USING CLASSIFICATION ON UPISI DATABASE

Kiril Kiroski
Institute of Informatics, Faculty of Natural Sciences and Mathematics
Skopje, Macedonia

Magdalena Kostovska
Institute of Informatics, Faculty of Natural Sciences and Mathematics
Skopje, Macedonia

ABSTRACT

Data mining plays a big role when analyzing data gathered over a period of time. It helps make sense out of a big collection of records, called dataset. Classification is one of the most common techniques of data mining, which occurs very frequently in everyday life. Classification involves diving up objects so that each of these objects will fall into one of mutually exhaustive and exclusive categories we call classes. In this paper, it will be discussed some of the most frequently used classification methods. These methods will be tested on a chosen dataset to serve as an example of which of these methods is most suitable for such dataset form. The dataset is extracted from Application “Upisi”, and treated with three classification approaches: Naive Bayes, Nearest Neighbor and Decision Trees.

I. INTRODUCTION

Humans have been "manually" extracting patterns from data for centuries, but the increasing volume of data in modern times has called for more automated approaches. Early methods of identifying patterns in data include Bayes' theorem (1700s) and Regression analysis (1800s). The proliferation, ubiquity and increasing power of computer technology has increased data collection and storage. As data sets have grown in size and complexity, direct hands-on data analysis has increasingly been augmented with indirect, automatic data processing. This has been aided by other discoveries in computer science, such as neural networks, clustering, genetic algorithms (1950s), decision trees(1960s) and support vector machines (1980s). Data mining is the process of applying these methods to data with the intention of uncovering hidden patterns.[1] According to the Gartner Group [2], “Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques.” There are other definitions:

- “Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner” (Hand et al. [3]).
- “Data mining is an interdisciplinary field bringing togther techniques from machine learning, pattern recognition, statistics, databases, and visualization to address the issue of information extraction from large data bases” (Evangelos Simoudis in Cabena et al. [4])

In data mining, data is analyzed from different perspectives and summarized into useful information - information that can be used to increase revenue, cuts costs, or both. We sometimes refer to Data mining as data discovery or knowledge discovery. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Data mining today is not only used in science circles, but everywhere we have big collections of data we can use to get some insight into their meaning. For example, almost every bigger sport collective uses data mining to discover where their strengths and weaknesses are, they use it to discover more information about their opponents, and find a way to play according to this newly acquired knowledge. It has been used for many years by businesses, scientists and governments to sift through volumes of data such as airline passenger trip records, census data and supermarket scanner data to produce market research reports.[5]

II. DATA PREPARATION

When beginning work on a data mining problem, it is first necessary to bring all the data together into a set of instances. This set is also referred as dataset, and it represents form of data suitable for data mining. With the correct dataset we then choose data mining method we will be using, and the starting goals for our data mining process. What we should be beware of, regarding data mining, is the fact that it is easy to get “preferred” results if you don’t apply data mining process in correct order, or on not well prepared data. Today there are a lot of very powerful software solutions that can help in the data mining process, but they can’t apply in every field and on every problem “out-of-the-box.” Therefore, data mining is not a process that should be taken as granted, or as a “black box” approach. Instead, it is always better to apply “white box” methodology, which emphasizes an understanding of the algorithmic and statistical model structures underlying the software. There are more models for data mining, which can be used as blueprints in the organization the process of gathering and analyzing data, disseminating and implementing results and monitoring improvements. [6] Some of the models are:

- CRISP (Cross-Industry Standard Process for data mining), which was proposed in the mid-1990s by a European consortium of companies to serve as a non-proprietary standard process model for data mining.
- The Six Sigma methodology - is a well-structured, data-driven methodology for eliminating defects, waste, or quality control problems of all kinds in manufacturing, service delivery, management, and other business activities. This model has recently become very popular (due to its successful implementations) in various American industries, and it appears to gain favor worldwide. It postulated a sequence of, so-called, DMAIC steps.
- SEMMA methodology is similar to Six Sigma, but it is focusing more on the technical activities involved in a data

mining project. The acronym SEMMA – sample, explore, modify, model, assess – refers to the core process of conducting data mining. Beginning with a statistically representative sample of data, SEMMA applies exploratory statistical and visualisation techniques, select and transform the most significant predictive variables, model the variables to predict outcomes, and confirm a model's accuracy.

