3.1
3.1.1

Table 1

Run NO.	Classifier	Parameters Parameters	Training Error	Cross-valid Error	Overfitting
1	ZeroR	Default	30.0%	30.0%	None
2	OneR	Default	25.7%	33.9%	lower significant
3	J48	Default	14.5%	29.5%	significant overfitting
4	IBK	Default	0%	28%	Extremely overfitting

The training error reflects the error of classification when testing the model on the instances from the training set, and in the test part, a part data which has been trained will be used to test. However, the testing set Cross-valid error rate put the data set into 10 folds, and each time, use 9 parts for training and 1 part for a test which has not been used in training, which reflect the true performance of the model. A higher gap between training error and cross-valid error can be determined as overfitting which means the model match or 'customize' training set so much.
Normally, the gap about 5% will be considered to not significant overfitting, and about 15% will be considered as significant overfitting, and higher that will be considered as extremely overfitting. In this test results, IBK and J48 do not have a good performance because of the gap of 15% and 28%.

3.1.2

Table 2

Run No.	C	M	Training Error	Cross-valid Error
1	0.1	16	24.5%	27.4%
2	0.25	21	23.2%	27%
3	0.35	20	22%	26.9%
4	0.5	20	21.9%	27%
5	1	59	25.5%	27%

C: confidence factor, which means the lower value, the more branch will be cut.
M: minimal number of values in the leaf node, which means, for a sub dataset, if the numbers lower than the threshold, that would not be split further.
After changed parameters, although the results have been increased a little bit, it decreases the situation of overfitting, which means this training model is more suitable for unseen data.

3.1.3

Table 3

Run No.	Percentage Split	Test Accuracy	Train Accuracy
1	40%	70.50%	85.50%
2	66%	72.65%	85.50%
3	80%	77%	85.50%
4	90%	74%	85.50%

40% split will separate the whole data set into two parts randomly, which is 40% for the training model and 60% for test. Normally, a higher percentage of data to train model will get a better model. However, when 90% data set was invoked to model, the accuracy rate has been decreased a little bit. Because it lets mode learn some 'unpopular' data, this is also a kind of overfitting. What is more, because train accuracy is tested by the data which has been used in a train, the accuracy rate is keeping in 85.5%.

3.1.4

Table 4

Run No.	Value of K	Test Accuracy	Train Accuracy
1	1	72.00%	100.00%
2	2	72.30%	85.20%
3	3	73.30%	86.00%
4	4	74.50%	80.60%
5	5	74.20%	82.30%
6	6	74.30%	79.70%
7	7	74.00%	80.90%
8	8	74.30%	78.60%
9	10	74.00%	77.90%
10	15	73.50%	77.60%
11	20	73.00%	75.00%
12	35	73.70%	74.70%

From the table, it is clear to see that, with the increase of the value of K, the training accuracy is keeping decrease and the highest test accuracy happened in k = 4, which is about 74.5%. After that, the test accuracy rate decreases slowly. Furthermore, the overfitting has been decreased with the increase of k. For example, in this table, k = 35 will make the lowest overfitting. However, test accuracy also needs to be considered.

3.1.5

Table 5

Run NO.	Classifier	Parameters	Training Error	Cross-valid Error
1	OneR	Default	25.70%	33.90%
2	OneR	Default	28.30%	28.90%
3	J48	C = 0.35, M = 20	22.00%	26.90%
4	RandomizableFilteredClassifier	None	0%	34%
5	randomforest	max depth = 7, seed = 25	0.30%	22.90%

From this table, it is clear to see that RandomizableFilteredClassifier have the highest cross-valid error rate and the most serious overfitting. What is more, the random forest got the best predicted cross-valid error with parameters-max depth = 7 and seed = 25. However, it got serious overfitting.

3.1.6

The error rate of the ZeroR is 30%, the best error rate of the OneR is 28.9% which is calculated by credit history, and the best IBK is 25.5%. Although there are 20 combinations of attributes which can be used to train model, they cannot get an extremely good result. Because the feature can be determined by some specific attributes, such as credit amount, credit history which do not have a strong relationship with each other that can contribute to a higher predict model, it is hard to get a very high result. What is more, because this dataset just has 30% people as a ‘bad’ feature, the ZeroR will process all the target as ‘good’, however, which, technically, should not be seen as a benchmark.

3.1.7

The attributes of credit history, credit amount and duration will have a higher ability to predict the target feature as a single attribute. However, all of them do not have a strong contribution with each other.

