Final Report Datamining UvA dec 2008

Andreas van Cranenburgh 0440949
Ricus Smid 9660208

Introduction

We applied several machine learning techniques to a pair of datasets concerning depression. We tried to make a model of the data in order to find correlations between the various attributes and the target functions *dropout* (dataset 1) and *remission* (dataset2). Although we worked together most of the time, Ricus focused somewhat more on dataset 1, while Andreas focused on dataset 2.

Previous Findings

In our previous report we described our initial findings concerning the two datasets. Because we will built on these findings, we provide a short summary of this report:

Dataset 1.
We decided to remove the following attributes because of their dependency on the target function *follow-up* : MADRS.2, BDI_TOT.2, BDI_AFF.2, BDI_COG.2, BDI_SOM.2. Furthermore we removed attributes that seemed to be unrelated to the target function: ID, YEARMINI, jaarmet1 and SOURCE. Also *farma* was removed because this attribute was only measured after 2005 and strangely correlated with follow up. The algorithms we used (J48, REPTREE etc) scored around 50 % accuracy for the modified dataset.

Dataset 2.
For our first report we used Quintiles-MADRS (converted to nominal values) as the target function. Quintiles-MADRS is dependent on Rverschil and on second measurements (MADRS.2, quintiles_BDI, BDI.2 etc), so we removed these attributes. We also removed another possible target function: Remissie-MADRS, because it strongly correlates with quintiles-MADRS for the wrong reasons: Remission (MADRS <10) always means a lower Madras score, since everyone starts with a MADRS > 10. The algorithms we used (J48, REPTREE etc) scored around 20% accuracy for the modified dataset

Algorithms

We want to use an algorithm that gives an output that can be translated in rules of some sort. Therefore algorithms like k-nearest neighbor, multilayer perceptron and naive Bayes are not very useful. Decision trees and rule-based algorithms on the other hand are very useful because they give output in the form of rules or trees that can easily be translated to rules. We therefore decided to focus on these algorithms. In our report we will focus mainly on the decision tree classifier J48, but throughout our experiments we used a range of other classifiers as well.

Preprocessing, Modeling and Results: dataset 1

Performance measures and cost matrices

We removed the attributes ID, farma, MADRS.2, BDI_TOT.2, BDI_AFF.2, BDI_COG.2 and BDI_SOM.2 for reasons described earlier. With the J48 classifier we got the following result:

Correctly Classified Instances			808	54.6685 %
FP Rate	**Precision**	**Recall**	**F-Measure**	**Class**
0.055	0.245	0.121	0.162	lost to follow-up: geen vervolgmeting….
0.241	0.524	0.471	0.496	geen vervolg meting. wel behandeling
0.513	0.59	0.707	0.643	wel followup

It may be worth mentioning that the instances in the provided dataset are ordered according to their classification. Therefore we had to make sure that splitting the dataset into a training and a test set would occur in a randomized way. Fortunately Weka takes care of this with both the percentage split and (of course) the cross-validation.

As can be seen in the table above, the overall accuracy is rather low. Moreover both the precision and recall for the class *lost to follow up* are very low. Precision is the proportion of instances that are classified as *a* that are indeed of class *a*, recall is the proportion of instances that are classified as *a* of all instances that are of class *a*. The F-measure reflects both the precision and the recall for a class.

Medics sometimes use the terms *sensitivity* and *specificity* to express the performance of a classifier. *Sensitivity* is the same as *recall*, while *specificity* is the proportion of instances that are not of class *a* that are indeed classified as negative for this class (this is the reverse of the *False Positive Rate*). The product of these terms (*s×s)* can be used as an alternative to the F-measure. Although we will focus on the F-measure, the values of the *S×S* measure are provided for completeness in the table below.

Specificity	**Sensitivity**	**S×S**	**Class**
0.945	0.121	0.114	lost to follow-up: geen vervolgmeting….
0.759	0.471	0.357	geen vervolg meting. wel behandeling
0.487	0.707	0.344	wel followup

Whether to aim for a high overall accuracy, F-measure, precision, recall, or S×S, depends on how costly one finds false predictions (false positives, false negatives) for a certain class. For example, we can imagine that hospitals are interested mainly in the class *lost to follow up* for reasons that are cost related. People that drop out of the therapy can be considered expensive because of the financial investments done by the hospital. Alternatively the 'health' cost for dropouts themselves could be considered high, since they didn't receive a treatment that helped them getting better. If the hospital wants to reduce investment in people that turn out to be dropouts, it would be ethical to aim for a very high precision for the class *lost to follow up*, since one wouldn't want to exclude people of another class from treatment. If the hospital wants to provide special treatment for people that have a reasonable chance to be *lost to follow up* in order to reduce the number of dropouts, one could instead *aim* for a high recall at the cost of precision, since a special treatment probably wouldn't hurt people of another class. When special treatment is costly, on could aim for a tradeoff between precision and recall: a high F-measure.

