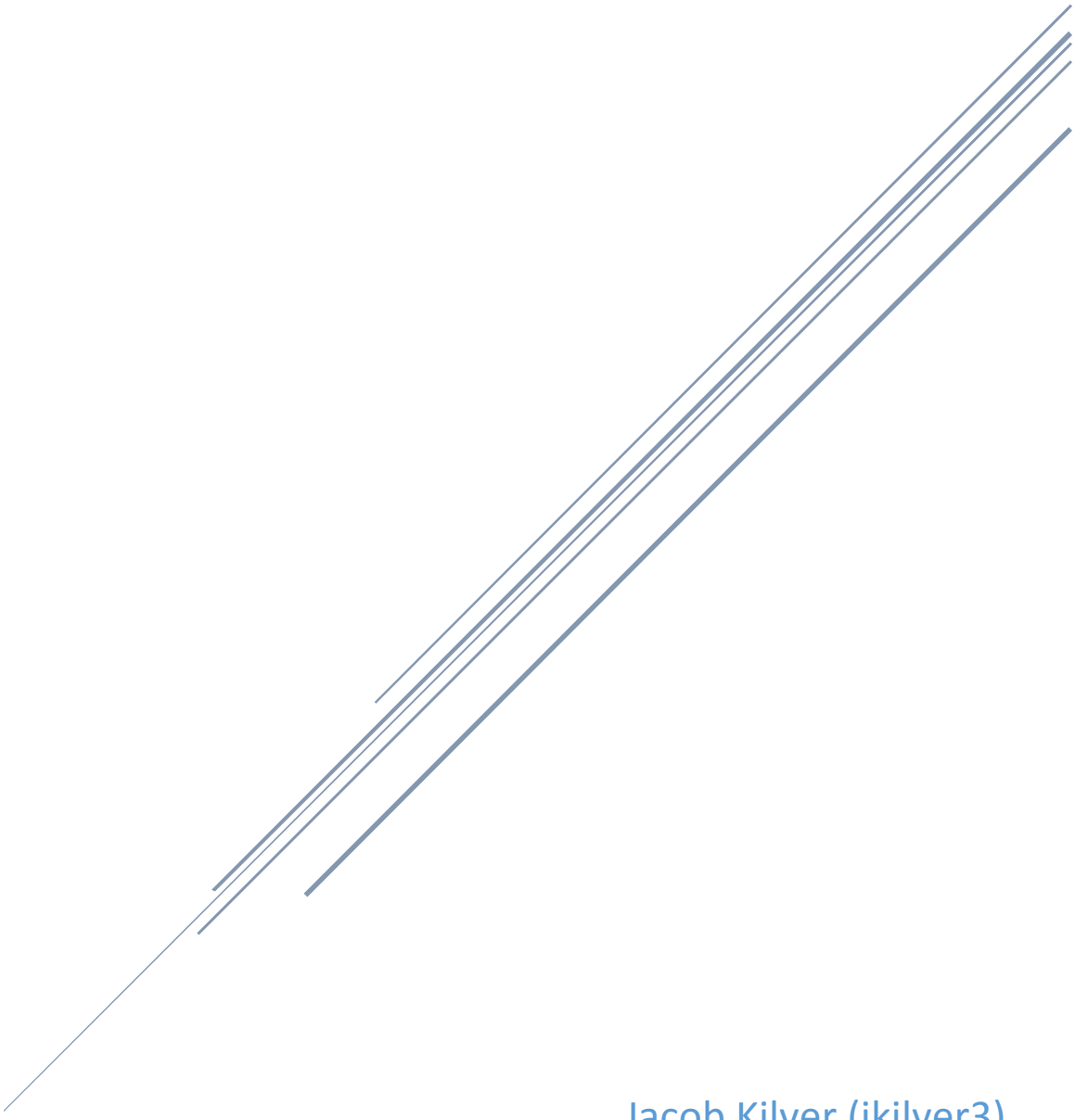
MC-3 PROJECT 1

KNN and Bagging



Jacob Kilver (jkilver3)
4 April 2016

1. Data better for Linear Regression

To create a data set that performs better for the linear regression learner than the KNN learner requires creating a data set that fits the model assumed by the linear regression learner. Namely this has to be a linear equation without noise since the linear regression learner assumes a linear equation. Any linear equation will work. For this report the following linear equation was used:

$$4 * X1 + 2 * X2 = Y$$

where X1 and X2 are the input data points and Y is the output variable. Since this is a linear equation in 2 dimensions it forms a plane. To adequately cover the input space, X1 and X2 were generated randomly over the interval [0, 1) and fed into the equation above. Figure 1 below shows a plot of the truth data points (red) versus the predicted values using the linear regulator learner (blue) on the out of sample data set. It should be noted that the truth values cannot be seen here because the linear regression learner *exactly* predicted the values. This is supported by the root mean square error computed in Table 1. The error is essentially zero and the correlation is exactly one, indicating that the linear regression learner exactly predicted the truth values.

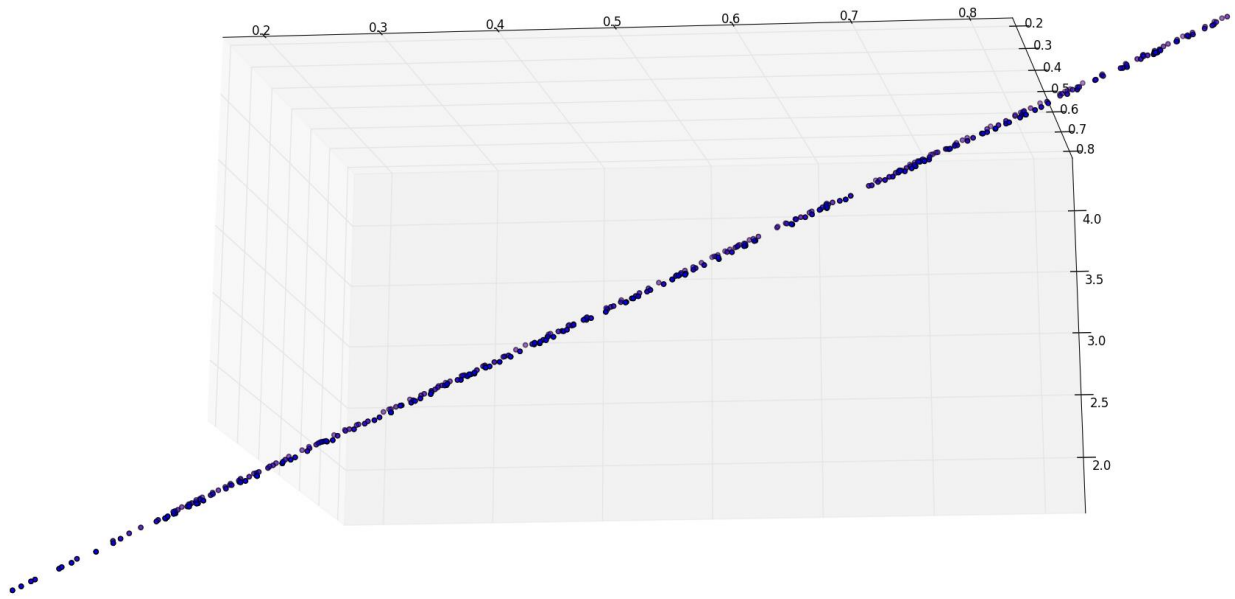


Figure 1: Plot of truth values versus predicted values using Linear Regression learner on out of sample data

Table 1: Metrics from out of sample test of linear regression learner

RMS Error	2.42E-15
Correlation	1.0

Given the results above, the KNN learner had a hard time matching these results. Figure 2 shows the truth values (in red) and the predicted values from the KNN learner (in green). As can be seen, the results are not as accurate as with the linear regression learner. The predicted values from the KNN learner do not match the truth values perfectly as with the linear regression learner. Furthermore, Table 2 shows that the error is significantly greater for the KNN learner than with the linear regression learner.

