Part1

1.

Run No	Classifier	Parameters	Training Error	Cross-valid Error	Over-Fitting
1	ZeroR	default	50 %	50.0095 %	0.0095% (none)
2	OneR	default	26.1297 %	29.9017 %	3.772% overfitting (slightly)
3	J48	Default (C 0.25 M 20)	10.6825 %	14.6436 %	3.9611% overfitting (slightly)
4	IBK	Default (K=1)	0 %	28.1244 %	28.1244% overfitting (heavily)

The results after running different classifiers with the default parameters are shown as above. ZeroR, as the simplest classification method, relies on the target and ignores all predictors. It determines baseline performance as a benchmark for other classification methods. However, the result is 50%, we cannot tell anything from the ZeroR classifier. OneR and J48 classifier are both slightly overfitting. The J48 gives us the highest accuracy. IBK gives the largest overfitting comparing to other classifiers. Nevertheless, using the training set for IBK classifier gives us nothing due to the algorithm.

2.

Run No	Classifier	Parameters	Training Error	Cross-valid Error	Over-Fitting
1	J48	C=0.1, M=10	13.443 %	15.069 %	1.626% (slightly)
2	J48	C=0.1, M=100	17.6782 %	18.132 %	0.4538% (slightly)
3	J48	C=0.1, M=1000	23.927 %	24.2012 %	0.2742% (slightly)
4	J48	C=0.01, M=10	15.8726 %	16.7045 %	0.8319% (slightly)
5	J48	C=0.001, M=10	15.9671 %	16.6383 %	0.6712% (slightly)

By changing the parameters C and M of J48 classifiers, we found the best combination for overfitting result is 0.01(C) and 1000(M). C is the confidence factor, it determines how aggressive the pruning process will be. So smaller value induce more pruning. It can affect classifier performance significantly. M determines what the minimum number of observations are allowed at each leaf of the tree. Therefore, it is obvious that by increasing M, overfitting will get better.

3.

Run No	Classifier	Training set (%)	Error rate
1	J48	80	15.1229 %
2	J48	70	14.7494 %
3	J48	66	14.8179 %
4	J48	50	16.1278 %
5	J48	40	16.2912 %
6	J48	30	16.3808 %

Changing the number in percentage split can change the error rate. When using a larger number of examples in the training set to build the model rather than the default 66, the error rate could be higher or lower. The error rate increases when the percentage of training set decreases in this case, but the error rate not always follow

this rule.

4.

IBK	K	Training Error	Cross-valid Error	Over-Fitting
1	2	12.1573 %	28.5309 %	16.3736% (heavily)
2	3	13.8117 %	26.6497 %	12.838% (medium)
3	4	15.3526 %	25.9406 %	10.588% (medium)
4	5	17.3284 %	25.6476 %	8.3192% (medium)
5	10	19.4933 %	24.532 %	5.0387% (medium)
6	50	24.2579 %	25.7421 %	1.4842% (slightly)
7	100	25.8745 %	26.9238 %	1.0493% (slightly)
8	500	28.5593 %	29.0887 %	0.5294% (slightly)

K is the number of nearest neighbors. Although the overfitting is in direct proportion to the value of k, we cannot say that they larger k is, the better performance we get. The class of the test set should be determined by the majority of the class of the nearest neighbors. It is important to get the appropriate k value to minimize the error. If the value of k is too low and too many noisy points in the training set, the performance could be bad. In addition, if the value of k is too high, it is easy to get errors because it may cross class boundaries.

5.

Run No	Classifier	Parameters	Training Error	Cross-valid Error	Over-Fitting
1	BayesNet	default	20.2023 %	20.8641 %	0.6618%(slightly)
2	Logistic	default	16.2034 %	16.4398 %	0.2364%(slightly)
3	IBK	K=2	12.1573 %	28.5309 %	16.3736%(heavily)
4	Bagging	default	9.3212 %	13.8211 %	4.4999%(slightly)
5	InputMappedClassifier	default	50 %	50.0095 %	0.0095% (slightly)

As the classifiers and parameters shown above, Bagging gives us the best predictive accuracy on Training Error. However, it is slightly overfitting. Logistic has better performance in over-fitting. The Training Error and Cross-validation Error of InputMappedClassifier is around 50%, we can not use this method to predict accuracy.