Weka provides the possibility to define cost matrices that assign costs to specific false predictions. In order to get a higher precision for the class *lost to follow up* the second and third row of the first column of the confusion matrix should get a lower value. The cost matrix that is shown left of the confusion matrix could be used for this. This matrix can be applied to J48 by using the meta classifier *CostSensitiveClassifier.*

0	1	1
1.5	0	1
1.5	1	0

Confusion Matrix			
a	**b**	**c**	**<-- Classified as**
23	48	119	lost to follow-up
30	251	252	geen vervolg meting. wel behandeling
41	180	534	wel followup

The result can be seen in the table below. The precision for class *a* is somewhat higher now, but at the cost of a lower recall and a lower F-measure.

Correctly Classified Instances	822	55.6157 %	
Precision	**Recall**	**F-Measure**	**Class**
0.288	0.079	0.124	lost to follow-up
0.521	0.493	0.507	geen vervolg meting.
0.591	0.721	0.649	wel followup

The cost matrix shown to the right should result in a higher recall for class *a*. The table below shows that the recall for class *a* is indeed now higher than before. Although the precision has dropped a bit, the resulting F-measure is now higher as well.

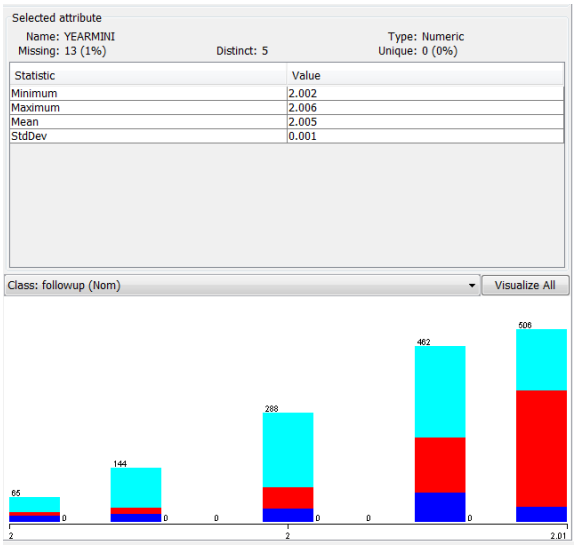
0	2	2
1.5	0	1
1.5	1	0

Correctly Classified Instances	785	53.1123 %	
Precision	**Recall**	**F-Measure**	**Class**
0.226	0.163	0.19	lost to follow-up
0.52	0.467	0.492	geen vervolg meting.
0.586	0.669	0.625	wel followup

Weka provides a meta classifier, *ThresholdSelector,* that optimizes the F-measure for a given class. Because this classifier only works for two-class problems, we experimented a bit by merging the other two classes (using the unsupervised filter *MergeTwoValues*) and then applying this classifier. The results were disappointing however: the F-measure was still pretty low, while the overall accuracy had dropped to a dramatic 12.9%. Moreover the classifier obtained a high recall by simply classifying all instances as *lost to follow* up. This could be a sign that this class might be very difficult to predict from the given data set.

Correctly Classified Instances	190	12.8552 %	
Precision	**Recall**	**F-Measure**	**Class**
0.129	1	0.228	lost to follow-up
0	0	0	geen vervolg meting + wel follow up.

Attribute Selection



YEARMINI

After removal of the earlier mentioned attributes, YEARMINI proved to be the best predictor for follow-up. This is somewhat strange, since MINI is a simple (telephone) interview to exclude depression and anxiety. The year this interview took place shouldn't have a major influence on the target attribute. We investigated the instances according to the attribute YEARMINI and saw a strange tendency starting from year 2003 (the second bar in the figure): the higher the year of YEARMINI the higher the proportion of instances with value *Geen vervolgmeting, wel behandeling* (red colour) and the lower the proportion of *wel follow-up* (light-blue). The discrepancy in class distribution over years can be seen even clearer in the second graph, where all bars are stretched to the same length.

We think it may be the case that people who have the classification *Geen vervolgmeting, wel behandeling* (which we'll call *GVWB* from now on) may include people who will have a follow-up eventually, but are simply not long enough under treatment to be ready for the follow-up. This means that the class assigned to these instances may change later on(!). Another explanation would be that people from who it is not yet decided if they will have a follow up or will be lost to follow up are not included in the data. In either case the data with YEARMINI ≥2005 is biased towards the class *GVWB*. A third explanation is that the data is actually correct. Since the misbalance in the class distribution is probably not caused by features of depression itself it may be caused by therapy, medicine, hospital policy or other factors we have no knowledge of.

