ROB 313: Assignment 1

Ali Seifeldin 1003894431

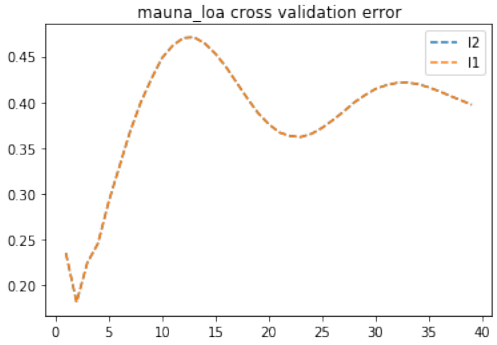
In this assignment, we want to investigate k-NN and SVD algorithm in order to solve regression and classification problems. The report includes all data that was asked for, which each individual coding requirement in its own individual iJupyter file.

1)

We start off by solving the regression problems using KNN algorithms.
First off, we combine X_train and X_val, shuffle them then and put them through a cross validation loop that creates 2 sets of n times, where 1/n of the set is the validation set and the rest is the training set. We parse the X values through the metric function, and then partition the smallest k elements in them. We then find the related Y results in those indexes and return the average result. Once we find the optimal K value (by minimizing the error), we use that K value and use all the X_train data in order to predict values for X_test. We then compare the predicted values to Y_test

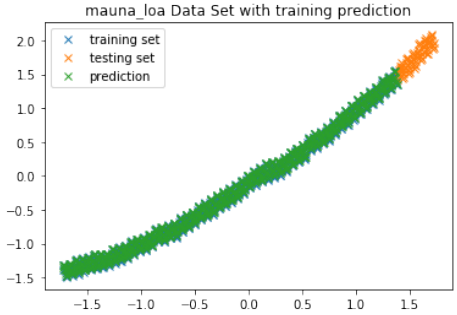
The table below shows the results from our models. The Rosenbrock and pumadyn32nm appear to not function as expected, due to error in my code. We know that since the cross-validation error is much higher than the test RMSE. In addition, the values are much higher than expected.

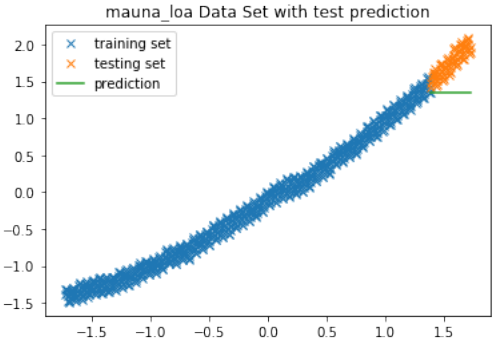
Dataset	Preferred K values	Preferred distance metric	Cross Validation RMSE	Test RMSE
Mauna loa	2	Same result	0.1811	0.4407
Rosenbrock	1	L2	0.0177	0.0075
pumadyn32nm	25	L1	0.0227	0.0290



For the Mauna_loa model, we can observe that that the l2 cross validation curve has a minimum at k = 2. The model drops when 2 neighbors are used (average between 2 points) and loses accuracy as more neighbors are taken into account as we start deviating from the query point.

In addition, as seen on the 2 graphs below, the predictions overlay well with the training set, but when we are using the training set to make predictions on the testing set, we hit a ceiling. Since we are trying to predict values that are higher than values we have seen, the model isn't able to predict them. We can solve this issue by better shuffling the data set throughout training, testing and validation





2)

For the Classification problem, the same concept is applied as the regression, but instead of averaging out all of the k elements predicted, we find the maximum index as use that at the predicted value. We then test for accuracy by comparing the % of correct indexes predicted. Using NumPy greatly speeds up the calculation from the parallelization of task by employing more matrix algebra.

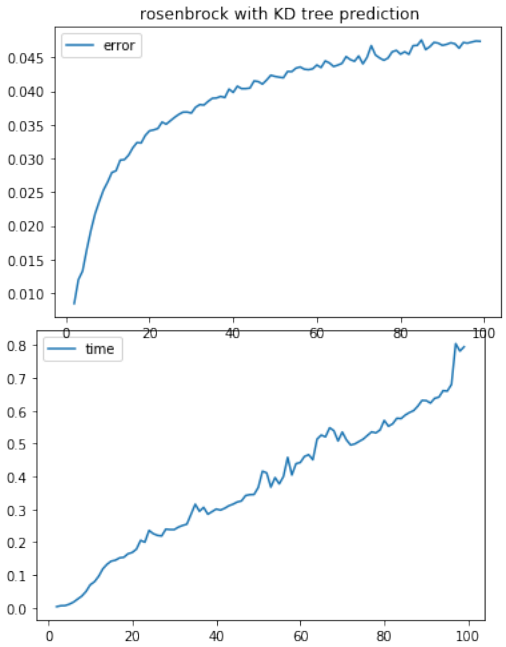
Below we summarize the results.

Dataset	Preferred K values	Preferred distance metric	Validation Accuracy	Test Accuracy
Iris	10	L1	1	1
Nmist-Small	1	L2	1	.958

3)

Using a KD tree greatly speeds up the process. Making predictions for all x_test dataset with d = 2 only takes less than 0.004 of a second, as opposed to 4-5 seconds using the brute force approach. These tests were conducted with n=5000 and k = 5

This greatly speeds up the process. Error and time were plotted, with respect to d. The larger d is, the more time it takes to compute. Even with d at 100, it still takes less than a second. We find the minimum error from the dataset to occur with d = 2, as see on the graph, before the error starts to increase again using.



We can also use this model to find the optimal k neighbors associated with the minimum error, just for fun. When k = 3 (d=2, n=5000), we get the minimum RMSE error with the testing set, which is close to the previous value we got of k=2 in part 1

4)

Using the SVD algorithm in order to predict the weights and create a model, we are able to make better predictions for the Mauna_loa, but worse predictions for the Rosenbrock and pumadyn32nm regression data set. Mauna_loa was the exemption in this case as the way the data set is formatted, using KNN would always plateau to a maximum value, as all the values in the test set were larger than all the training sets. This allows the linear model made using SVD to be better, but the same cannot be said for the other 2 regression problems.

Using SVD for classification, we can see that both models have an above 85% accuracy, so they do relatively well, but not as well as the nearest neighbor equivalent.

Dataset	Test RMSE
Mauna loa	0.3072
Rosenbrock	0.0311
pumadyn32nm	0.0403

Dataset	Test Accuracy
Iris	0.867
Nmist-Small	0.858