JOZEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Data Mining and Knowledge Discovery

Case Study on the use of Data Minig Techniques in Food Science using Honey Samples

Matej Gašperin
matej.gasperin@ijs.si

Professor: Prof. Nada Lavrač

Ljubljana, February 2007

Abstract

The objective of this work was to preform a case study of the use of advanced data mining techniques in the field of food science, on the case of honey samples. The attributes with the higher classification value have been identified and three different techniques have been tested to classify a honey sample based on 7 numeric attributes. The results show, that the accuracy of the classification given our dataset is between 70 and 80%. We also suggest improvements so the accuracy can be increased.

1 Introduction

In recent years, data mining techniques have been widely applied in food science, a number of examples were reported in the literature concerning a variety of products. The objective of this work is to preform a case study of some of these techniqes on the dataset describing different sorts of honey.

Honey is the natural sweet substance produced by *Apis mellifera* bees from the nectar of plants or excretions of plant-suckinginsects on the living parts of plants. honey contains many different substances [1], mainly sugarssuch as fructose and glucose. Water in general can be present in amount lower than 20%. In honey there are also present various organic and inorganic acids, proteins, amino acids, enzymes, vitamines and hormones.

Untill now, the research has minly been focused on classifying the honey sample based on the minerals and metals present in the sample. We evaluated the use of other sensoric data for the classification.

In the second chapter, the dataset used in this study will be presented and analized. Third chapter describes the data mining techniques used in this work and in the fourth chapter the results will be presented. The final chapter contains the conclusion and ideas for future work.

2 Data

The dataset used in this analysis has been provided by the Department of Food Science and Technology from Biotechnical Faculty, University of Ljubljana. The laboratory has been involved in the honey analysis for the past few years and has assembled a large dataset of honey samples in that time. The presented dataset represents a part of this data collected from honey samples dating from 2000 and 2001.

Sampling and classification has been done by beekeepers in Slovenia and further analysis of the samples has been done by researches at the

department. For the purpose of our study, it can be assumed that all the values and classifications are correct.

2.1 Data description

The dataset consist of 527 instances which represent different honey samples. Each instance has 9 numeric attributes and a nominal class value. The class values represent different types of honey. In our dataset the types included are acacia, floral, lime, chestnut, forest, spruce, fir and honeydew honey from *M. pruinosa*. The attributes corespond to different ingredients in the honey and its physical properties, namely diastase number, prolin, water, electrical conductivity,free acids, lactones, total acids, ash and pH value. The diastase numer and prolin attributes have 12% and 14% missing values respectively. The missing values are equaly distributedamong all class vlaues and therefore they are not expected to cause major irregularities.

The distribution of the instances over class values is shown in Fig. (1) and the distribution of all attributes in Fig. (2).

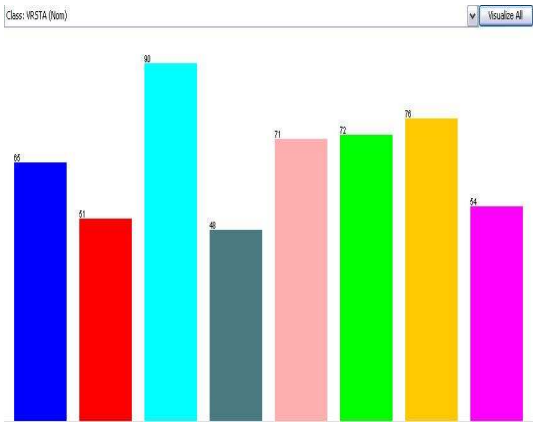

Fig. 1: Distribution of instances over the class variable

For better understanding of the relations between attributes, the data can also be visualized in Weka, using the visualization tab. The result is shown in Fig. (3)

It can bee seen, that some attributes exibit strong corelations, namely for example free acids and total acids or electrical conductivitiy and ash.