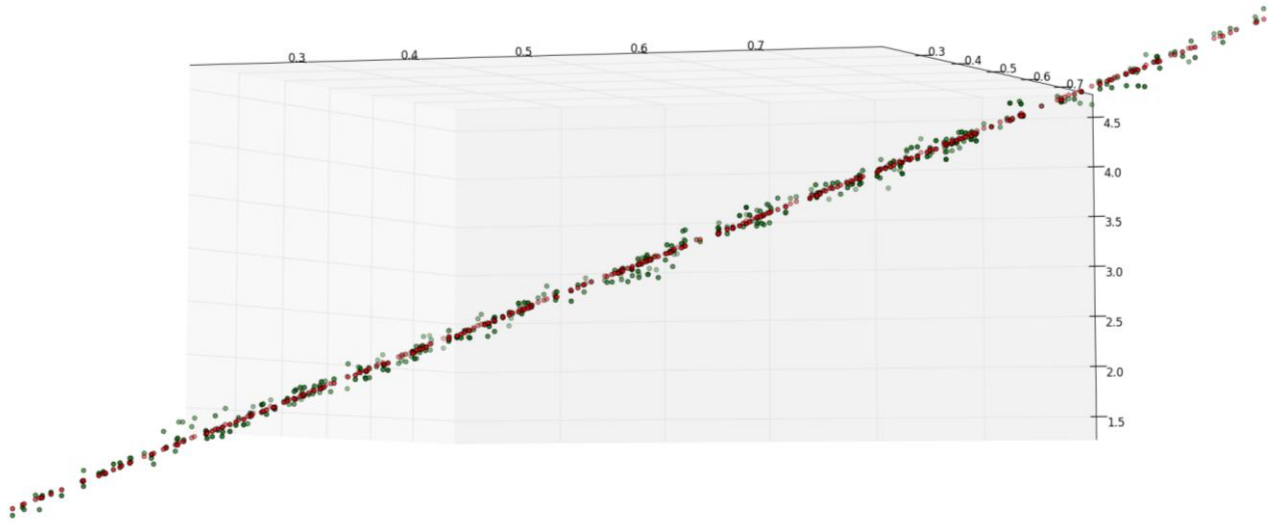


Figure 2: Plot of truth values versus predicted values using KNN learner on out of sample data

Table 2: Metrics from out of sample test of KNN learner

RMS Error	0.066027
Correlation	0.998732

2. Data better for KNN Learner

Since the linear regression learner assumes a linear equation, any non-linear equation (e.g., a polynomial) can be learned better by a KNN learner than a linear regression learner. The KNN learner does not make any assumptions on the data beforehand, so it can learn the polynomial better than a learner that incorrectly assumes the data has a linear form. For this report, the following polynomial was used:

$$X1^2 + 2 * X2 = Y$$

where X1 and X2 are the input variables and Y is the output variable. This forms a parabola in three dimensions. To adequately cover the input space, X1 and X2 were generated randomly over the interval [0,1) and fed into the equation above to generate the truth output variables. Figure 3 shows a plot of the truth values (red) and the predicted values using the KNN learner (green) on the out of sample data set. The predicted data clearly follows the path of the parabola. This is much better than the predicted values from the linear regression learner in Figure 4. As can be seen, the predicted values from the linear regression learner (blue) all lie on a plane. Thus, the linear regression learner is unable to successfully predict values in some regions of the input space because the underlying function does not fit its assumptions. Table 3 and Table 4 contain summaries of important metrics for these tests. The KNN learner is approximately 4 times more accurate than the linear regression learner on this data set.

