Designing a Machine Learning method to distinguish between Alzheimer patients and healthy controls

TM10007: Machine Leaning

Mark van Leeuwen - *4670655*
Marloes Jager - *4807960*
Puck Noorlag - *4560833*

April 15, 2022

1 Introduction

Dementia is the name for a combination of symptoms in which the brain can no longer process information properly and has an estimated worldwide prevalence of about 24 million. [1] Up to 80% of all dementia diagnoses are Alzheimer's disease (AD). [2] AD is diagnosed by both neurological and neuropsychological examinations and/or imaging such as a Magnetic Resonance Imaging (MRI) scan. Multiple studies show that the pathology of AD is present years before the manifestation of clinical symptoms and the diagnosis. Early detection of AD would allow for early intervention and possibly delay or prevent the manifestation of clinical symptoms. [3] By detecting changes in structure, cerebral blood flow and blood oxygenation, (functional) MRI data could be utilised to distinguish between AD, mild cognitive impairment (MCI) and healthy controls (HC). [4]

The Alzheimer's Disease Neuroimaging Initiative (ADNI) was founded in 2004 to contribute to AD research and facilitate the sharing of data all around the world [5]. In this report, machine learning will be applied to a part of the ADNI database to distinguish between Alzheimer's Disease and healthy controls based on features extracted from T1-weighted MRI scans.

2 Method

Google Colab was used to write a Python code which performs machine learning on a part of the ADNI dataset. The code consists of multiple parts: loading and describing the data, preprocessing the data, feature selection and extraction, training different classifiers and evaluating the results. This process can be seen in figure 1. The Python code can be accessed by the following GitHub link: https://github.com/vleeuwenmark/TM10007_GR13

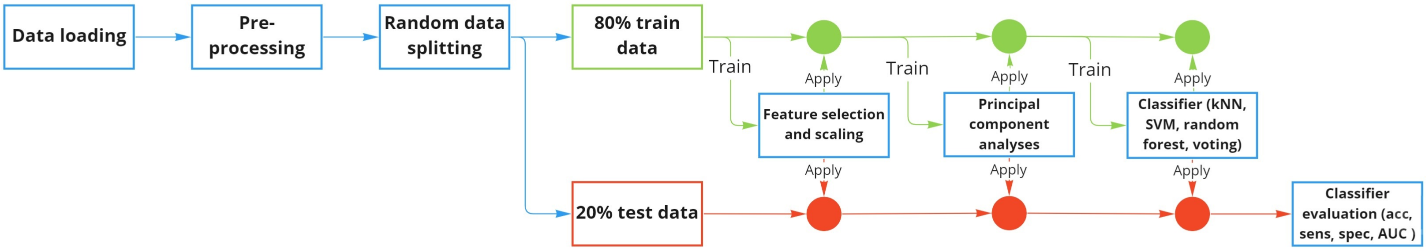
Data loading
Pre-processing
Random data splitting
80% train data
20% test data
Train
Apply
Feature selection and scaling
Apply
Train
Apply
Principal component analyses
Apply
Train
Apply
Classifier (kNN, SVM, random forest, voting)
Apply
Classifier evaluation (acc, sens, spec, AUC)

Figure 1: The experimental set-up of the research.

2.1 Loading and describing the data

The dataset was loaded into Google Colab from GitHub. The amount of features and samples were obtained from the data. Also, the number of samples labelled as Alzheimer disease (AD) and as control group (CN) were obtained.