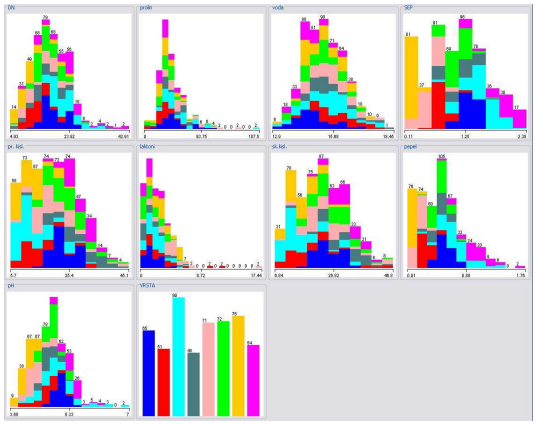

Fig. 2: All attributes visualization

The fact that the electrical conductivity is proportional to the amount of ash in the sample can be explained by the fact that particles of ash conduct electricity. Also the corelation between total amount of acids and the amount of free acids was expected. These corelations may prove useful when building classifiers because they suggest that the classifier may be build using less attributes.

2.2 Data understanding

The classification problem in our case is to be able to classify a new instance (i.e. honey sample) based on the known attribute values (i.e. analysis results). Until now, most reasearches focused on the classification based on the contents of the minerals and metals in the sample. In our work, we evaluate different approaches of classification based on other sensoric data to construct different classifications or to complement the existing ones.

Our dataset consists of 9 numeric attributes, from which we are trying to predict a nominal class. The drawback of our data is, that we do not have any information about the methodology used to determine the values. It has been sugested [1], that in further research sampling should be done systematically and by qualified personnel rather than beekeepers.

The first attribute represents the diastase number. Diastase is an enzyme, belonging to amylase which at high temperatures decomposes sac-

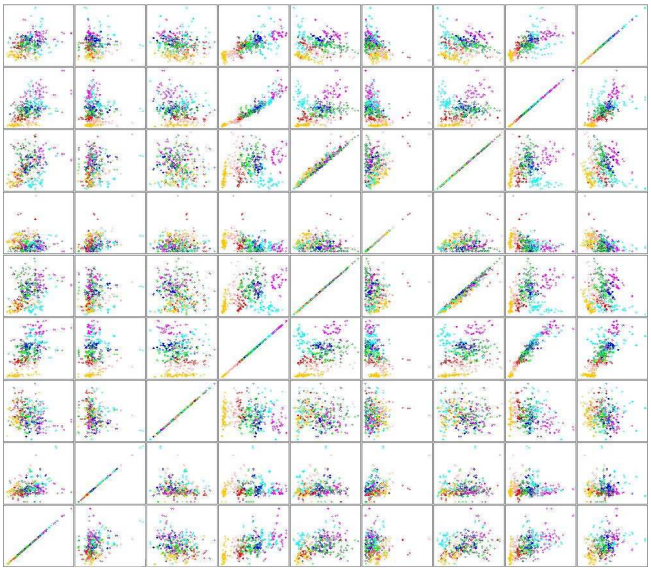

Fig. 3: Visualization of attribute relations

charosis to glukosis and fructosis. The activity of the enzyme is expressed with diastase number. It represents the volume of the 1% starch solution, which the enzyme hydrolises from 1g of honey in one hour at temperature of 40^0C. Second attribute is the amount of proline in the sample measured in $mg/100g$. Next attribute value is amount of water in the sample, given in [%]. The attribute SEP (Specifična električna prevodnost) is a slovene abreviation for electrical conductivity in mS/cm. Next two attributes are connected with the amount of acids in the sample, "pr.kisl" denoting free acids ans "sk.kisl." the total acids, both measured in mekv/kg. Attribute lactones describes the ammount of lactones in the sample and is also measured in mekv/kg. The finaly two attributes represent the quantaty of ash [%] and the pH value of the honey sample.

The complete dataset includes many aditional attributes, such as colour, minerals and metals in sample etc., but at this moment it was not possible to aquire them. Eventhough the complete dataset is not provided, this dataset is ocnsidered rich enough for the need of this case study.

3 Machine learning techniques

In this study, a few different techniques have been considered, based on different approaches, Bayes based classifier, decision tree and decision rules. All three classifiers will be described in this chapter.