All of these models are concerned with the process of how to integrate data mining methodology into an organization, how to "convert data into information," how to involve important stake-holders, and how to disseminate the information in a form that can easily be converted by stake-holders into resources for strategic decision making. Some software tools for data mining are specifically designed and documented to fit into one of these specific frameworks.

In the dataset we use, we will first preprocess our data to remove incomplete entries, obsolete or redundant fields and outliers, if there are any. Since the data we are using are intended to be used by a specific application, we will first have to transform data into more suitable form for our analysis, consistent with our policy. This process of data preparation is known as *data cleaning* and *data transformation*. Preparing input for a data mining investigation usually consumes the bulk of the effort invested in the entire data mining process. According to [6], "data preparation alone accounts for 60% of all the time and effort expanded in the entire data mining process."

III. Classification

Classification is a task that occurs very frequently in everyday life. Essentially it involves dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as classes. The term 'mutually exhaustive and exclusive' simply means that each object must be assigned to precisely one class, i.e. never to more than one and never to no class at all. [7]

Classification is learning a function that maps (classifies) a data item into one of several predefined classes. This data item will be the target for classification process, and it will contain data interesting for the researchers. The data mining model examines a large set of records, each record containing information on the target variable as well as a set of input or predictor variables. Researcher should be able to classify data records which are still not in the dataset, on base of the characteristics associated with the predictor variables.

Classification methods require thorough data preparation, so as to be able to give more precise prediction. Then, the data set containing predictor variables and target variable is examined, which is the way the algorithm (software) "learns" the ground rules how predictor variables are associated with the target classes. The data set on which learning is performed, is called *training set.* When these rules are set, algorithm is ready to proceed with assessing the accuracy of the classification procedure. This assessment is performed through using established classification on the *test data*. Accuracy of the classification is given with the following equation:

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}} \tag{1}$$

For the purposes of this analysis, we will use following three classification methods:

Naive Bayes
Decision Tree
k-nearest neighbor

A. Naïve Bayes

Naive Bayes is a classification method which uses no rules, decision tree or some other explicit classifier representation. Instead, Naive Bayes uses *probability theory* to find the classification with greatest likelihood. It is based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a vehicle can be considered as a "car", if it has at the most 4 tires, it is not higher than 2 meters, and can accommodate from 4 to 8 passengers.

The probability of this "class" can be a value between 0 (impossible) and 1 (certain), and it is improved with a longer series of trials. Since we are not interested in only one class, but in a set of alternative classes, they must be *mutually exclusive* and *exhaustive*, so one and only one will always occur. The probability model for the classifier is the following conditional model:

$$p(C|F1,\dots,Fn) \tag{2}$$

for the dependent class variable C with small number of outcomes (classes), conditional on variables F1 through Fn. Using Bayes' theorem, it can be written as:

$$p(C|F1,\dots,Fn) = \frac{p(C)p(F1,\dots,Fn|C)}{p(F1,\dots,Fn)} \tag{3}$$

This equation can be written also as:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \tag{4}$$

B. Decision Tree

One of the most popular classification methods is the construction of a *decision tree*, which contains *decision nodes*, connected by *branches*. These branches are extending from the *root node*, and they terminate with *leaf nodes*. The root node, by convention, is placed at the top of the decision tree diagram, and it describes the attribute which is best suited for initial split according to the decision tree algorithm. Attributes at the decision nodes are tested, and each possible outcome is represented by a branch. Each branch in turn leads either to another decision node, or to a terminating leaf node.

