Ahmad Aldabbagh

KNN Stock Learning Strategy

1. Strategy Overview:

The basis of my strategy is to use a knn-learning algorithm to predict future changes in the price and make trading decisions accordingly. For a knn-learner there are two main parameter that I needed to adjust to get a good strategy:

1. The type of features used
2. The value of K which was set to be equal to 3

First are the type of features that I used, for a given stock I used three main features that are calculated and used as input to the predictor. The technical features are: **Bollinger band** feature [2], **momentum** [2] and **Kairi relative index** [1].

1. The **Bollinger Band** feature was calculated as mentioned in the project's page which is:

   bb_value[t] = (price[t] - SMA[t])/(2 * stdev[t])

   Where SMA is a moving average with window = 20; and stdev is the standard deviation over the same window.

2. The **momentum** feature was also based on the mentioned formula in the project's page which is:

   momentum[t] = (price[t]/price[t-N]) - 1

   Where 'N' was chosen to be 5

3. The third feature I decided to use the **Kairi Relative Index [KRI]** that I happened to stumble upon in Investopedia. As described in [1], the KRI is a Japanese leading indicator that indicates momentum oscillations and have been over shadowed by the Relative Strength Index. The way KRI is calculated is as follows (where I changed the multiplication by 100 to 10.0):

   10.0 x (Price – simple moving average)/(simple moving average)

   The SMA could be over a period of 20 or 10 days and I have decided to go with 20 days just like the Bollinger band indicator.

   As for the predicted output, as mentioned earlier, it is the expected change in the price after five days which was calculated as follows:

   Y[t] = (price[t+5]/price[t]) - 1.0

Now that the inputs and outputs have been explained, I will explain the main strategy to buy or sell. In my strategy I decided to either buy or sell where I do not stay out of the market so my Entry Exit chart will only show buying and selling entries since the short and long exits are overlapping with the entries of the following transactions. In other words, I execute an action and hold until the indicator changes and then execute the alternate action accordingly and this is done over the whole trading period. To summarize the strategy:

I iterate over my days and predict the value of 'ypredict' and the following conditions apply which in python code.

```
Entryl, Entrys = False
if ypredict >= 0.01 and not Entryl and not Entrys:
    print 'here'
    orders['Order'][i] = 'BUY'
    orders['Shares'][i] = 100.0
    orders['action'][i] = 'long entry'
    Entryl = True
    Entrys = False

elif ypredict <= -0.01 and not Entryl and not Entrys:
    #print 'here1'
    orders['Order'][i] = 'SELL'
    orders['Shares'][i] = 100.0
    orders['action'][i] = 'short entry'
    Entrys = True
    Entryl = False

elif ypredict <= -0.01 and Entryl and not Entrys:
    #print 'here2'
    orders['Order'][i] = 'SELL'
    orders['Shares'][i] = 100.0
    orders['action'][i] = 'long exit and short entry'
    Entryl = False
    Entrys = True

elif ypredict>=0.01 and not Entryl and Entrys:
    #print 'here3'
    orders['Order'][i] = 'BUY'
    orders['Shares'][i] = 100.0
    orders['action'][i] = 'short exit and long entry'
    Entrys = False
    Entryl = True
```