3.1 Decision tree: J48

Decision tree learning is one of the most widely use and practical methods for classification. We used the algroithm developed by Quinlan, 1993, which is implemented in weka under the name J48.

The learning algorithm decides which attribute to put in the node of the tree based on its *information gain*. The idea of this approach is, that if we always select the attribute that carries the most information, this will minimize the size of the tree. In information theory, the amount of information is measured using entropy. The concept of entropy in information theory describes how much information there is in an event. The idea of entropy was introoduced by Shannon in 1948. An intuitive understandingof information entropy relates to the amount of uncertanty about an event associated with a given probability distribution. Shannon defined the entropy in terms of discrete random variable X, with possible states $x_1, x_2, ..., x_n$ as:

$$H(X) = \sum_{i=1}^{n} p(x_i) log_2(\frac{1}{p(x_i)}) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i). \quad (1)$$

That is, the entropy of the variable X is the sum, over all possible outcomes x_i of X of the product of the probability of outcome x_i times the log of the inverse of the probability of x_i.

Using the entropy as a measure for information gain we can describe the approach as:

- Chose an attribute that has the highest entropy value.
- Create a separate tree branch for each value of the chosen attribute.
- Divide the instances into subgroups so as to reflect the attribute values of the chosen node.
- For each group, temrinate the attribute selection process if all members of a subgroup belong to the same class or a subgroup contains a single node.

- For each subgroup that has not been labeled as terminal repeat the above process.

The main issue with the decision tres is, that the algorithm is making new branches deep enough to perfectly fit all the training data. This results in the decision tree that perfectly fits the trianing data but decreases its preformance over the test examples. To avoid this overfiting, a post-pruning approach will be used. With this approach, we let the algorithm to grow a full tree and than remove the branches that will cause the overfitting.

3.2 Naive bayes classifier

Naive bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong independance assumptions. A more descriptive term for the underlying probability model would be independant feature model.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers often work much better in many complex real-world situations than might be expected. Recently, careful analysis of the Bayesian classification problem has shown that there are sound theoretical reasons for the apparently unreasonable efficacy of naive Bayes classifiers.

The probabilistic model for a classifier is a conditional model over a dependent class variable v with a small number of outcomes or classes (2), conditional on several attributes a_1 through F_n.

$$P(v|a_1, a_2, ..., a_n) \tag{2}$$

The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more treatable. Using Bayes' theorem, we write:

$$P(v|a_1, a_2, ..., a_n) = \frac{P(v)P(|a_1, a_2, ..., a_n|v)}{P(a_1, a_2, ..., a_n|v)} \tag{3}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on v and the values of the features a_i are given, so that the denominator is effectively constant. The relative frequencies of the class in the training data are easy to calculate and present no probelm.

Using the definition of conditional probability the numerator can be written in the following form:

$$P(a_1, a_2, ..., a_n|v) = P(v)P(a_1|v)P(a_2|v, a_1)P(a_3, ...a_n|v, a_1, a_2) \quad (4)$$

Eq. (4) is not feasible to estimate unless we have a very large set of training data. The number of terms is equal to the number of possible instances times number of possible target values. Therefore we need to see every instance many times in order to obtain reliable estimates.

The naive bayes classifier is based on the simplifying assumption that the attribute values are conditionaly independant given the target value. Using that in Eq. (4) we obtain:

$$P(a_1, a_2, ..., a_n|v) = \sum_i P(a_i|v) \quad (5)$$

Now the model can be written as:

$$P(v|a_1, a_2, ..., a_n) = \frac{1}{Z}P(v)\sum_i P(a_i|v), \quad (6)$$

where Z is the scaling factor and is efectivles a constant. In this way the number of distinct conditional probabilites that must be estimated is just the number of distinct attribute values times the number of distinct target values.

Classifier now combines the model with a decision rule. Bayesian approach to classifying the new instance is to assign the most probable target value. The v_{nb} denotes the target value output by the classifier.

$$v_{nb} = arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j) \quad (7)$$

We see, that estimating $P(a_i|v_j)$ instead of $P(a_1, a_2, ..., a_n|v_j)$ greatly decreases the number of parameters and although the assumptions made here are often inaccurate it it still very usefull in practice. One of the explanations may be, that bias in estimating the probabilites often may not make a difference because it is the order of the probabilites, not their exact values that determine the classifications.

