I want you to submit the code you used for the project.  It can be a zip file of your directory if you like.

code.zip

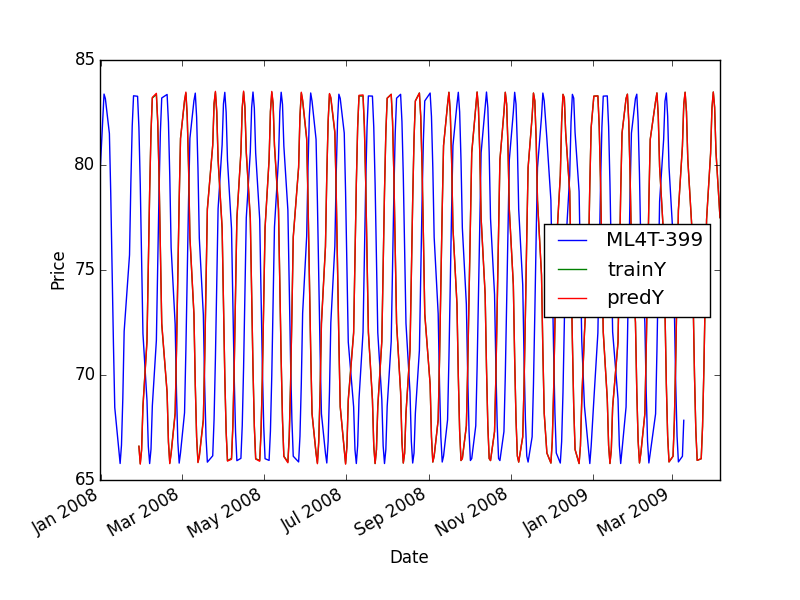
**My posts**

KNN goes flat when it goes out of range, so you're very likely seeing indicators outside of what it has ever seen before in your last 73 days. Make sure you normalize the indicators and that shouldn't happen.

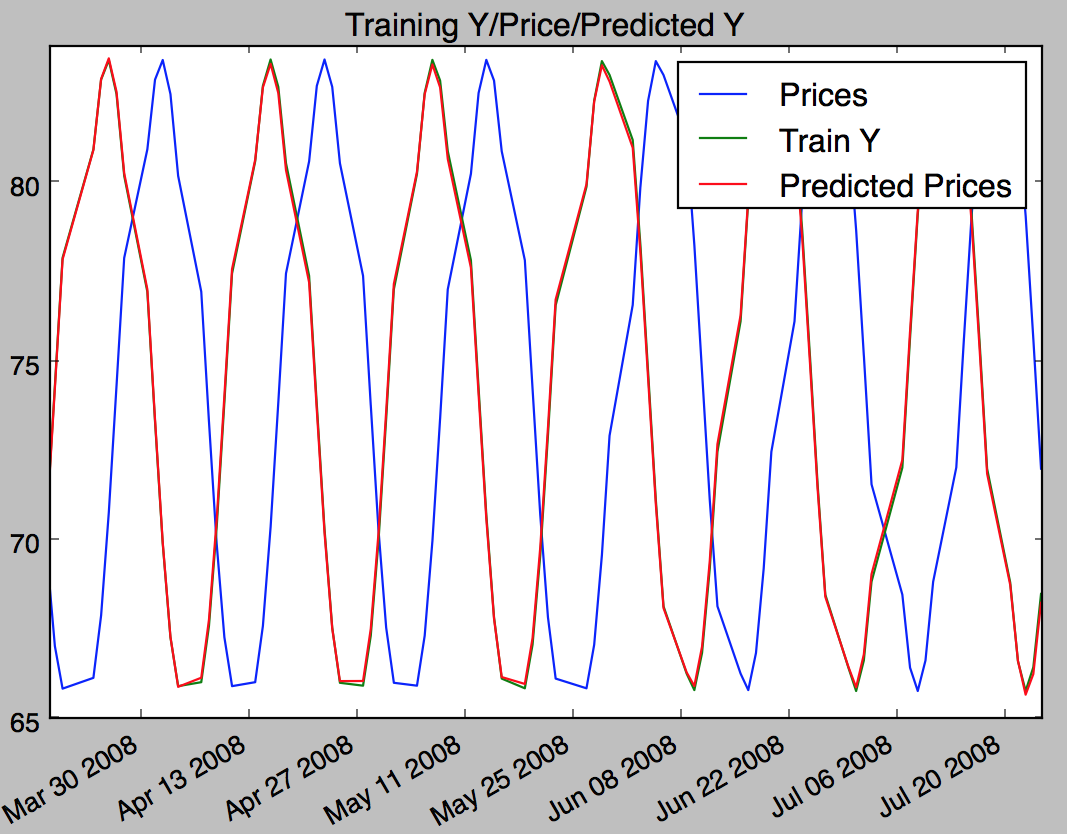
**Getting started**

 I missed the step to add the price back to trainY and predY before graphing.  I was also graphing test and train values before, so the dates were way off.

