# Advanced Topics in Machine Learning Programming Task

Jannis Becke, Christian Lausberger and Maik Riestock
Otto von Guericke University Magdeburg - Data and Knowledge Engineering Group

## 1. APPROACH

Our general approach to solve the task was as follows:

- analyse the data set and its various attributes
- cleane the data (removed attributes, missing values)
- use different libraries which provide different machine learning algorithms
- evaluate the quality of selected algorithms with accuracy metric and cross-validation

In order to implement the classification algorithms we used different libraries:

**Weka**[1] - collection of machine learning algorithms for data mining tasks which provides a lot of basic functionality like data storage, filtering, visualisation, and additionally many classifier and clustering algorithms

**Collective Classification**[2] - provides semi-supervised algorithms and extends the Weka library

**LIBSVM**[3] - is an integrated software for support vector classification and provides an implementation of an SVM classifier for Weka

For our solution we selected these four classifiers:

- NaiveBayes
- Decision Table
- SVM
- YATSI

## 2. PREPROCESSING

Analyzing the data leads to the conclusion that preprocessing steps have to be done. At first attributes are selected, that are expected to yield the most improvement in classification accuracy. Then missing values are replaced by mean/mode values. Data was not preprocessed semantically. We did not feel to be able to judge on that without further background knowledge in the field of medicine.

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://www.cms.waikato.ac.nz/~fracpete/projects/collective-classification/
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

This documentation was created in the context of the course Advanced Topics in Machine Learning summer term 2014/15. This course was held by: Prof. Dr. Andreas Nurnberger, M.Sc. Tatiana Gossen; Research group Data and Knowledge Engineering Group, Otto-von-Guericke-University of Magdeburg, Germany.

### 2.1 Attributes

In total 22 attributes were removed for the following reasons:

(1) **Uniqueness**
The first two attributes are unique IDs for the instances. These will not improve classification accuracy, but they could make it worse.

(2) **Missing Values**
Two of the attributes are having a lot of missing values: Patient weight has got 97% missing, payer code 40%. On that account they were removed.

(3) **Unequal Distribution**
A few attributes distributions are heavily biased towards one attribute value. For these attributes instances are having the same attribute value 99% of the time. Needless to say they were removed too.

(4) **Single Attribute**
Two of the attributes only have a single attribute value. Without divergence there is no classification, so these are removed.

### 2.2 Instances

During preprocessing no instances were removed. Because the main attributes with missing values were removed, all instances retained enough data to be of any value for building a classification model. Less than 1% of the data was missing. These missing values were replaced by means for numeric and modes for nominal data.

### 2.3 Conclusion

Affords were made to semantically split the data and analyze the value of different types of attributes. The removal of such groups or single attributes did not lead to an observable improvement in accuracy, therefore all remaining attributes were left in the training data.

## 3. CLASSIFICATION

The attribute to be classified from the data set is *Readmitted* and exist in three characteristics, *No*, *<30* and *>30*. These values denote if the patient was readmitted in more or less than 30 days, and No for no record of readmission. [**?**]

The distribution of the values in these three instances differs greatly from 54864 instances in *No* and 35545 instances in *<30* to 11357 instances in *>30*. These unequal distribution leads to a problem that is addressed in section One Class Problem.

When selecting classifiers the following criteria were important: performance, type specification from task description and feasibility. In addition, many classifiers omitted because they could not solve a 3 classes problem.

Used classification algorithms:

- NaiveBayes, a supervised algorithm

- DecisionTable, a supervised algorithm
- LibSVM, a algorithm based on SVMs
- YATSI, a semi-supervised algorithm

For the evaluation of the trained classifiers, we used two common evaluation methods:
**Percentage Split** - used with a 50%/50% split
**Cross-Validation** - used with 10 folds

By means of the presented classification and evaluation methods, we achieved the following results:

Table I. Classifier result for each evaluation method

| Algorithm | Evaluation | Correctly Classified |
|---|---|---|
| NaiveBayes | Percentage Split | 57.32% |
| NaiveBayes | Cross-Validation | 57.14% |
| DecisionTable | Cross-Validation | 57.75% |
| SVM | Percentage Split | 60.4765% |
| SVM | Cross-Validation | 60.4186% |
| YATSI | Percentage Split | 53.22% |
| YATSI | Cross-Validation | 53.30% |

As seen in Table I there are no significant differences in the evaluation methods used. *Support Vector Machines* are outperforming the other classification algorithms, yielding a result just above 60%. Surprisingly *YATSI* is very close to a random classifier. At this point we haven't got an explanation for it's poor performance. *Naive Bayes* and *Decision Tables* are performing better with around 57%.