6.At cross-validation option, ZeroR accuracy is 50.01 %, OneR is 70.10%, Best J48 is 83.36%. ZeroR gives us nothing but can be used as a baseline performance benchmark when comparing with other classification methods. The accuracy of OneR depends on the best attribute found, so sometimes the accuracy could be lower as other classification methods may take account of other attributes. Different C value and M value gives us different accuracy in J48. Getting the appropriate and reasonable combination of C and M can give us the best accuracy. Therefore, from the bank-balanced1 data, we can see that there is a single dominant attribute, but some combination of all of the attributes is also needed.

7. From the result above, we found that Bagging can give us the best accuracy of data prediction, but medium overfitting. However, J48 gives us both high accuracy and reasonable overfitting rate compare with other classifiers.

When the duration of the last contact is bigger than 503, the client is likely to subscribe a term deposit, but when the duration is smaller than 503, the possibility of a client to subscribe a term deposit is mainly determined by month and day.

8.

Run No	Classifier	Attributes	Set	Cross-valid Accuracy	Time taken (seconds)
1	J48	Full attributes	Full set	85.3564 %	0.24
2	J48	Age,contact,day, month,duration,previous, poutcome,y	Reduced set	85.8574 %	0.09

By using AttributeSelection filter and apply attribute selection with WrapperSubsetEval, BestFirst and J48 as the classifier in WrapperSubsetEval, we found that there is no big difference between using full set and using reduced set. Using an attribute selection algorithm helps us to find the most relevant attributes, and reduce process time from 0.24 to 0.09 seconds.

Part2

1.

Run No	Classifier	Parameters	Training Mean absolute error	Cross-valid Mean absolute error	Over-Fitting
1	ZeroR	default	1688.8122	1689.1791	0.3669(slightly)
2	M5P	default	1562.7685	1645.8208	83.0523(medium)
3	IBK	Default(k=1)	0	2091.5265	2091.5265(heavily)

ZeroR gives us the baseline performance benchmark for other classification. However, in this scenario, the performance of IBK is worse due to its algorithm and the parameters selected that we can not learn anything from it with the default parameters. Among these three, even though M5P gives us lowest mean absolute error, it is close to the result of ZeroR and the overfitting is high.

2.

Run No	Classifier	Parameters	Training Mean absolute error	Cross-valid Mean absolute error	Over-Fitting
1	M5P	M=10	1605.1162	1623.529	18.4128(slightly)
2	M5P	M=100	1608.4825	1618.6813	10.1988(slightly)
3	M5P	M=1000	1620.616	1625.435	4.819(slightly)
5	IBK	K=100	1603.2384	1622.0227	18.7843(slightly)
6	IBK	K=1000	1633.3081	1636.6517	3.3436(slightly)

In terms of 20 runs for each Classifier, the classifier IBK with K equals 100 gives us the lowest training mean absolute error and the M5P with M equals 100 gives us the lowest cross-validation mean absolute error. For overfitting, IBK with K parameter is 1000 gives the smallest value of overfitting rate. Overall, M5P has the best performance in this scenario balancing errors and overfitting. However, M, which is the minNumInstances, it might not appropriate to set M equals 1000 in some cases.

3.