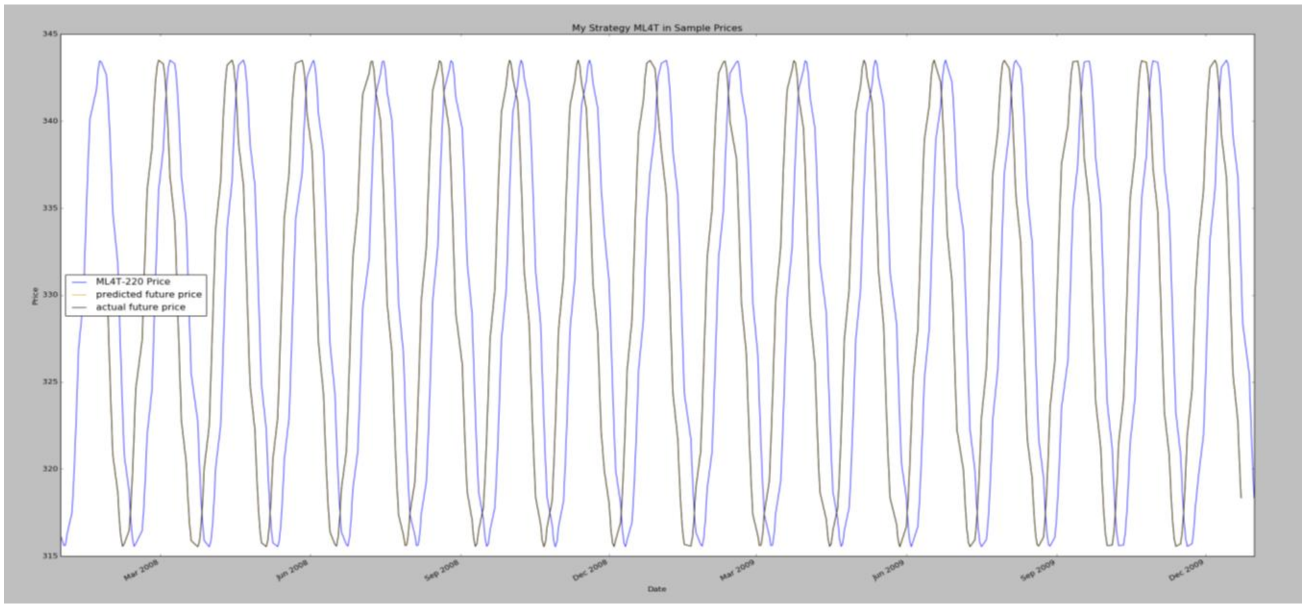
We can see above that if I am predicting y to change positively by 1%, I buy and hold until I am predicting that it will decrease by 1% so I sell and hold. This is done iteratively. Where I iterate between longs and shorts while not holding onto more than +100 or not less than -100 shares at a given moment.

Note that the chosen parameters were based on tweaking the learning on the ML4T and seeing how it performs in both the in-sample and out-sample phases. Even though the performance is quite acceptable as I will discuss below, I still believe there room for improvement. Finally, all portfolio initial values are $10,000.

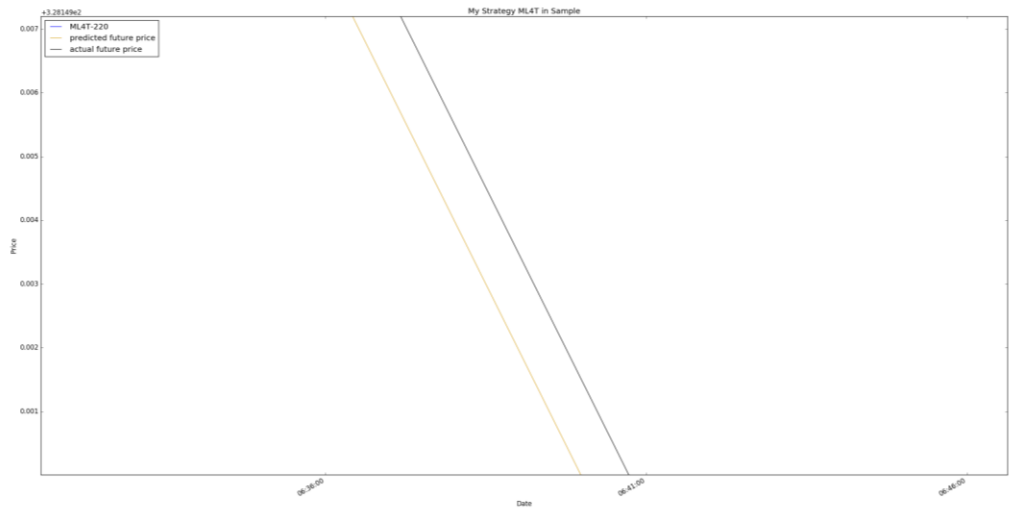
2. Plots and Results:

1. Training Y/Price/Predicted Y (This is for the sine data the one for IBM can be found at the end of the report in the appendix as an additional chart):

   It can be seen in the figure that the predicted and actual price for the in sample ML4T sine data is almost identical and this is quite understandable due to the small value of my k which is set to 3. But if we zoom in we will find that they are very close.