2.2 Preprocessing the data

The data was analysed and checked for missing data points. Features with more than half of the samples consisting of zeroes were removed from the data treating these zeroes as missing values. This threshold was chosen based on an approach that was taken in previous studies on this dataset. [6,7,8,9]

Two new dataframes were created, one with the labels only (labels) and one with all the other data without the labels (features). These dataframes were then randomly split up into train data (80%) and test data (20%). The choice of these ratios was based on previous studies. [6]

2.3 Feature selection and extraction

To reduce the amount of features, feature selection and principal component analysis (PCA) were applied to the train data. Due to the type of data (numerical input, categorical output, n_samples $>$ n_features) the univariate approach was chosen for the feature selection, using ANOVA to analyse the variance. [10,11] The performance of the different classifiers on train data was evaluated for different thresholds (ranging from 10 to 100) for the F-value. The leftover features were scaled using min-max standardization. The choice for this scaler was made due to the non-gaussian distribution of data and the lack of influence of outliers. [12] After this a PCA was performed. The number of principal components was chosen by using a threshold of 95% for the cumulative variance. [13] Again the performance of the classifiers on train data was evaluated with and without usage of PCA. Feature selection, feature scaling and PCA were afterwards applied to the test data.

2.4 Classifiers

Four machine learning classifiers were chosen and trained: the k-nearest neighbour (k-NN) classifier, the Random Forest (RF) classifier, the Support Vector Machine (SVM) classifier and the voting classifier (using the previous three classifiers). These classifiers were chosen based on a flow chart published by skicit learn, implying that these classifiers were the best fit for our type of dataset. [14]

k-Nearest Neighbour Classifier

The first classifier that is evaluated is k-NN. This classifier was chosen because it can compete with other, more complex classifiers because it can make highly accurate predictions and no specific assumptions about the data have to be made [15, 16].

To find the optimal number of neighbours a grid search was carried out, using the area under the curve (AUC) of the Receiver operating characteristic (ROC) as scoring method. For a 10-fold cross-validation the optimal number of neighbours (between 1 and 30) was sought and the median was used as the optimal number. This number of neighbours was subsequently used to train the k-NN classifier.

Random Forest Classifier

RF is the second classifier used in this research. This classifier has shown to have high performance for binary classification in neuroimaging. [17]

RF has a lot of hyperparameters that can be tuned to improve outcomes and reduce overtraining on a dataset. To find the optimal hyperparameters a gridsearch was performed, using a 5 fold cross validation. The found hyperparameters were then used to train the classifier.

The parameters included in the gridsearch were: number of trees in the forest, the number of features for each split, the minimum samples at each split, the minimum samples at each leaf node and the maximum number of levels in a tree. Creating more decision trees will result in a higher accuracy, but will increase computation time. The other parameters are used as criteria for creating new splits at the nodes. These will increase accuracy up to a certain point, after which the tree will get overtrained. [18]

Even though RF creates its own feature selection, it was decided that the classifier would use the principal components following the above feature selection and PCA. This choice was made because the accuracy of the classifier did not differ for the different features, but it did improve the computation time.

Support Vector Machine

The third classifier that was trained is the SVM. The SVM was chosen for two reasons. Next to the flowchart, the SVM classifier is the most used classifier on the ADNI dataset and the SVM “demonstrated its utility in neuroimaging-based applications, especially in the classification of future clinical outcomes”. [19]

First, the optimal combination of hyperparameters is determined by performing a grid search. These parameters include coefficient C, kernel parameter gamma and the polynominal kernel parameter degree. Besides, it is determined which type of kernel performs best.

The coefficient C is defined as ‘a tuning parameter that controls the trade off between maximising the margin and classifying without error’. [20] A typical range for C is $0.1 < C < 100$. [21] The gamma parameter defines the width or slope of kernel function and has an effect on the polynomial, radial basis function (RBG) and sigmoid kernel. [22] A typical range for gamma values is $0.0001 <$ gamma < 10. [21] The kernels that were included in the hyperparameter grid search are the linear kernel, the RBF kernel, the polynomial kernel and the sigmoid kernel.

Once the optimal combination of parameters and kernel was determined, the SVM was trained on the train data using 5-fold-cross-validation.

