Exercise 4: Data Mining II – Ensembles

1) Part I – Warm-Up

To get you started with ensemble learning, have a look at the dart dataset that is provided in ILIAS. This dataset contains the positions of darts thrown by four different people. Use "dart_train.csv" to train to simple classifiers and check their performance on "dart_test.csv". As classifiers, use a k-NN with k=1 and an SVM with polynomial kernel and a degree of 2.

Inspect the classifications on the training data (connect the "tes" port of the cross validation operator). What can you say about the decision boundaries?

Now combine the two classifiers by stacking (use the "X Stacking" operator from the Mannheim RapidMiner Toolbox). Create a new attribute "distance from centre" and use a decision tree as meta learner, so you can see how the decisions of the base learners are combined. Can you improve the performance on the test set?

2) Part II – Data Mining Cup 2006

In this exercise we will focus on all kinds of ensembles you have learned about in the theoretical part of this lecture. Please remember the methodology of Voting, Bagging, Boosting and Stacking before you start working on the tasks.
We will use the DMC 2006 dataset for this exercise. For time reasons, please use the SPLIT DATA (0.7/0.3) operator to create a training set and a test dataset up front (For the real task, we also could use X-Validation, but this is more time consuming). First convert the attribute gms_greater_avg into a binominal attribute. Then set this attribute as label. Exclude the gms attribute and as well as listing_end_date, listing_start_date, listing_subtitle, listing_title.

Setting up a baseline:
Now use the 0.7 split of the data to train a classifier. Please remember, you just want to set up a baseline, so optimization of all parameters is not necessary. You should first get a feeling on how good you can do with a standard classifier.

Using Bagging to create a combined model:
Now that we have used different classifiers and learned meta-learners we want to focus on using bagging to increase the accuracy for our data set. Apply the BAGGING operator (0.5 / 10) to your process (same as before – still use SPLIT DATA) and test different classifiers. Alternatively you can use a RandomForest.

Chaining classifiers using Boosting:
Use the ADABOOSTING operator with 10 iterations and apply the different classifiers you have tested before. Compare the results to bagging and the default variation of the classifier.

Combining classifiers using Voting and Stacking:
Select the best performing single classifier and combine them within the VOTE operator of

RapidMiner. Play around with different combinations of the best performing classifier. Think about, if it would help, to include also a bad performing operator into the VOTE operator. When you think, you find a good combination; try to improve using STACKING instead of simply voting. Try to learn a meta-classifier based on the outcome of the base-classifiers only. Then use all attributes to learn the meta-classifier.

What does it cost? MetaCost Operator:
As we want to make money from our dataset, we want to find articles which will sold under the average GMS to know beforehand on which auction we need to focus to make a “schnäppchen”. Set up the process, using the METACOST operator and initialize the misclassification cost matrix in the way, that if an item is sold under the average but is classified as “to be sold over gms” we lose 5 and in the other way around we just lose 1, as we bid maximum the GMS. The meta-cost operator works similar to the cost performance operator, you should know from DataMining I but instead of calculating the model, its automatically manipulating the created model in the right direction. What is the best classifier you can find?