We decided to adopt the hypothesis that the data of the years 2005 and 2006 is not very usable due to the first reason we gave above. We therefore decided to consider only data with YEARMINI≤2005 (at the cost of losing a large part of the data). We could have removed instances from the year 2004 also, but then there would have been very little data left. We also removed the attributes YEARMINI and jaarmet1 from the set.

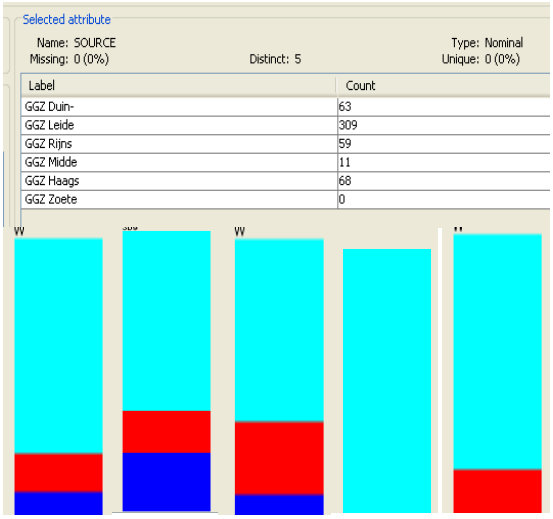
J48 now had an accuracy of 68% (see table). However since the vast majority of instances now has the class 'wel follow-up' the decision tree is strongly biased towards this class. The resulting rule therefore simply always predicts this class (this behavior can be changed by modifying the amount of pruning J48 does).

Correctly Classified Instances	348	68.2353 %
Precision	**Recall**	**F-Measure**	**Class**
0	0	0	lost to follow-up
0	0	0	geen vervolg meting.
0.69	0.983	0.811	wel followup

Source

In our first report we described how we removed the attribute *Source* because we didn't expect this attribute to have influence on the target function. However after having removed the previously described attributes, *Source* became the best predictor for the follow up class. We inspected the class distribution and found that this distribution varies greatly between the different sources.

In the table the stretched bars are shown to reveal the differences in class distribution. Two hospitals don't have any instances that are *lost to follow up (blue)* while GGZ Leiden has a relatively large proportion of this class.



We can only guess what the reasons behind this discrepancy in class distribution might be. It could be the kind of patients the different hospitals treat, it could be the way they treat patients, it could be the administration that differs from hospital to hospital. If we would know, we might get a better insight in the data. Since we don't we decided to remove this attribute.

Attribute Evaluation

In Weka there are several methods to find out which attributes are the best predictors for a dataset. The attribute evaluator *CfsSubsetEval* creates a subset of attributes considering both the predictivity of the attributes and the redundancy between them. *GainRatioAttributeEval* determines for every attribute the information gain.

```
                                    Ranked attributes:
etniciteit                           0.049765   46 pijnstoorn
burgstaat                            0.032103   48 alcafhe
werk                                 0.026746   47 BDD
geslacht                             0.024879   43 GAS
SUIC                                 0.021271   10 BDI_AFF.1
BDI_AFF.1                            0.013183    7 geslacht
ernstdephe                           0.011944   38 paniekstzag
paniekstmag                          0.009739   37 paniekstmag
GAS                                  0.009084   36 dystNAOve
pijnstoorn                           0.008335    1 etniciteit
alcafhe                              0.008055   45 hypoch
```

The subset chosen by *CfsSubsetEval* after removal of the earlier mentioned attributes can be seen to the left. Four out of eleven attributes are demographic attributes . The other ones are more directly related to depression. The ranked list of attributes according to *GainRatioAttributeEval* shows a decent overlap with the subset to the left. It does seem however that depression related attributes (diagnoses, tests) are more predictive than demographic attributes.

Resampling

Unbalanced distribution of classes in the instances gives the problem that more information is available for the dominant class(es). This results in a decision tree that is based mainly on attributes that are significant for the dominant class. To overcome the problem of the unbalanced data set, biased towards the dominant clas(ses), we again applied the trick with the *ThresholdSelector* to aim for a high F-measure for the *lost to follow up* class. The table shows that the classifier still performs not very good, but somewhat better than it did for the previous dataset (shown in brackets).

Correctly Classified Instances	108	21.1765 %	
Precision	**Recall**	**F-Measure**	**Class**
0.143(0.129)	0.892(1)	0.247(0.228)	lost to follow-up
0.84(0)	0.096(0)	0.173(0)	geen vervolg meting + wel follow up.

We preferred to consider all three classes instead of just two however. We thus had to find a way to make the training set more balanced, while keeping the test set representative for the provided data set. To make the training set more uniform, while retaining the original class distribution for the test set, we decided to split the set of instances in a training set and a test set and then take a subsample of the training set with a uniform distribution of the classes.

