3.1 Part 1: Classification

1.We have calculated training error and cross validation error for the ZeroR, OneR, J48 and IBk classifiers. For ZeroR both the error rates are same and therefore we can say that there is no overfitting. The error differences of OneR and J48 is greater than 1, therefore we can say that there is no overfitting. For IBk the error differences is more than 4% which leads to overfitting.

Run No	**Classifier**	**Parameters Parameters**	**Training Error**	**Cross validation Error**	**Over-Fitting**
1	ZeroR	None	7.7147 %	7.7147 %	None
2	OneR	None	2.9958 %	3.7646 %	None
3	J48	None	0.1856 %	0.4242 %	None
4	IBK	None	0.0000 %	8.4836 %	Yes

2.

C Value	**M Value**	**Training Accuracy**	**Cross-Validation Accuracy**	**Difference in Accuracy**
C = 0.7	M = 5	99.7084%	99.5228%	0.1856 %
C = 0.3	M = 4	99.6819%	99.5228%	0.1591 %
C = 0.5	M = 8	99.4433%	99.3372%	0.1061 %
C = 0.25	M = 10	99.3637%	99.3107%	0.053 %
C = 0.01	M = 7	99.3902%	99.3637%	0.0265 %

Here C value is the 'confidence Factor' and M value is 'Minimum number of object'. For C = 0.01 and M = 7 the minimum difference accuracy is 0.0265 % which doesn't overfit whereas when C = 0.7 and M = 5 the minimum difference accuracy is 0.1856 % which shows that higher the confidence value the bigger is the difference in accuracy.

3.

Percentage Split %	**Accuracy**
54	99.4236 %
56	99.4578 %
58	99.4318 %
60	99.4036 %
62	99.4417 %
64	99.4109 %
66	99.4540 %

The percentage split divides the data into two parts: Training set and testing set. By default the % split is 66 which is the training split whereas the rest 34 % is testing split. The more the training data the more accurate is the result as less unseen values are tested. With the % decrease as obvious the accuracy decreases but some % splits show higher accuracy. Due to this the data split should be proper with testing data atleast half of the training data. This is to ensure that most of the unseen labels get predicted.

4.

Value of K	Training Accuracy	Cross-Validation Accuracy	Difference in Accuracy
1	100 %	91.5164 %	8.4836 %
2	94.9629 %	93.2927 %	1.6702 %
3	95.334 %	93.2131 %	2.1209 %
5	94.3531 %	93.2662 %	1.0869 %
10	93.4783 %	93.2397 %	0.2386 %
15	93.2131 %	92.948 %	0.2651 %
20	92.895 %	92.8685 %	0.0265 %
30	92.8685 %	92.6564 %	0.2121 %
50	92.3118 %	92.1792 %	0.1326 %
60	92.1792 %	92.1792 %	0
95	92.2853 %	92.2853 %	0
100	92.2853 %	92.2853 %	0

Using different values of K we have calculated the accuracy for 1Bk classifier. For k = 1 the difference in the accuracy is8.4836 % which is > 5 i.e. overfitting. As K increases the difference in accuracy also decreases which minimizes the chances of overfitting. However, extremely high k values such as from k = 60 to 100 we can see that the difference in accuracy is zero i.e. imbalanced.

5.

Classifier	Parameters	Accuracy
NaiveBayes	Percentage Split = 66 % useSupervisedDiscretization = False	95.3978 %
NaiveBayes	Percentage Split = 66 % useSupervisedDiscretization = True	98.2059 %
NaiveBayes	Percentage Split = 75 % useSupervisedDiscretization = False	95.9703 %
NaiveBayes	Percentage Split = 75 % useSupervisedDiscretization = True	98.4093 %
NaiveBayes	Percentage Split = 80 % useSupervisedDiscretization = False	96.4191 %

Classifier	Parameters	Accuracy
RandomForest	Percentage Split = 66 % numFeatures = 0	98.6739 %
RandomForest	Percentage Split = 66 % numFeatures = 10	99.376 %
RandomForest	Percentage Split = 75 % numFeatures = 0	99.3637 %
RandomForest	Percentage Split = 75 % numFeatures = 10	99.7879 %
RandomForest	Percentage Split = 80 % numFeatures = 0	99.4695 %

NaiveBayes and RandomForest are the two algorithms which were used with different parameters. Comparing the results of both, we found that random forest has the highest accuracy of 99.7879 %

this is because the % split is appropriately distributed between training and testing data and also with increase in number of features at each node the accuracy was slightly improved. For NaiveBayes as the % split increases the accuracy also got increased. Moreover, using Supervised Discretization (converts numeric values to nominal) which works better when it is set to true as the whole dataset contains categorical data where classifiers tend to classify data more accurately.

