PART 1: CLASSIFICATION

Classification of chronic-kidney-disease-2016.arff

Task 2)

Run No	Classifier	Parameters	Training Error Incorr. class. Instances (R.A.E.)	X-valid Error Incorr. class. Instances (R.A.E.)	Overfitting
1	ZeroR	default	37.5% (100%)	37.5% (100%)	None
2	OneR	default	7.5% (15.9973 %)	8.25% (17.5968%)	None
3	J48	default	2% (7.9994%)	2.625% (10.227%)	Some
4	IBK	default	4 (8.6544%)	4 (8.7621%)	None

Using additionally the Relative absolute error (R.A.E.) besides the value of incorrectly classified instances the conclusion is that the training error is in any case lower as the cross validation error. In case of IBk, ZeroR, and OneR I wouldn't assume a overfitting problem. But in case of J48 the difference in training error to Cross Validation Error (X-Valid Error) is around 2.2276 (in R.A.E.) and with OneR it's 1.5995. Question here is if this counts already as overfitting as it is still quite a low difference. Still tree structures are prone to a overfitting bias.

Task 3.)

J48 Run	Confidence C	MinNumObj	Result Training	Result X-Valid.	Difference
1	0.35	3	11.1899%	11.1078%	-0.082%
2	0.25	10	19.9889%	19.9909%	0.002%
3	0.30	3	11.1899%	11.4565%	0.267%
4	0.25	3	11.1899%	11.4565%	0.267%
5	0.60	2	6.9886%	8.2679%	1.279%

Yes, I can find such values (comparing in R.A.E.) as you can see with all 5 runs but in many cases it results in higher error rates for both the training set run and the cross validation. Only in one case I find a situation in which the cross validation is better than the training set.

Task 4.)

J48 Round	Percentage of training set	Relative absolute error result
default	100%	7.9994%
1	90%	16.462%
2	80%	14.0826%
3	70%	10.9122%
4	66%	12.1941%
5	50%	15.2563%

The effect is that with lesser instances to handle the error rate increases. At first it increases its error rate 14% - 16% but then starts lowering the error rate when 70 to 60 percent of the original data set I used. Starting from 60 % of the total amount of instances the error increases again. For this reason a recommended training split is 66% as it is the best compromise when working with reduced data sets..

Task 5.)

Test Run	Value K	Result Training (R.A.E)	Result X-Valid. (R.A.E)	Difference	Wrong class. Instances Training	Wrong class. Instances X- Valid
default	1	8.6544	8.7621	0.108	4	4
round 1	2	8.6544	9.6619	1.008	4	4.625

round 2	3	10.1919	10.6136	0.422	3.25	4.25
round 3	4	10.1919	11.9899	1.798	3.25	4.625
round 4	15	23.1425	24.9257	1.783	80	101

No, I can't find a better value of k, because there is actually no real overfitting. If I change the value of k to any higher number than the default one, which is 1, than the actually overfitting starts appearing as the difference between trained data and validated data grows. Therefore I conclude that the default value is the best for this situation.

Task 6.)

Classifier	Accuracy Training	Accuracy X-Valid.	Difference
ZeroR	62.5%	62.5%	0.000%
OneR	92.5%	91.75%	-0.750%
J48	98%	97.375%	-0.625%
IbK	96%	96%	0.000%
JRip	100%	98.875%	-1.125%
Random forest	100%	100%	0.000%
KStar	100%	98.5%	-1.500%
DecisionTable	99.75%	99%	-0.750%
RandomTree	100%	99.75%	-0.250%

It seems that tree structures have the best predictive accuracy, Random forest, RandomTree and J48 are highly accurate in training with up to 100% correct classified objects. The differences in the cross validation shows that there is less than 1% difference and in case of random forest both training data and validation data is 100%. On the other hand ZeroR, IBK have the worst accuracy in comparison.