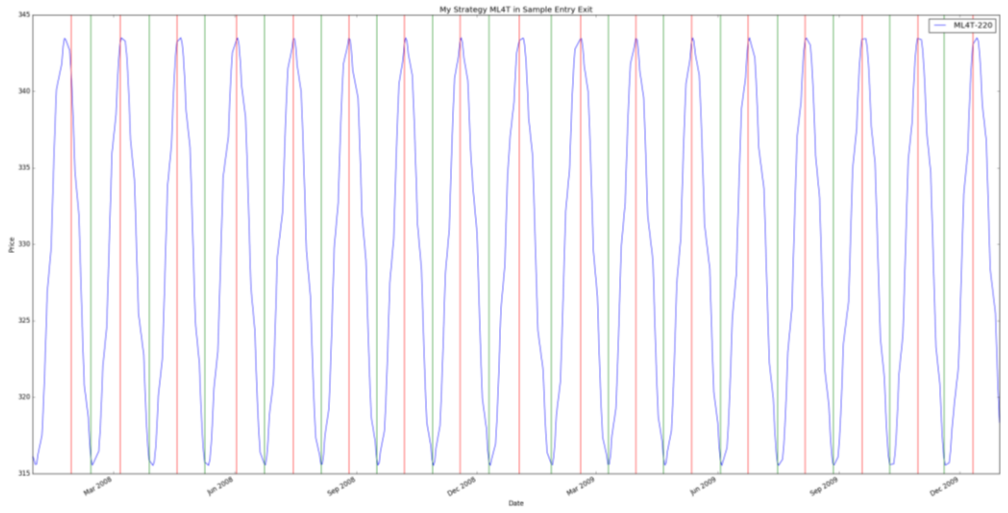


Zoomed in:



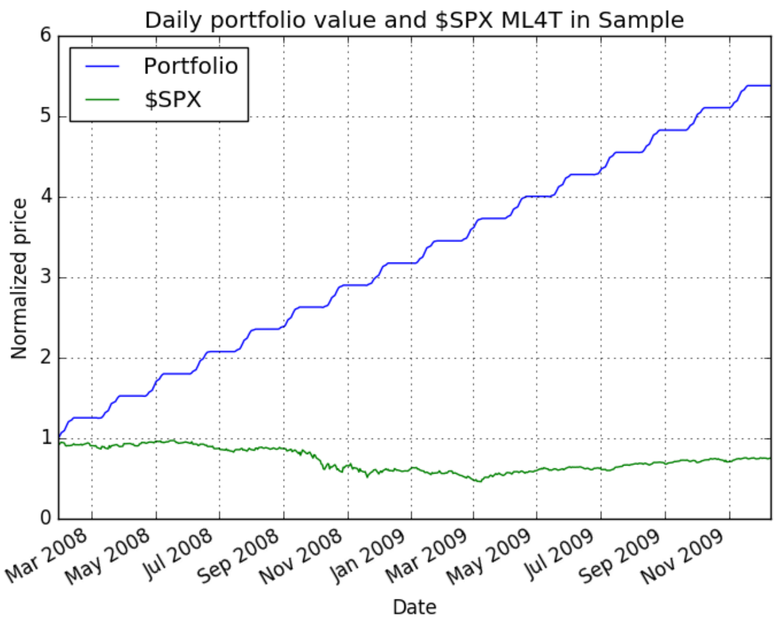
We can see here that the predicted vs. actual are very close but difficult to see on the large chart.