During the review of the results we noticed, that some classification algorithms assigned the same label to every instance. The problem arises from the unequal distribution of instances in the target class. To tackle this problem, two approaches were validated presented in section One Class Problem.

## 4. ONE CLASS PROBLEM

During experimental evaluation of different classifiers some classification algorithms assigned the same label to every instance. The problem arises from an unequal distribution in the target class. To tackle this problem, we have tried two approaches presented in the following.

### 4.1 Even Classes

In this approach, we have adjusted the number of instances for each attribute value in the target class. We have first determined the class with the least number of instances. Then we have removed instances from other classes until an equal distribution was obtained.

With this newly created data set we have carried out our classifications again. Surprisingly, the results were showing the same problem again. However the classifiers assigned the label *>30*, with the former fewest occurrences, to every instance. Because of these poor results, we did not pursue this approach any further.

### 4.2 2 Class Problem

In this approach, we converted the 3-class problem to a 2-class problem. We have got the target attribute *Readmitted* changed in order to summarize the characteristics with fewer occurrences, *<30* and *>30*. The goal is to reduce the expressiveness of the attribute. With this change the only possible statement about the readmission of a patient is whether it took place or not.

On this newly created data set we have carried out our classifications again, yielding better results than using the original one. As this is not compatible with the tasks description, we did not pursue this approach.

## 5. CONCLUSION

While solving the task of evaluating different classifiers on the basis of the provided *UCI* data set we came across different problems. A lot of thought was put into preprocessing and the selection of attributes, with the best results coming from a sparsely edited set.

Experiments showed no surprise with *Support Vector Machines* getting the best results with about 60.4%. *YATSI* has performed poorly in comparison to the other classifiers. The focus of additional work could be the analysis of it's behavior to explain or even improve it's performance.

Easing up the classification to a binary problem improved the results classifiers achieve. While this is not a method applicable in this task, it could come in handy for future ones.

## List of Tables



## APPENDIX

## A. TASK DESCRIPTION

Use the Diabetes 130-US hospitals for years 1999-2008 Data Set. This dataset contains records with 55 features for more than 100000 patients. 55 features include information about the diabetic encounters, including demographics, diagnoses, diabetic medications, number of visits in the year preceding the encounter, and payer information. Note that even though some attributes are codified using numeric values, they are nominal (not numeric) attributes. Browse additional information about the attributes in the paper cited on the dataset website. The goal of this assignment is to find suitable methods in the area of machine learning to determine the readmission attribute of a patient and to estimate the quality of selected approaches.

In order to achieve this goal, the following tasks have to be completed:

- analyse the data set and its various attributes
- clean the data (e.g. missing values)
- select an appropriate subset of the attributes and explain your choice
- use different suitable machine learning algorithms (either implement them, or use existing libraries, e.g. Weka)
- determine the quality of your model (e.g. through cross-validation, log loss3, confusion matrix)
- compare your results between different algorithms

Specifically, different suitable machine learning algorithms should include:

- at least two classification algorithms, one of which, SVM, you learn during the course
- at least one algorithm for semi-supervised classification. Use a part of the dataset as unlabeled data for learning (omit the label)

## B. PROGRAM EXECUTION

In order to execute our program you need to open a console and enter the following command:

```
java -jar <PATH TO JAR>\ATMLProgramming.jar
```

The database file has to be saved in the same directory as the jar file. After executing the jar file the menu will guide you through the whole program and ask for some parameters before the classification starts.