Task 7)

Classifier (using default parameters)	Accuracy Training	Accuracy X-Valid	Difference
ZeroR	62.5%	62.5%	0.000%
OneR	92.5%	91.75%	-0.750%
J48	98%	97.375%	-0.625%

ZeroR is quite in accurate in terms of absolute classifying the data since has no difference between training and new (validation data). The highest difference between training accuracy and valuation accuracy is with OneR. While J48 as a tree structure like in question 6 is quite high in classification while the difference to validation is the second highest in the three.

Task 8.) Medical research needs high accuracies as life or wellbeing of people might be dependent on it. It is crucial to their business to reflect correct statements. This given any medication research will have to make sure that overfitting doesn't appear as well as that training sets and cross validation sets are not similar and reflect a true statement about the accuracy of classification algorithms. With caution on overfitting I would recommend tree structures like e.g. J48 or random forest for developing medical application out of it.

Task 9.) No golden nuggets found.

PART 2: NUMERIC PREDICTION

Numeric Prediction of the Age attribute in the kidney disease data.

Task 1.)

Classifier	Training Prediction	Cross Valid Prediction	Difference	Overfitting

	Mean Abs. Error	R.M.S. Error	Mean Abs. Error	R.M.S. Error	Mean Abs. Error	
ZeroR	13.8138	17.1477	13.8365	17.1731	0.023	None
M5P	7.664	9.9717	12.0864	15.3044	4.422	Significant
IBK	6.6598	14.4337	7.8824	15.5485	1.2226	Some

ZeroR is not very well suited for a numeric prediction as it only calculates one value and compares all cases to it. This is insufficient and for that reason increases the error rate. On the other hand the IbK algorithm has out of the three the lowest Error means (R.M.S = Root Mean Squared Error) and doesn't seem to have overfitting. M5P seems to have an overfitting issue with default values.

Task 2.)

Classifier	Parameters	Training Prediction		Cross Valid Prediction		Difference		Overfitting
		Mean Abs. Error	R.M.S. Error	Mean Abs. Error	R.M.S. Error	Mean Abs. Error	R.M.S. Error	
M5P	minNumInst = 5	11.68	14.7568	12.0277	15.1755	0.348	0.419	None
	minNumInst = 6, unpruned = true	7.757	10.117	9.8873	13.035	2.130	2.918	Some
	minNumInst = 4, unpruned = true	7.664	9.9717	9.7759	12.8347	2.1119	2.863	Some
	minNumInst = 8, unpruned = true	7.9286	10.4547	10.1554	13.3592	2.2268	2.905	Some
	minNumInst = 4, unpruned = true useUnsmoothed = true	0.1228	0.2905	6.1368	10.3254	6.014	10.035	Significant
IBK	KNN = 5	11.3444	15.7131	12.6555	17.136	1.3111	1.423	Some
	KNN = 5, distance Weight = 1/distance	7.9727	13.8569	9.288	14.995	1.3153	1.138	Some
	KNN = 2, distance Weight = -1/distance	6.6598	14.4337	13.2007	17.9946	6.5409	3.561	Significant
	KNN = 10, distance Weight = -1/distance	11.6278	15.5082	12.9045	16.846	1.2767	1.338	Some
	KNN = 10	11.7407	15.6182	12.9649	16.9143	1.2242	1.296	Some

For IbK it appears that any change of K from default only results in higher error means and in some cases increases the likelihood of overfitting. Tampering with additional parameters like weight distance doesn't show any improvement. In case of M5P turning on unpruned and useUnsmothed reduces the error means but also increases the likelihood of overfitting. Increasing the number of minimum instance improves in the error means in comparison to the default but stagnates after that.