2. Sine data in-sample Entries/Exits



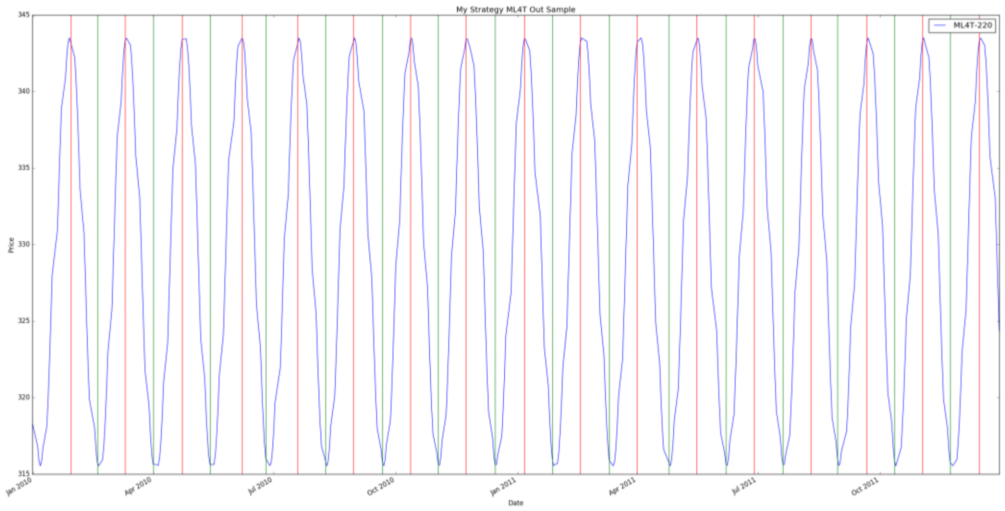
It can be seen above that my strategy is buying when it is expecting an increase and selling when it is expecting a decrease which what I intended it to do.

3. Sine data in-sample backtest



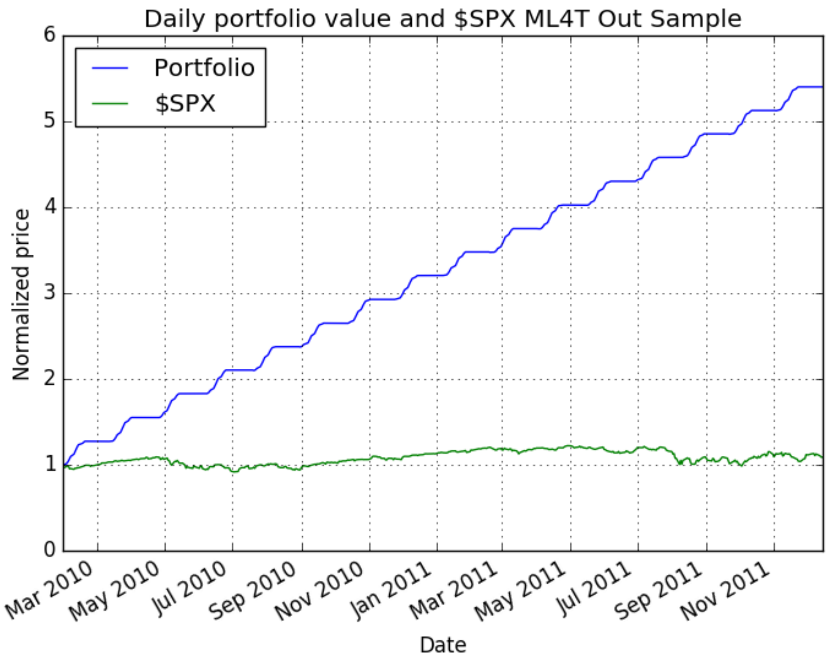
It can be seen above that the in sample performance is quite good with the following stats:
RMSE: 5.87540443032e-06
corr: 0.999999981249
Profit: 43804.9299
Even though this looks promising, we cannot tell until we try it on the out of sample.