Voting classifier

Finally, the previous three classifiers, using the optimal hyperparameters, were used to reduce variance in the outcome and to balance out the weaknesses of the individual trained classifiers. A soft voting classifier was used to give more weight to more confident votes. [23]

2.5 Evaluation

The classifiers were applied to the test data to evaluate the performance. The accuracy, sensitivity and specificity were determined. Also, ROC curves were created.

3 Results

3.1 Loading and describing the data

The ADNI dataset contained 855 samples with 268 features. 519 samples were classified as AD and 336 samples were classified as CN.

3.2 Preprocessing the data

The dataset did not contain any missing values so none were removed or adapted. After removing the features with more than half of the samples consisting of zeros from the data, the remaining dataset contained 855 samples with 261 features.

3.3 Feature selection and extraction

Different thresholds (ranging from 10 to 100) for a minimum F-score were evaluated in the different types of classifiers, all resulting in minimal differences in the result. A threshold of 20 was chosen to exclude the features of which it could be concluded with certainty that the label does not depend on it. For the used random train data, this resulted in 129 features. This is displayed in figure 2a. The PCA resulted in a cumulative variance of 95% when using approximately 30 principal components. This is displayed in figure 2b.

The determination of the feature selection threshold and the PCA components was an iterative process including a lot of testing with multiple combinations. The combination described above was the optimal combination and was used in the further process.

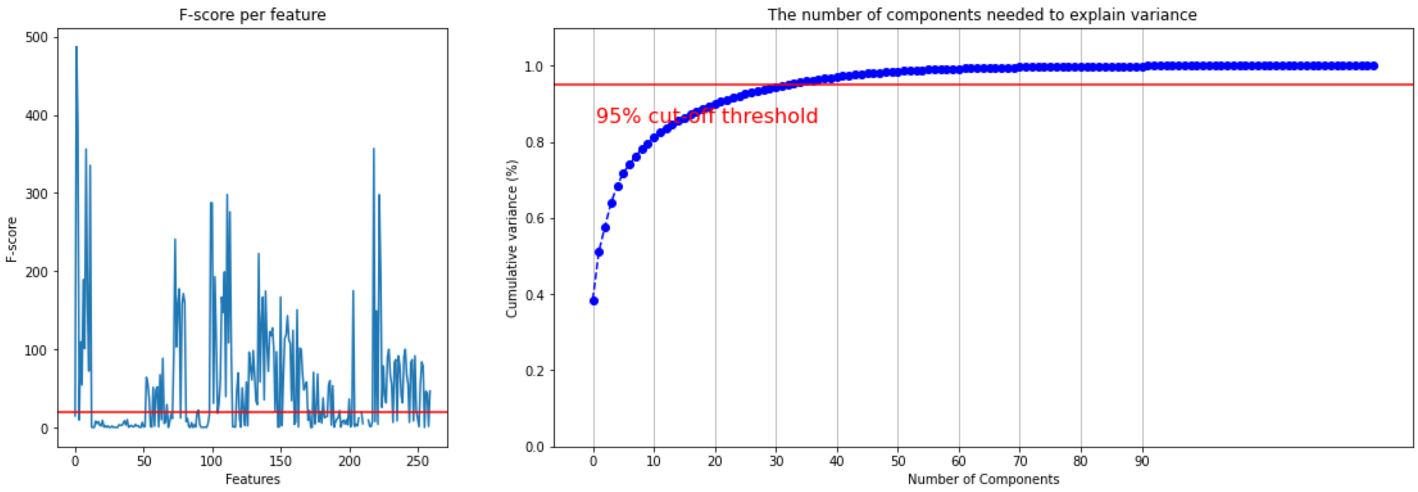

Figure 2: Left (2a): The F-score per feature and the threshold. Right (2b): The number of PCA components determined with a 95% cut off threshold.

3.4 Classifiers

k-Nearest-Neighbour Classifier

Following the grid search with a 10-fold cross-validation, the median optimal number of neighbours was determined to be

25. The distribution of the resulting AUC scores for these optimal numbers, can be seen in figure 3. The results are shown for the test and validation data.

