MATRICULATION NUMBER – s1368177

Introductory Applied Machine Learning - Assignment 2.

PART 1

1.a)

Technically speaking, clusters do correspond to classes. SimpleKMeans clustering is run with the following configuration:

weka.clusterers.SimpleKMeans -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

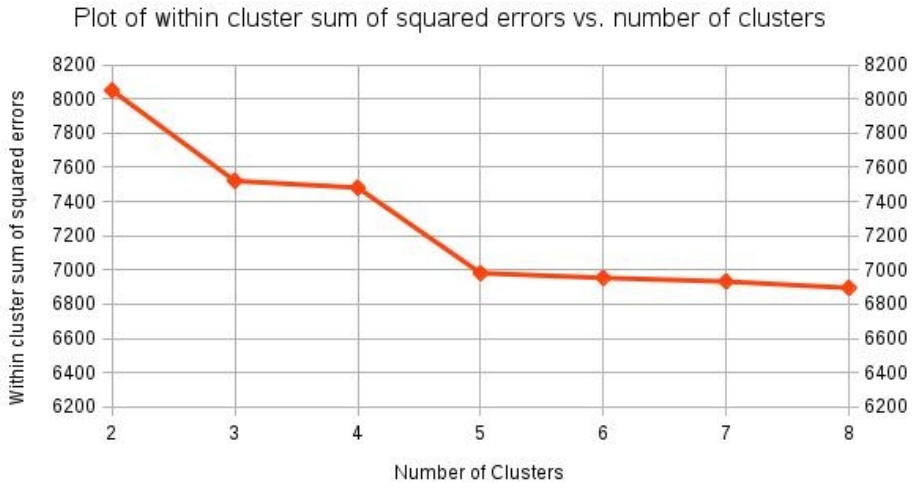
And the resulting confusion matrix is shown below.

0	1	2	3	4	<-- assigned to cluster
354	9	0	20	0	| 1
8	442	0	3	0	| 2
6	432	0	6	0	| 3
5	23	35	401	9	| 4
0	24	172	58	220	| 5

We can see from the above confusion matrix that the class 5 is the most confused one, it has assigned 0, 24, 172, 58, 220 instances to each cluster respectively in the order (none of them with a majority). As for being wrong and confused with a different class, class 3 has its clustering almost similar to that of class 2 (i.e. it has most of its instances assigned to the cluster belonging to class 2). We can arrive to this conclusion because an ideal confusion matrix would have 0 on its non-diagonal entries. This happens because as stated in the description of the assignment, there are two highly related classes comp.sys.ibm.pc.hardware and comp.sys.mac.hardware and are represented by class 2 and class 3.

1.b)

Below is the required plot for number of clusters 2 through 8. Looking at this graph, one would select number of clusters as 5. It agrees with my expectation that as the number of clusters increase, the within cluster sum of squared errors decreases at a decreasing rate.



The elbow method states that the optimal k is when the error decreases abrubtly i.e. forms an elbow in the line plot. The elbow is indeed formed at k=5 and would be the optimal selection.
N.B. Elbow method is just a heuristic, that selection may or may not work well.

The seed value in SimpleKMeans is a value that is used to initialize the random number generator. SimpleKMeans sets initial cluster centroids by randomly selecting instances from the training data, as long as its the same seed value for all our above test runs, it is safe to make the assumption. Although, to account for variance, we must use different seed values and then average out the results.

1.c)