6.

Classifiers	**Accuracy**
ZeroR	92.1217 %
OneR	96.0218 %
J48	99.454 %

The Accuracy of ZeroR (92.1217 %) is less than OneR (96.0218 %) and J48 (99.454 %) because only one attribute is used and the data is roughly normalized that evenly predicts the classes. Also, OneR has better accuracy than ZeroR as it selects one rule with the smallest error. The J48 has the highest accuracy among others is because it works well on classifying categorical data.

7.J48 works well on this dataset but as RandomForest is the collection of decision trees which yields highest accuracy among all. As the dataset is roughly normalized the ZeroR classifier will always have the lowest accuracy. For IBk as the value of k increases initially overfitting decreases and as the value increases after certain limit accuracy suffers. ZeroR is worst classification algorithm as it learns nothing, the model does not train very much. Classifiers give better accuracy on attribute set whereas on whole set the accuracy reduces because of the all unnecessary attributes therefore attribute selection is important.

8.

Classifier	**Full Set Accuracy**	**Attribute Set Accuracy**
IBK	90.1656 %	91.6537 %
J48	99.454 %	97.1919 %
OneR	96.0218 %	96.0218 %
ZeroR	92.1217 %	92.1217 %

Generally, attribute selection is done so that the model gives the best accuracy using values of important features. In Weka BestFirst search method and cross validation mode was used for attribute selection. BestFirst selects the best attributes. Accuracy was computed using model with 10 features. IBk classifier had 1% of increased accuracy from reduced attribute set whereas J48 classifier decreased by 2%. The accuracy of J48 might have decreased due to the outliers which were present in the full dataset. On the other hand ZeroR and OneR had the same accuracies where ZeroR doesn't need predictors or attributes and in OneR only one attribute with smallest error is selected.

Part 2: Numeric Prediction

1.

Classifiers	Training Error	Cross-validation Error	Overfitting
ZeroR	100 %	100 %	None
M5P	14.7976 %	15.6194 %	None
IBk	0 %	23.7602 %	Yes

Comparing the difference accuracies (Training Error - Cross-validation Error) of ZeroR, M5P and IBK we found that the difference accuracy was 0 in ZeroR (i.e. no overfitting), M5P had 1% accuracy difference (i.e. no overfitting)whereas, IBk had 23% difference in accuracy which leads to overfitting.

2.

Classifiers	Parameters	Correlation Coefficient (For Training set)	Correlation Coefficient (Cross validation)	Training Error (Relative Absolute Error)	Cross validation Error (Relative Absolute Error)	Overfitting
M5P	batchSize = 50	0.9862	0.9766	14.7976 %	15.6194 %	None
M5P	batchSize = 100	0.9862	0.9766	14.7976 %	15.6194 %	None
M5P	batchSize = 200	0.9862	0.9766	14.7976 %	15.6194 %	None
M5P	batchSize = 400	0.9862	0.9766	14.7976 %	15.6194 %	None
M5P	batchSize = 800	0.9862	0.9766	14.7976 %	15.6194 %	None

Classifiers	Parameters	Correlation Coefficient (For Training set)	Correlation Coefficient (For Cross Validation)	Training Error (Relative Absolute Error)	Cross validation Error (Relative Absolute Error)	Overfitting
IBk	K = 32	0.8679	0.8574	45.2159%	46.0705%	None
IBk	K = 34	0.8681	0.857	44.9717%	456146%	None
IBk	K = 36	0.8707	0.8552	44.6355%	44.5797%	None
IBk	K = 38	0.8691	0.8564	44.2214%	45.5107%	None
IBk	K = 40	0.8678	0.8599	44.1639%	45.2975%	None

M5P and IBk classifiers were used with different parameters to compute the predictive accuracy and to check the overfitting. Relative absolute error is the ratio, comparing a mean error with the actual value observed(Percent error / actual value observed). From the above table of M5P classifiers we got to know that as the parameters for batchSize changes i.e. increases the Correlation Coefficient (For Training set and validation), Training error and cross validation error remains same for all the batch

size. Therefore, by changing the batch size all the results remains same. Correlation coefficient is the correlation between expected and the predicted output. In the table the correlation coefficient is 0.9 which means that most of the values are predicted correctly. There is no overfitting in M5P classifier as the distance between the errors is very less. In IBK classifier as the parameters of K are changed from 32 to 40 we can see that there is a slight decrease in correlation coefficientsand also in the training errors and the cross validation error. There is no overfitting as we see in training error and cross validation error that there is no huge difference in %. In short, M5P classifier performs best as compared to IBk classifier.