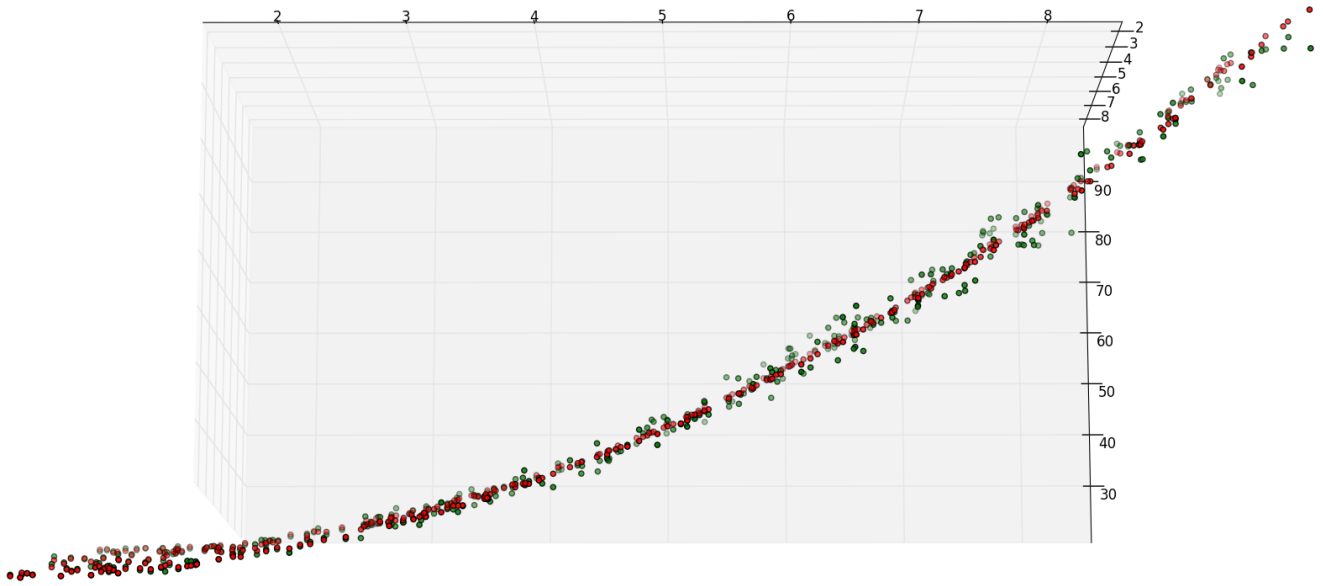


Figure 3: Plot of truth (red) and predicted values using KNN learner (green)

