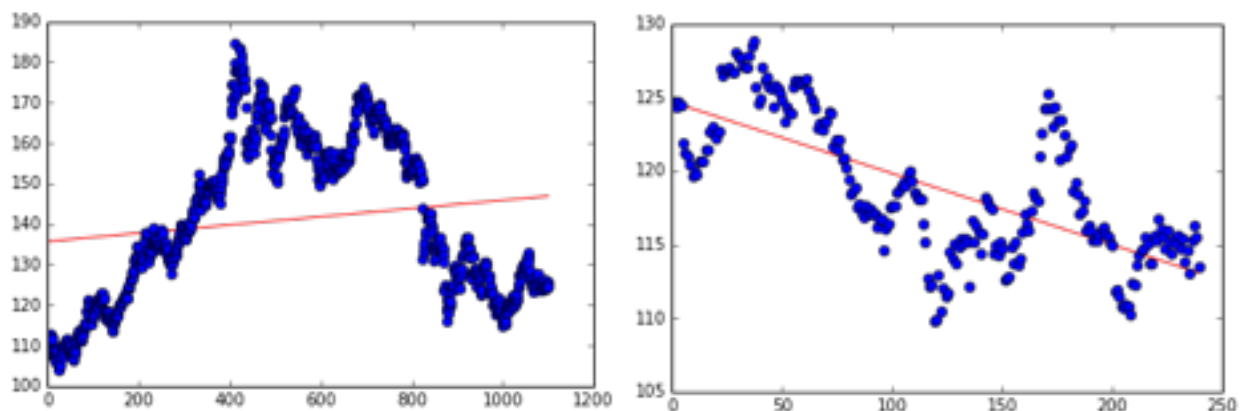


Technical Analysis on Gold ETF

We trained the model on from on data from dates 2010-01-04 to 2014-05-19. Did a quick test of the predictions from 2014-05-19 to 2015-4-30. Results of out of sample data predictions. 125/240. That is, the model correctly predicted the direction the stock would go ~52% of the time.

One way we could go about acting on these results is we could buy when the the prediction says the stock will go up tomorrow (label: “UP”) and not buy when prediction says it will go down or stay the same (label: “DOWN”). However since we aren’t predicting the magnitude of the moves this doesn't give us much insight into the profitability of the strategy. If however we got someone to take our bet at even money on if stock will go up or down the next day we could’ve made some money over this time period.

This is a very small sample size which would lead us to discounting the results of the out of sample test. However, it’s interesting to note that the data we trained the model on had 510 “DOWN” labels and 575 “UP” labels which we’d believe would bias the model to predicting “UP” (Figure on left).



But, it’s encouraging to see that even though the trend of the out of sample data(Figure on right) is going down, the model still accurately predicted more than 52%.

Google’s prediction API is a total blackbox. We can’t be sure what methods or algorithms they are using. Some quick research suggests they try a couple different methods and ends up going with whichever one performed best on your data. In training the model the prediction API gives us: "classificationAccuracy": “0.54”. Since our data is ordered time-series data, of course depending on how they perform the cross-validation, the results most likely aren’t helpful. Unfortunately all we know from Google is, “The API performs multiple rounds of cross-validation to calculate an estimated error for your dataset.”[1]

Another issue is analyzing the model there are a couple red flags. ie: "WARNING": "Feature column 1 always has the same value." When looking at the data given to the model, this isn’t the case.

Also, feature 6 has "mean": "NaN", this is worrisome and warrants some investigation. It could be that the API can't handle very small numbers and at some points treats sufficiently small numbers just as zero which makes it a useless feature.

Finally, we have the issue of no real basis on which exact features were chosen.. The features include the returns, the ratio of returns and slope of trend line over 2, 5, and 15 days. There's no reason to believe these are the optimal numbers to use or there should be some predictive ability from those specific time frames since they were just arbitrarily chosen.

Some things we could try in the future. The model does give you a "score" for each label each adding up to 100, which suggests it's assigning the probability that, that particular list of features is a certain label. One thing we could try is only acting on predictions the model deems very strong, eg. 0.70+. Could backtest a strategy that only generates a "buy" signal that model the model assigns a high probability of going up.

Also we do not know if Google's Prediction API normalizes the data at all, and we can try doing that ourselves to check if this would help improve the accuracy of the predictions

So although going in I was very skeptical of any form of "technical analysis" the findings show that maybe there is something there. On the other hand, when attempting to predict financial outcomes we have to be very wary, since there are so many other factors that could be influencing the events that our simple model isn't taking into account.

[1] <https://cloud.google.com/prediction/docs/faq>