3.3 Decision rules:NNge

The third classifier used is Non-Nested Generalised Exemplars (NNGE), which is an algorithm introduced by Brent, 1995. It preforms generalisation by merging exemplars, forming hyperrectangles in attribute space that represent conjunctive rules with internal disjunction. The algorithm forms a generalisation each time a new example is added to the database, by joining it to its nearest neighbour of the same class.

The algorithm learns incrementally by first classifying, then generalising each new example. When classifying an instance, one or more hyperrectangles may be found that the new instance is a member of, but which are of wrong class. The algorithm prunes these so that the new example is no longer a member.Once classified, the new instance is generalised by merging it with the nearest exemplar of the same class, which may be a single instance or a hyprerectangle.

It has been shown [4], that generalising exemplars results in improved classification preformance over standard nearest neighbour. The only thing that may pose a problem is, that the algorithm tends to produce rules that test a large number of attributes. Because of this they are not very intelligible to people.

4 Evaluation

All algorithms were tested using Weka softvare, version 3.4.10. First they were all tested on full dataset and later using only 7 attributes as suggested by the Fig. (3).

4.1 Evaluation of the J48 decision tree

The J48 algorithm was first tested using the default parameter values, meaning we diddnt prune the tree. The results of 10-fold cross validations are shown in Fig. (4):

The algorithm achieved 78.75% accuaracy, but the size of the tree is 101, which is well more than what can be visualized. To make the decision tree practicaly usefull the tree will have to be pruned.

For the next run, we set the algorithm parameters to limit the minimum number of instances in one leaf to 15 and decrease confidence factor to 0.1. The weka output of 10-fold cross validation is shown in Fig. (5).

We see that the percentage of the accuratley classified instances has decreased to 70.78%, but the tree is now of managable size (23). The visu-

```
Correctly Classified Instances         415               78.7476 %
Incorrectly Classified Instances       112               21.2524 %
Kappa statistic                          0.7558
Mean absolute error                      0.0568
Root mean squared error                  0.2112
Relative absolute error                 26.1297 %
Root relative squared error             64.0593 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.723     0.03       0.77      0.723     0.746    H
  0.686     0.034      0.686     0.686     0.686    L
  0.9       0.025      0.88      0.9       0.89     K
  0.667     0.046      0.593     0.667     0.627    S
  0.718     0.031      0.785     0.718     0.75     C
  0.681     0.048      0.69      0.681     0.685    G
  0.947     0.013      0.923     0.947     0.935    A
  0.889     0.015      0.873     0.889     0.881    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 47  0  3  4  0  9  0  2 |  a = H
  0 35  3  4  7  2  0  0 |  b = L
  3  0 81  1  2  1  0  2 |  c = K
  3  2  2 32  0  8  0  1 |  d = S
  0 11  0  1 51  2  6  0 |  e = C
  5  3  3  9  1 49  0  2 |  f = G
  0  0  0  0  4  0 72  0 |  g = A
  3  0  0  3  0  0  0 48 |  h = SS
```

Fig. 4: Results of 10-fold cross validation using J48 algorithm

alization of the tree is shown in Fig. (6).

Following the idea from the chapter 2, we decide to remove some attribute values. we use the genetic search algorithm from weka togeather with **CfsSubsetEval** evaluator, which evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. the result of this attribute selection algorithm are seven attributes that best describe the whole dataset. The two attributes removed are lactones and free acids, as is suggested from Fig. (3). The decision tree learning algorithm with 10-fold cross validation has been used on the new dataset. The results are shown in Fig. (7).

