Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner

Part3: Experiments, Custom Data

Intuition

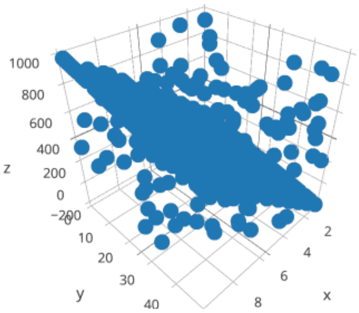
Both linear regression learner and KNN learner aim to predict value of Y given X inputs, where Y is a function of X. However, both methods take a different approach.

Linear regression assumes data falls on a straight line, plus or minus some noise, and hence the value of Y is equal to the value of f(X) multiplied by slope of the line. Linear regression will have an idea of f(X) for any given X without any other information.

On the other hand, K-Nearest Neighbor does not worry about how the data looks like and it does not model the data. The data representing Y could be anything from a line to a circle. The algorithm assumes that f(X) will be similar to its k number of nearest neighbor; f(X1), f(X2), etc. This assumption would be roughly true for most models including for linear models.

Linear Regression Learner

Using the above intuition, Linear Regression Learner would perform better on linear data as it would be able to better fit a model over this type of data unlike KNN Learner which makes an assumption. KNN Learner would be able to make good prediction over linear data however with much larger RMSE than Linear Regression. On top of that if the linear data contains noise, KNN Learner is bound to perform worse. The below generated data contains 5% noise.



Data:

X1 = Inches of snow from 1-10

X1 = Temperature in Fahrenheit from 1-50

Y = Chance of school cancellation, calculated as (10*(10*S))- (3*(2*T)). Note, this is not a percentage. Higher the number, more the chance.

Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner

In the above 3D model X represents inches of snow, Y represents temperature and Z represents chance of school closure. As you notice more it snows there are more chances of school closure and the closure chances increase even higher as temperature drops.

Result:

Linear Regression Learner
In sample results
RMSE: 163.558088895
corr: 0.855472982344

Out of sample results
RMSE: 148.399824734
corr: 0.870815868928

KNN Learner
In sample results
RMSE: 132.076829493
corr: 0.86615405167

Out of sample results
RMSE: 169.647592248
corr: 0.827247153418

We can see that on out of sample data, Linear Regression learner performs about 5% more accurately than KNN learner. For in sample data, KNN performs just slightly better and this is expected as 1 out of 3 k points will contain the actual value itself.

Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner

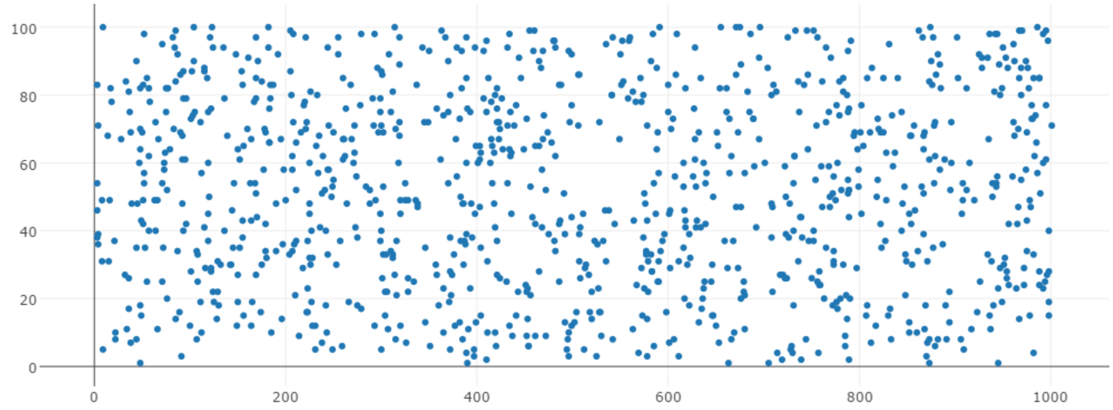
KNN Learner

KNN learner does not care how the data looks like. It could be line, circle, parabola, or sin wave. Assumptions are made using nearest k points. On the other hand Linear Regression learner would struggle with non-linear data.

Data:

Many factors decide how much rain a particular region in country receives. Two of the factors are average temperature and distance from ocean. We assume the country is surrounded by ocean on all four sides. Distance is measured from center of country.