C. K-nearest Neighbor

Nearest neighbor classification is most often used for classification, although it can be used for other data mining approaches, such as estimation and prediction. It is mainly used when all attribute values are continuous, but it can be modified to deal with categorical attributes. Nearest neighbor estimates the classification of an unseen instance using classification of the instances that are closest to it, in a space that need to be defined. In this method, the attributes are called *dimensions*, and they can be diagrammatically shown if they are small in number. Distance between two instances X and Y can be annotated as dist(X, Y), and three rules apply:

The distance of any point A from itself is zero, i.e. dist(A,A) = 0.

The distance from A to B is the same as the distance from B to A, i.e. dist(A,B) = dist(B,A) (the symmetry condition).

The triangle inequality - the condition says that for any points A, B and Z: dist(A,B) ≤ dist(A,Z) + dist(Z,B).

The distances are usually measured using Pythagora's theorem. If there are two instances in an n-dimensional space, distance between these two instances will be determined using the formula:

$$\sqrt[2]{(a_1 - b_1)^2 + (a_2 - b_1)^2 + \cdots + (a_n - b_n)^2} \qquad (5)$$

A major problem when using the Euclidean distance formula is that the large values frequently swamp the small ones. To overcome this problem we generally *normalise* the values of continuous attributes. The idea is to make the values of each attribute run from 0 to 1. In general if the lowest value of attribute *A* is *min* and the highest value is *max*, we convert each value of *A*, say *a*, to (*a* − *min*)/(*max* − *min*).

IV. EXPERIMENT

Data we need for these classification tasks are found in the ISII database used for application "Upisi". To be able to use this data we first needed to extract fields we need. First step was creating the view "v_skoro_site_podatoci_za_student" which includes 29 fields. Using of this view, we created our initial data set which includes eighteen attributes: dossier number for students (used for grouping students and other attributes, program code (label for study program student is attending), information if student is part time or full time, and three fields (code for the course, highest grade and number of times student is attending to the course) for each of the five subjects we were interested in (represented by their code – OP, OVP, AK, SPA, BP). The reason we choose this courses is their direct relationship (OP->AK; OP->OVP->SPA->BP). After initial discussion we decided to use them to try to predict number of times the student will need to attend to the BP course to be able to pass it.

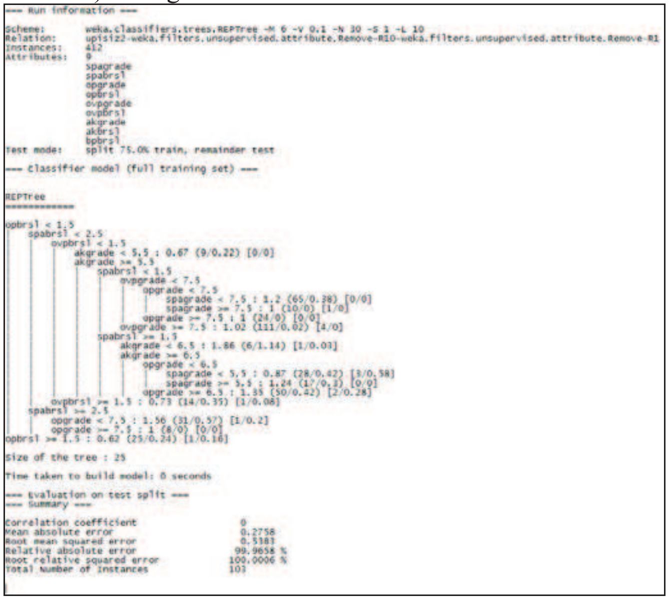
After that we proceeded with data cleaning, by removing entries which doesn't hold information for every link in this chain of dependences. Also, we removed entries indicating that course was attended for the first time, but not passed (because it means that the student has not yet taken any exam in BP course). With preprocessing finished, we got 412 instances for the data set.

A. *Using continuous values for data mining*

For the first part of our data mining process, we started without categorization of our fields, but we used continuous values, as are in the database. Since no Bayes' method works with such values, we experimented only using Nearest Neighbor and Decision Trees. Application we use for these experiments is Weka 3.6, which can be found at [8].