Task 3)

Classifier	Training Prediction		Cross Valid Prediction		Difference		Overfitting
Default	Mean Abs. Error	R.M.S. Error	Mean Abs. Error	R.M.S. Error	Mean Abs. Error	R.M.S. Error	
Decision Table (default)	0.2796	2.1667	2.0014	6.0488	1.722	3.882	Some

RandomTree (K = 10)	0.9023	2.5118	3.177	7.0188	2.2747	4.507	Some
RandomTree (K = 25)	0.6713	1.9274	3.1581	7.4882	2.4868	5.561	Some
RandomTree (default)	1.2256	3.1529	3.7297	7.5339	2.504	4.381	Some
RandomForest (K = 4)	2.7101	4.0217	6.1902	8.4947	3.4801	4.473	Significant

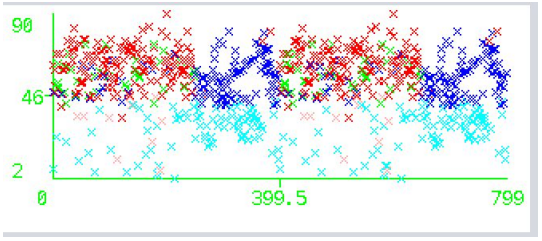
I worked with RandomTree, RandomForest and DescisionTable to explore different Classifiers. It seems that decision table is the better algorithm for prediction in comparison as it has a lower error means and lower difference between Training and Cross Validation set, unlike RandomForst that seems to have a overfitting problem.

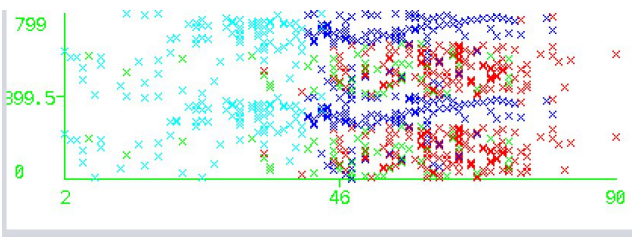
Task 4.) So far no golden nuggets.

PART 3: CLUSTERING

Clustering of the chronic kidney disease data with the attributes Age,bp,rbc,pc and hemo.

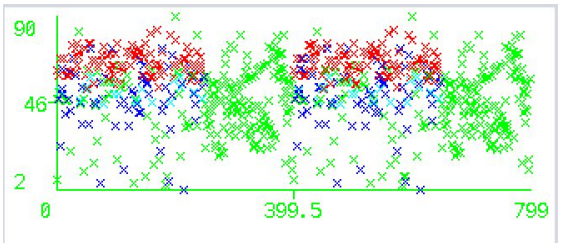
Task 1.) The more the clusters grow the more groups the instances are broken into. Looking at the visualisation there is between 4-10 (as the two pictures show) the most accurate clusters amounts. The amount of 20 clusters becomes to confusing as clusters overlap to much, while with too less the algorithms overlooks abnormal values of rbc and pc which should be clustered separately. Left picture (5 clusters) and right (4 clusters) show clusters separated by age - the line at 40.





Task 2.) I use the value of K = 5 for this task. There seems to be no obvious change in the visualisation (data points remain) except for that the color for the custers change. It seems the cluster centroids change depending on seed value, but there is no system behind it. I have used seed value 1, 2, 3, 4, 5, 10, 15 and can't see a pattern behind it. I assume it's random. But it seems also the coloring and data points between 0 and 399.5 instances are structurally the same as between 399.5 and 799.

Task 3.) It creates initially 4 clusters, the data points of the visualisation looks like same like the 5 cluster visualization of the K-means run(compare left pictures above). Even though the coloring (assignments of instances to clusters) of the cluster appears different than in the above example the same pattern emerges between 0 and 399.5 instances and as between 399.5 and 799.



Task 4.) The standard deviation and mean for bp and age change to a much smaller values and the log likelihood also from -11,1 to - 1,75. Other values remain and the visualisation doesn't change.

Task 5.) The closer minLogLikelihoodImprovementCVminStdDev gets to 1 (starting at 0.1) the less cluster get created. On the other hand if this value gets smaller than a certain point (0.001) than it has no effect on the amount of cluster for this case. With minLogLikelihoodImprovementIterating it's the reversed case, the closer it gets to 1 the more cluster it creates. The smaller the value gets the less cluster are created to a certain point (in our case 1.0E-5). Below this point only mean and SD change.