Run No	Classifier	Parameters	Training Mean absolute error	Cross-valid Mean absolute error	Over-Fitting
1	RandomTree	K=1	8.6317	2146.3947	2137.763 (heavily)
2	RandomTree	K=5	12.195	2210.4122	2198.2172 (heavily)
3	M5Rules	M=100	1608.4825	1618.8923	10.4098 (slightly)
4	M5Rules	M=1000	1620.616	1625.435	4.819 (slightly)
5	Vote	S=1	1688.8122	1689.1791	0.3669 (slightly)

Vote has the lowest number of overfitting. RandomTree has the smallest Training Mean absolute error with K equals 1, but the cross-validation Mean absolute error is very high, so the overfitting is heavy. In terms of predictive accuracy and overfitting, M5Rules with parameter M equals 1000 has the best performance in this scenario.

4. From the result above, we cannot find any significant improvement by using different classifier compare with ZeroR, which means that we cannot predict the balance competently with this data.

Part 3:

1. Compare with the result of different values of K parameter, with the increasing of K, we can get lower cluster sum of squared errors, but the result cannot give a distinct cluster distribution with default seeds parameter is 10. In addition, if K is too big some of the clusters will be very similar and can be manually combined.
2. The seed number is the randomization for initial K points. K represents the number of the Clusters. Changing the seed value will result in the different initial cluster. In this scenario, it is obvious that the value of seeds does not directly affect the cluster sum of squared errors when we set the parameter K to 4. Because there is a fluctuation in cluster sum of squared errors when running seed number from 10 to 1000.
3. Running the EM algorithm with the default parameters gives us 9 clusters, from Cluster 0 to Cluster 8. However, by observing the mean and standard deviation of the age of nine clusters. They are largely overlapped which indicates that the cluster boundaries are not clear between each other. However, we can say most people fall in Cluster 7, their mean age is 46. Most of them have secondary school qualification in this cluster same as Cluster 0, Cluster 2, Cluster 3, Cluster 5 and Cluster 6. Within all Clusters, most people's marital status is married. Same as their age, their value of balance is also largely overlapped.
4. There has a normalized display on numeric data(age & balance), of which the minimum is 0 and maximum is 1, but not change to marital and education data. And the likelihood of normalized cluster is 1.83 compared with the original data -13.85.
5. The Number of clusters selected by cross validation decrease when the value of minLogLikelihoodImprovementCV increase. The Number of iterations decreases when the value of minLogLikelihoodImprovementIterationg or minStdDev increase. For the likelihood, there has a slight decrease when the values of minLogLikelihoodImprovementCV, minStdDev and minLogLikelihoodImprovementIterationg increase.
6. 9 cluster from EM algorithm. One of them corresponds to people who age 51.59 with 14.9 standard deviation. Among these people, most of them are married and they are secondary education above background, and they have the richest balance with mean 12101.90, but the standard deviation is also the highest 10158.53.
7. EM algorithm is better than Kmeans. Since Kmeans algorithm considers the relationship between the value of K and squared error. EM assigns objects according to the probability of occurrence of members relationship between objects and cluster, and it gives more accuracy and detail, although it was more time-consuming than Kmeans algorithm.
8. By using Kmeans with setting the value of K parameter to 4 and the value of seeds to 50, we can get a distinct cluster distribution with 4 clusters between age and education.