1) *Decision Tree*

After experimenting with a number of decision trees, namely, DecisionStump, MP5, REPTree and UserClassifier, the best results we achieved are with using 75% of instances as training set, which was then tried on the rest for evaluation, as shown on Figure 1. We used 8 attributes as predictors (number of times course is attended and grade achieved for every course preceding BP in the chain, namely OP, OVP, AK and SPA), and ninth (number of times BP will be attended) as target class.



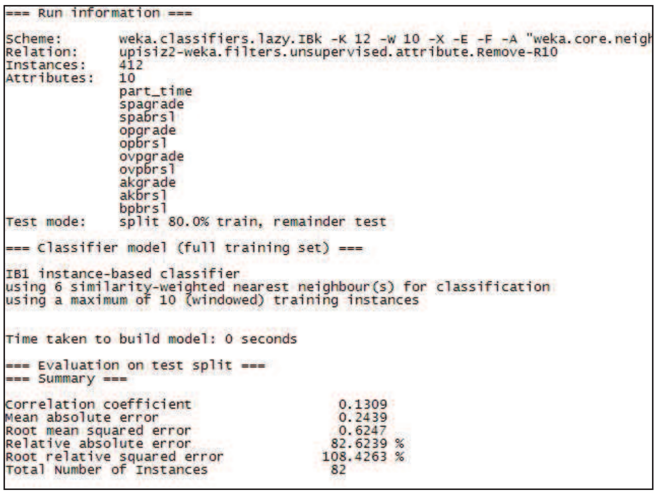
```
=== Run information ===

Scheme:       weka.classifiers.trees.REPTree -M 6 -V 0.1 -N 30 -S 1 -L 10
Relation:     upisi22-weka.filters.unsupervised.attribute.Remove-R10-weka.filters.unsupervised.attribute.Remove-R1
Instances:    412
Attributes:   9
              spagrade
              spabrsl
              opgrade
              opbrsl
              ovpgrade
              ovpbrsl
              akgrade
              akbrsl
              bpbrsl
Test mode:    split 75.0% train, remainder test

=== Classifier model (full training set) ===


REPTree
============

opbrsl < 1.5
|   spabrsl < 2.5
|   |   ovpbrsl < 1.5
|   |   |   akgrade < 5.5 : 0.67 (9/0.22) [0/0]
|   |   |   akgrade >= 5.5
|   |   |   |   spabrsl < 1.5
|   |   |   |   |   ovpgrade < 7.5
|   |   |   |   |   |   opgrade < 7.5
|   |   |   |   |   |   |   spagrade < 7.5 : 1.2 (65/0.38) [0/0]
|   |   |   |   |   |   |   spagrade >= 7.5 : 1 (10/0) [1/0]
|   |   |   |   |   |   opgrade >= 7.5 : 1 (24/0) [0/0]
|   |   |   |   |   ovpgrade >= 7.5 : 1.02 (111/0.02) [4/0]
|   |   |   |   spabrsl >= 1.5
|   |   |   |   |   akgrade < 6.5 : 1.86 (6/1.14) [1/0.03]
|   |   |   |   |   akgrade >= 6.5
|   |   |   |   |   |   opgrade < 6.5
|   |   |   |   |   |   |   spagrade < 5.5 : 0.87 (28/0.42) [3/0.58]
|   |   |   |   |   |   |   spagrade >= 5.5 : 1.24 (17/0.3) [0/0]
|   |   |   |   |   |   opgrade >= 6.5 : 1.35 (50/0.42) [2/0.28]
|   |   ovpbrsl >= 1.5 : 0.73 (14/0.35) [1/0.08]
|   spabrsl >= 2.5
|   |   opgrade < 7.5 : 1.56 (31/0.57) [1/0.2]
|   |   opgrade >= 7.5 : 1 (8/0) [0/0]
opbrsl >= 1.5 : 0.62 (25/0.24) [1/0.16]

Size of the tree : 25

Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correlation coefficient                  0
Mean absolute error                      0.2758
Root mean squared error                  0.5383
Relative absolute error                 99.9658 %
Root relative squared error            100.0006 %
Total Number of Instances              103
```

Figure 1: Classification results using REPTree as Decision Tree algorithm for prediction of number of times student will need to attend to the BP course before passing it.