To create the test set we took a (1/3) sample of the data using the supervised filter *StratifiedRemoveFolds*. After saving the test set, we took a the inverted (2/3) sample as training set. We then made the training set uniform by applying the filter *SpreadSubSample* with distributionspread 1.

The results for the J48 classification are shown in the table. Unfortunately the performance for the class *lost to follow up* (where we aimed for) did not improve. This may be due to the greatly reduced size of data we used for this experiment: small data sets with many attributes are more sensitive to overfitting.

Correctly Classified Instances	57	35.2941 %	
Precision	**Recall**	**F-Measure**	**Class**
0.105	0.16	0.127	lost to follow-up
0.164	0.407	0.234	geen vervolg meting.
0.692	0.381	0.492	wel followup

Instance Selection

We continued our experiments with the unsampled version of the data set and now took a look at the instances using the *edit* function of Weka. This revealed some interesting points.

We noticed that there was a lot of sparse data in our set. For example 100 out of 309 instances (remember we did remove a large amount of instances earlier) didn't have an attribute value for the demographic attributes *etniciteit, etnioud, burgstaat, kind, werk* and *opleiding*. Furthermore no less than 205 instances (including almost all 100 instances with missing demographic values) didn't have values for the 18 personality features diagnosed at the first measurement. Very few instances had missing values for the other attributes. We also noticed that GGZ Leiden and GGZ Rijnsweerd accounted for almost all the sparse demographics data and together with GGZ DHaag for all sparse personality features as well.

We thought it might be interesting to measure the performance of J48 on a non sparse data set. We first did this by removing all attributes that had a lot of missing values, which didn't lead to any performance increase. (see tables below).

Result after removing the demographic attributes

Correctly Classified Instances	168	54.3689 %
Precision	**Recall**	**F-Measure**	**Class**
0.211	0.123	0.155	lost to follow-up
0.167	0.087	0.114	geen vervolg meting.
0.632	0.788	0.701	wel followup

Result after removing the personality feature attributes

Correctly Classified Instances	168	47.5728 %
Precision	**Recall**	**F-Measure**	**Class**
0.167	0.123	0.142	lost to follow-up
0.095	0.087	0.091	geen vervolg meting.
0.616	0.682	0.647	wel followup

We then wanted to remove all instances with (many) missing attributes from the data set, instead of removing the attributes themselves. We first added the instances with YEARMINI > 2004 back to our dataset in order to keep a decent amount of data, and then removed all instances with sparse data. The tables below show that removing the instances with missing demographic attributes improves the overall accuracy with 4%, while removing all instances with missing personality decreases the performance with 1%. Initial result J48 *(with attributes MADRS.2, BDI_TOT.2, BDI_AFF.2, BDI_COG.2, BDI_SOM.2, ID, YEARMINI, jaarmet1 and SOURCE removed)*

Correctly Classified Instances	168	42.7605 %
Precision	**Recall**	**F-Measure**	**Class**
0.139	0.111	0.123	lost to follow-up
0.373	0.362	0.368	geen vervolg meting.
0.516	0.554	0.534	wel followup

Result after removing instances with missing demographic attributes

Correctly Classified Instances	168	46.167 %
Precision	**Recall**	**F-Measure**	**Class**
0.147	0.132	0.139	lost to follow-up
0.353	0.342	0.348	geen vervolg meting.
0.576	0.597	0.586	wel followup

Result after removing instances with missing personality features

Correctly Classified Instances	168	41.0825 %
Precision	**Recall**	**F-Measure**	**Class**
0.094	0.094	0.094	lost to follow-up
0.398	0.406	0.402	geen vervolg meting.
0.496	0.488	0.492	wel followup

Because of the strong correlation between some of the hospitals and the abundance of sparse data, these results are difficult to interpret without additional knowledge.

Modifications of Attributes

We continued with the earlier dataset (without instances > 2004) and did some experiments with attribute modification. The results can be compared to J48 results before modification:

Correctly Classified Instances	190	61.4887 %
Precision	**Recall**	**F-Measure**	**Class**
0.091	0.015	0.026	lost to follow-up
0	0	0	geen vervolg meting.
0.636	0.955	0.764	wel followup

Merging classes

We merged the values *Geen vervolgmeting & Wel follow up* of the target attribute earlier. We also tried the two other combinations and measured the J48 performance (with a main interest in the single class)

Merged Classes: *Lost to follow up & Geen vervolgmeting*

Correctly Classified Instances	199	64.4013 %
Precision	**Recall**	**F-Measure**	**Class**
0.517	0.135	0.214	lost to follow-up & geen vervolg meting.
0.657	0.929	0.77	wel followup

Merged Classes: *Lost to follow up & Wel follow up*

Correctly Classified Instances	180	58.2524 %
Precision	**Recall**	**F-Measure**	**Class**
0.828	0.643	0.724	lost to follow-up & wel followup
0.105	0.239	0.146	geen vervolg meting.