3.

Classifiers	**Parameters**	**Correlation Coefficient (For Training set)**	**Correlation Coefficient (For Cross Validation)**	**Training Error (Relative Absolute Error)**	**Cross validation Error (Relative Absolute Error)**	**Overfitting**
RandomForest	numIterations = 100	0.9955	0.9515	6.7393%	15.6212%	Yes
RandomForest	numIterations = 250	0.9959	0.9708	6.3801%	15.6234%	Yes
RandomForest	maxDepth = 10	0.9955	0.9515	6.7393%	15.6443%	Yes
RandomForest	batchSize = 400	0.9955	0.9515	6.7393%	15.6212%	Yes
RandomForest	batchSize = 800	0.9955	0.9515	6.7393%	15.6212%	Yes

Classifiers	**Parameters**	**Correlation Coefficient (For Training set)**	**Correlation Coefficient (For Cross Validation)**	**Training Error (Relative Absolute Error)**	**Cross validation Error (Relative Absolute Error)**	**Overfitting**
LinearRegression	batchSize = 50	0.963	0.9262	32.5055%	41.9542%	Yes
LinearRegression	batchSize = 100	0.963	0.9262	32.5055%	41.9542%	Yes
LinearRegression	batchSize = 200	0.963	0.9262	32.5055%	41.9542%	Yes
LinearRegression	batchSize = 400	0.963	0.9262	32.5055%	41.9542%	Yes
LinearRegression	batchSize = 800	0.963	0.9262	32.5055%	41.9542%	Yes

RandomForest and LinearRegression classifiers were used with different parameters to compute the predictive accuracy and to check the overfitting. From the above table of RandonForest Classifier, as numIterations increases there is an increase in correlation coefficient using training set and validation set. For the different batch size values the accuracy was constant using both sets. There exist overfitting in all the runs of RandomForest. In LinearRegression as batch size increases the accuracy remains constant throughout the runs. Other parameters were also experimented but the output had no

difference.Correlation coefficient is the correlation between expected and the predicted output. In the table the correlation coefficient is 0.9 which means that most of the values are predicted correctly. There is an Overfitting while running LinearRegression Classifier where difference between Training error and cross validation error was around 9%. Therefore, RandomForest classifier performs best as compared to LinearRegressionclassifier.

4. We found thatM5P classifier perform betters among other classifiers.M5P works better on continuous numerical values as it is a binary regression tree model which does classification via regression.

Part 3: Clustering

1. The percentage split of 66% was used for different k values. The output of K clusters had different initial random starting points. Out of all the K values, the k =3 had roughlyequal clustered instances such as 33%, 37% and 31%. For TSH and TT4, the values of cluster 1 and 2 were very close to each other. For the age, the cluster 1 is separated from the others. Visualizing the results, we found that for x=age and y=TT4 variables, the clusters were formed for 1 to 5 values of K. Whereas, for 10 and 20 K values the clusters were overlapping with one another.We also got 22 iterations and the average distance between each point in the cluster with respect to centroid was 812.73794.The Initial starting points of 3 clusters were 27, 44 and 68.

2. The value of K is 3 with default seed value 10 and with seed value 11. We checked the outputs and found that the instances were roughly balanced for seed 10 and for seed 11, only two clusters had roughly similar instances. Moreover, visualization output with seed value 10 showed clusters close to each other for age and TT4 features. Whereas, the clusters were overlapping with each other for seed value 11.

3. After running EM algorithm we got 7 clusters using cross validation. Consider the age, cluster 6 is separated from other clusters. The closest one to it is cluster 0 where the mean difference is 16 and with the Standard deviation of 9.Moreover, cluster 4 and cluster 6 are very close to each other with their centroids within a distance of around 0.40, while their variance difference is 1.3, which suggest that the data points are not overlapping with each other.

4. There happens to be no difference except that the value of log likelihood which was -12.0736 and after normalize it was around 0.3737.From normalizing we get a change in log likelihood which shows the purity of the clusters.After normalizing the numeric attributes, clusters were formed same.

5. For EM the value of minStdDev correspond roughly to the number of decimal places in the data. In this case for minStdDev 0.1 there were total 9 clusters whereas for minStdDev 100 there were only two clusters. When increasing minLogLikelihoodImprovementIterating from 10E-6,0.1, and 1.0, we notice that the EM took less time to run (17.36, 3.71 and 38.72) sec respectively. Thus setting a larger value to it terminates EM quicker. For minLogLikelihoodImprovementCV values the EM terminates quicker but this time however, the log likelihood decreases a lot when parameter is set to 1.0.