It can be seen, that reducing of the attribute number decreased the accuracy a bit more to 69.45% and the tree size remained about the same as in prevoius experiment.

```
Correctly Classified Instances         373               70.778  %
Incorrectly Classified Instances       154               29.222  %
Kappa statistic                          0.6642
Mean absolute error                      0.098
Root mean squared error                  0.2323
Relative absolute error                 45.07   %
Root relative squared error             70.4434 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.662     0.039      0.705     0.662     0.683    H
  0.647     0.046      0.6       0.647     0.623    L
  0.822     0.046      0.787     0.822     0.804    K
  0.521     0.069      0.431     0.521     0.472    S
  0.648     0.024      0.807     0.648     0.719    C
  0.444     0.07       0.5       0.444     0.471    G
  1         0.027      0.864     1         0.927    A
  0.815     0.013      0.88      0.815     0.846    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 43  0  9  5  0  6  0  2 |  a = H
  0 33  3  5  8  2  0  0 |  b = L
  0  2 74  1  0 11  0  2 |  c = K
  4  4  1 25  2 11  0  1 |  d = S
  0 10  0  1 46  2 12  0 |  e = C
  8  6  5 19  1 32  0  1 |  f = G
  0  0  0  0  0  0 76  0 |  g = A
  6  0  2  2  0  0  0 44 |  h = SS
```

Fig. 5: Results of 10-fold cross validation using J48 algorithm and pruning

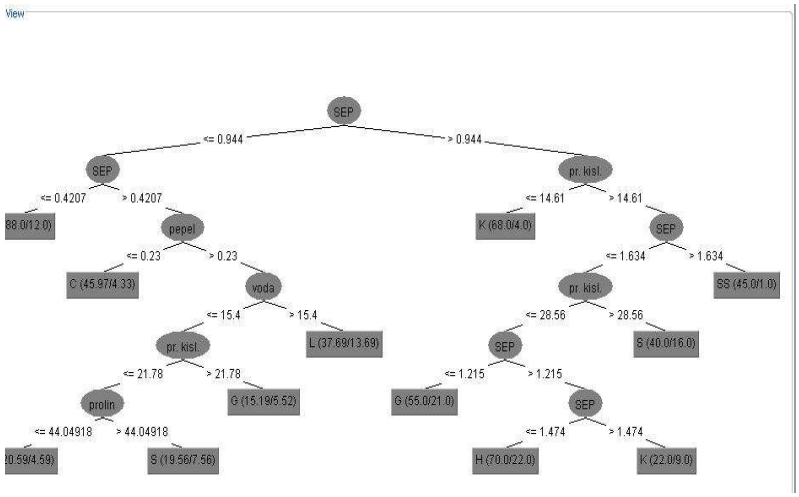

Fig. 6: Pruned decision tree obtained with J48 algorithm

```
Correctly Classified Instances         366               69.4497 %
Incorrectly Classified Instances       161               30.5503 %
Kappa statistic                          0.648
Mean absolute error                      0.0988
Root mean squared error                  0.2328
Relative absolute error                 45.4257 %
Root relative squared error             70.5933 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.692     0.026      0.789     0.692     0.738    H
  0.608     0.04       0.62      0.608     0.614    L
  0.8       0.071      0.699     0.8       0.746    K
  0.354     0.046      0.436     0.354     0.391    S
  0.676     0.031      0.774     0.676     0.722    C
  0.431     0.088      0.437     0.431     0.434    G
  1         0.027      0.864     1         0.927    A
  0.852     0.023      0.807     0.852     0.829    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 45  0  8  1  0  8  0  3 |  a = H
  0 31  3  4 10  3  0  0 |  b = L
  0  2 72  2  0 12  0  2 |  c = K
  3  4  4 17  3 15  0  2 |  d = S
  0  8  0  1 48  2 12  0 |  e = C
  5  5 12 14  1 31  0  4 |  f = G
  0  0  0  0  0  0 76  0 |  g = A
  4  0  4  0  0  0  0 46 |  h = SS
```

Fig. 7: Results of 10-fold cross validation using J48 algorithm and pruning on dataset using 7 attributes