Resulting tree has 25 leaf nodes. The mean absolute error we achieved was 0.2758, with worst ranging between 35 and 40%, depending on the type of the Decision Tree algorithm.

2) *Nearest Neighbor*

For Nearest Neighbor approach we had choice out of three algorithms (IBk, KStar and LWL). We used 80% of the instances as training set, and achieved best results with IBk algorithm, with 9 predictor fields (number of times course is attended and grade achieved for every course preceding BP in the chain - OP, OVP, AK and SPA; and part or full time student). Results are shown on Figure 2.



```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 12 -W 10 -X -E -F -A "weka.core.neigh
Relation:     upisiz2-weka.filters.unsupervised.attribute.Remove-R10
Instances:    412
Attributes:   10
              part_time
              spagrade
              spabrsl
              opgrade
              opbrsl
              ovpgrade
              ovpbrsl
              akgrade
              akbrsl
              bpbrsl
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 6 similarity-weighted nearest neighbour(s) for classification
using a maximum of 10 (windowed) training instances


Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correlation coefficient                  0.1309
Mean absolute error                      0.2439
Root mean squared error                  0.6247
Relative absolute error                 82.6239 %
Root relative squared error            108.4263 %
Total Number of Instances               82
```

Figure 2: Classification results using IBk as Nearest Neighbor algorithm for prediction of number of times student will need to attend to the BP course before passing it.

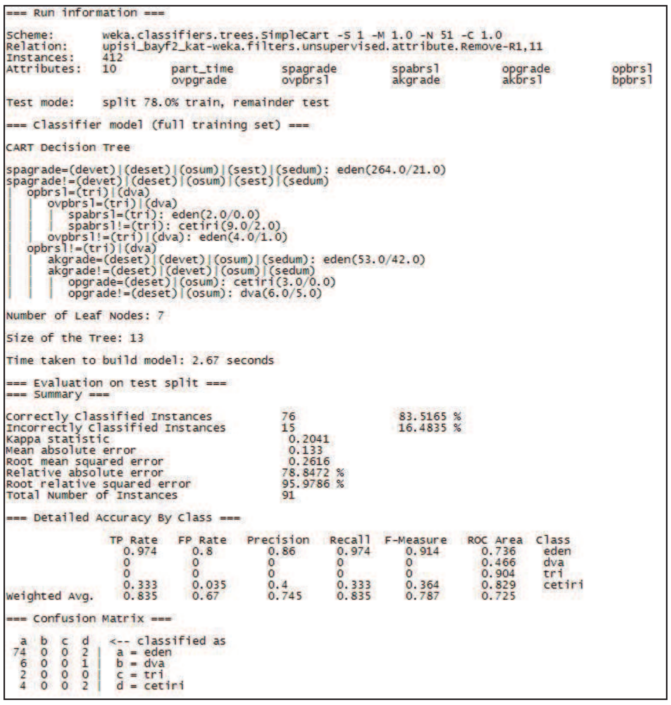
We used 12 nearest neighbors with cross validation using mean squared error, distance method used was Euclidean Distance with 1-distance. Achieved minimum mean absolute error was 0.2439, which is by 3% better than using best Decision Tree with continuous values.

B. *Using categorized values for data mining*

Second part of our experiment includes working with categorized data. For this purpose we translated our values to categories (for example, 1 to “eden”, 2 to “two” and so on), and again used different classification methods on this customized data set. Since all three methods work with discrete attributes, we were able to use all of them on our data set.

1) *Decision Tree*

After experimenting with a number of decision trees, namely, BFTree, ID3, J48, FT, REPTree and so on, the best results we achieved are using 78% of instances as training set, which was then tried on the rest for evaluation, as shown on Figure 3. We used 8 attributes as predictors (number of times course is attended and grade achieved for every course preceding BP in the chain, namely OP, OVP, AK and SPA), and ninth (number of times BP will be attended) as target class.