6. There are 7 clusters in the data. Cluster 2 is mostly female with age ranging from about 45-60 with low chances of underactive thyroid having TSH levels of 0.11 and TT4 levels of 133.14 which indicates thyroid disease.

7. EM tries to maximize the likelihood after each iteration which makes it so special then the k-means. On the other hand k-means calculates Euclidian distance for clustering the data points where it assumes that the clusters have equal covariance matrices. EM is time consuming as compared to k-means.*K*-means algorithm differs in the method used for calculating the Euclidean distance while calculating the distance between each of two data items; and EM uses statistical methods. The EM algorithm is often used to provide the functions more effectively. K-means hard assign a data point to one particular cluster on convergencewhereas EM Soft assigns a point to clusters (so it give a probability of any point belonging to any centroid).

8. We can get golden nuggets by looking at the Clusters:-

Females with age 46 have low chances of underactive thyroid having TSH levels of 1.4928 and females with age 46 have high chance of TT4 levels of 178.0771 which indicates thyroid disease

Part 4: Association Finding

1. The datasets have similar no of attributes but vary in the number of observations. The supermarket1-small.arff dataset has binary values f (for false) and t (for true) present for each of the attributes. Whereas, the supermarket2-small.arff has t (for true) values and missing values (?) for each of the present attributes. This means that the missing values (?) could be an f (for false) values where attributes might have binary values just like the first dataset.

2.As the dataset was huge in size, the Weka couldn't process the dataset. Hence, 7 attributes (Bread and Cake, Frozen foods, Milk, Biscuits, Fruit, Party snack foods and vegetables) were selected for finding association.
Apriori algorithm was used with the minimum support: 0.1, minimum metric <confidence>: 0.9.
Three association rules were found. Rule 1 is:

- *biscuits=t frozen foods=t party snack foods=t milk-cream=t fruit=t vegetables=t 512 ==> bread and cake=t 466 <conf:(0.91)> lift:(1.26) lev:(0.02) [97] conv:(3.05)*

As we can see the rule is presented in antecedent ==> consequent format. No rules have coverage less than 0.91 i.e. 95% (466/512). The support is 466/978 or 47% for the first rule. The no besides antecedent is the coverage of the dataset instances whereas the no besides the consequent is the matching instances of both sides. All the rules have a consequent of 'bread and cake' and the biscuits, frozen foods, fruit and vegetables appear in all the 3 rules.

3.Lift metric type was used with minimum support of 0.35 and minimum metric of 1.1. The cut off was 1.15 for lift and the fruit as the consequent appear most as compared to other parameters. The bread and cake, fruit and vegetables appear most in the antecedent of the rules.

4.The supermarket2 dataset had 1281 instances with 7 attributes (same as supermarket1). The minimum support was 0.1 (128 instances) with confidence metric type. The confidence interval was between 91-93% on different rules. The 'break and cake' was the consequent in all the 10 rules. Also, biscuits, frozen foods and fruits were the most which appear in the rules.

5.Lift metric type was used with minimum support of 0.4 (512 instances) and minimum metric of 1.1. The cut off was 1.11 ranging from 1.11 to 1.15. Bred and cake, fruit and biscuits appeared most in both antecedent and consequent of the rules.

6.The FPGrowth algorithm was used (on supermarket2 dataset) with confidence interval between 91-93% for different rules. The 'bread and cake' was presented in all the 10 rules as a consequent. Moreover, fruits, biscuits and frozen foods were mostly visible in the antecedent of all the rules. The output of both the Apriori and FPGrowth shows similar readings. There is no significant difference between the findings of both algorithms on supermarket2 dataset.

7. Based on frequent item set buying, such as in this case, we found biscuits, frozen foods, fruit and vegetables are set of items people buy on which we can convince people to buy bread and cake along with the other item set so that the transactional amount is maximized.

8.Apriori algorithm was used by selecting some nominal attributes from hypothroid dataset. There was no association or any significant findings as the result were the obvious and already known i.e. if a person is no sick then surgery is not required.

Pair work submission

My Name is Mark Pereira and my other member name is Shonil Dabreo. We both worked as a pair. First of all I would like to say that we both contributed equally to this assignment. Firstly in classification we solved each question by running results into the Weka, while I was running the dataset on the Weka and running classifiers my group member Shonil also noted all the calculations which were needed and later we both compared our results. Like these we try to complete the whole classification part and at the end discussed the golden nuggets. Now after completing part 1 to part 4 and discussing all the results we began with the report. For the part 5 we discussed about ethical issues and made a presentation. Some slides were made by me while some slides were made by shonil and finally both of us gave presentation by distributing the slides.