w471_thanks:0.0603	w487_drive:0.0655	w499_hockey:0.068	w493_he:0.0736	w496_god:0.0926
w211_my:0.0471	w468_scsi:0.0556	w500_team:0.0649	w495_baseball:0.0694	w442_we:0.0545
w250_com:0.045	w497_mac:0.0553	w498_game:0.0543	w486_year:0.0559	w459_keith:0.0543
w386_you:0.0418	w471_thanks:0.048	w439_ca:0.0522	w480_his:0.0501	w445_people:0.054
w94_does:0.0388	w485_card:0.0449	w493_he:0.0488	w489_games:0.0484	w386_you:0.0532
w487_drive:0.0385	w482_mb:0.0424	w488_play:0.0483	w422_runs:0.0465	w380_your:0.0511
w447_monitor:0.037	w355_video:0.0401	w492_nhl:0.0465	w420_was:0.0462	w417_say:0.0487
w497_mac:0.0356	w211_my:0.0398	w388_go:0.0448	w500_team:0.0447	w219_not:0.0485
w491_apple:0.0354	w213_cd:0.0396	w420_was:0.0415	w329_edu:0.0443	w290_by:0.0477
w385_problem:0.0353	w446_use:0.0393	w490_players:0.0387	w454_who:0.0431	w474_religion:0.0457
w329_edu:0.0344	w406_floppy:0.0393	w494_season:0.0384	w323_they:0.0429	w472_atheism:0.0454
w219_not:0.0342	w353_memory:0.0391	w475_cup:0.0372	w498_game:0.0422	w118_as:0.0453
w189_do:0.0337	w362_speed:0.0376	w323_they:0.0372	w419_last:0.0397	w189_do:0.045
w28_it:0.0331	w410_mhz:0.0375	w478_playoffs:0.0366	w490_players:0.0392	w431_bible:0.0449
w312_advance:0.0327	w111_driver:0.035	w489_games:0.0362	w494_season:0.0373	w146_are:0.044
w452_software:0.0325	w453_system:0.0349	w454_who:0.0351	w466_pitching:0.037	w329_edu:0.0417
w446_use:0.032	w455_disk:0.0348	w464_leafs:0.0349	w414_braves:0.0365	w128_so:0.0417
w146_are:0.032	w425_motherboard:0.0346	w486_year:0.0347	w423_hit:0.0362	w421_morality:0.0404
w485_card:0.0318	w94_does:0.0336	w450_rangers:0.0345	w470_him:0.0359	w400_being:0.0402
w323_they:0.0318	w250_com:0.0335	w480_his:0.0344	w407_article:0.0358	w344_must:0.0395

We have classes 1:alt.atheism, 2:comp.sys.ibm.pc.hardware, 3:comp.sys.mac.hardware, 4:rec.sport.baseball and 5:rec.sport.hockey. By looking at the table above that shows top 20 attributes in each class by normalised importance weights, one can safely assume that the fifth column is related to class 1 (atheism) since the words like god, people, religion and atheism are the most important ones. And other classes don't have slightest correlation to god and religion. Columns third and fourth are respectively class 4 (baseball) and class 5 (hockey) because hockey

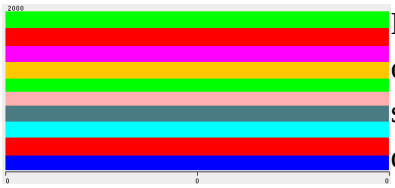
and baseball are within the top two most important attributes in each and no mention of each other in each other's classes. Other words like season, playoffs, team and game tell us that they are recreational sports and are one of the two sports. The remaining first two columns of clusters correspond to either ibm pc (class 2) or mac (class 3) and are the most confusing because they are highly correlated, both are computers, and have a high mention of each other and computer terms in each other, such that its hard to differentiate without computational power.

PART 2

2.a)

Naive Bayes classifier accuracy (percent correct) – 71.15%

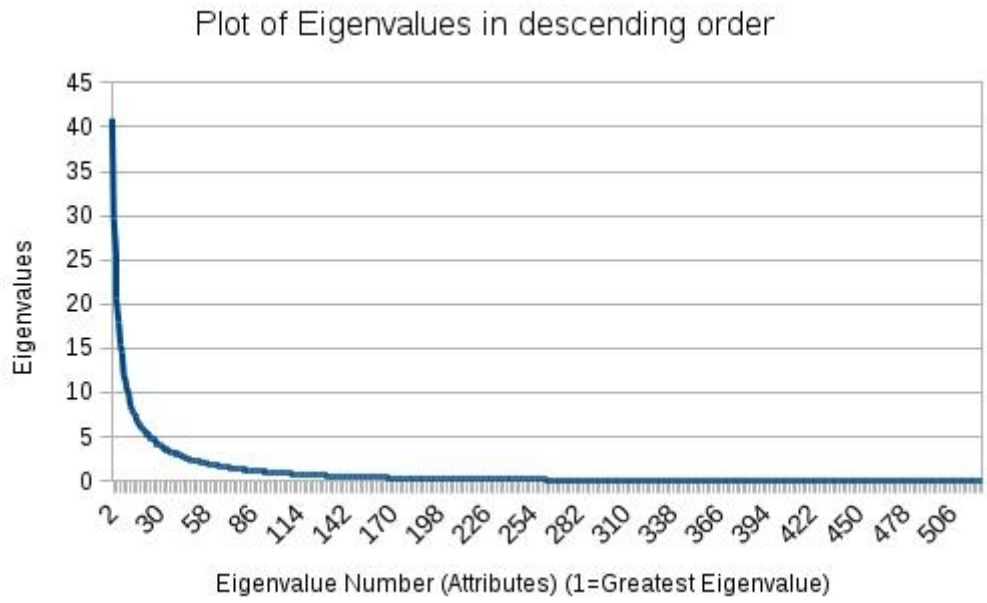
Linear Support Vector Machine classifier accuracy (percent correct) – 88.5%



Histograms like the one on the left contribute no information to the classification task because they have no information. If we look at the statistic box for the attribute, all values, min, max, mean and standard deviation are zero and hence are useless.

