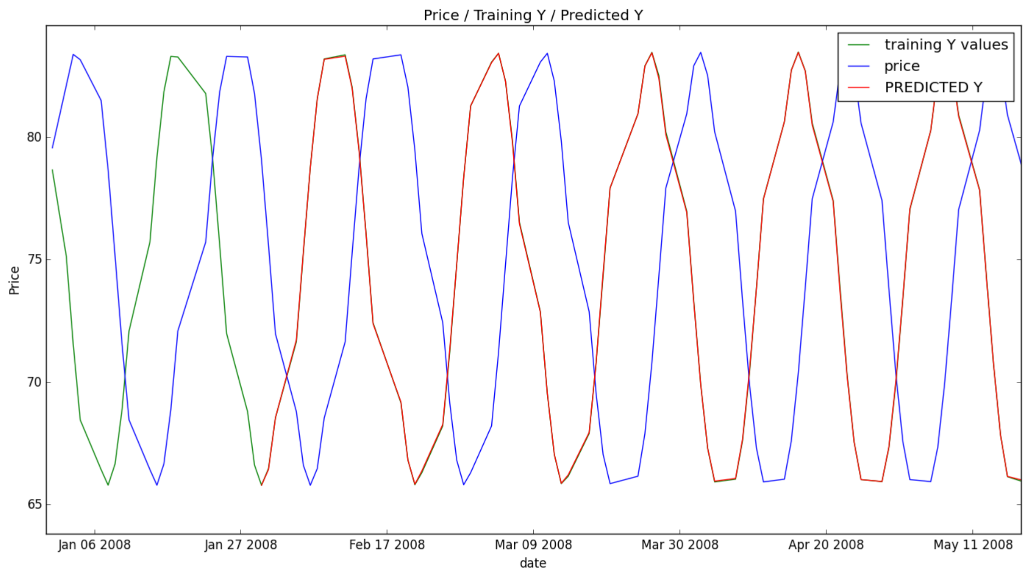
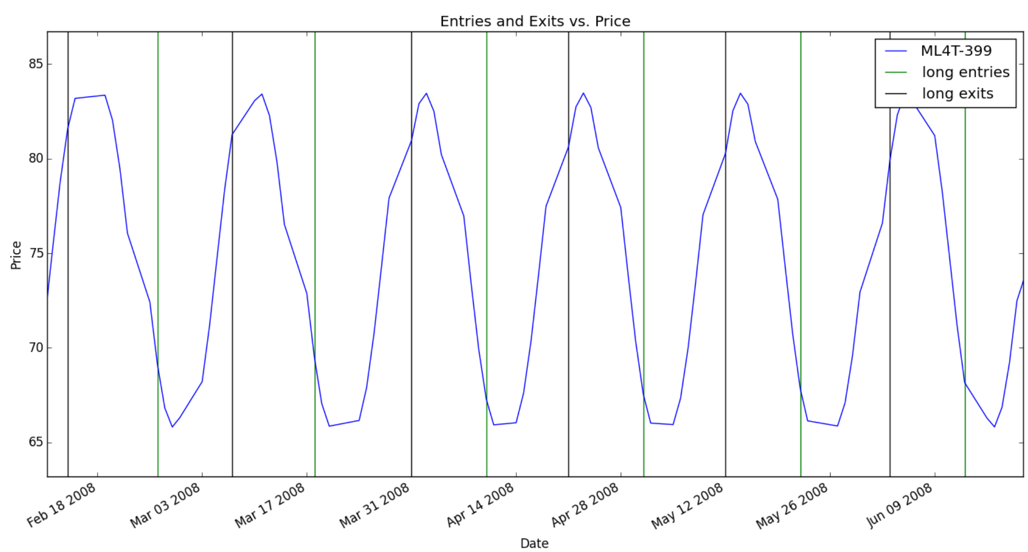
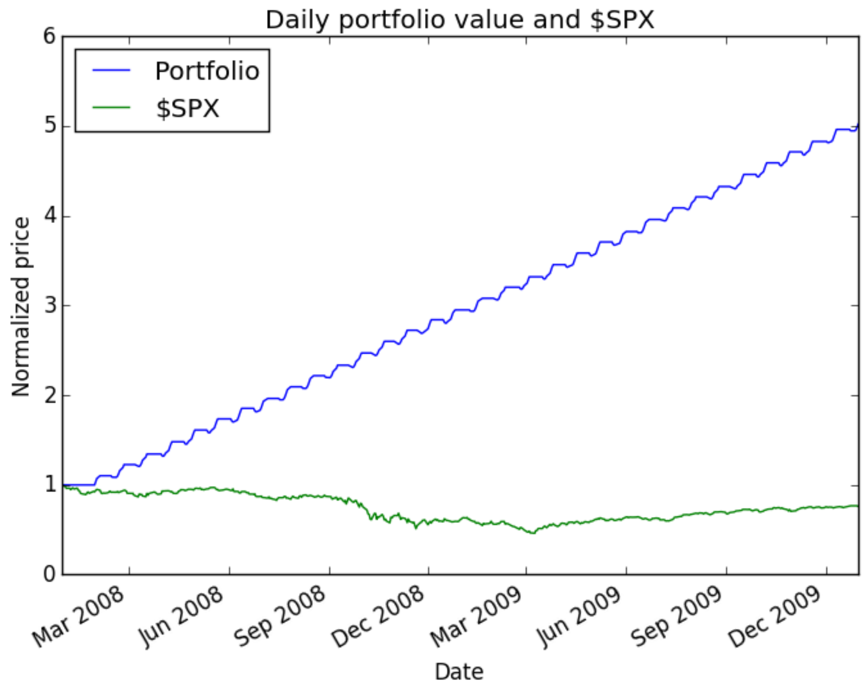
ML4T-399 Charts **1.** Training Y/Price/Predicted Y:



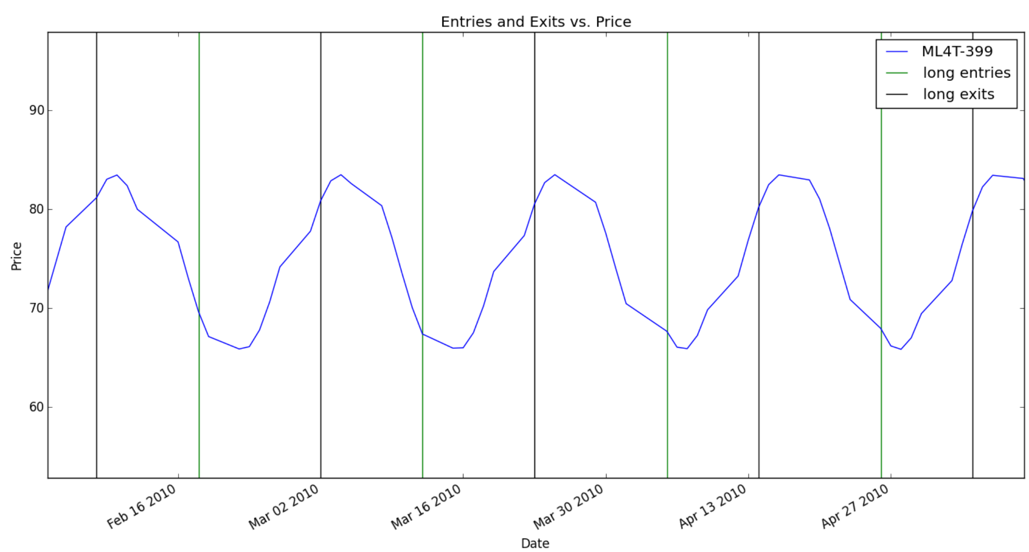
2. Sine Data In Sample Entries/Exits:



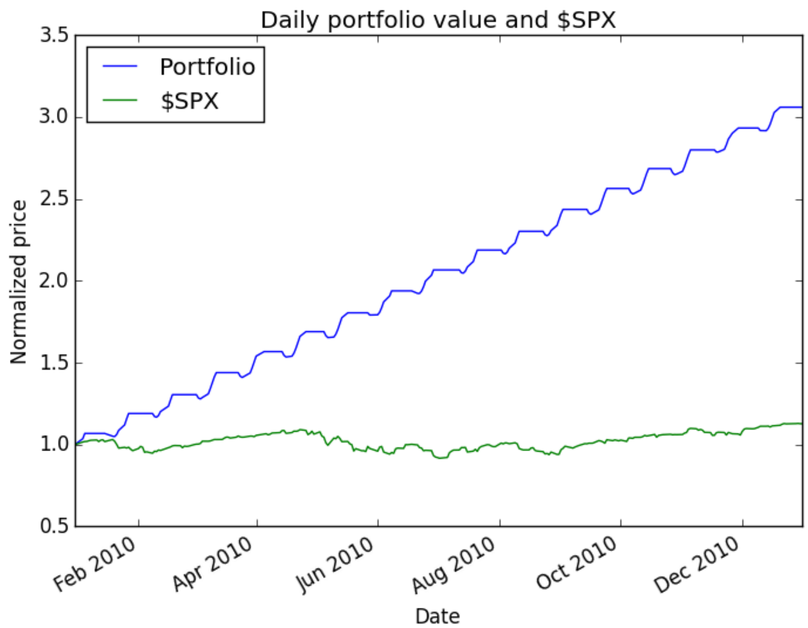
3. Sine Data In Sample Backtest



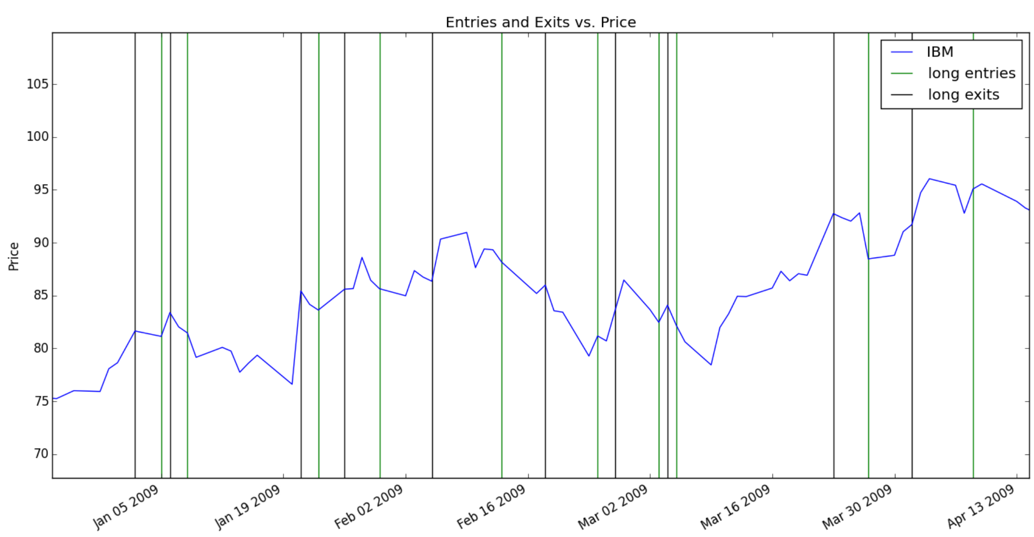
4. Sine Data Out of Sample Entries/Exits:



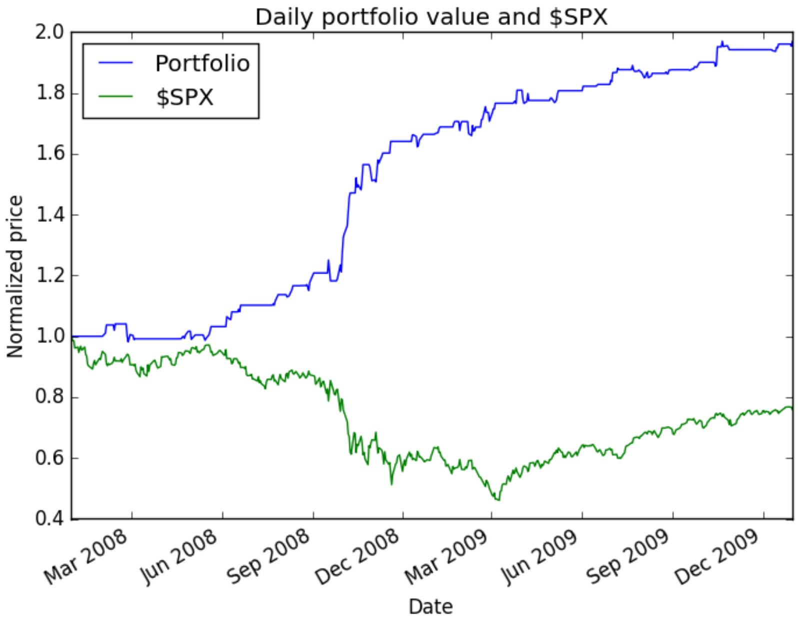
5. Sine Data Out of Sample Backtest



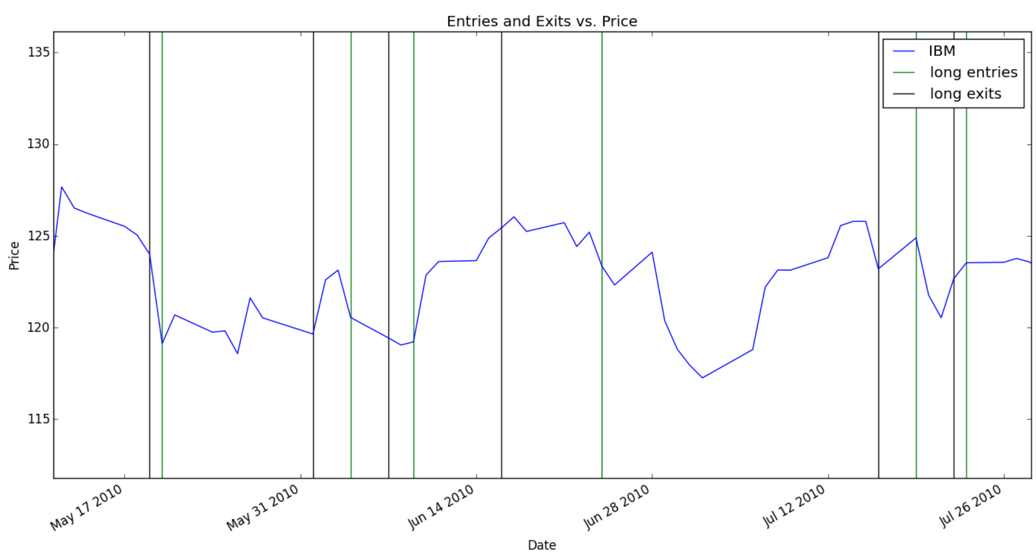
IBM Charts 6. IBM IN SAMPLE ENTRIES/EXITS



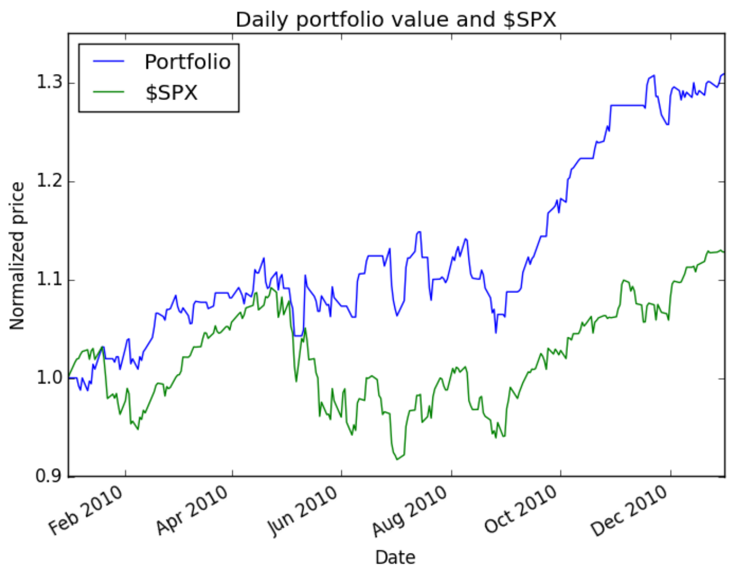
7. IBM IN SAMPLE BACKTEST



8. IBM OUT OF SAMPLE ENTRIES and EXITS



9. IBM OUT OF SAMPLE BACKTEST



Describe each of the indicators you have selected in enough detail that someone else could reproduce them in code.

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

code: bb_value = (price - SMA) /(2*std)

Momentum: *momentum[t] = (price[t]/price[t-N]) - 1   N = 10 # days*

code: momentum = price/price.shift(10) - 1

Volatility: *volatilty[t] = stddev(dailyreturns)*

code: volatility = pd.rolling_std(daily_returns, window=20, min_periods = 20)

- Describe your trading policy clearly.

I used my bag learner with my knn learner with k = 5 and number of bags = 20 to forecast the 5 day future price. When 5 day future price increased > 1%, I made a long entry. I held the position until it decreased < -1% , and made a long exit and repeated.

