CS 760

Homework 2

Submitted by: Sneha Rudra

Part 1

The file kNN.py implements a k-nearest neighbor learner for both classification and regression.

- kNN.py reads the training set and test set files in the ARFF format. The file kNN.py can parse ARFF files that are similar in format to the example ARFF files provided (wine_train.arff, wine_test.arff, yeast_train.arff, yeast_test.arff). The example ARFF files provided did not have any comments (lines starting with %) and hence kNN.py assumes that any ARFF files that need to be read in do not contain any comments. As mentioned in the guidelines, all the features must be assumed to be numeric and hence kNN.py assumes all attributes except the last one to be numeric attributes.
- kNN.py is callable from the command line and accepts three arguments - training set file, test set file and k.
- To run the file kNN.py using the script file kNN, the above three arguments should be provided in line 2 of the script file. For example, if the program is being run for the yeast dataset and k=3, the line 2 of the script file kNN should read (Please note the format in which the arguments should be provided):

  python kNN.py yeast_train.arff yeast_test.arff 3
- Subsequently the script file kNN can be executed through the following command (in the appropriate path where kNN.py, the training set arff file and test set arff file are located):

  [sneha@royal-18] (25)$ kNN

Part 2

Explore the effect of the parameter ‘k’ on the predictive accuracy.

- **For the yeast data set, draw a plot showing how the test accuracy varies as a function of k. Your plot should show the accuracy for k = 1,5,10,20, 30**

$$\text{Test Accuracy} = \frac{Numberofcorrectlyclassifedtestinstances}{Totalnumberoftestinstances}$$

Table ***Error! No sequence specified.*** Yeast Test Set Accuracy for different values of k

k	Yeast Test Accuracy
1	0.4831460
5	0.5483146
10	0.5685393
20	0.5730337
30	0.5617977

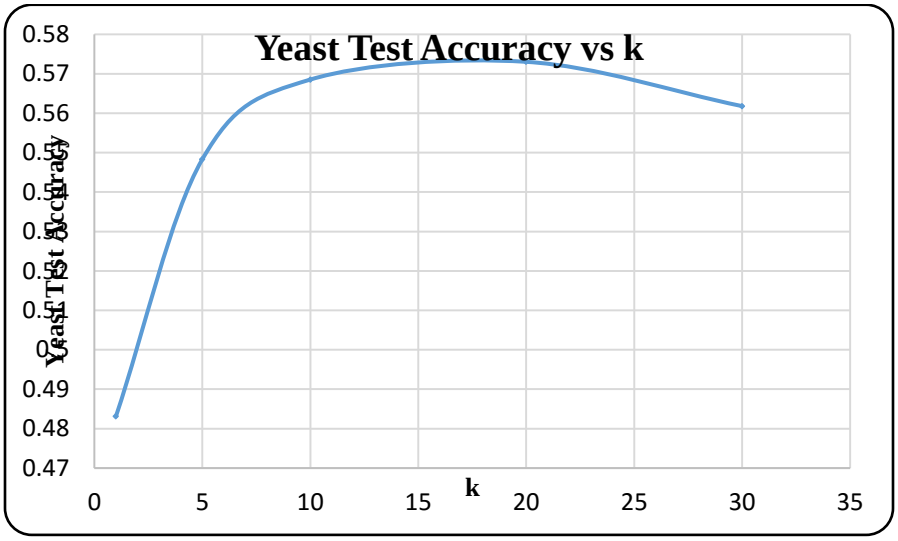

Figure ***Error! No sequence specified.*** Variation of Yeast Test Accuracy as a function of k

As can be seen from Figure 1, the yeast test accuracy increases upto a certain k after which it starts decreasing.

- **For the wine data, draw a similar plot showing the test-set mean absolute error as a function of k, for k = 1,2,3,5,10**

$$\text{Test Set Mean Absolute Error} = \frac{\sum_{i=1}^{n} f_i - y_i \vee}{n}$$

Where:

n = Number of test set instances

f_i = Predicted wine score

y_i = True response/wine score

Table ***Error! No sequence specified.*** Wine test set MAE for different values of k