4. Sine data out-of-sample Entries/Exits



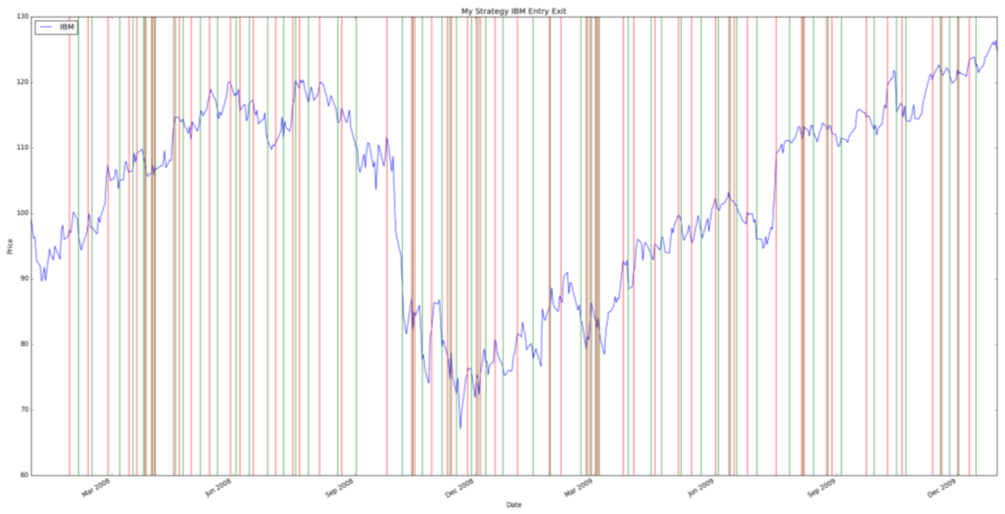
We can see that the data follows the same behavior as the in-sample which is expected due to the repetitive nature of the data.

5. Sine data out-of-sample backtest



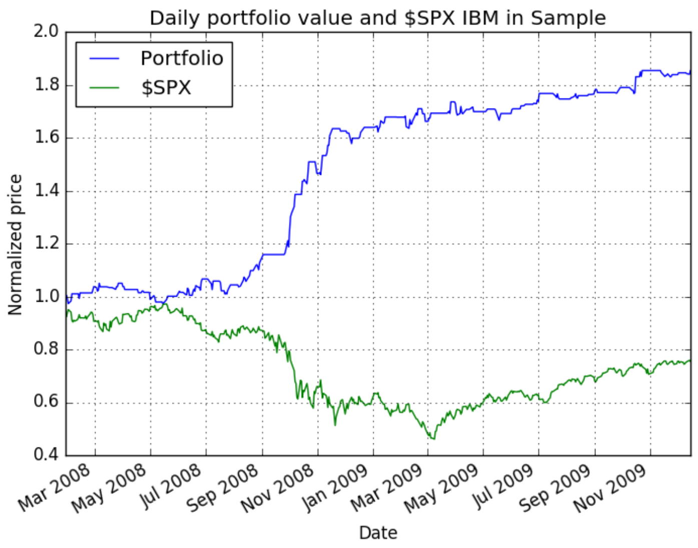
Here we can see that the performance follows a similar trend for the in-sample period which is also expected as mentioned earlier due to the repetitive nature of the data. The results are as follows:
RMSE: 2.5709009665e-05
corr: 0.999999639395
Profit: 44014.7992
The profit is consistent with what we had earlier.

6. IBM data in-sample Entries/Exits



We can see that the same strategy is executed here where I only buy and sell based on the predicted behavior. It can be noticed from both charts that alternation between buys and sells where I either buy or sell. Note that this is based on the same parameters used for the sine data which were frozen when testing on IBM both in sample and out of sample.