In both cases the performance for the single class was boosted a little bit. It is furthermore remarkable that the class consisting of the two supposedly opposite classes (*lost to follow up* and *wel follow up*) is still relatively easy to predict.

Classifier parameters

The amount of pruning that J48 does can be modified with a parameter. Less pruning generally results in a lower overall accuracy (overfitting), but a higher performance for the least frequent classes in an unbalanced data set (extensive pruning results in very simple rules such as: always predict the dominant class). Raising the *confidenceFactor* to 0.75 (= less pruning) has the following results (compare with the table at the top of the page)

Correctly Classified Instances	171	55.3398 %
Precision	**Recall**	**F-Measure**	**Class**
0.2	0.108	0.14	lost to follow-up
0.167	0.087	0.114	geen vervolg meting.
0.64	0.808	0.714	wel followup

Using *binarySplits* of nominal attributes (instead of multiple splits) also improved the performance of the least frequent:

Correctly Classified Instances	161	52.1036 %
Precision	**Recall**	**F-Measure**	**Class**
0.204	0.154	0.175	lost to follow-up
0.188	0.13	0.154	geen vervolg meting.
0.636	0.732	0.681	wel followup

Finally, applying a cost matrix as we have showed before can boost the performance for a particular class even more (*lost to follow up* in this case):

0	15	20
10	0	1
5	1	0

Correctly Classified Instances	115	37.2168 %
Precision	**Recall**	**F-Measure**	**Class**
0.211	0.4	0.277	lost to follow-up
0.1	0.109	0.104	geen vervolg meting.
0.618	0.424	0.503	wel followup

Other Classifiers

We tried several classifiers on our datasets. Some results are shown below. The performance is not optimized for any class. A class with a * is not interpretable as a rule or a set of rules. The best result in a column is shown in red.

Classifier	**Class**	**Accuracy**	**F-measure lost to follow-up**	**F-measure geen vervolg meting**	**F-measure wel followup**
J48	Tree	61.5	0.026	0	0.764
Dec. Stump	Tree	60.2	0.21	0	0.749
REPTree	Tree	62.1	0.051	0.043	0.768
ConjunctiveRule	Rule	64.1	0	0.042	0.78
DecisionTable*	Rule	60.2	0.274	0.164	0.758
JRip	Rule	64.1	0.111	0.07	0.785
NNge	Rule	54.7	0.084	0.169	0.718
OneRule	Rule	**65.4**	0.16	0.073	**0.795**
IBk (k=3)*	Lazy	40.1	0.287	0.208	0.531
MLPerceptron*	Function	56.6	**0.302**	0.175	0.732
NaiveBayes*	Bayes	57.9	0.132	**0.242**	0.743

A remarkable fact is that the OneRule classifier performs pretty well with its overall accuracy (although not for the two minor classes). The classifiers that obtain decent results for all three classes are unfortunately only those classifiers that are not interpretable as rules.

Preprocessing, Modeling and Results: dataset 2

Many techniques we applied to the first data set could off course also be applied to the second data set. However, we only describe the experiments that we think are most relevant here.

Predicting remission

The most straightforward target class is that of remission. This is a binary attribute stating whether the disorder is still present during the second test, or whether the disorder has gone into remission (ie., the test indicates the patient is no longer suffering from depression). We will be using J48, an algorithm to generate decision trees based on selecting the most informative

attributes. Other algorithms have been considered, such as JRip (rule-based), NNge (nearest-neighborlike rule-based) and NBtree (tree with naive bayes classifiers at leaves) but they did not perform as well or their results were harder to interpret.

The following attributes have been removed so as not to confound the results:

```
ID, MADRS.2, BDI_[...].2, SOURCE, Rverschil, remissie_MADRS, quintiles_MADRS, quintiles_BDI.
```

One problem is that there is an uneven distribution of patients having remission and those that do not (the latter represents 80% of the instances). Regardless of the actual remission rates with depression, it seems that the class of patients without remission is over-represented. The results of this is decision trees like this:

```
J48 pruned tree
------------------
: geen remissie (601.0/115.0)
```

Such a tree will, of course, perform as well as the prior probability of "no remission." Note that the preceding decision tree represents a pruned tree. The unpruned version contains many more branches and actually performs better on predicting remission, although it has a lower overall performance -- which is what pruning optimizes over. This means that pruning could run counter to our purposes. The confusion matrix for an unpruned tree on the data as-is (classifying for remission_BDI):

```
   a   b   <-- classified as         F-Measure
 414  72 |   a = geen remissie       0.836
  91  24 |   b = remissie            0.227
```

This outcome is, prima facie, not very hopeful. From the instances which are predicted to undergo remission, the majority will actually be an erroneous prediction! Other algorithms, such as rule-based learners, display similar performance. This means that it is not meaningful to compare different methods using the overall performance unless it exceeds the prior probability of no remission; we will employ confusion matrices instead.