Task 6.)I think there are three main clusters because both KMeans as well as EM show this through overlapping standard various as soon as more than three clusters are chosen. One are people under 45 years old who have rbc and pc normal

Task 7.) Kmeans is certainly faster than EM, while EM takes 4-5 seconds to compute Kmeans has results in 0.01 seconds. The results of Kmeans are easier to read while EM results can show standard deviation and those can give hints of overlapping clusters that belong together. EM defines it clusters by itself while for Kmeans it has to be set. In general I think both have their uses and depending on situation can be handy. Optimal is to use both to get a better picture of clusters.

Task 8.) What I can see is that the data is duplicated - 399 instances are the same. Otherwise no golden nuggests.

PART 4: ASSOCIATION FINDING

Association finding in the files bakery-data1.arff and bakery-data2.arff

Task 1.) The difference is that one has has its representations in clear binary statements (Yes and No): bakery-data1, while the other baker-data only has positive values - “Yes” and if Yes doesn’t apply than a question mark is set instead of a “No” (or null for that matter).

Task 2.) Seemingly nobody likes the Almond Bear Claws, because people that don’t buy, chocolate eclairs also don’t those Bear Claws and that with very high confidence levels. Almost all association that goes :
“Chocolate Eclair=no Cherry Soda=no” end in Almond Bear Claw=no. Almost all the results are negative meaning that if this is not bought than this will also not be bought. Meaningful associations are harder to get out of this.

Task 3.) In case of Lift as metric type the two “best” rules are this: Mutually excluding associations.
1. Truffle Cake=no Cherry Tart=no 818 ==> Gongolais Cookie=no Apricot Danish=no 751
2. Gongolais Cookie=no Apricot Danish=no 823 ==> Truffle Cake=no Cherry Tart=no 751
If some doesn’t buy truffle cake and no cherry tart then they also don’t buy Apricot Danish and Gongolais Cookies
If I deviate from minMetric I need to be sure that the value I enter will be determined at the operation otherwise I exclude all results. Same goes for lowerBoundMinSupport and UpperBoundMinSupport. These basically work like filters if the result set values are out of the bound no result is shown, but it’s a good way to lower the result set.

Task 4.) It shows that people like to buy Coffee Eclairs and Apple Pies on the weekends together with Almond Twist or Hot Coffee. 1. Almond Twist=yes Hot Coffee=yes Weekend=yes ==> Coffee Eclair=yes
2. Almond Twist=yes Hot Coffee=yes Weekend=yes ==> Apple Pie=yes
3. Apple Pie=yes Almond Twist=yes Hot Coffee=yes Weekend=yes ==> Coffee Eclair=yes
Knowing that the setup of the file is different from previous one, this time only yes/positive values appear. Additionally there is a huge difference in occurrences in comparison to the previous file. In the previous one up to 800 occurrences of rule can be found, here only 10.

Task 5.) Basically using different metrics result in the same statement in Task 4. People that buy Almond Twists, Apple Pies or Coffee Eclairs also buy Hot Coffee on Weekends and vice versa. All other metrics confirm this picture. Otherwise the associated parameter work similar like in Task 2.

Task 6.) Try the other associators. What are the differences to Apriori?
Aside from the rules found hot coffee and weekend, FPGrowth also found three higher occuring rules about
1. [Hot Coffee=yes, Apricot Croissant=yes]: 32 ==> [Blueberry Tart=yes]: 32
2. [Raspberry Cookie=yes, Raspberry Lemonade=yes]: 29 ==> [Lemon Cookie=yes]: 29
3. [Vanilla Frappuccino=yes, Walnut Cookie=yes]: 18 ==> [Chocolate Tart=yes]: 18
On the FilteredAssociator I can’t get a result. It runs forever.

Task 7.) What golden nuggets did you find, if any?
Yes I found the following golden nugget: Offer on weekends Hot Coffee to special prices to attract customers if they buy Apple Pies, Coffee Eclairs or Almond Twists, that way sales could be increased as it seems people are interested in this. In the same way offer on weekdays specials where for each person that buys Apricot Croissant and Blue Blueberry Tart gets a reduced coffee. In general encourage your customer buying more products through the knowledge of the rules that have been generated in Task 4 and 6.