I limited the plot to the first 300 entries so that I could visually observe the shift.



**Expected chart**



**Indicator tips**

**[David Hills](https://piazza.com/class/idadrtx18nie1?cid=1220)** [1 day ago](https://piazza.com/class/idadrtx18nie1?cid=1220)

KNN with k  = 3.   However, I have several indicators I'm using:  SMA, Bolinger value, STD, long term momentum (20 day), short term momentum (4 day), one day SMA momentum.      The long term and short term momentum ones are from our earlier create your own indicator exercise so they are not the definition I've seen in the forums for momentum - just   (S(t) - S(t-w))/S(t-w).    S(t) = stock price at time t and S(t-w) is stock price w days before t.  I've since found places on the web where this formula is used as a momentum indicator. So I apply it with w = 20 and 4.   It's really just rate of change.   The 4 day one is just the number of days I found worked well in the previous exercise.  The SMA momentum was one I found on the internet  SMA(t) - SMA(t-1)   and then modified to scale properly.   I've checked it several way and it's really just on top of each other.  I even double checked that I wasn't accidently including Y actual as one of my Xs.

**Normed SMA values**

Start with

    SMA\_value[t] = (price[t]/SMA[t]) - 1

Then take the mean and stdev of that value and use

    SMA\_normed[t] = (SMA\_value[t] - mean)/stdev

**Normalization**

I think having the value at 1.5 is not such a terrible thing. 150% is a huge jump and should be weighted as so. I think the idea of normalization is to get the values CLOSE to the range [-1, 1], not absolutely inside. But I may be wrong.

**~ An instructor (Tucker Balch) endorsed this answer  ~**

**the instructors' answer,**

The comment on the wiki about normalization is a suggestion that should improve the performance of your learner.  I don't care much about how you implement it specifically (or if you implement it at all).

X and Y Values

Your xValues have always been a multidimensional list; this is just saying that rather than two x values to describe each point for x, you'll be giving 3 x values. And the yValue, rather than being the price, would be the future five day return. In other words, for date d, your row in xTrain might look like this:

Date:                BollingerVal:               Momentum:                    Volatility:

2008-01-05      .09                            .05                                     .03

(The values are completely made up)

And the Y value for that particular date index, rather than being that days price, is the return for five day's ahead of that date.

Results : KNN and LinReg on ML4T-399

https://piazza.com/class/idadrtx18nie1?cid=1203

Below are the results for my analysis of KNN and Linreg on ML4T-399. Any views or suggestions?

**<b>**KNN **</b>**

In sample results (2008-2009)

RMSE: 0.00723369210084

corr: 0.99877245721

Out of sample results (2010)

RMSE: 0.0246193223584

corr: 0.986134091556

Time to execute KNN: 6.28100013733 secs

**<b>** Linear Regression **</b>**

In sample results (2008-2009)

RMSE: 0.0176006249429

corr: 0.992706603662

Out of sample results (2010)

RMSE: 0.0270382106194

corr: 0.983269214366

Time to execute Linear Regression: 0.0 secs

EDITED  
  
IBM Results

KNN

In sample results (2008-2009)

RMSE: 0.0333094738726

corr: 0.634691541075

Out of sample results (2010)

RMSE: 0.0303651227864

corr: 0.073906590106

Time to execute KNN: 6.65599989891 secs

Linear Regression

In sample results(2008-2009)

RMSE: 0.0426893226964

corr: 0.133096095555

Out of sample results(2010)

RMSE: 0.022077944431

corr: 0.242832259378

Time to execute Linear Regression: 0.00100016593933 secs

Bagging without Boosting (with KNN)

In sample results

RMSE: 0.031272031846

corr: 0.713271128055

Out of sample results

RMSE: 0.0280231002327

corr: 0.112257039356

Time to execute Bagging without Boosting (with KNN): 186.44299984

Balch: what are your features?

Would you expect out of sample to be the same or different than in sample for ML4T-399?  What about IBM?  If you have different expectations for the two, why?

 correlation > 0.12 is excellent.