3.1.8

Table 6

Run NO.	Classifier	Parameters	Training Error	Cross-valid Error	dataset
1	ZeroR	Default	30.00%	30.00%	full
			30.00%	30.00%	dataset 1
			30.00%	30.00%	dataset 2
2	OneR	Default	25.70%	33.90%	full
			28.30%	28.90%	dataset 1
			25.70%	33.90%	dataset 2
3	J48	Default	14.50%	29.50%	full
			24.80%	29.50%	dataset 1
			20.40%	25.70%	dataset 2
4	IBK	Default	0.00%	28.00%	full
			20.00%	29.60%	dataset 1
			0.00%	32.40%	dataset 2

Dataset1: checking_status, duration, credit_history and class
Dataset2: checking_status, duration, credit_history, credit_amount, other_parties, age and class

Because huge of attributes may not have a positive influence on a classifier model, which may add some invalid data items that reduce the accurate rate for training model rather than contribute to training a model, properly reducing the data set helps to make the model more simple, efficient and accurate. Or, although dataset got a similar accurate rate with a simpler training model.
Thus, in this data set, a filter has been used and the parameters of attribute selection are CfsSubsetEval and the best first (dataset1). This algorithm is to assess attribute based on general characteristics of the data before entering data into a machine learning algorithm. From the result, we can see that the accuracy did not decrease so much in J48, increase a little bit in OneR and the rest of them remain unchanged.
In the second one(dataset2), the wrapper has been chosen as the feature selection algorithm and the parameters are wrapperSubsetEval, greedyStepwise, and the correct results will be tested by J48. Because learning method is part of procedure for wrapper, and there is loop for system choose the best combination with the highest correct rate by greedy step ways, and in this data, the algorithm will be taken as J48, in the results, J48 gets a lower cross-valid error rate, which is about 25.7%.

3.2
3.2.1

Table 1

Run NO.	Classifier	Parameters	Training Error	Cross-valid Error
1	ZeroR	default	39.31	39.46
2	M5P	default	36.77	39.9
3	IBK	default	0.77	54.46

This data set has 14 attributes with 7 categorical values and 7 numerical values and many outliers in 'cholesterol' which is a feature that needed to be predicted. Basically, for the ZeroR, this algorithm just takes the average value as a standard to predict the target value, which is 247.747. What is more, for M5P, this algorithm should be especially suitable for the dataset with a combination of numerical attributes and categorical attributes. However, this algorithm gets a higher cross-valid error, after visualizing the tree, it is clear to see that there are only two branches with one root and split by sex into two leaf nodes. There is no strong relationship for all attributes to create a linear relationship. Finally, although IBK got a lower training error, the cross-valid error is extremely high, which means the training model is not suitable for test data. This is extremely overfitting.
A correlation coefficient shows a very lower correlation, which means low accuracy. Furthermore, the uneven density of data and sparse data does not help the model to learn enough.
3.2.2

Table 2

Run NO.	Classifier	Parameters	Training Error	Cross-valid Error
1	M5P	m = 14, unpruned = true	34.03	38.95
2	M5P	m = 30, unpruned = true	34.1	38.57
3	M5P	m = 35, unpruned = true	34.88	38.94
4	M5P	m = 68, unpruned = true	36.62	38.62
5	M5P	m = 88, unpruned = true	37.01	38.21
6	IBK	k = 22	37.17	38.45
7	IBK	k = 25, distanceWeight =1-distance	36.60	38.43
8	IBK	k = 65, distanceWeight = 1/distance	6.48	38.14
9	IBK	k = 75, distanceWeight =1-distance	37.59	38.37
10	IBK	k= 85, distanceWeight =1-distance	37.7	38.31

Unpruned means allowed the tree to grow naturally and min number instance will cut the branch when instances meet the minimal number.
K means how many neighbors near the target value. 1-distance will process this model as regression, which will normalize the values of numeric attributes to the 0-1 range to solve the hug value of cholesterol, compared with other attributes.