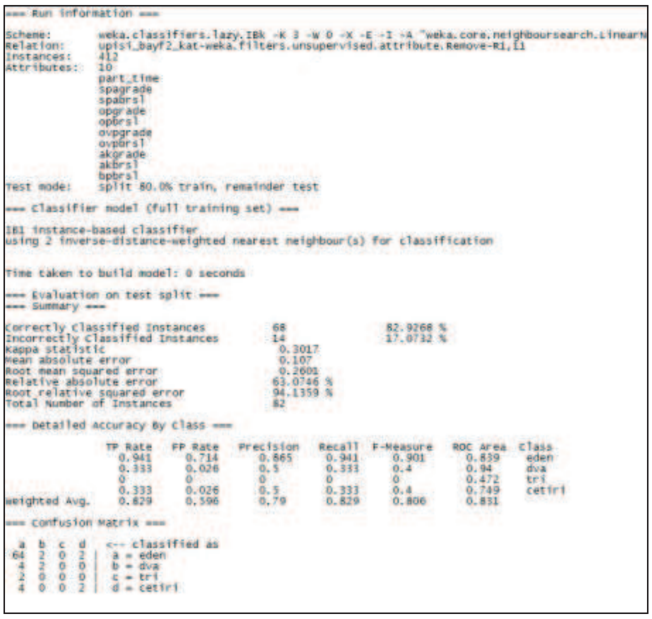
```
=== Run information ===

Scheme:       weka.classifiers.trees.SimpleCart -S 1 -M 1.0 -N 51 -C 1.0
Relation:     upisi_bayf2_kat-weka.filters.unsupervised.attribute.Remove-R1,11
Instances:    412
Attributes:   10         part_time      spagrade       spabrsl        opgrade        opbrsl
                         ovpgrade       ovpbrsl        akgrade        akbrsl         bpbrsl

Test mode:    split 78.0% train, remainder test

=== Classifier model (full training set) ===

CART Decision Tree

spagrade=(devet)|(deset)|(osum)|(sest)|(sedum): eden(264.0/21.0)
spagrade!=(devet)|(deset)|(osum)|(sest)|(sedum)
|  opbrsl=(tri)|(dva)
|  |  ovpbrsl=(tri)|(dva)
|  |  |  spabrsl=(tri): eden(2.0/0.0)
|  |  |  spabrsl!=(tri): cetiri(9.0/2.0)
|  |  ovpbrsl!=(tri)|(dva): eden(4.0/1.0)
|  opbrsl!=(tri)|(dva)
|  |  akgrade=(deset)|(devet)|(osum)|(sedum): eden(53.0/42.0)
|  |  akgrade!=(deset)|(devet)|(osum)|(sedum)
|  |  |  opgrade=(deset)|(osum): cetiri(3.0/0.0)
|  |  |  opgrade!=(deset)|(osum): dva(6.0/5.0)

Number of Leaf Nodes: 7

Size of the Tree: 13

Time taken to build model: 2.67 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          76               83.5165 %
Incorrectly Classified Instances        15               16.4835 %
Kappa statistic                          0.2041
Mean absolute error                      0.133
Root mean squared error                  0.2616
Relative absolute error                 78.8472 %
Root relative squared error             95.9786 %
Total Number of Instances               91

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.974     0.8        0.86      0.974     0.914      0.736    eden
                 0         0          0         0         0          0.466    dva
                 0         0          0         0         0          0.904    tri
                 0.333     0.035      0.4       0.333     0.364      0.829    cetiri
Weighted Avg.    0.835     0.67       0.745     0.835     0.787      0.725

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 74  0  0  2 |  a = eden
  6  0  0  1 |  b = dva
  2  0  0  0 |  c = tri
  4  0  0  2 |  d = cetiri
```

Figure 3: Classification results using SimpleCart as Decision Tree algorithm for prediction of number of times student will need to attend to the BP course before passing it.