There are two ways of dealing with this problem. The first is to resample the data with a uniform class distribution, the other is to specify a higher misclassification cost for instances incorrectly classified has not having remission. Both appear to be problematic. Resampling could be problematic because that data being trained on has an unrealistic distribution. Cost-sensitive evaluation might be problematic because the costs might be rather arbitrarily chosen to fit the sample.

Results using FilteredClassifier with spread subsampling with a uniform, 50-50 distribution of remission and no remission (note that the filter was only applied to the training data, hence the unchanged number of instances in the confusion matrix):

```
   a   b   <-- classified as         F-Measure
 317 169 |   a = geen remissie       0.746
  47  68 |   b = remissie            0.386

(Percent correct equals 64%).
```

This result is much better than the previous unpruned tree, because the number of correctly predicted remissions is more than twice as large. However, the problem of incorrectly predicted remissions has increased!

Similar tricks can be performed using cost-sensitive classification. Here we choose the MetaCost classifier with a cost matrix which punishes classifications for the over-represented class, hoping to obtain a more balanced result:

```
Cost Matrix    a   b   <-- classified as     F-Measure
 0 1           430  56 |   a = geen remissie 0.859
 3 0           85   30 |   b = remissie      0.229

(Percent correct equals 76%).
```

It is apparent that a trade-off can be made between false positives and false negatives. Unfortunately it seems that reducing one increases the other. Endless fiddling with cost matrices probably increases the risk of overfitting to the data at hand.

The full decision tree:

```
BDI_TOT.1 <= 24
|   jaarmet1 <= 2.005
|   |   ernstdephe = onbekend
|   |   |   MADRS.1 <= 18: remissie (3.0)
|   |   |   MADRS.1 > 18: geen remissie (5.02)
|   |   ernstdephe = gedeeltelijk in remissie: remissie (1.26/0.26)
|   |   ernstdephe = licht
|   |   |   midafve <= 0
|   |   |   |   hoofdiagnose = somatoforme stoornis: geen remissie (1.26/0.26)
|   |   |   |   hoofdiagnose = status afwezig: remissie (3.0/1.0)
|   |   |   |   hoofdiagnose = stemmingsstoornis: remissie (24.0/5.0)
|   |   |   |   hoofdiagnose = angststoornis: geen remissie (6.51)
|   |   |   |   hoofdiagnose = geen klinische diagnose: geen remissie (1.0)
|   |   |   |   hoofdiagnose = anders: remissie (2.0)
|   |   |   midafve > 0: geen remissie (2.0)
|   |   ernstdephe = matig
|   |   |   diagangst <= 0
|   |   |   |   BDI_TOT.1 <= 17: remissie (9.41/2.37)
|   |   |   |   BDI_TOT.1 > 17
|   |   |   |   |   specfob <= 0
|   |   |   |   |   |   kindthuis = nee: geen remissie (23.08/4.21)
|   |   |   |   |   |   kindthuis = ja
|   |   |   |   |   |   |   BDI_SOM.1 <= 9: remissie (3.05/0.05)
|   |   |   |   |   |   |   BDI_SOM.1 > 9
|   |   |   |   |   |   |   |   BDI_AFF.1 <= 4: remissie (3.43/0.43)
|   |   |   |   |   |   |   |   BDI_AFF.1 > 4: geen remissie (7.49/1.0)
|   |   |   |   |   specfob > 0: geen remissie (5.84)
|   |   |   diagangst > 0
|   |   |   |   BDI_AFF.1 <= 1: geen remissie (2.09)
|   |   |   |   BDI_AFF.1 > 1
|   |   |   |   |   BDI_AFF.1 <= 4: remissie (13.55/1.55)
|   |   |   |   |   BDI_AFF.1 > 4: geen remissie (3.13/1.0)
|   |   ernstdephe = ernstig zonder psychotische kenmerken: remissie (6.26/0.26)
|   jaarmet1 > 2.005
|   |   changemod1 = additie
|   |   |   VOLG.1 <= 2.88: remissie (2.0)
|   |   |   VOLG.1 > 2.88: geen remissie (3.0)
|   |   changemod1 = geen wijziging bekend
|   |   |   GEMPH.1 <= 2: remissie (22.47/0.47)
|   |   |   GEMPH.1 > 2
|   |   |   |   tijdROM12wek <= 16.43: remissie (2.04/0.04)
|   |   |   |   tijdROM12wek > 16.43: geen remissie (2.0)
|   |   changemod1 = switch: geen remissie (1.0)
|   |   changemod1 = additie + switch: remissie (1.0)
BDI_TOT.1 > 24
|   BDI_AFF.1 <= 5
|   |   BDI_COG.1 <= 9: geen remissie (44.98/0.47)
|   |   BDI_COG.1 > 9
|   |   |   specfob <= 0
|   |   |   |   AFFLAB.1 <= 3.75
|   |   |   |   |   ID <= 10004060: geen remissie (5.78/0.4)
|   |   |   |   |   ID > 10004060: remissie (7.15/0.13)
|   |   |   |   AFFLAB.1 > 3.75: geen remissie (19.05/0.98)
|   |   |   specfob > 0: geen remissie (7.76)
|   BDI_AFF.1 > 5: geen remissie (381.4/0.61)
```