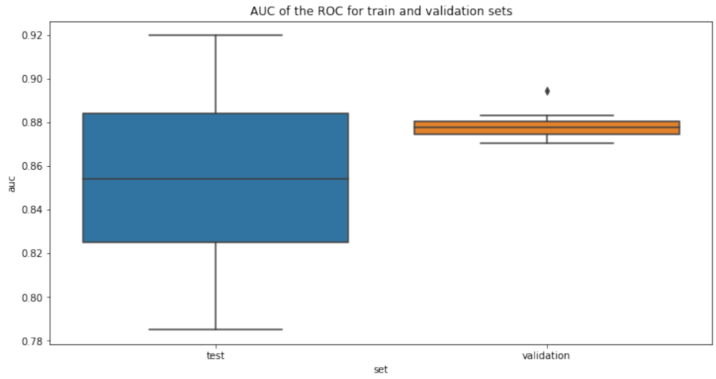

Figure 3: AUC scores of the k-NN classifier for the train and test data following the grid search.

Random Forest
The hyperparameters that followed from the gridsearch were numbers of trees in the forest = 15, numbers of features for each split = square root of the numbers available, minimum samples at each leaf node = 10, minimum samples at each split = 5 and maximum numer of levels in a tree = 10.

Support Vector Machine
The best scoring hyperparameters following the grid search were C = 100, gamma = 0.01, degree = 1 and kernel = rbf.

3.5 Evaluation

The train accuracy, test accuracy, sensitivity and specificity of each classifier can be seen in table 2. The ROC curves are displayed in figure 4.

Classifier	Train accuracy	Test accuracy	Sensitivity	Specificity
k-nearest neighbour	81%	81%	83%	78%
Random forest	81%	80%	85%	72%
Support vector machine	82%	80%	84%	73%
Voting classifier	81%	79%	83%	73%

Table 1: Statistic results per classifier.

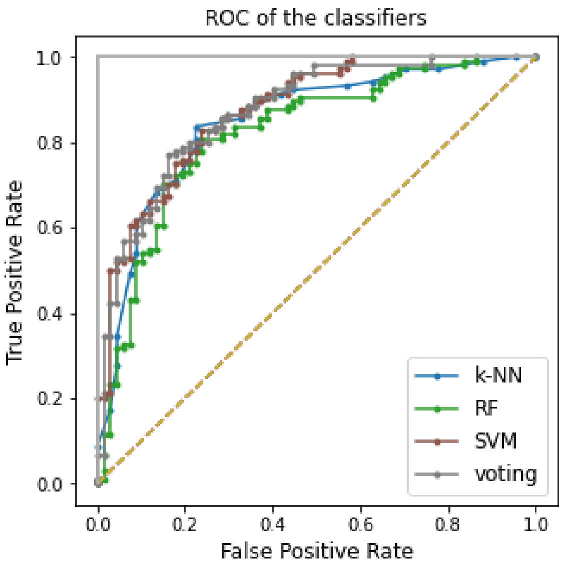

Figure 4: The ROC curves of each classifier.

4 Discussion

From our testing data it can be concluded that the k-Nearest Neighbour classifier, with a test accuracy of 81% and the best ROC curve, is the best classifier for this dataset. However, as can be seen from the results, each classifier seemed to perform comparably, ranging from 79% to 81% for the test accuracy. As can be seen, the train accuracy is very close to the test accuracy, implying the classifiers are not very overtrained and generalize well.

4.1 Limitations of the research and recommendations for the future

During the preprocessing of the data, it appeared that several features contained a number of zeros. Based on literature, it was then decided to remove the features that consisted of zeroes for more than half of the samples. However, there were multiple features consisting of a considerable number of zeroes, for example feature *vf_Frangi_inner_min_SR(1.0, 10.0)_SS2.0* with 140 zeroes. In future research with an extended period of time, the zero removal strategy could be looked at more critically. Both whether more or less features should have been removed. Sometimes, features with many zeroes are correlated with the patient category and can be very useful for classification.