4.2 Evaluation of naive Bayes classifier

The next algorithm that has been tested is the naive Bayes classifier. Its preformance depends on the conditional independance of the attribute values, therefore high accuracies on this datased are not expected. The results of the classifier using 10-fold cross validation are shown in Fig. (8).

```
Correctly Classified Instances         383               72.6755 %
Incorrectly Classified Instances       144               27.3245 %
Kappa statistic                          0.6861
Mean absolute error                      0.0769
Root mean squared error                  0.2287
Relative absolute error                 35.3429 %
Root relative squared error             69.3422 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.815     0.058      0.663     0.815     0.731    H
  0.765     0.044      0.65      0.765     0.703    L
  0.822     0.021      0.892     0.822     0.855    K
  0.396     0.05       0.442     0.396     0.418    S
  0.718     0.046      0.708     0.718     0.713    C
  0.486     0.073      0.515     0.486     0.5      G
  0.921     0.013      0.921     0.921     0.921    A
  0.778     0.006      0.933     0.778     0.848    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 53  1  1  0  0 10  0  0 |  a = H
  0 39  1  0  8  3  0  0 |  b = L
  5  0 74  6  2  3  0  0 |  c = K
  4  5  0 19  3 15  0  2 |  d = S
  0  8  2  2 51  2  6  0 |  e = C
 10  7  3 14  2 35  0  1 |  f = G
  0  0  0  0  6  0 70  0 |  g = A
  8  0  2  2  0  0  0 42 |  h = SS
```

Fig. 8: Results of 10-fold cross validation using naive Bayes classifier

The accuracy of the classifier is in fact lower than the one acchieved using decision tree without pruning, but higher than the one acchieved after pruning. For the next experiment, again the same two attributes have been removed from the dataset and the algorithm has been applied again.

In this case, the classifier has acchieved higher accuracy than when using the full dataset. This suggest that the classifier preforms better on the datasets where attributes are conditionaly independant.

```
Correctly Classified Instances         384               72.8653 %
Incorrectly Classified Instances       143               27.1347 %
Kappa statistic                          0.6883
Mean absolute error                      0.0801
Root mean squared error                  0.2243
Relative absolute error                 36.842  %
Root relative squared error             68.0161 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.831     0.065      0.643     0.831     0.725    H
  0.784     0.048      0.635     0.784     0.702    L
  0.789     0.018      0.899     0.789     0.84     K
  0.354     0.046      0.436     0.354     0.391    S
  0.746     0.039      0.746     0.746     0.746    C
  0.514     0.073      0.529     0.514     0.521    G
  0.921     0.013      0.921     0.921     0.921    A
  0.778     0.006      0.933     0.778     0.848    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 54  1  1  1  0  8  0  0 |  a = H
  0 40  0  0  7  4  0  0 |  b = L
  6  0 71  7  2  4  0  0 |  c = K
  4  4  1 17  2 17  0  3 |  d = S
  0  8  2  2 53  0  6  0 |  e = C
 12 10  2 10  1 37  0  0 |  f = G
  0  0  0  0  6  0 70  0 |  g = A
  8  0  2  2  0  0  0 42 |  h = SS
```

Fig. 9: Results of 10-fold cross validation using naive Bayes classifier using 7 attribute values

4.3 Evaluation of Non-nested generalized exemplars algorithm

The last algorthm to be evaluated is the NNge algortihm, implemented in weka. The highest accuracy has been obtained using the default parameter values. The results of the decision rules classification are shown in Fig. (10).

```
Correctly Classified Instances         425               80.6452 %
Incorrectly Classified Instances       102               19.3548 %
Kappa statistic                          0.7775
Mean absolute error                      0.0484
Root mean squared error                  0.22
Relative absolute error                 22.2474 %
Root relative squared error             66.7057 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.815     0.035      0.768     0.815     0.791    H
  0.804     0.032      0.732     0.804     0.766    L
  0.911     0.027      0.872     0.911     0.891    K
  0.604     0.038      0.617     0.604     0.611    S
  0.775     0.018      0.873     0.775     0.821    C
  0.583     0.053      0.636     0.583     0.609    G
  0.947     0.011      0.935     0.947     0.941    A
  0.944     0.008      0.927     0.944     0.936    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 53  0  4  0  0  7  0  1 |  a = H
  0 41  2  2  3  3  0  0 |  b = L
  2  3 82  0  0  2  0  1 |  c = K
  3  2  1 29  1 10  0  2 |  d = S
  0  8  0  2 55  1  5  0 |  e = C
 10  2  5 13  0 42  0  0 |  f = G
  0  0  0  0  4  0 72  0 |  g = A
  1  0  0  1  0  1  0 51 |  h = SS
```