Table 3: Metrics from out of sample test of KNN learner

RMS Error	1.791349
Correlation	0.998075

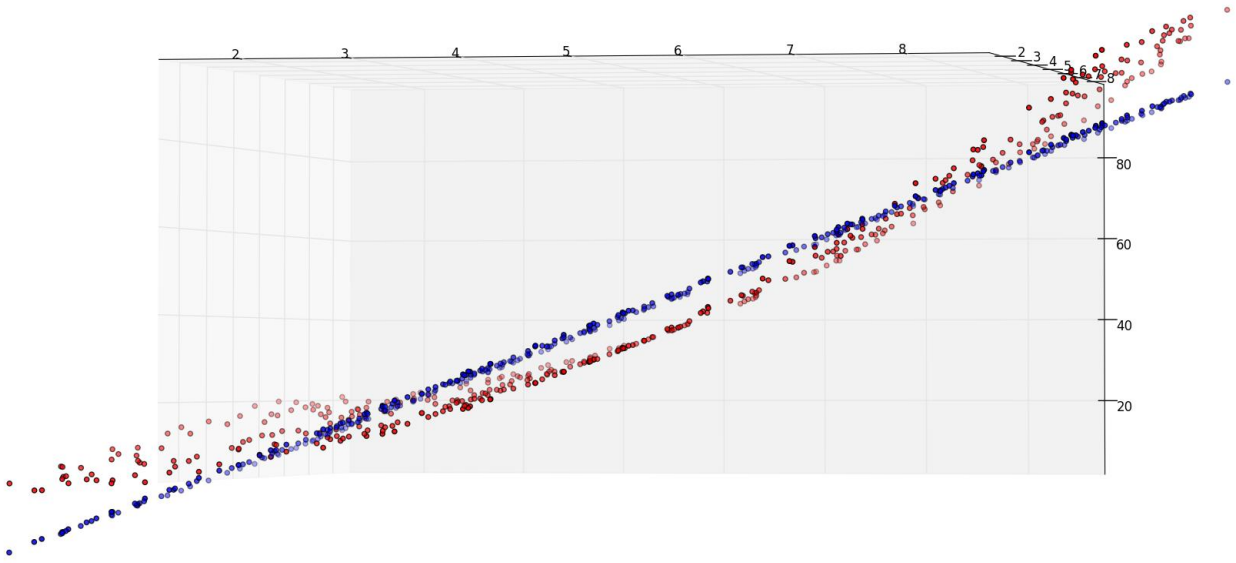


Figure 4: Plot of truth (red) and predicted values using linear regression learner (blue)

Table 4: Metrics from out of sample test of linear regression learner

RMS Error	7.431321
Correlation	0.967672

3. Analysis of KNN Learner with fixed data set

For these tests the ripple data set provided by the instructor was used.

a. Without bagging

Figure 5 and Figure 6 show the in sample out of sample error respectively using the ripple data set provided by the instructor with the KNN learner. As expected, the in sample error for k=1 was zero, since each query point returned the corresponding training point. As the value of k increased, more and more training data points were averaged for each query point. This averaging resulted in higher error as the value of k increased. While k=1 has the lowest error, it is actually over fit when viewed with the out of sample data.

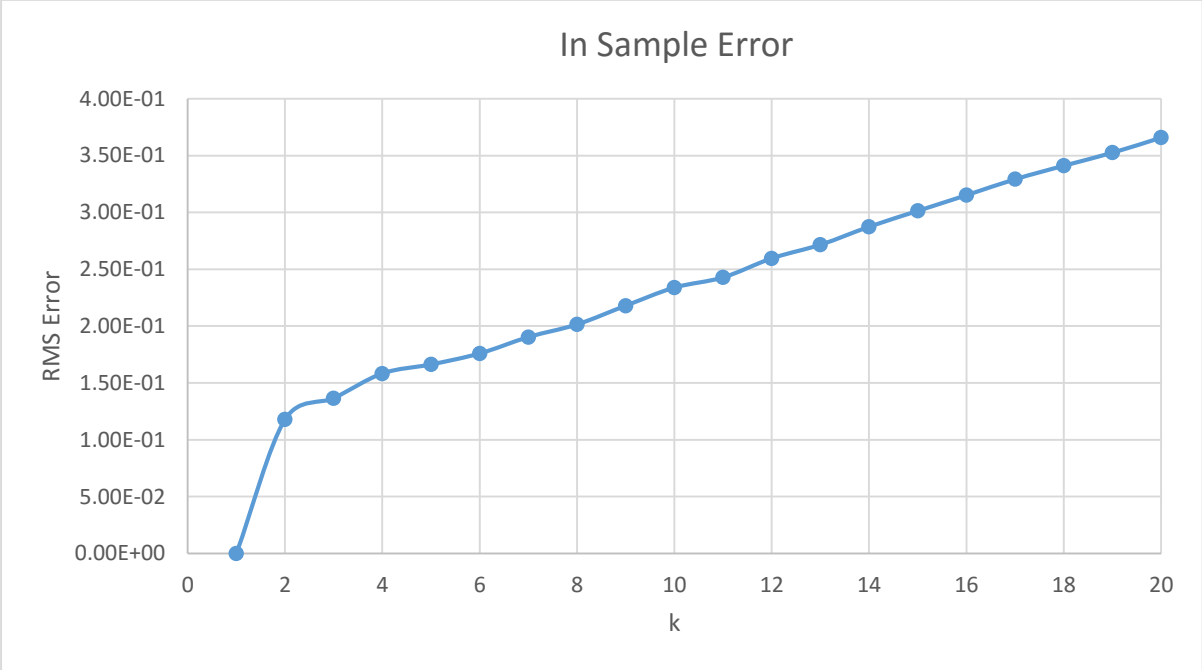


Figure 5: Plot of root mean squared error as a function of k using the in sample data set

Overfitting is when the data is believed to be more accurate than it actually is, leading to higher error on data that has not been observed. For the out of sample data set, overfitting occurs for k = 1 and 2. At k = 3, we have the minimum error. For values of k greater than three, we have underfitting. In this case the data is believed to be less accurate than it actually is, and so some information that is contained in the data is thrown away (through averaging), again leading to higher error.