In the part of the code where the data is split randomly, a random state has been added to the line of code to prevent the results from differing from those noted in this report. In real practice, it is only logical that the classifier results differ each time due to the random splitting of the data and actually the random state should be removed.

During this research, one method of feature selection and extraction method was used for every classifier, this was an univariate feature selection with ANOVA followed by a PCA. All classifiers were tested with different thresholds for the feature selection and with and without PCA, but this did not result in considerably different performances. Therefore, the method which resulted in the least features (so, with PCA) was chosen, due to less computational cost. [24] Due to time constraints, it was not possible to look deeper into a better fitting feature selection and extraction method for each individual classifier and this could be improved by a more extensive literature review in future research.

For all classifiers, a grid search was used to evaluate the best hyperparameters. This evaluation was done on the basis of performance and the complexity and generalizability are disregarded. Therefore, it is possible that other hyperparameters, with a slightly lower performance, but a considerably better complexity and generalizability, are overseen. In future research, this possibility could be assessed. Also, the possibility of using a random grid search instead of or in combination with the used grid search could be evaluated in the future.

Another thing that could be implemented in the future is that a weighted random forest could possibly be carried out to improve the performance. Following the clinical needs, a choice could be made in prioritising the diagnosis of Alzheimer or identifying healthy people. By using a weighted random forest the performance for this class could increase.

4.2 Performance on similar unseen data

The different components of the learning process are not trained on the test data, but only applied after being trained, as could be seen in figure 1. Consequently, the test data was only used for the classifiers for the final evaluation. Therefore, the classifiers are expected to have a similar accuracy on similar unseen data as the accuracy of the test data.

4.3 Application of this Machine Learning method in clinical setting

At this time, due to the above mentioned limitations, some adjustments will have to be made to optimise the classifiers and before this method can be applied in the clinic.

The sensitivity of the classifiers is found to range between 83% and 85% and the specificity is found to range between 72% and 78%. This indicates that the classifiers are better in correctly identifying patients with AD than correctly identifying patients without AD. Although further research is necessary, these results correspond or are superior to results of other studies. [6, 25]

One application possibly useful in the clinic in the feature, could be looking into which features are used in the classifiers and are therefore important for the detection of Alzheimer disease. This could contribute to a better understanding of the disease to help optimise classifiers or be of value in clinical practice.

One thing about machine learning that has become very clear is that there is no gold standard and all choices depend on the data set and there is a lot of trial and error involved. Each (feature selection and extraction) method and classifier has its own advantages and disadvantages and it is up to the researchers to use these (dis)advantages to make well-founded choices.

5 References

1. Mayeux R. Epidemiology of Alzheimer Disease. Cold Spring Harbor Perspectives in Medicine. 2012;2(8):a006239.

2. Weller J, Budson A. Current understanding of Alzheimer's disease diagnosis and treatment. F1000Research. 2018;7:1161.

3. Mortimer J, Borenstein A, Gosche K, Snowdon D. Very Early Detection of Alzheimer Neuropathology and the Role of Brain Reserve in Modifying Its Clinical Expression. Journal of Geriatric Psychiatry and Neurology. 2005;18(4):218-223.

4. Yang W, Lui R, Gao J, Chan T, Yau S, Sperling R et al. Independent Component Analysis-Based Classification of Alzheimer's Disease MRI Data. Journal of Alzheimer's Disease. 2011;24(4):775-783.

5. ADNI — About [Internet]. Adni.loni.usc.edu. 2022 [cited 3 April 2022]. Available from: https://adni.loni.usc.edu/about/

6. Muhammed Niyas K. P., Thiyagarajan P., Alzheimer's classification using dynamic ensemble of classifiers selection algorithms: A performance analysis, Biomedical Signal Processing and Control, Volume 68, 2021, 102729, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2021.102729.