Another problem is that there are two possible outcomes to predict: remission according to MADRS, and remission according to BDI. A possible solution is to filter the data to retain only those instances where the remission according to MADRS and BDI is equal, ie., to filter out conflicting outcomes. This removes about 17% of the original instances. It did not, however, improve the results significantly at first glance.

A last straw was to employ boosting. Boosting attempts to improve the performance of a weak algorithm by repeatedly training on misclassified instances. This theoretically allows algorithms to approach perfect performance, given enough iterations. In this case however it did not yield significantly better results, and has the added downside of producing numerous cascading decision trees which are more difficult to interpret than a single decision tree. The results, using AdaBoostM1 on a uniform spread subsample:

```
   a   b   <-- classified as         F-measure
 315 171 |   a = geen remissie       0.744
  46  69 |   b = remissie            0.389

(75% percent correct)
```

Predicting improvement

Instead of predicting which patients will go into remission, it is also interesting to predict which patients generally improve. The difference is that this will include patients going from severe to mild depression. A way to do this is to group the last four remission quintiles into one class (these show improvement), and contrast it with the group in the first quintile (who show little to no improvement according to the tests). Although patients generally improve over time, whether treated or not (Posternak et al, 2001; as cited on Wikipedia), it can be useful to discover what kind of patients form an exception to this tendency.

Results using uniform spread subsampling :

```
   a   b   <-- classified as                F-measure
 348 122 |   a = twee_vijf_drie_vier 0.777
  78  39 |   b = een                        0.281

(65% percent correct)
```

The decision tree:

```
paniekstmag <= 0
|   midafve <= 0
|   |   switch1far = niet van toepassing
|   |   |   BDI_COG.1 <= 2: een (8.84/0.92)
|   |   |   BDI_COG.1 > 2: twee_vijf_drie_vier (156.21/53.38)
|   |   switch1far = snri: een (3.49/0.31)
|   |   switch1far = geen
|   |   |   jaarmet1 <= 2.004: twee_vijf_drie_vier (2.55/0.19)
|   |   |   jaarmet1 > 2.004: een (4.43/1.27)
|   |   switch1far = tca: twee_vijf_drie_vier (0.0)
|   |   switch1far = snri en benzo: twee_vijf_drie_vier (0.0)
|   |   switch1far = ssri: twee_vijf_drie_vier (2.32/0.12)
|   |   switch1far = tca en benzo: twee_vijf_drie_vier (1.16/0.06)
|   |   switch1far = benzodiazepinen: twee_vijf_drie_vier (0.0)
|   midafve > 0: een (9.0/2.0)
paniekstmag > 0
|   jaarmet1 <= 2.003: twee_vijf_drie_vier (2.0)
|   jaarmet1 > 2.003
|   |   BDI_AFF.1 <= 3: twee_vijf_drie_vier (3.0/1.0)
|   |   BDI_AFF.1 > 3: een (9.0)
```

After some attempts it appears that this classification task is even more difficult than the previous one.

Attribute selection

Besides class prediction it is also interesting to look at what subset of attributes is most predictive of the actual class. The simplest method of doing this is to iteratively add the attribute most correlated with the class to evaluate on, while simultaneously selecting attributes which are least correlated with those already selected. The result is a small subset of relatively independent and predictive attributes:

```
Attribute Subset Evaluator (supervised, Class (nominal): 93 remissie_BDI):
       CFS Subset Evaluator

Selected attributes: 2,3,4,5,23,34,46,47,54 : 9
                     MADRS.1
                     BDI_AFF.1
                     BDI_COG.1
                     BDI_SOM.1
                     wijz1psy
                     switch2psy
                     COGVER.1
                     IDENT.1
                     NINTIM.1
```

The test results (MADRS, BDI) are indeed good predictors FOR remission, but the rest of the attributes are a somewhat arbitrary selection (the peformance of this set of attributes was not noteworthy). Much of the other attributes might have been equally important predictors, such as attributes describing addictions, other disorders and marital and job status. The fact that these are not part of this attribute subset demonstrates the difficulty of making non-arbitrary inferences from this data.