Resulting tree has seven nodes and size of thirteen. Percent of correctly classified instances is 83.5% with mean absolute error of 0.133, which is by more than 0.14 better than best result achieved using continuous variables.

2) *Nearest Neighbor*

For Nearest Neighbor approach with categorized values we had choice out of five algorithms (IB1, IBk, KStar, LBR and LWL). We used 80% of the instances as training set, and achieved best results with IBk algorithm, with 9 predictor fields (number of times course is attended and grade achieved for every course preceding BP in the chain - OP, OVP, AK and SPA; and part or full time student). Results are shown on Figure 4.



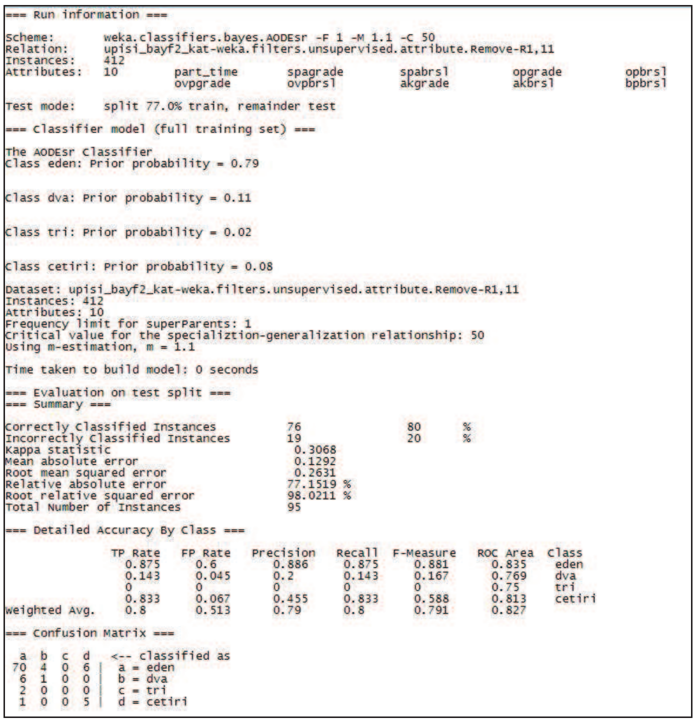
```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 3 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearN
Relation:     upisi_bayf2_kat-weka.filters.unsupervised.attribute.Remove-R1,11
Instances:    412
Attributes:   10
              part_time
              spagrade
              spabrsl
              opgrade
              opbrsl
              ovpgrade
              ovpbrsl
              akgrade
              akbrsl
              bpbrsl
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 2 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          68               82.9268 %
Incorrectly Classified Instances        14               17.0732 %
Kappa statistic                          0.3017
Mean absolute error                      0.107
Root mean squared error                  0.2601
Relative absolute error                 63.0746 %
Root relative squared error             94.1359 %
Total Number of Instances               82

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.941     0.714     0.865     0.941     0.901      0.839    eden
                 0.333     0.026     0.5       0.333     0.4        0.94     dva
                 0         0         0         0         0          0.472    tri
                 0.333     0.026     0.5       0.333     0.4        0.749    cetiri
Weighted Avg.    0.829     0.596     0.79      0.829     0.806      0.831

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 64  2  0  2 |  a = eden
  4  2  0  0 |  b = dva
  2  0  0  0 |  c = tri
  4  0  0  2 |  d = cetiri
```

Figure 4: Classification results using IBk as Nearest Neighbor algorithm for prediction of number of times student will need to attend to the BP course before passing it.

We can see that using categorized values gives almost 83% chance of successful predicting the number of times student will need to successfully pass BP course, with mean absolute error of 0.107. That is almost 0.14 better result when using discrete values.