7. Y. Dong, C.-Y. Joanne Peng, Principled missing data methods for researchers, SpringerPlus, 2 (1) (2013), p. 222

8. C. Curley, R.M. Krause, R. Feiock, C.V. Hawkins, Dealing with missing data: a comparative exploration of approaches using the integrated city sustainability database, Urban Aff. Rev., 55 (2) (2019), pp. 591-615

9. L.L. Brockmeier, J.D. Kromrey, C.V. Hines, Systematically missing data and multiple regression analysis: an empirical comparison of deletion and imputation techniques, Mult. Linear Regres. Viewp., 25 (1998), pp. 20-39

10. Sklearn.Feature_selection.SelectKBest. (n.d.). Scikit-Learn. Retrieved April 10, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

11. Brownlee, J. (2020, June 4). How to perform feature selection with numerical input data. Machine Learning Mastery. https://machinelearningmastery.com/feature-selection-with-numerical-input-data/

12. Aniruddha. (2020, April 3). Feature scaling. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

13. Mikulskibartosz. (2019, June 3). PCA — how to choose the number of components? Bartosz Mikulski. https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components/

14. Choosing the right estimator. (n.d.). Scikit-Learn. Retrieved April 12, 2022, from https://scikit-learn.org/stable/tutorial/machine_learning_map/

15. Blokdyk, G. (2018). IBM docs: Complete self-assessment guide. Createspace Independent Publishing Platform.

16. Kulkarni, R. (2020, May 23). Summary of KNN algorithm when used for classification. Analytics Vidhya. https://medium.com/analytics-vidhya/summary-of-knn-algorithm-when-used-for-classification-4934a1040983

17. Sarica, A., Cerasa, A., Quattrone, A. (2017). Random Forest algorithm for the classification of neuroimaging data in Alzheimer's disease: A systematic review. Frontiers in Aging Neuroscience, 9, 329. https://doi.org/10.3389/fnagi.2017.00329

18. Probst, P, Wright, MN, Boulesteix, A-L. Hyperparameters and tuning strategies for random forest. WIREs Data Mining Knowl Discov. 2019; 9:e1301. https://doi.org/10.1002/widm.1301

19. Grueso, S., Viejo-Sobera, R. (2021). Machine learning methods for predicting progression from mild cognitive impairment to Alzheimer's disease dementia: a systematic review. Alzheimer's Research Therapy, 13(1), 162. https://doi.org/10.1186/s13195-021-00900-w

20. Dioşan, L., Rogozan, A., Pecuchet, J.-P. (2012). Improving classification performance of Support Vector Machine by genetically optimising kernel shape and hyper-parameters. Applied Intelligence, 36(2), 280–294. https://doi.org/10.1007/s10489-010-0260-1

21. Yıldırım, S. (2020, May 31). Hyperparameter tuning for support vector machines — C and gamma parameters. Towards Data Science. https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167

22. Al-Mejibli, I. S., Alwan, J. K., Abd, D. H. (2020). The effect of gamma value on support vector machine performance with different kernels. International Journal of Electrical and Computer Engineering (IJECE), 10(5), 5497. https://doi.org/10.11591/ijece.v10i5.pp5497-5506

23. Aurelien, G. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.

24. Cheng, A. (2020, February 10). Machine learning: Step-by-step. Towards Data Science. https://towardsdatascience.com/machine-learning-step-by-step-6fbde95c455a

25. Lombardi, G., Crescioli, G., Cavedo, E., Lucenteforte, E., Casazza, G., Bellatorre, A.-G., Lista, C., Costantino, G., Frisoni, G., Virgili, G., Filippini, G. (2020). Structural magnetic resonance imaging for the early diagnosis of dementia due to Alzheimer's disease in people with mild cognitive impairment. Cochrane Database of Systematic Reviews, 3, CD009628. https://doi.org/10.1002/14651858.CD009628.pub2