k	Wine Test Set

	MAE
1	0.69553805
2	0.69568387
3	0.68844172
5	0.68148148
10	0.65823855

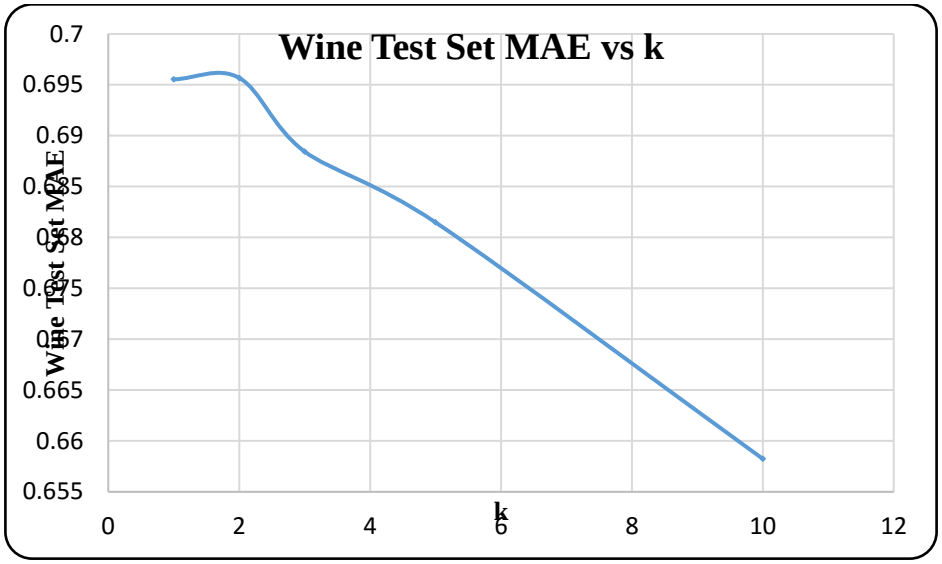

Figure ***Error! No sequence specified.*** Variation of wine test-set MAE as a function of k

From Figure 2, for the wine test set, the MAE, in general decreases as k increases (for the values of k in the question).

- **For the yeast data set, construct confusion matrices for the *k* = 1 and *k* = 30 test-set results. Show these confusion matrices.**

Table ***Error! No sequence specified.*** k=1 Yeast Test Set Confusion Matrix

		Actual Class									
		CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL
Predicted Class	CYT	69	46	26	3	3	0	0	3	1	0
	NUC	39	57	12	9	0	0	0	0	1	0
	MIT	21	13	34	2	2	1	1	1	1	0
	ME3	4	6	2	32	3	0	0	0	0	0
	ME2	1	0	3	1	6	3	1	0	0	0
	ME1	0	0	0	0	1	6	3	1	0	0
	EXC	1	0	2	0	2	3	5	0	0	0
	VAC	2	1	0	0	0	0	0	1	0	0
	POX	1	0	2	0	1	1	0	0	5	0
	ERL	0	0	0	0	0	0	0	0	0	0

Table ***Error! No sequence specified.*** k=30 Yeast Test Set Confusion Matrix

		Actual Class									
		CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL
Predicted Class	CYT	89	48	25	3	4	0	0	2	6	0
	NUC	36	59	5	2	2	0	0	3	0	0
	MIT	11	13	45	2	3	1	1	1	2	0
	ME3	1	3	1	40	3	0	0	0	0	0
	ME2	0	0	1	0	3	1	0	0	0	0
	ME1	1	0	3	0	3	9	4	0	0	0
	EXC	0	0	1	0	0	3	5	0	0	0
	VAC	0	0	0	0	0	0	0	0	0	0
	POX	0	0	0	0	0	0	0	0	0	0
	ERL	0	0	0	0	0	0	0	0	0	0

Briefly discuss what the matrices tell you about the effect of *k* on the misclassifications.

As can be seen from the above two confusion matrices for the yeast test set, we have more entries on the diagonal for k=30 compared to k=1, meaning that we have fewer misclassifications for k=30 compared to k=1, consistent with the observation in Figure 1.

Part 3

Using the k-d tree and the training set, **show how the nearest neighbor for the query point x^q = (7,10) would be found**. Assumptions: For all instances, the features have integer values. Distance metric is Euclidean and coordinates of instances in the figure are obtained from the table.

Solution

The nearest neighbor for the query point x^q can be found by using the following algorithm (from instance-based lecture slides):

NearestNeighbor(instance x^q):

PQ ={}

Best_dist = ∞

```
PQ.push(root,0)
while PQ is not empty:
        (node, bound)= PQ.pop()
        if bound >= best_dist:
                return best_node.instance
        dist = distance(x^q, node.instance)
        if dist < best_dist:
                best_dist = dist
                best_node = node
        if (q[node.feature] – node.threshold)>0:
                PQ.push(node.left, x^q[node.feature]-node.threshold)
                PQ.push(node.right,0)
        else:
                PQ.push(node.left,0)
                PQ.push(node.right, node.threshold-x^q[node.feature])

return  best_node.instance
```

For every step in the search, show the distance to the current node, the best distance found so far, best node found so far and the contents of the priority queue

	Distance	**Best Distance**	**Best Node**	**Priority Queue**
		∞		{(f,0)}
Pop f	$\sqrt{(6-7)^2+(3-10)^2}=7.07$	7.07	f	{(c,1), (h,0)}
Pop c	$\sqrt{(5-7)^2+(10-10)^2}=2$	2	c	{(h,0), (e,0), (b,0)}
Pop h	$\sqrt{(12-7)^2+(5-10)^2}=7.07$	2	c	{(e,0), (b,0),(g,5),(i,0)}
Pop e	$\sqrt{(2-7)^2+(4-10)^2}=7.81$	2	c	{(b,0),(g,5),(i,0),(d,0)}
Pop b	$\sqrt{(3-7)^2+(12-10)^2}=4.47$	2	c	{(g,5),(i,0),(d,0),(a,4)}
Pop g	return c			

Hence the nearest neighbor for the given query point is c = (5,10) which is at a distance of 2 units away from the query point.