It could be argued that for k = 2, 3, and 4 that neither overfitting nor underfitting occur since all these errors are almost equal. This specific behavior could also be caused by the particular data set that was used here. The general principle here is that overfitting occurs when too few data points are averaged and underfitting occurs when too many data points are averaged.

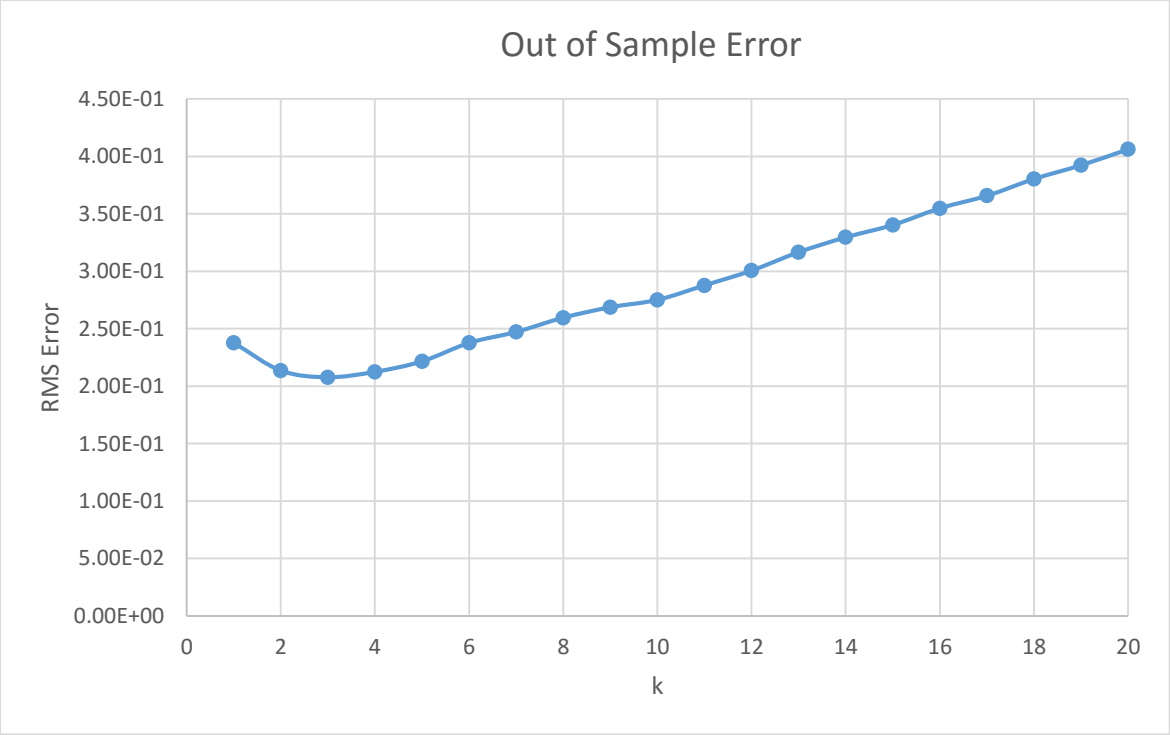


Figure 6: Plot of root mean squared error as a function of k using the out of sample data set

b. With bagging

Bagging is a process whereby multiple learners are trained simultaneously. At query time, each learner is consulted individually and the results from the collection of learners are averaged to produce the final answer. Figure 7 and Figure 8 show the in sample and out of sample errors respectively for the KNN learner on the ripple data set as the number of bags is changed for k=1, 3, 5, 10, and 20.