3.2.3

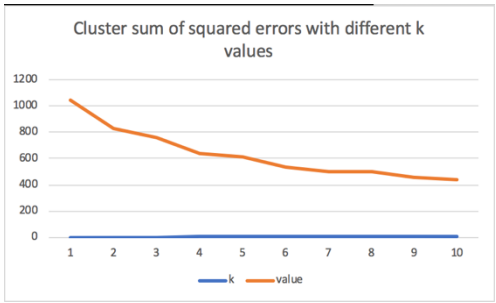
Table 3

Run NO.	Classifier	Parameters	Training Error	Cross-valid Error
1	Random forest	default	14.7	40.4
2	Random forest	depth = 6	26.22	39.82
3	Kstar	default	0.26	47.86
4	Kstar	globalBlend=80	22.38	39.47
5	LWL	default	36.38	39.31

These three algorithms are used in this step, which includes random forest, kstar and LWL.
In the random forest, the parameter maximum depth can be used to prune branches which help to control overfitting. The Kstar algorithm with default value has obvious overfitting. However, it can be fixed by setting different parameters such as globalBlend. The last algorithm celled LWL that works well with the default parameter, which has the best result, compared with the other two algorithms, and almost without overfitting.
3.2.4
The Apriori algorithm could be used to find an association between those attributes. It shows that if ‘cp’ equals to 4 and ‘thal’ equals to 3, ‘fbs’ will have 96% possibility to equal to 0. Also,s if ‘sex’ equal to 0 and ‘slope’ equal to 1, ‘thal’ will have 96% possibility to equal to 3.

3.3.1
Firstly, with the increase of k values, the cluster sum of squared errors is decreasing from about 1039.7 to 258.338. The K value means how many clusters in the dataset which you want. Because there are two data features in this data set which is ‘good’ and ‘bad’, the best situation is two clusters with two kinds of data. However, it is easy to see from the visualization that two kinds of features were merged with different kinds of attributes. Thus, the clusters cannot be separated directly as we thought. What is more, with the increasing of k value, there are two ways to judge does it divide into good clusters or not. The first one is to calculate the average distance between data points and centroids with different k. If the average distance falls rapidly until the right k, then falls much more slowly (strong bend), that is the best k value. The second one is to inspect the cluster means and standard deviations. If two generated clusters with means that are so close to each other, those should be merged to the same cluster, which means the value of k needs to be decreased.
3.3.2
Different seeds will influence the first location for initialized centroids which may affect the final output because the K-means is sensitive with the first random selected centroid.
3.3.3
The instances fall into 7 roughly equal-sized groups. For example, for cluster 0, there are 100 test instances and the credit amount is about 1254 and the age is about 31. For cluster 1, the mean age is about 54, which ranges from about 46.63 to 62.6 (1 standard deviation) which may have an intersection with cluster 2. For cluster 0,3,5,6, they are fairly similar and could possibly be merged into one cluster.
3.3.4
Normalisation is used to put attributes onto roughly the same scale which puts the values ranged from 0 to 1, and reduce the influence of the different data scale for numerical values.
3.3.5
minStdDev: this value can help prevent arithmetic overflow resulting from multiplying large densities, and the bigger value may combine all the clusters, and on the contrary, it will separate more clusters. Normally, this value should correspond roughly to the number of decimal places in the data.
MinLogLikehoodImprovementCV: help to find the best number of clusters when cross-validation by increasing the number of clusters.
MinLogLikehoodImprovementIterating: minimum improvement in log likelihood, which determines the necessary iterations to find the best model.
3.3.6
The sum of squared errors with different k values will decrease quickly from 1 to 4, and after 4, the squared errors reduce slowly. Thus, the k needs to be selected as 4.

Chart 3



3.3.7
k-means is a variant of EM, with the assumptions that clusters are spherical. For EM, there are two steps. E- step: each object is assigned to the centroid in a most likely cluster. M-steps: recompute the centroid. After iterating these two steps, it will get a training model. Because, in this dataset, different datasets have combined in each other without a specific cluster shape, EM will be more suitable.
3.3.8
According to EM, we can get a conclusion that more than half percent of new car has been sell to the age between 27 to 39, which means young people will have a higher chance to buy a new car.

3.4

3.4.1

In the file groceries1.arff, the information is stored as true and false. If a person brought a product, this information was stored as true. Otherwise, false was stored. In the file groceries2.arff, only true value is recorded.

3.4.2

Because there are too many attributes in this data set and the processed data subset could be 2^n -1, it cannot finish the running. Thus, I removed some attribute which contains less than 100 true variables. The minimum support is set as 0.05, minimum confidence is set as 0.7.

According to the results, many values are false, which will have higher support and confidence, but it is meaningless because people only choose what they want.