3) Naive Bayes

Using Naive Bayes Algorithms in Weka is possible only with categorized values. During experiments we worked with several Naive Bayes methods, including AODE, AODEsr, HNB, NaiveBayes, BayesNet, etc. We used 77% of the instances as training set, and achieved best results with AODEsr algorithm, with 9 predictor fields (number of times course is attended and grade achieved for every course preceding BP in the chain - OP, OVP, AK and SPA; and part or full time student). Results are shown on Figure 5.



```
=== Run information ===

Scheme:       weka.classifiers.bayes.AODEsr -F 1 -M 1.1 -C 50
Relation:     upisi_bayf2_kat-weka.filters.unsupervised.attribute.Remove-R1,11
Instances:    412
Attributes:   10        part_time       spagrade        spabrsl         opgrade         opbrsl
                        ovpgrade        ovpbrsl         akgrade         akbrsl          bpbrsl

Test mode:    split 77.0% train, remainder test

=== Classifier model (full training set) ===

The AODEsr Classifier
Class eden: Prior probability = 0.79


Class dva: Prior probability = 0.11


Class tri: Prior probability = 0.02


Class cetiri: Prior probability = 0.08

Dataset: upisi_bayf2_kat-weka.filters.unsupervised.attribute.Remove-R1,11
Instances: 412
Attributes: 10
Frequency limit for superParents: 1
Critical value for the specializtion-generalization relationship: 50
Using m-estimation, m = 1.1

Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          76               80      %
Incorrectly Classified Instances        19               20      %
Kappa statistic                          0.3068
Mean absolute error                      0.1292
Root mean squared error                  0.2631
Relative absolute error                 77.1519 %
Root relative squared error             98.0211 %
Total Number of Instances               95

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.875     0.6       0.886     0.875     0.881      0.835    eden
                 0.143     0.045     0.2       0.143     0.167      0.769    dva
                 0         0         0         0         0          0.75     tri
                 0.833     0.067     0.455     0.833     0.588      0.813    cetiri
Weighted Avg.    0.8       0.513     0.79      0.8       0.791      0.827

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 70  4  0  6 |  a = eden
  6  1  0  0 |  b = dva
  2  0  0  0 |  c = tri
  1  0  0  5 |  d = cetiri
```

Figure 5: Classification results using AODEsr as Naive Bayes algorithm for prediction of number of times student will need to attend to the BP course before passing it.

Results show successful prediction rate of 80%, with 0.1292 mean absolute error, which is worse than other classification techniques, but better than classification without categorized values.

V. CONCLUSION

Classification is one of the most common methods of data mining, and it helps putting results into "brackets" we use to call classes. In this paper we showed how we can use classification to help us predict student behavior when attending BP course. We can see that classification used on our database of students can produce fairly accurate prediction about student success at a course using student's previous experience with related courses. What we hope to achieve is to give teachers "heads-up" about what can they expect from the students just signing on their course. We hope that teachers can use our work to "look into" their new student course and decide if students need more basics to help them more easily master course material, or they can start with more advanced topics. Of course, we used only a small fraction of the "Upisi" database, to help us go through with our experiments, but there are a lot of other relationships between different courses that can be exploited and used in betterment of the teaching process.

We are aware of the fact that the dataset we can use for now is fairly small, and therefore cannot provide enough insight into the student database, but it is a good start, and we will build on it in the future.

REFERENCES

[1] Kantardzic, Mehmed (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons.

[2] The Gartner Group, www.gartner.com.

[3] David Hand, Heikki Mannila, and Padhraic Smyth, Principles of Data Mining, MIT Press, Cambridge, MA, 2001.

[4] Peter Cabena, Pablo Hadjinian, Rolf Stadler, Jaap Verhees, and Alessandro Zanasi, Discovering Data Mining: From Concept to Implementation, Prentice Hall, Upper Saddle River, NJ, 1998.

[5] Discovering knowledge in data: An Introduction to Data Mining, Daniel T. Larose, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

[6] Statsoft Electronic Statistics Textbook, http://www.statsoft.com/textbook/data-mining-techniques.

[7] Principles of data mining, Max Bramer, Springer-Verlag London Limited, 2007.

[8] Weka Software, The University of Waikato, http://www.cs.waikato.ac.nz/ml/weka/.