We can remove these attributes by using the RemoveUseless filter with maximumVariancePercentageAllowed set to 0. This reduces the number of attributes to 525 from original 670.

2.b) The plot of Eigenvalues in descending order after performing Principal Component Analysis on the dataset is shown below:



We can notice from the graph that the eigenvalues form an inverse relationship which means that there are few eigenvectors with an eigenvalue >1 and significantly more which are <1. They decrease in the amount of variance in the originals. They account for x First component captures most of the variance, 2nd second most and so on until all the variance is accounted for. The eigenvector with the highest eigenvalue is the principal component and denotes how much spread

the data has in vector space.

2.c) The results after performing Principal Component Analysis compared with before is shown below: (in percent correct)

Classifier	After PCA	Before PCA
Naive Bayes	82.8%	71.15%
Linear Support Vector Machine	88.75%	88.5%

The performance of Naive Bayes improved because PCA redistributes all variance into orthogonal components and uses all variable variance and treats it as true variance. It also facilitates dimensionality reduction and trivially data-transformation in pre-processing improves performance imminently as noticed in the previous coursework. On the contrary, in case of Linear SVM, there is an improvement of .25%, this is because we removed some attributes and this is not a legitimate improvement i.e. performance unchanged. SVM is a hyperplane separator mechanism, if you move the points not on the marginal hyperplanes, solution don't change - therefore those points don't matter and neither does transformation which is exactly what PCA does.

PART 3

3)

Linear SVM (with PolyKernel and set exponent to 1.0 for pair conjunction and then 2.0 for the default dataset) applied on the following datasets with percent correct accuracy statistic shown below:

Dataset ->	Pair Conjunction	Default
Lin. SVM with above config.	90.2%	89.55%

Pair conjunction is an important technique, it can be viewed as parsing a block of text into words and phrases instead of characters. Some methods have difficulty in precisely classifying for problems where examples from the same class have few common features. Since similarities between examples from the same class become minute, classifiers fail to distinguish some pairs from others. To overcome this, we can use conjunctions of features across datasets. Using a kernel on pair instances, our method can use feature conjunctions without using large computing resources. This also gives us higher precision and recall statistics. This also helps avoiding no correlation and overcoming orthogonality.

PART 4

For the mini-challenge, I first removed the outliers above the value of 0.5 to make sure they don’t skew our computation. I ran that with the following configuration on the relation.

Relation: all_train-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.attribute.Remove-R501-514,516-519-weka.filters.unsupervised.instance.RemoveWithValues-S0.5-C1-Lfirst-last-V

Next I decided to use an attribution selection classifier with the configuration shown below.

Scheme: weka.classifiers.meta.AttributeSelectedClassifier -E "weka.attributeSelection.SymmetricalUncertAttributeEval " -S "weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1" -W weka.classifiers.meta.RotationForest -- -G 3 -H 3 -P 50 -F "weka.filters.unsupervised.attribute.PrincipalComponents -R 1.0 -A 5 -M -1 -D" -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2

RotationForest classifier was chosen here with SymmetricalUncertAttributeEval evaluator. This evaluator Evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.

To create the training data for a base classifier, the feature set is randomly split into K subsets (K is a parameter of the algorithm) and Principal Component Analysis (PCA) is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Thus, K axis rotations take place to form the new features for a base classifier. The idea of the rotation approach is to encourage simultaneously individual accuracy and diversity within the ensemble. Diversity is promoted through the feature extraction for each base classifier. Decision trees were chosen here because they are sensitive to rotation of the feature axes, hence the name "forest." Accuracy is sought by keeping all principal components and also using the whole data set to train each base classifier.[Ref1]

By doing this procedure we achieved an accuracy of 66.487% up from an original 52.65%.

Ref1: http://www.ncbi.nlm.nih.gov/pubmed/16986543