Summary Data of Results:

• ML4T-399 IN SAMPLE	ML4T-399 OUT OF SAMPLE
k = 5	
Number of bags: 20	
In sample results	Out of sample results
RMSE: 0.000740068149779	RMSE: 0.00070504090674
corr: 0.999987242947	corr: 0.999988374019
Data Range: 2007-12-31 to 2009-12-31	Data Range: 2009-12-31 to 2010-12-31
Sharpe Ratio of Fund: 6.94438547065	Sharpe Ratio of Fund: 7.54908475589
Sharpe Ratio of $SPX: -0.21996865409	Sharpe Ratio of $SPX: 0.756512754402
Cumulative Return of Fund: 4.02166813	Cumulative Return of Fund: 2.06004713
Cumulative Return of $SPX: -0.240581328829	Cumulative Return of $SPX: 0.127827100708
Standard Deviation of Fund: 0.00737807995534	Standard Deviation of Fund: 0.00944609267468
Standard Deviation of $SPX: 0.0219524869863	Standard Deviation of $SPX: 0.0113715303326
Average Daily Return of Fund: 0.00322757919013	Average Daily Return of Fund: 0.00449206708097
Average Daily Return of $SPX: -0.000304189525556	Average Daily Return of $SPX: 0.000541919649169
Final Portfolio Value: 50216.6813	Final Portfolio Value: 30600.4713

IBM IN SAMPLE	
k = 5	IBM OUT OF SAMPLE
Number of bags: 20	
In sample results	Out of sample results
RMSE: 0.0361430563998	RMSE: 0.0275997887905
corr: 0.561835434391	corr: 0.0602149310415
Data Range: 2007-12-31 to 2009-12-31	Data Range: 2009-12-31 to 2010-12-31
Sharpe Ratio of Fund: 2.62400664839	Sharpe Ratio of Fund: 1.73320251849
Sharpe Ratio of $SPX: -0.21996865409	Sharpe Ratio of $SPX: 0.756512754402
Cumulative Return of Fund: 0.9693	Cumulative Return of Fund: 0.3089
Cumulative Return of $SPX: -0.240581328829	Cumulative Return of $SPX: 0.127827100708
Standard Deviation of Fund: 0.0083314785834	Standard Deviation of Fund: 0.0102675013926
Standard Deviation of $SPX: 0.0219524869863	Standard Deviation of $SPX: 0.0113715303326
Average Daily Return of Fund: 0.00137716742955	Average Daily Return of Fund: 0.00112102116312
Average Daily Return of $SPX: -0.000304189525556	Average Daily Return of $SPX: 0.000541919649169
Final Portfolio Value: 19693.0	Final Portfolio Value: 13089.0

- Discussion of results. Did it work well? Why? What would you do differently?

My learner setup worked extremely well with ML4T-399 data, with both in and out of sample RMSE almost 0 and correlation almost 1. Almost increased 5 times in portfolio value in sample and 3 times out of sample. This is because the data was a totally predictable sine wave so that the KNN learner had perfect training data to refer to. This enabled extremely well performance on sine data as we could predict to buy near the bottom at the curve and sell near the top of the curve very accurately.

My learner setup worked quite well with IBM in sample, with in sample RMSE .036, correlation .56 and it almost doubled in portfolio. For out of sample, RMSE was even slightly lower at RMSE .027 which I didn’t expect, but perhaps it’s close enough to in sample. However, correlation was quite lower at .06 which was to be expected for out of sample prediction. The portfolio did perform quite well with a 31% increase. It appears that the predictions of the KNN learner were accurate enough to predict 1% future price increases or decreases in order to enable my system to properly implement the policy of buying before price increases and selling before price decreases. Combine this with a large number of trades, more often than not the system was accurate, amounting to an increase in portfolio.

I also removed the nans from the x data and ensured that the x data where normalized values in order to optimize KNN learner accuracy. Using the bag learner at 20 bags helped remove potential overfitting. I also found that k=5 gave larger accuracy to the predictions than k=3. Perhaps because our aim was to predict 5 day price change.

Additionally the x data used momentum, volatility, and Bollinger value seemed to be good choices for indications of future price change. Momentum tends to continue into the future for a while, volatility indicates how much price change, and Bollinger value indicates how likely the price is going to go back the other direction.

A 1% increase and decrease threshold seemed to be a fair choice since the IBM price rarely fluctuated more than 2% in real data. Therefore too high of a threshold would result in no long entries and exits. Too low of a threshold would result in incorrect long entries and exits, resulting in decreased portfolio.

"What would you do differently?":

I would've increased the number of KNN learners, with different k's, with increased bagging to achieve higher accuracy and therefore better results with more appropriate entries and exits. I would not have combined these with LinReg learners as I found them to be much less accurate with IBM data.

I also would've spent more time researching indicators and trying out different combinations for more optimal results.