In this project we worked mainly with pre-built functions. It was easy to achieve an accuracy of ~87%, but we struggled for quite some time to reach 90%.

We started off using Random Forest in R. The package that gave the best results was RFSRC. Its multilabel task scored 88% accuracy.

Still in R, we went for a C-classification SVM

with radial kernel, and used cross validation to optimize for cost(=exp(0.5)) and gamma/influence parameter (=0.1). This gave us an accuracy of almost 89%. We also tried to utilize cross validation to find features that we can leave out of the classification task (since for example x12 and x13 are equal).

Afterwards we tried to improve the RFSRC result by ignoring features that we found to be not important by “brute force”: looking at the correlation matrix and variance inflation factors to sort out, step by step, the feature adding the least prediction power.

Now using Python, we went through multiple classification models given by the sklearn library. We split the training set into a training and a validation part and employed 10-fold cross validation to search for the best accuracy score. The best score was obtained using the BaggingClassifier (bootstrap aggragating: drawing random subsets several times and performing classifications on each of them; then averaging the result) as a meta-classifier with 120 ExtraTreesClassifiers as base-classifiers. Interestingly we got a better score by drawing the samples without replacement.