Of most interest for the in sample set are the results for k = 1. This still has the lowest error for all values of k. The error for 1 bag was not expected to be zero because the samples for each "bag" were drawn with replacement, so necessarily not every training sample exists in the single bag for the KNN learner. For this reason the error for bags = 1 was expected to be the highest for all the learners, which is the case. As the number of bags is increased, the likelihood of the in sample data point being returned is increased, so for k=1 there should be some learners that have no error and some that have some error. When averaged together, the error is between zero and the maximum error from any single learner. For all other values of k, the error is between some non-zero value and the maximum error from any single learner, so their error is necessarily greater.

As the number of bags is increased, error tends to decrease, but eventually the averaging of the learners fails to produce significantly different results. It is interesting to note that for all except k=1, the error in the limit as the number of bags is increased settles on a value very close to the in sample error without bagging for that value of k.

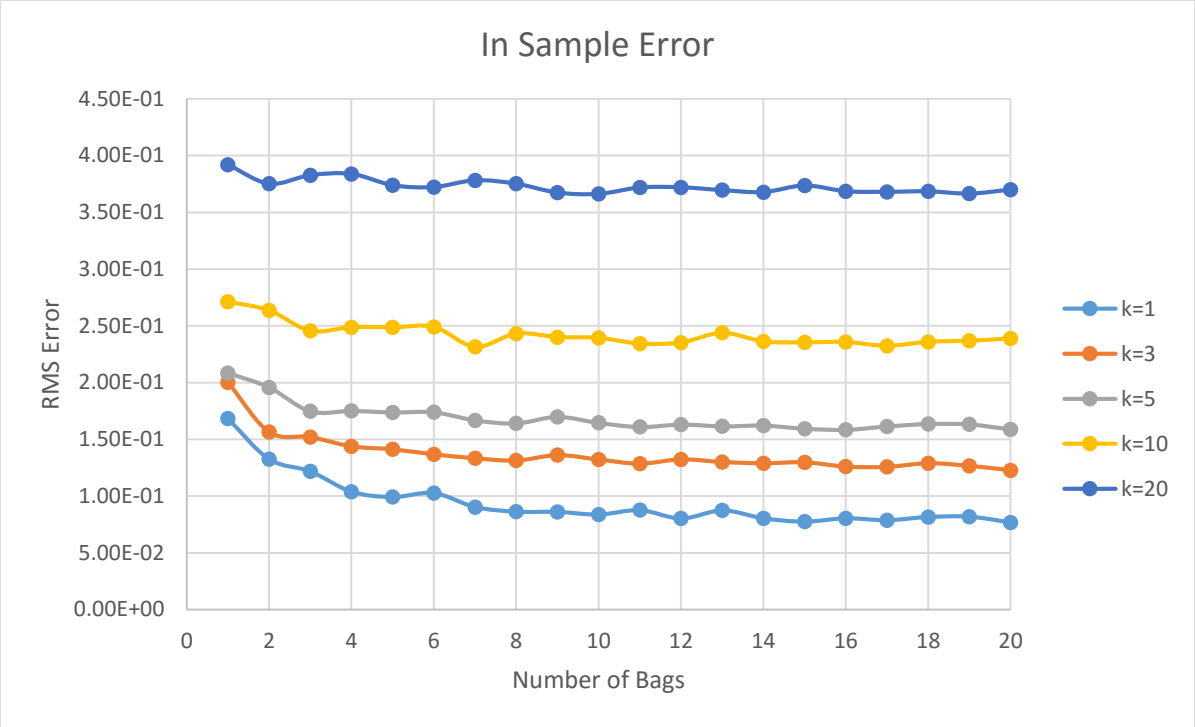


Figure 7: Plot of in sample error as a function of the number of bags used with a KNN learner

For the out of sample data set, for values of k greater than 3 the error still depends on the value of k selected. Choosing too many data points for each query dilutes the information contained in the data, yielding poorer results. In this sense bagging cannot eliminate underfitting for a particular value of k. However, bagging seems to eliminate overfitting quite well. For k=1 with bagging, the error in the limit is the same as the minimum error without bagging. In this case bagging accomplishes what the KNN learner does without bagging with k=3. Only the closest single point for each learner is consulted and the output from each is averaged together. This is somewhat like taking the three closest points and averaging those together for the output. Thus, k=3 and k=1 have almost identical errors in the limit of number of bags.