Fig. 10: Results of 10-fold cross validation using NNge algorithm

The resulting model accuracy achieved using full dataset has been 80.65%, which is significantly higher than the accuracy of the previous models. Even when using only 7 attribute values, the accuracy still remains 80.45%, which is effectively the same (Fig. 11)

```
Correctly Classified Instances         424               80.4554 %
Incorrectly Classified Instances       103               19.5446 %
Kappa statistic                          0.7754
Mean absolute error                      0.0489
Root mean squared error                  0.221
Relative absolute error                 22.4655 %
Root relative squared error             67.0319 %
Total Number of Instances              527

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.738     0.035      0.75      0.738     0.744    H
  0.804     0.034      0.719     0.804     0.759    L
  0.867     0.039      0.821     0.867     0.843    K
  0.625     0.046      0.577     0.625     0.6      S
  0.775     0.015      0.887     0.775     0.827    C
  0.681     0.035      0.754     0.681     0.715    G
  0.974     0.011      0.937     0.974     0.955    A
  0.907     0.008      0.925     0.907     0.916    SS

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h   <-- classified as
 48  0  9  2  0  5  0  1 |  a = H
  0 41  2  1  3  4  0  0 |  b = L
  4  4 78  1  0  1  0  2 |  c = K
  5  2  5 30  2  4  0  0 |  d = S
  0  9  0  2 55  0  5  0 |  e = C
  6  1  1 14  0 49  0  1 |  f = G
  0  0  0  0  2  0 74  0 |  g = A
  1  0  0  2  0  2  0 49 |  h = SS
```

Fig. 11: Results of 10-fold cross validation using NNge algorithm and 7 attributes

Algorithm	Full dataset	Reduced dataset
J48	70.78%	69.46%
Bayes	72.68%	72.86%
NNge	80.64%	80.46%

Tab. 1: Classifier results comparison

4.4 Results comparison

From the summary of the algorithms accuracy in the Table 1, it can be seen, that eventhough the decision tree classifier is the most conveniant to visualize, it offers lower accuracies than other two algorithms considered. Of course this is only true in case of pruned decision tree, but also when using the full tree, the accuracy is still lower than the one acchieved by NNge classifier. The thing that may pose a problem with NNge classifier is that the decision rules test large numbers of attributes and is not very intelligible to people. Naive Bayes classifier on one hand offers a simpler model, which is also extremely fast to learn, but for the cost of its preformance.

5 Conclusion

In this work we have presented a different approach to classification of food samples on the example of different sorts of honey. We have shown how the number of attributes used for classification can be systematicaly reduced. Three diferent classifiers have been presented and compared in preformance. For a more detailed interpretation of the results some consultations with the researches from food science will be needed.

Better results could be acchieved by expanding the dataset with the attributes describing the minerals and metals in the sample, but as mentioned before we were unable to obtain that data for now. Some authors like Latorre *et. al.*, 1998 suggest, that in the case of honey classification accuracies of 90% and more can be acchieved using only three attributes (Li,Cu and Mn).

We also do not have any information of how the year of production affects these results. Some authors [1] suggest that the year of production does not have a significant influence on elemental data, although it is possible there is influence on other attributes. The samples used here have been collected over the period of 2 years, but we do not have the information which sample belonges to particular year.

References

[1] Golob T, Dobersek U, Kump P, Nečemer P "Determination of trace and minor elements in Slovenian honey by total reflection X-ray flouorescence spectroscopy," *Food Chemistry, Elsevier Ltd., 2006*

[2] Latorre M J, Pena R, Pita C, Botana A, Garcia S, Herrero C "Chemometric classification of honey according to their type. II. Metal content data," *Food Chemistry, Elsevier Ltd., 1999*

[3] Mitchell T M, "Machine learning," *1997*

[4] Brent M, "Instance Based Learning: Nearest Neighbour with Generalisation," *University of Waikato, Hamilton, 1995*