X1 = Average temperature from 0-100
X2 = Distance to Ocean from 1-1000
Y = Amount of Rain calculated as (Average Temperature/Distance from Ocean) * 100



Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner

Result:

In the above we expect KNN Learner to do much better than Linear Regression as it is hard to fit a line that model this type of data. We expect the coasts to get more data than inner parts of the country.

Linear Regression Learner
In sample results
RMSE: 114.958911903
corr: 0.335086603318

Out of sample results
RMSE: 38.5666126689
corr: 0.601890964949

KNN Learner
In sample results
RMSE: 36.2800366919
corr: 0.954777654012

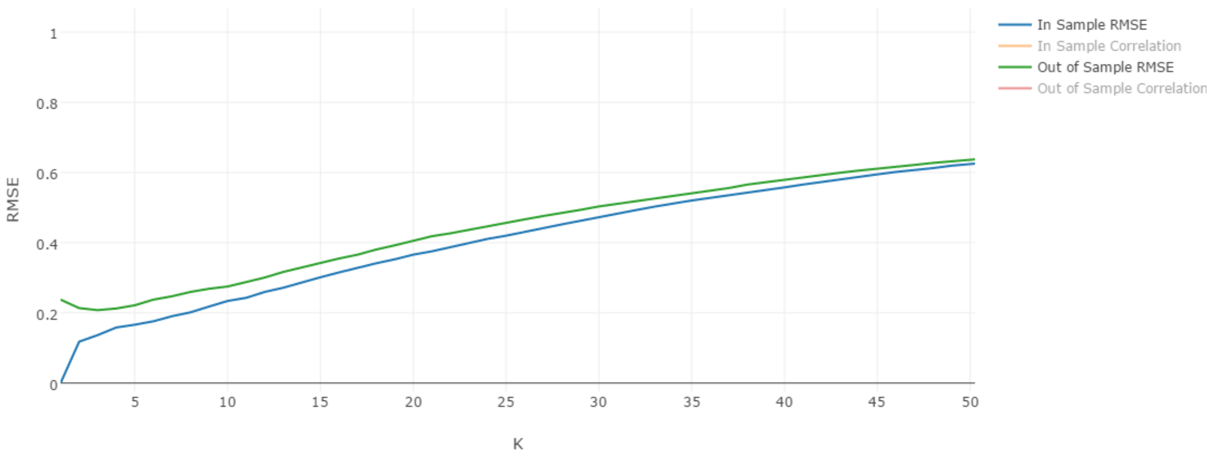
Out of sample results
RMSE: 25.395180953
corr: 0.926370115329

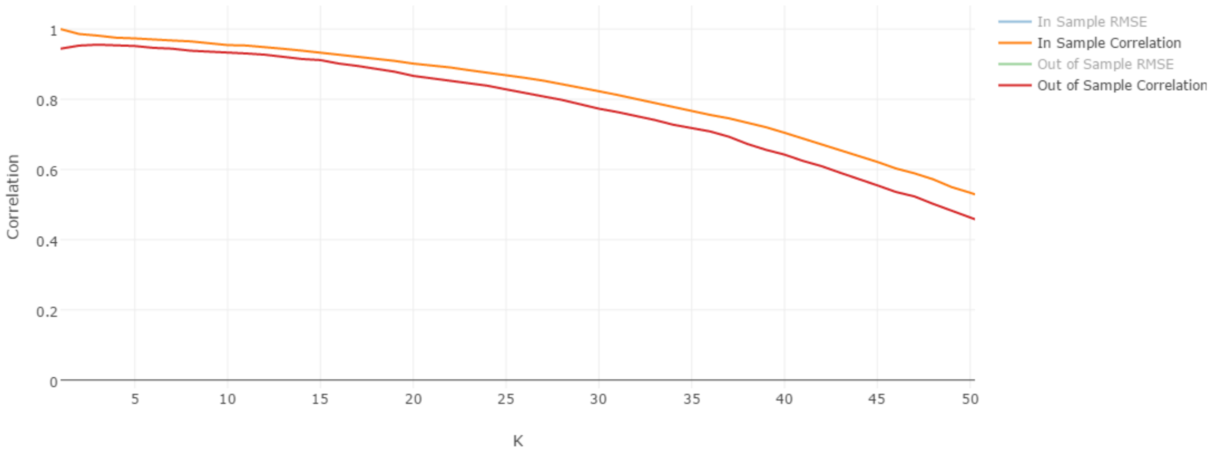
Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner

Questions

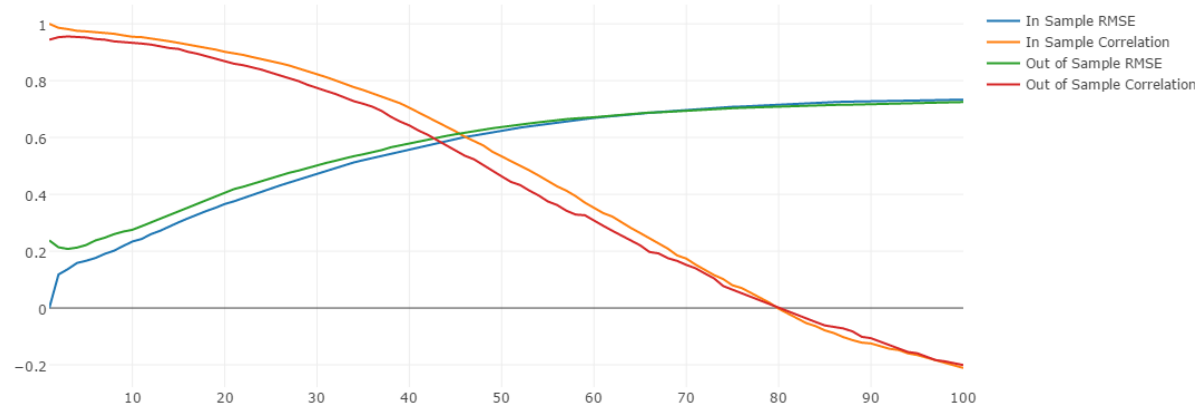
Consider the dataset ripple with KNN. For which values of K does overfitting occur? (Don't use bagging).

I ran test for k=1 to k=100. Here is the result.





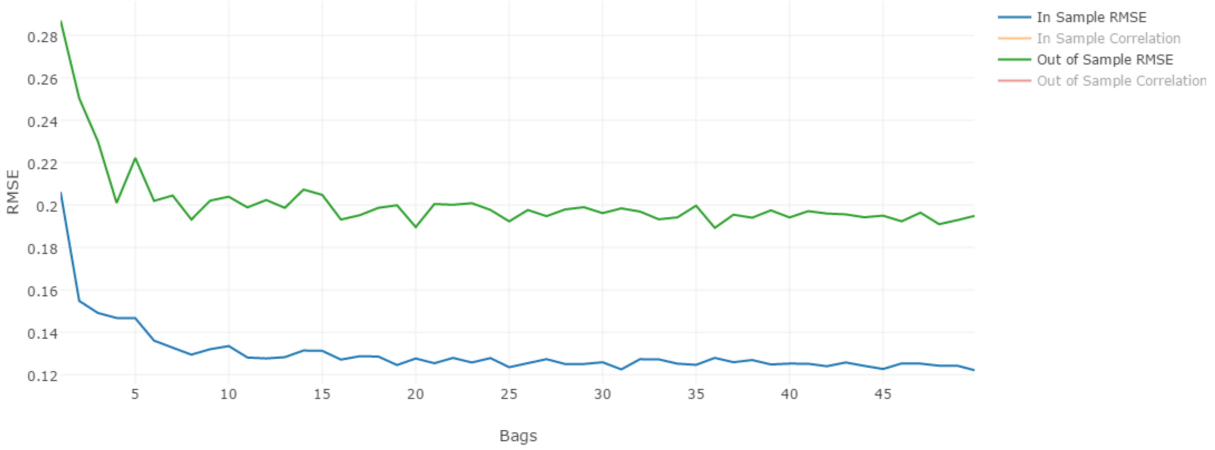
Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner



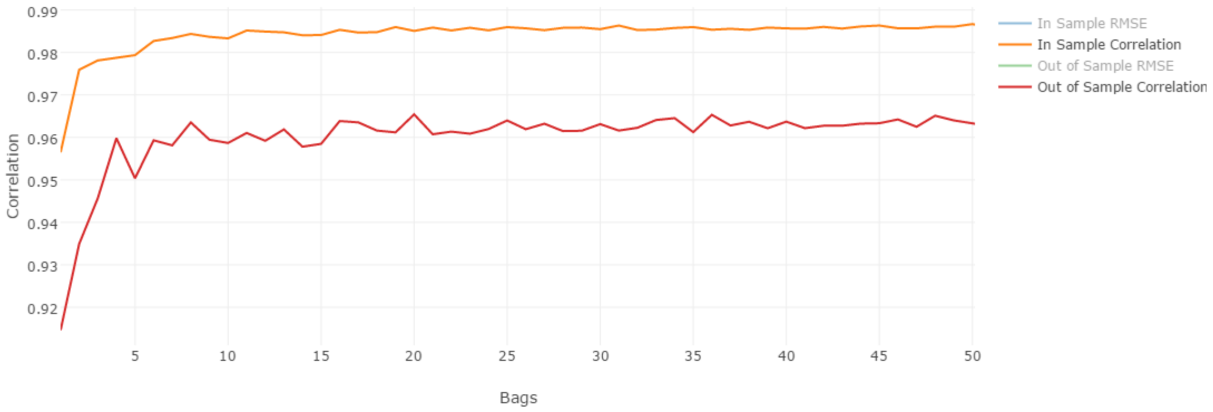
From the above result, for ripple dataset k=3 gives highest correlation and lowest RMSE. Any higher value for k increases RMSE and also decreases correlation.

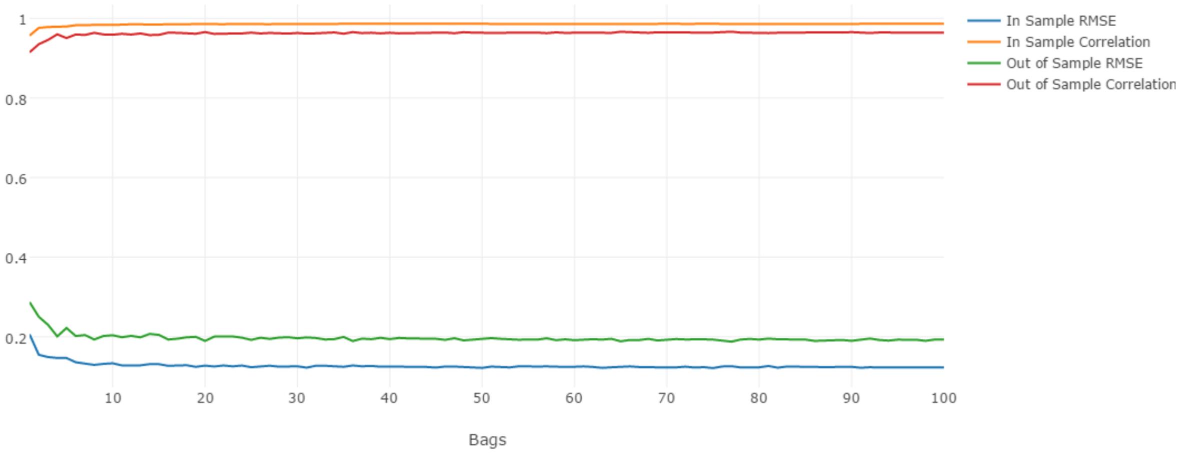
Now use bagging in conjunction with KNN with the ripple dataset. How does performance vary as you increase the number of bags? Does overfitting occur with respect to the number of bags?

I ran test for bags=1 to bags=100 keeping k=3. Here is the result.



Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner



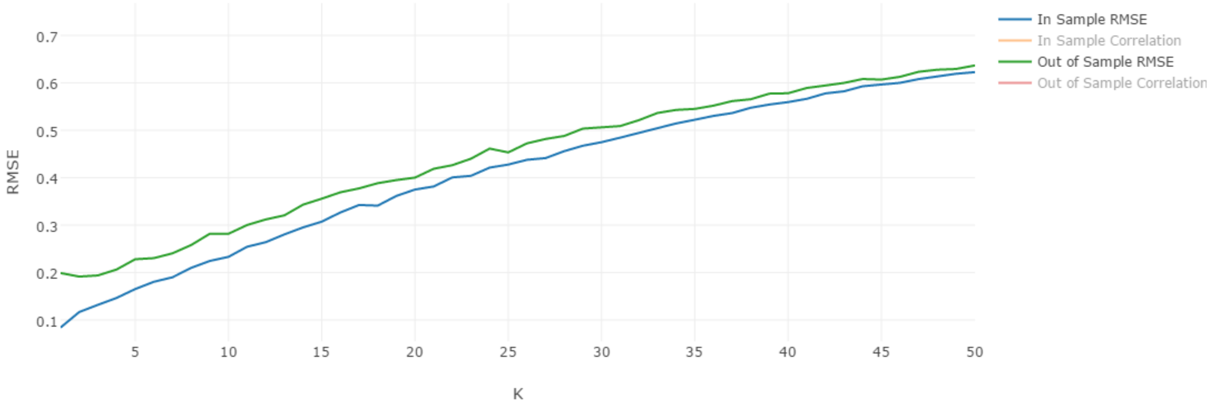


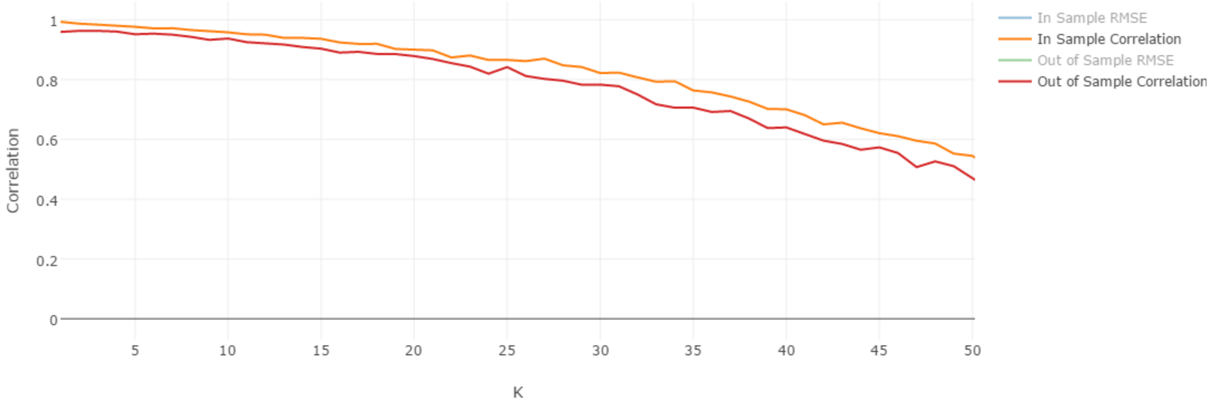
From the above result, we can see that overfitting does not occur with respect to the number of bags. However, at some point there is also no extra value gain by increasing number of bags and the algorithm will perform slower. In my opinion, for ripple dataset ideal number of bags is 10.

Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner

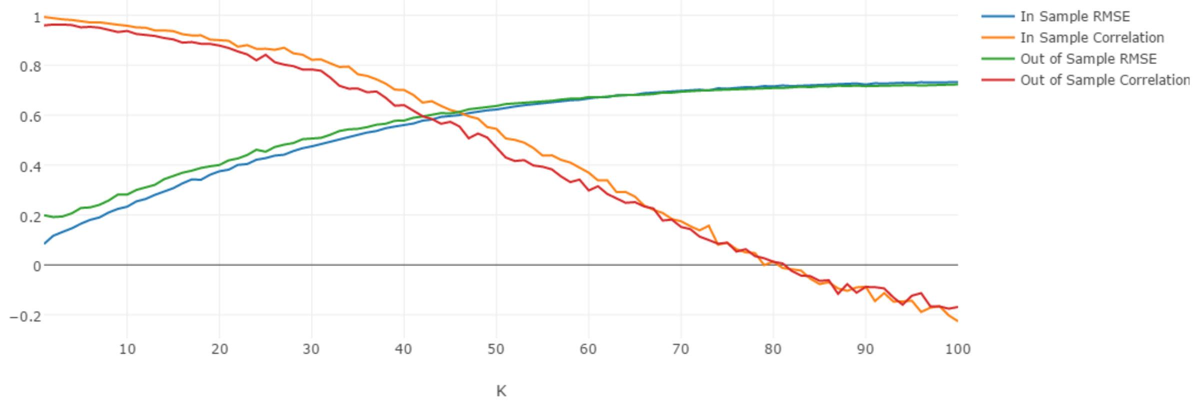
Can bagging reduce or eliminate overfitting with respect to K for the ripple dataset?

I ran test for k=1 to k=100 keeping bags=10. Here is the result.





Nakul Patel
Machine Learning For Trading
MC3-Project-1 – Linear Regression Learner & KNN Learner



From the above result, we can see that overfitting does occur when we increase K even with bagging.