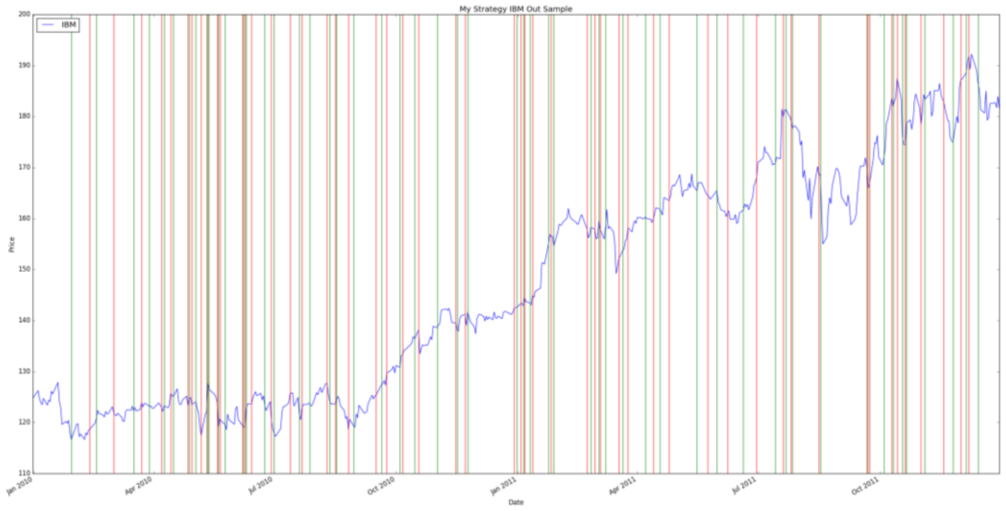
7. IBM data in-sample backtest



While the performance here is not as good as the sine data due to the variant and non-repetitive nature of the data, it still performs better than the manual method which achieved a profit of around 4800 using the KRI index. The summary of results are as follows:
RMSE: 0.0329227585838
corr: 0.649480669644
Profit: 8531.0

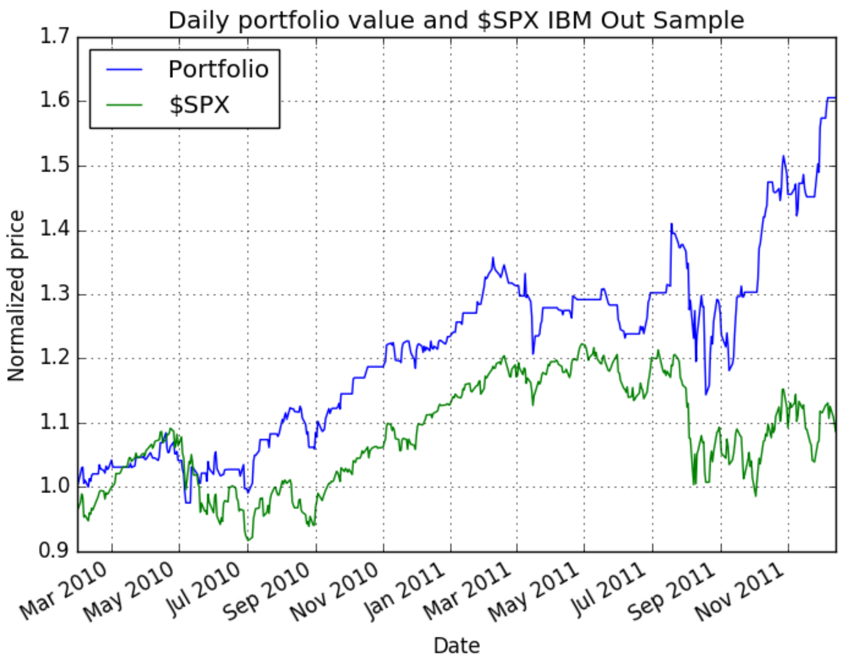
The mediocre correlation here is expected due to the variance in the data and that I am averaging the lowest 3 values but this is based on the parameters used/tweked in the sine data.

8. IBM data Out-of-sample Entries/Exits



We can see here that in some of the flat area like that in December, 2010, the trader is just holding because there is no high increase or decrease and it looks like quite flat. This is consistent with the intent of the strategy as discussed earlier.

