1. Part One (SVM and perceptron)

   Here, we are providing accuracy which is just 1 - error (or error rate).

   Linearn Kernel (0)

		Prediction outcome	
		One (1)′	**Zero (0)′**
actual value	**One (1)**	14	2
	Zero (0)	3	16

   So, the accuracy is 85.7143 % (30 out of 35).

   Polynomial Kernel (1)

		Prediction outcome	
		One (1)′	**Zero (0)′**
actual value	**One (1)**	9	7
	Zero (0)	2	17

   So, the accuracy is 74.2857 % (26 out of 35).

   Radial Basis (2)

		Prediction outcome	
		One (1)′	**Zero (0)′**
actual value	**One (1)**	11	5
	Zero (0)	3	16

So, the accuracy is 77.1429 % (27 out of 35).

Sigmoid function (3)

		Prediction outcome	
		One (1)′	**Zero (0)′**
actual value	**One (1)**	16	0
	Zero (0)	19	0

So, the accuracy is 45.7143 % (16 out of 35).

Perceptron (from Homework 3)

Parameter settings: iteration number 10000, learning rate 0.01, and initial weight -3.0

		Prediction outcome	
		One (1)′	**Zero (0)′**
actual value	**One (1)**	12	4
	Zero (0)	1	18

So, the accuracy is 85.7143 % (30 out of 35).

From above results, linear SVM and Perceptron are doing well for this data set. Seems like, the decision boundary is linear instead non-linear one.

2. Part Two (Boosting)

   Data Set 1 (Chess):

   `https://archive.ics.uci.edu/ml/datasets/Chess+%28King-Rook+vs.+King%29`

Base Class	**Vanilia**	**Bagging**	**Boosting**
Decsion Tree (J48)	68.4691	71.3099	73.3417
Naive Bayes	34.0046	33.9369	34.0046
KNN (IBK)	56.0542	58.8309	56.0542

Table 1: Accuracy for data set one

Data Set 2 (Seismic Bumps):

`https://archive.ics.uci.edu/ml/datasets/seismic-bumps`

Base Class	**Vanilia**	**Bagging**	**Boosting**
Decsion Tree (J48)	93.3437	92.8019	92.1053
Naive Bayes	86.726	86.9969	87.113
KNN (IBK)	89.3963	89.7059	89.3963

Table 2: Accuracy for data set two

Data Set 3 (Bank Note Authentication):

`https://archive.ics.uci.edu/ml/datasets/banknote+authentication#`

Base Class	**Vanilia**	**Bagging**	**Boosting**
Decsion Tree (J48)	98.3224	98.76	99.8541
Naive Bayes	83.9533	83.9533	99.8541
KNN (IBK)	99.8541	99.8541	99.8541

Table 3: Accuracy for data set three

Question 1: Which algorithms+dataset combination is improved by Bagging?

Answer:

For dataset one, Bagging with Decision tree and KNN has improved the performace: Decision tree (from 68.4691% (Vanilia) to 71.3099% (Bagging)) and KNN (from 56.0542% (Vanilia) to 58.8309%).

For dataset two, with Bagging with Naive Bayes performance (86.9969%) increases slightly from Vanilia (86.726%).

For dataset three, with Bagging with Decision tree performance (98.3224%) increases slightly from Vanilia (98.76%).

Question 2: Which algorithms+dataset combination is improved by Boosting?

Answer:

For dataset one, boosting with decision tree has better performance than Vanilia (from 68.4691% (Vanilia) to 73.3417% (Bagging)).

For dataset two, boosting with Naive Bayes has better performance than Vanilia (from 68.4691% (Vanilia) to 73.3417% (Bagging)).

For dataset three, boosting with Decision Tree and Naive Bayes has better performance than Vanilia one. Decision tree: from 98.3224% to 99.8541%, Naive Bayes: from 83.9533% to 99.8541% (improvement is considerable).

Question 3: Can you explain these results in terms of the bias and variance of the learning algorithms applied to these domains? Are some of the learning algorithms unbiased for some of the domains? Which ones?

Answer:

In our experiments, we use one unstable classifier: Decision Tree and two stable classifiers: Naive Bayes and KNN.

However, it is not always bagging and boosting improve the vanilia's result. That means, bagging will improve accuracy by reducing variance where boosting will improve accuracy by reducing both bias and variance. For ustable learner or classifier (decision tree), performace does not always improve for example for dataset two (Seismic Bumps). Stable classifiers are not always hurt by bagging or improved by boosting. In some case, bagging hurts stable classifier. For example, Naive Bayes classifier for dataset one (Chess). In case of boosting, for some cases, performace reduces (Dataset 2 with Decsion tree). This may for outliers in training dataset which are given importance by boosting.

Yes. For some cases, classifier is indefferent or unbiased for vanilia or Bagging or Boosting. For example, for dataset 3, KNN provides same accuracy for all cases (vanilia or Bagging or Boosting).

3. Part Three (K-means Clustering)

Problem 1: Display the images after data compression using K-means clustering for different values of K (2, 5, 10, 15, 20).

Answer:

We did the experiments for 5 (five) times for each K value. And they are included in .zip.

Problem 2: What are the compression ratios for different values of K? Note that you have to repeat the experiment multiple times with different initializations and report the average as well as variance in the compression ratio.

Answer:

Original size of the files are:

Koala.jpg (780.8kB)

Penguins.jpg (777.8kB)

We run Kmeans with these figures five (5) times.

All the results and conmpression ratio are give from table 4 to table 9.

Compression Ratio is defined as = (Size for K value/Original Size) * 100.

K	Koala.jpg Size (kB)	Penguins.jpg Size (kB)
2	130.2	85
5	175.6	105.7
10	166.4	117.6
15	159.2	116.1
20	157.1	116.4

Table 4: output1 with size for Koala.jpg and Penguins.jpg with various K

K	Koala.jpg Size (kB)	Penguins.jpg Size (kB)
2	130.9	85
5	176.6	107.3
10	164.7	117.6
15	157.4	116.7
20	155.8	116.8

Table 5: output2 with size for Koala.jpg and Penguins.jpg with various K

K	Koala.jpg Size (kB)	Penguins.jpg Size (kB)
2	130.9	85
5	176.6	106.2
10	163.5	117.8
15	159.1	116.9
20	155.2	116.6

Table 6: output3 with size for Koala.jpg and Penguins.jpg with various K

Problem 3: Is there a tradeoff between image quality and degree of compression. What would be a good value of K for each of the two images?

Answer:

No. As we can see, increasing the value of K has not linear relation with size of the compressed file. And by observation (instead of some standard method), we get for K=20 for both of figure, the quality is highest. However,

K	Koala.jpg Size (kB)	Penguins.jpg Size (kB)
2	130.9	85
5	176.6	107.3
10	163.0	118.1
15	157.2	117.7
20	156.8	117.0

Table 7: output4 with size for Koala.jpg and Penguins.jpg with various K

K	Koala.jpg Size (kB)	Penguins.jpg Size (kB)
2	130.2	85.0
5	175.6	108.3
10	163.7	116.6
15	156.7	115.6
20	156.8	116.7

Table 8: output5 with size for Koala.jpg and Penguins.jpg with various K

K	Koala Avg. Size (kB)	Penguins Avg. Size (kB)	Koala CRatio	Penguins CRatio
2	130.62	85	16.72	10.92
5	176.4	106.96	22.59	13.75
10	164.26	117.46	21.03	15.10
15	157.92	116.6	20.22	14.86
20	156.34	116.7	20.02	15.00

Table 9: Compression Ratio (K-value/Original Size)*100

it may relative issue and varies from person to person.