This prior knowledge of which attributes are relevant also allows us to make a manual selection of attributes, and see how it performs:

```
              Selected attributes:
                    MADRS.1
                    BDI_AFF.1
                    BDI_COG.1
                    BDI_SOM.1
                    burgstaat
                    kindthuis
                    opleid
                    werk
                    aantdiag
                    ernstdephe
                    aantangst
                    alcafhe
                    deprrecid
```

Using J48 with uniform spread subsampling, the performance is 67.7%:

```
   a   b   <-- classified as        F-measure
 344 142 |   a = geen remissie      0.78
  52  63 |   b = remissie           0.394
```

J48 pruned tree:

```
------------------

BDI_AFF.1 <= 5
|   BDI_SOM.1 <= 12
|   |   deprrecid = enkelvoudig: remissie (20.69/2.81)
|   |   deprrecid = recidiverend
|   |   |   BDI_COG.1 <= 11: remissie (32.74/7.0)
|   |   |   BDI_COG.1 > 11: geen remissie (3.07/0.07)
|   BDI_SOM.1 > 12
|   |   deprrecid = enkelvoudig
|   |   |   MADRS.1 <= 25: geen remissie (4.51/1.0)
|   |   |   MADRS.1 > 25: remissie (4.56)
|   |   deprrecid = recidiverend: geen remissie (26.51/8.51)
BDI_AFF.1 > 5
|   werk = ziektewet/WAO: geen remissie (46.03/7.56)
|   werk = geen betaald werk
|   |   aantangst <= 0: geen remissie (22.27/5.19)
|   |   aantangst > 0
|   |   |   deprrecid = enkelvoudig: geen remissie (6.0/2.0)
|   |   |   deprrecid = recidiverend: remissie (8.82/2.63)
|   werk = betaald werk
|   |   BDI_COG.1 <= 10
|   |   |   burgstaat = partner verloren door scheiding/overlijden: remissie (0.0)
|   |   |   burgstaat = samenwonend/gehuwd
|   |   |   |   deprrecid = enkelvoudig: geen remissie (2.5/0.12)
|   |   |   |   deprrecid = recidiverend
|   |   |   |   |   MADRS.1 <= 25: remissie (4.78/1.0)
|   |   |   |   |   MADRS.1 > 25: geen remissie (2.23)
|   |   |   burgstaat = niet samenwonend/gehuwd: remissie (14.54/3.36)
|   |   BDI_COG.1 > 10: geen remissie (10.77/1.23)
```

Discussion

As may have become clear from the report, we found it very hard to model the data in such a way that a decent performance could be obtained. We therefore consider our report mainly as a description of a useful exploration of machine learning techniques. Of those techniques we tried many in our experiments but, of course, never enough. The number of possible paths to improvement are numerous and it is impossible to follow them all. However, we have some ideas for possible further work on these data sets.

A major problem is that we encountered several unexpected correlations between the target function and attributes for which we have no good explanation. In our report we mentioned the enormous discrepancy in class distribution with regard to the year of the first telephone interview (yearmini). If our hypotheses is correct that either the data set is incomplete (instances with certain classes are missing) or incorrect (instances may be assigned a different class later on) then the whole data set will be in fact useless, unless someone can explain exactly what data is missing/incorrect. We also encountered major differences in class distribution between the different hospitals, which also could use some explanation in order to get a better insight in the data. Furthermore several attributes that we expected to correlate strongly with each other failed to do so (for example BDI-AFF didn't correlate well with diagnoses for affective disorders). Again, we can't explain this, but it makes us wonder how reliable the diagnoses are and how useful they are for making predictions. Getting reliable data in the domain of mental disorder is, we assume, not easy. We imagine that it is very important that the gathering of data happens in the same way over time, place and persons. We are not sure if this has been the case,

and if not, what has gone wrong. It would therefore be necessary to discuss the data with an expert for further progression.

Obviously, better results may be obtained if we try more techniques and refinements and apply them on data sets that are modified in another way. For example, we did experiment with clustering techniques but not enough to make it to good use. We also tried transforming numeric attribute values to nominal values or merging nominal values to get a more natural representation for some attributes and we see potential for some of these techniques to lead to a better performance in the end.

Finally, finding relations between attributes and other target values than we focused on in this report, may prove to be interesting as well.

References

Posternak MA, Miller I (2001). "Untreated short-term course of major depression: A meta-analysis of outcomes from studies using wait-list control groups". Journal of Affective Disorders ***66 (2–3): 139–46.***

Major depressive disorder. (2008, December 6). In *Wikipedia, The Free Encyclopedia. Retrieved 14:31, December 6, 2008, from* http://en.wikipedia.org/w/index.php?title=Major_depressive_disorder&oldid=256226847

http://en.wikipedia.org/wiki/Digital_object_identifier

I. Witten, E. Frank (2005). Data Mining - Practical Machine Learning Tools, Techniques 2d ed.