Part4

1. Both supermarket1.arff and supermarket2.arff list the number of attributes. The only difference is that they have different values. Supermarket1.arff has two values (t and f) for each attribute except attribute total. Supermarket2.arff just has one value(t) or none attribute except attribute total.
2. We separated all the attributes in 15 runs and finally finished running the Apriori algorithm with default setting on this data. For each run, we got 10 best rules. By observing the rules, we can see all the Confidence value, Lift value, Leverage value and Conviction value are the same, they are 1, 1, 0, 0, 0 respectively. Even though the confidence value is one, the association is still not useful. For example, picking one of the rules, if salads not occur, then department 57 not occur, which we can not understand the meaning from it.
3. We run 4 different possibilities of the metric type and associated parameters, which is minMetric 0.3 for metricType Confidence, minMetric 0.01 for metricType Lift, minMetric 0.01 for Leverage and minMetric 0.3 for metricType Conviction. The results are very similar to the result we got in 4.2.
4. Using the default setting of Apriori algorithm, we got 10 best rules with confidence all above 0.9. The confidence of the best top 1 rule is 0.92. It can be interpreted as the 92% of the occurrence for buying bread and cake will also buy biscuits, frozen foods, fruit with a high total. However, it is not acceptable involving the labeled class total when getting association rule. The data should be unsupervised.
5. After we remove the class attribute total, the MetricType Confidence with minMetric 0.9 gives us the following result:

```
Best rules found:

 1. biscuits=t frozen foods=t pet foods=t milk-cream=t vegetables=t 516 ==> bread and cake=t 475    <conf:(0.92)> lift:(1.28) lev:(0.02) [103] conv:(3.44)
 2. baking needs=t biscuits=t milk-cream=t margarine=t fruit=t vegetables=t 505 ==> bread and cake=t 464    <conf:(0.92)> lift:(1.28) lev:(0.02) [100] conv:(3.37)
 3. biscuits=t frozen foods=t milk-cream=t margarine=t vegetables=t 585 ==> bread and cake=t 537    <conf:(0.92)> lift:(1.28) lev:(0.03) [115] conv:(3.35)
 4. biscuits=t canned vegetables=t frozen foods=t fruit=t vegetables=t 536 ==> bread and cake=t 492    <conf:(0.92)> lift:(1.28) lev:(0.02) [106] conv:(3.34)
 5. baking needs=t frozen foods=t milk-cream=t margarine=t fruit=t vegetables=t 517 ==> bread and cake=t 474    <conf:(0.92)> lift:(1.27) lev:(0.02) [101] conv:(3.29)
 6. biscuits=t frozen foods=t pet foods=t milk-cream=t fruit=t 511 ==> bread and cake=t 468    <conf:(0.92)> lift:(1.27) lev:(0.02) [100] conv:(3.26)
 7. biscuits=t frozen foods=t tissues-paper prd=t milk-cream=t vegetables=t 575 ==> bread and cake=t 526    <conf:(0.91)> lift:(1.27) lev:(0.02) [112] conv:(3.22)
 8. biscuits=t frozen foods=t beef=t fruit=t vegetables=t 536 ==> bread and cake=t 490    <conf:(0.91)> lift:(1.27) lev:(0.02) [104] conv:(3.2)
 9. baking needs=t biscuits=t frozen foods=t cheese=t fruit=t 538 ==> bread and cake=t 491    <conf:(0.91)> lift:(1.27) lev:(0.02) [103] conv:(3.14)
10. biscuits=t frozen foods=t milk-cream=t margarine=t fruit=t 592 ==> bread and cake=t 540    <conf:(0.91)> lift:(1.27) lev:(0.02) [113] conv:(3.13)
```

When we change the parameters within Confidence MetricType from 0.1 to 0.8, they give us different results. When we try different metric types like Lift, Leverage and Conviction with associated parameters, all results of the best 10 rules are also totally different.
6. By using FPGrowth associator with supermarket2 data, we can get the same rules found in Apriori but without the detail of sets of large itemsets. When using the FPGrowth with supermarket1 data, it will run out of memory which is different from using Aprior that require to a higher computer configuration.
For FilteredAssociator, it runs to failed on both supermarket1 and supermarket2 files.
7. The best rule we found with running Apriori algorithm(default parameters) is people who buy biscuits, frozen foods, pet foods, milk-cream and vegetables have high probability buying bread and cake. In addition, the probability of occurrences of buying biscuits is very high. However, it is normal that people buy groceries including all kinds of food.
8. By using the Apriori algorithm, we get 10 best rules with confidence from 0.98 to

0.99. It can be interpreted from the second best rule that people who not the personal loan and using cellular as the contact communication type are not likely to have credit in default which accounts for 99% occurrence. Overall, it is hard to find any meaningful associations in this bank data. However, it is possible that if we narrow the dimension for each attribute, the association rule can be better. For instance, the value of 'unknown' and 'other' can be categorized into one group in poutcome.