DATA MINING: Quiz 4

1. Assume you have 5 *independent* classifiers, each of them with an accuracy of 0.7. Compute which is the accuracy for the *Majority Vote* algorithm for those 5 classifiers.
   For this problem we rely on the following formula:

$$P_{maj} = \sum_{m=0}^{\lfloor L/2 \rfloor} \binom{L}{m} p^{L-m}(1-p)^m$$

   Being L = 5 and p = 0.7:
   =0.16807 + 0.36015 + 0.3087 = 0.83692

2. Briefly explain if each of the following claims is true or not and why:
   a. The larger the number of iterations in the bagging method, the lower the variance of results and the larger the accuracy obtained
      True, this is what bagging does, when we combine the outputs of different classifiers trained on different samples of the training, it helps reducing the overall variance and results in a higher accuracy.

   b. Boosting cannot be applied to support vector machines because the linear combination of hyperplanes is another hyperplane.
      False, boosting can also be applied to support vector machines. In cases when the data is hard SVM becomes a weak learner, making it applicable to boosting.

   c. When the "a" parameter in random forests is set to the number of features, random forest is equivalent to bagging with decision trees.
      True, when using decision trees, we can choose the number of features (using the parameter 'a') we consider making the split between the nodes. Bagging, on the other hand, uses all the features possible.

d. Diversity of classifiers is the source of success in meta-method. In order to ensure this diversity, we always train classifiers with different training datasets.

True, we can use different classifiers using different methods in order to provide more diversity in our results. For example, we can use bagging with *knn*, *naïve bayes* or *decision trees* to train our datasets even more.

3. When implementing the main loop of the *Adaboost* procedure, what should we do when the error produced by the classifier on the training set (feed with a set of examples according to the current iteration weights) is equal to 0? Briefly explain why you think so.

   a. Stop the boosting iterations and return the weighted ensemble of classifiers built until that moment.
   b. Return that last classifier as the final classifier.
   c. Remove that classifier and continue the boosting loop until the limit number of iterations is achieved.
   d. Reduce the confidence on that classifier (with respect to its theoretical confidence) and continue the boosting loop until the limit number of iterations is achieved.
   e. Boosting cannot be applied in that case.

   When a classifier is equal to 0 as a training error, this is going to happen in the first round of boosting. Therefore, what we need to do is stop the boosting and take the last classifier that, until that moment, will be the one that did perfect classification.

4. After building a support vector machine with a linear kernel with a given C, we found the number of support vectors is very large. If we want to decrease the number of support vector, what should we do? Explain why.
    a. Decrease the C value
    b. Increase the C value
    c. Change to the RBF kernel
    d. Try a Polynomic kernel
    e. None of the above

    When increasing the value and range of the *C* parameter, apart that the execution time of the algorithm increases by a lot as it has a cubic value in respect to the number of rows, it increases the bar of the margin that it is set to the patterns that violate the margin constraint. So, that means that when we increase *C*, the number of support vectors decreases because they are less likely to fit it since the margin of violation is narrower.

5. We have a linear SVM trained on a dataset of 100,000 observations. The SVM shows 60,000 supports. Comment each of the following statements:
    *sections b, c, d and e in this exercise can be justified with the explanation in exercise 4.

    a. SVM is Ok. We probably will have a low error on the small test set

        No, the SVM is not OK, the proportion of supports is not acceptable because is above 0.5, so it will not perform well.

    b. We need to increase the number of supports to reduce the error in the test set. Therefore, we should increase C.

        No, by increasing the *C* parameter we reduce the number of supports which results in a better accuracy, meaning less error, if it is used in the same samples that has been trained. Although it is not the only parameter that helps reducing the error.

c. We have to reduce the number of supports to reduce the error in the test set. Therefore, we should increase the C parameter.

   True, if we want to reduce the number of supports, we should reduce the *C* parameter.

d. We need to increase the number of supports to reduce the error in the test set. Therefore, we should decrease the C parameter.

   Correct, If we want to increase the number of supports and therefore reduce the error in the test set, we should decrease the *C* parameter.

e. We have to reduce the number of supports to reduce the error in the test set. Therefore, we should decrease the C parameter.

   Incorrect, when decreasing the *C* parameter, the number of supports increases. It can happen that the error is reduced, but the number of supports is not the only parameter that takes part in the accuracy value.

6. Briefly explain if each of the following claims is true or not and why:
   a. In the *apriori* algorithm, given a rule, we will say that it is a good rule if its support and its confidence are above the required thresholds

      True, a good association rule mining like *apriori* algorithms have the goal to find all the rules having its support greater or equal than the minsup threshold and the confidence greater or equal than the minconf threshold (Being minsup and minconf the required thresholds for each parameter).

b. The support required for rules should be always independent of the elements that belong to the itemset.

No, that situation would be counter-productive, if the itemset is not related the value is arbitrary so there is no way to know if the thresholds are too restrictive or too permissive.

c. While finding frequent itemsets, in the main iteration of the algorithm, the itemsets below minimum support of iteration "*i*" should be kept to do pruning in iteration "*i+1*"

True, while an itemset is frequent in one iteration, they should keep doing pruning in the next one. The apriori pruning principle says that if there's any itemset which is infrequent, its superset should not be generated/tested. In this case, that would be the other way around.

d. The *apriori* algorithm can learn causal rules that explain the behavior of customers.

Yes, as it is an association rule it can explain the implications of different itemsets. Those itemsets can be the type of food a customer buys, so by doing implications between those itemsets the algorithm can learn causal rules that tell us about the behavior of customers.