Picture 1

```
 1. citrus_fruit=F margarine=F 8523 ==> semi-finished_bread=F 8389    <conf:(0.98)> lift:(1) lev:(0) [16] conv:(1.12)
 2. citrus_fruit=F 9021 ==> semi-finished_bread=F 8871    <conf:(0.98)> lift:(1) lev:(0) [9] conv:(1.06)
 3. margarine=F 9259 ==> semi-finished_bread=F 9105    <conf:(0.98)> lift:(1) lev:(0) [9] conv:(1.06)
 4. citrus_fruit=F semi-finished_bread=F 8871 ==> margarine=F 8389    <conf:(0.95)> lift:(1) lev:(0) [37] conv:(1.08)
 5. citrus_fruit=F 9021 ==> margarine=F 8523    <conf:(0.94)> lift:(1) lev:(0) [30] conv:(1.06)
 6. semi-finished_bread=F 9661 ==> margarine=F 9105    <conf:(0.94)> lift:(1) lev:(0) [9] conv:(1.02)
 7. citrus_fruit=F 9021 ==> margarine=F semi-finished_bread=F 8389    <conf:(0.93)> lift:(1) lev:(0) [37] conv:(1.06)
 8. margarine=F semi-finished_bread=F 9105 ==> citrus_fruit=F 8389    <conf:(0.92)> lift:(1) lev:(0) [37] conv:(1.05)
 9. margarine=F 9259 ==> citrus_fruit=F 8523    <conf:(0.92)> lift:(1) lev:(0) [30] conv:(1.04)
10. semi-finished_bread=F 9661 ==> citrus_fruit=F 8871    <conf:(0.92)> lift:(1) lev:(0) [9] conv:(1.01)
```

Treat zero as a missing parameter as true has been set. Thus, all the rules come with true information, which is easier to find golden nugget for the business.

Although many rules are coming out with high support and confidence, the lift is only valued 1, which means that the item is independent. Thus, those rules do not have a contribution at all.

3.4.3

Because most of the items in this data set are independent, it is really hard to get the meaningful rules from it. The value of lift should be greater than 1 to determine those items are relative. However, Weka cannot give me the rule even I set the minMetric as 1.1. thus, it is better to use groceries2.arff to get more useful rules.

3.4.4

In this step, some attributes which have a lower number of instances also need to be removed. However, it shows no large itemset or rule found because, compared with the whole dataset, the value of true is very rare. If the parameter of lower bound minimal support can be set to 0.001, some relationship can be found.

3.4.5

The minimum support as 0.05 and minimum confidence as 0.7 has been set. However, there are no results. It can be assumed that because the confidence of item is not higher than 0.7. Thus, the confidence as 0.3 has been set.

Picture 2

```
1. whole_milk=T 2513 ==> yogurt=T 551    conf:(0.22) < lift:(1.57)> lev:(0.02) [200] conv:(1.1)
2. yogurt=T 1372 ==> whole_milk=T 551    conf:(0.4) < lift:(1.57)> lev:(0.02) [200] conv:(1.24)
3. whole_milk=T 2513 ==> other_vegetables=T 736    conf:(0.29) < lift:(1.51)> lev:(0.03) [249] conv:(1.14)
4. other_vegetables=T 1903 ==> whole_milk=T 736    conf:(0.39) < lift:(1.51)> lev:(0.03) [249] conv:(1.21)
```

As we can see the rules are coming out, which shows the confidence are 0.4, 0.39 and 0,31.

Then the metric type as lift and minMatric as 1.3 has been set.

Picture 3

```
Best rules found:

1. yogurt=T 1372 ==> whole_milk=T 551    <conf:(0.4)> lift:(1.57) lev:(0.02) [200] conv:(1.24)
2. other_vegetables=T 1903 ==> whole_milk=T 736    <conf:(0.39)> lift:(1.51) lev:(0.03) [249] conv:(1.21)
3. rolls_buns=T 1809 ==> whole_milk=T 557    <conf:(0.31)> lift:(1.21) lev:(0.01) [94] conv:(1.07)
```

After that the metric as leverage and conviction have been set, however, the rules are almost the same. The confidence of those rules is low, it can be tried to improve by setting the support to the lower value.

Picture 4

```
1. flour=T root_vegetables=T whipped_sour_cream=T 17 ==> whole_milk=T 17    <conf:(1)> lift:(3.91) lev:(0) [12] conv:(12.66)
```

Although the confidence became to 1, the support is only 17, which does not have any meaningful information.

3.4.6

The parameters in another associator celled FPgrowth is almost the same as Apriori. However, it is much faster than Apriori algorithm, because it scans only twice from the data set.