9. IBM data out-of-sample backtest



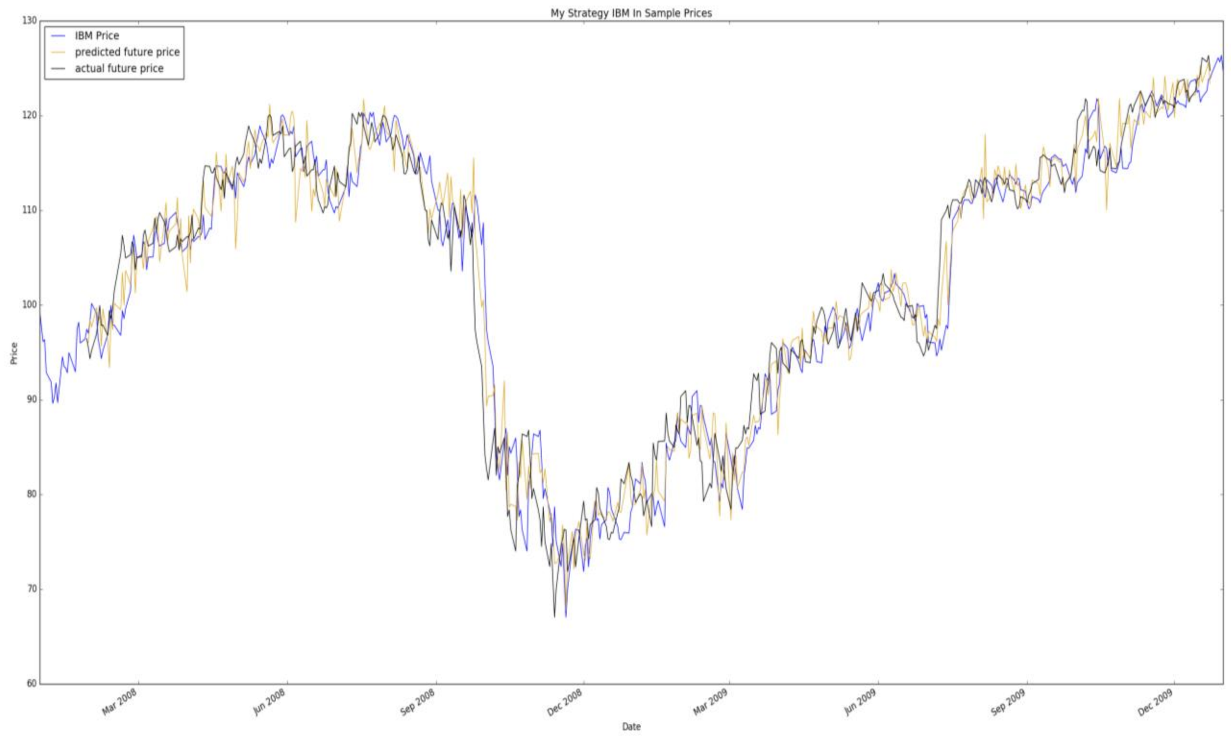
It can be seen here that the trader performs also quite well as compared to my manual method which achieved a profit of around 4000. The summary of results is as follows:

RMSE: 0.0348800981992
corr: 0.0256035177727
Profit: 6056.0

The achieved results are expected to have this lower performance when compared with the sine data which as shown earlier follow a predictable repetitive trend which not the case in this data. One possible area of improvement might be the use of 'stock specific' parameters to avoid overfitting the learner for the sine data. The stock specific parameters can be trained on the respective companies' historical data and tweaked to achieve better results. It is could also be the case that the small value of k=3 lead the knn learner to become overfitted where it was seen in the past project that the learner was overfitted for lower values of k. Another possible area of improvement, is to choose a different type of learner or to use bagging if knn was to be used.

Appendix:

The following chart is an additional chart that show Training Y/Price/Predicted Y for IBM data



References:

1. Kairi Relative Index: The Forgotten Oscillator | Investopedia. (2009). Retrieved March 26, 2016, from http://www.investopedia.com/articles/forex/09/kairi-relative-strength-index.asp
2. Indicators in the project page: http://quantsoftware.gatech.edu/MC3-Project-2#Choosing_Technical_Features_--_Your_X_Values
3. CS 7646 Video Lectures
4. The line of method1 to optimize the calculation of the RMS error as posted on piazza https://piazza.com/class/ij9yiif53l27fs?cid=1143