**Assignment MSB1011 – Machine Learning and Multivariate Statistics**

***Report***

2020-2021

Koot, MEA (i6084689)

# Background

This report accompanies the Python classification script ‘MSB1011\_Koot\_I6084689’. The script is created and run with Python version 3.7.3 in the free and open source ‘Scientific Python Development Environment’ (Spyder), accessed through Anaconda 3.0. The goal of the script is to correctly predict for a 1000 subjects (i.e., samples) whether each is a patient or a healthy control subject. This classification is based on the gene expression levels of 400 genes (i.e., features) and the classes are mutually exclusive. For this binary classification problem I have trained and compared two discriminative classifiers: a support vector classifier (SVC) and logistic regression (LR) algorithm.

An SVC is a support vector machine (SVM) used for classification problems. An advantage of SVMs is that they are robust against high dimensional spaces, and they are effective when the number of dimensions exceeds the number of samples (Géron, 2019). However, it is important to choose the right regularization parameter and the right kernel to prevent overfitting. LR is a linear model which can be used for classification, for this method it is also important to select the regularization parameter (Géron, 2019).

For training and selection of the best model, these classifiers are applied to a set of labeled training data. This data consists of expression levels of the same 400 genes for a set of 200 samples. After loading packages and setting gene expression and subject label variables, the first step is to inspect the balance of the labeled data. 105 out of 200 samples are patients ( 52.5 %), and 95.0 are controls ( 47.5 %). Thus, the data is almost perfectly balanced. Because the classes are almost balanced, bias because of underlying population proportions is well-nigh non-existent. Therefore, accuracy is an appropriate measure for the performance of the classifier, and other measures were not be considered (Thakur, 2020). Accordingly, there is no information on the disease. Therefore, I am equally interested in correctly identifying patients and healthy controls, respectively (i.e., specificity and sensitivity are identical in terms of importance.

Then, the scikit-learn library is used to apply a nested cross-validation (CV) method to optimize regularization parameter C, train the models, and yield the estimated prediction accuracies of the models (Krstajic, Buturovic, Leahy, & Thomas, 2014; Pedregosa et al., 2011). For each iteration of the outer loop the best inner model – i.e., the model with a C that yields the best inner accuracy estimate – is selected and evaluated on the outer test set. Hence, the outer layer estimates the quality of the models trained on the inner layer. Based on this procedure a new instance of the best model with the best C is created and is fit to the whole outer training set.

For both LR and SVC a grid search method is used in the inner loop of the nested CV to test for various values of C. The value of C represents the penalty parameter of the error term (Hsu, Chang, & Lin, 2003). The closer (the positive) value of C is to zero the bigger stronger the regularization of the model. In other words, a model with a high C can result in overfitting, because it tries to classify all training examples correctly, while a low C results in a smoother decision surface (Hsu et al., 2003; Pedregosa et al., 2011). Therefore, SVC and LR are tested with various values for C, with the highest tested C = 1.

Both the inner and the outer fold layers of the nested CV are created with a stratified nested K-fold, this prevents overlap between the folds and ensures that the class proportions for each fold are equal (Géron, 2019). The outer split is created with a 5-fold (160 outer train vs. 40 outer test samples) and the inner split with a 4-fold (120 inner train vs. 40 inner test samples). Hence, for both SVC and LR there are 20 trained inner models, of which 4 models are selected as the best model. In order to prevent a high complexity and overfitting, the SVC kernel parameter is set to linear (Hsu et al., 2003). Throughout the script, all random states are set to the same number (i.e., 23) to ensure reproducibility. Furthermore, within each inner fold, a z-score standardization that is fit to the training data is applied to both training an test folds.

This procedure is not only applied to the data with all features, but also to sets with the 300 (75%), 200 (50%), and 100 (25%) of the best features. In addition to the fact that this reduces computational time, it can also improve the performance of the model because it can remove redundant features and/or noise. The selection of the K best features is implemented within the inner CV layer, which prevents overfitting.

# Results

Chart, bar chart

Description automatically generatedThe two barplots below show the accuracy scores of the best model as determined by the inner layer for each of the 5 test folds in the outer layer. Each subplot shows this information for a different amount of K-best features. Each barplot also has an inset which presents the mean accuracy and the respective standard deviation (SD) over the outer folds. The highest mean accuracy (0.735) is yielded by fitting an SVC to the 200 (50%) best features. The 95% confidence interval for the accucacy is 0.589 – 0.881. The SD doesn’t differ largerly from the other SDs. In fact, the LR fit on 300 best features yields a higher SD (0.090), while its mean is lower. For the majority of SVC on 200 best features, the best value for parameter C was 0.005. Therefore, SVC with C=0.005 on 200 selected best features is chosen as the best classifier.

Chart, bar chart

Description automatically generated

In order to further validate this model, a new instance of the model created and fit on the whole outer train set. Again, a stratified 5fold was used, and the search for the best 50% of features was part of the cross-validation. The result can be found in the bar chart below.

# Chart, bar chart Description automatically generated

The mean accuracy for the folds in this validation step was 0.725 (SD=0.081) which is only 0.001 lower than the accuracy estimated earlier. However the SD is slightly bigger, yielding a 95% confidence interval 0.563 – 0.887. Based on this result, the classification accuracy of this model is expected to be approximately 0.725.

What was not taken into account in this is algorithm is that, more often than not, the expression levels of genes are not independent. However, in this algorithm each gene was treated as an independent feature, which could negatively affect the performance of the model. Furthermore, in parameter optimization only the parameter that represents regularization parameter was varied. The performance might benefit from optimization of other parameters as well. Lastly, the number of outer and inner folds affects the model performance. Respectively, a K of 5 and 4 was chosen, because a sample size is quite small. When the sample size is bigger, the classifier can benefit from bigger values for K, but in this case bigger values would have resulted in problematically small test set size.

# REFERENCES

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*: O'Reilly Media.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification. In: Taipei.

Krstajic, D., Buturovic, L. J., Leahy, D. E., & Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics, 6*(1), 10. doi:10.1186/1758-2946-6-10

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Thakur, A. (2020). *Approaching (almost) any machine learning problem*: Abhishek Thakur.