The filtered associator can filter the data set before we use an algorithm to analyze the data.

3.4.7

There is no really strong association between those items in the data set, the confidence is too low to determine the association. But there is one information we can get is that the whole mile item is really popular in the

market, which means it is a necessity for life. Thus, we can put this product into a deep location in the market, when people come to buy it, they need to walk through more product, which could make them to purchase more products.

3.4.8

In the credit dataset, the Apriori algorithm cannot work directly. Thus, all the numerical attributes have been removed to make the algorithm work.

Picture 5

```
 1. other_parties=none 907 ==> foreign_worker=yes 880    <conf:(0.97)> lift:(1.01) lev:(0.01) [6] conv:(1.2)
 2. other_parties=none other_payment_plans=none 742 ==> foreign_worker=yes 718    <conf:(0.97)> lift:(1) lev:(0) [3] conv:(1.1)
 3. other_payment_plans=none 814 ==> foreign_worker=yes 782    <conf:(0.96)> lift:(1) lev:(-0) [-1] conv:(0.91)
 4. other_payment_plans=none foreign_worker=yes 782 ==> other_parties=none 718    <conf:(0.92)> lift:(1.01) lev:(0.01) [8] conv:(1.12)
 5. foreign_worker=yes 963 ==> other_parties=none 880    <conf:(0.91)> lift:(1.01) lev:(0.01) [6] conv:(1.07)
 6. other_payment_plans=none 814 ==> other_parties=none 742    <conf:(0.91)> lift:(1.01) lev:(0) [3] conv:(1.04)
 7. other_payment_plans=none 814 ==> other_parties=none foreign_worker=yes 718    <conf:(0.88)> lift:(1) lev:(0) [1] conv:(1.01)
 8. other_parties=none 907 ==> other_payment_plans=none 742    <conf:(0.82)> lift:(1.01) lev:(0) [3] conv:(1.02)
 9. other_parties=none foreign_worker=yes 880 ==> other_payment_plans=none 718    <conf:(0.82)> lift:(1) lev:(0) [1] conv:(1)
10. foreign_worker=yes 963 ==> other_payment_plans=none 782    <conf:(0.81)> lift:(1) lev:(-0) [-1] conv:(0.98)
```

The rules show as default parameters, as we can see the rules have high support and confidence. They meet the requirement of a strong association. However, the lift value is only 1, which means those events are independent. Thus, the metric type as lift and minMetric=3 has been set.

Picture 7

```
 1. other_parties=none property_magnitude=no known property foreign_worker=yes 139 ==> housing=for free 100    conf:(0.72) < lift:(6.66)> lev:(0.08) [84]
 2. housing=for free 108 ==> other_parties=none property_magnitude=no known property foreign_worker=yes 100    conf:(0.93) < lift:(6.66)> lev:(0.08) [84]
 3. housing=for free 108 ==> other_parties=none property_magnitude=no known property 100    conf:(0.93) < lift:(6.57)> lev:(0.08) [84] conv:(10.31)
 4. housing=for free foreign_worker=yes 108 ==> other_parties=none property_magnitude=no known property 100    conf:(0.93) < lift:(6.57)> lev:(0.08) [84]
 5. other_parties=none property_magnitude=no known property 141 ==> housing=for free 100    conf:(0.71) < lift:(6.57)> lev:(0.08) [84] conv:(2.99)
 6. other_parties=none property_magnitude=no known property 141 ==> housing=for free foreign_worker=yes 100    conf:(0.71) < lift:(6.57)> lev:(0.08) [84]
 7. other_parties=none housing=for free 103 ==> property_magnitude=no known property foreign_worker=yes 100    conf:(0.97) < lift:(6.39)> lev:(0.08) [84]
 8. property_magnitude=no known property foreign_worker=yes 152 ==> other_parties=none housing=for free 100    conf:(0.66) < lift:(6.39)> lev:(0.08) [84]
 9. property_magnitude=no known property foreign_worker=yes 152 ==> housing=for free 104    conf:(0.68) < lift:(6.34)> lev:(0.09) [87] conv:(2.77)
10. housing=for free 108 ==> property_magnitude=no known property foreign_worker=yes 104    conf:(0.96) < lift:(6.34)> lev:(0.09) [87] conv:(18.32)
```

In this time, the results show that the lower support, but really high confidence and lift. All of the rules here have the lift value greater than 6, which shows a strong dependent relation between those items.