As the number of bags is increased the error tends to decrease. The difference is more pronounced for lower values of k since the averaging that is done for higher values of k was already being done. For all values of k besides k=1, the error in the limit as the number of bags is increased settles on a value very close to the results observed for that value of k without bagging. For k=1, the overfitting is virtually eliminated and settles on an error very close to that of minimum error observed.

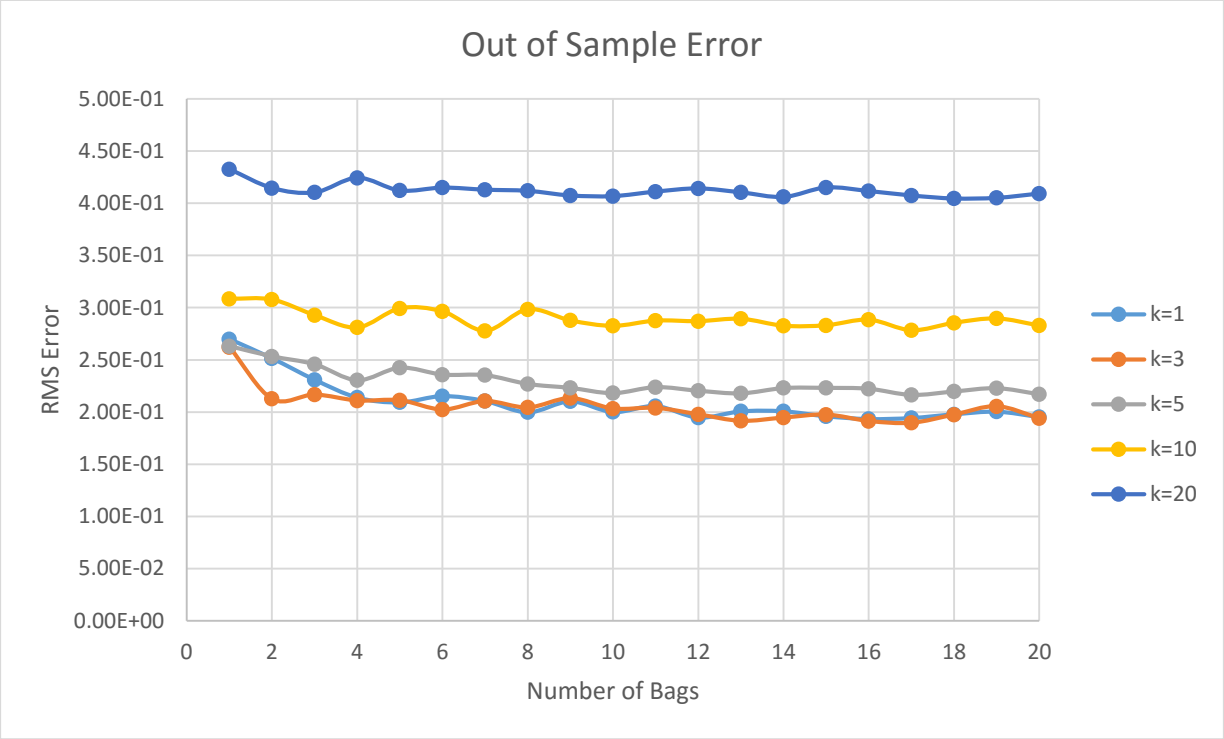


Figure 8: Plot of out of sample error as a function of the number of bags used with a KNN learner