TM10007: Machine Learning ADNI Report

Group 2

Link to Github Project

Kerkhof, Enzo 44488555

Ramsodit, Karan 4363043

Gerritse, Bram 4462599

Bijl, Saskia 4439457

April 11, 2020

Contents

1 Introduction

Alzheimer's Disease (AD) is a slowly evolving cognitive disease characterized by progressive brain atrophy. This process is difficult to visualize in most imaging modalities, though it is essential for diagnosis and treatment planning. In most cases this degenerative process is already present before cognitive impairment occurs in the patient. This makes detection even more important, because it will allow a prognosis and a treatment planning to be made at a very early stage. (1) The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a North American consortium of universities and medical centres that collect large amounts of data from Mild Cognitive Impaired (MCI) patients, AD patients and healthy elderly control persons. This data is combined in a public database available for all other researchers on their website. The database contains various types of data, ranging from genetic profiles and cognitive assessments to MRI and PET imaging. (2) This study uses a subset of the ADNI database with the aim of distinguishing AD patients from healthy elderly controls. The goal is to perform a classification based on machine learning methods, to predict whether a subject belongs in the AD group or the healthy control group. The classifier will be trained using only T1-weighted MRI images from the ADNI study. To find the optimal classification pipeline various different methods will be tested and compared.

2 Method

2.1 Data

To train the classifier the ADNI dataset is be used. Specifically from the ADNI dataset, a subset of image features from T1-weighted MRI of healthy controls and AD patients will be used. The set contains 855 samples. For each sample 267 different features have been extracted from the T1-weighted MRI image. These images were all taken at baseline using three different types of scanners at 1.5T. The data is split in a test and training set with the latter being 67% of the samples.

2.2 Preprocessing

First the data is preprocessed to extract relevant features and reduce the size of the feature space. This makes finding a relationship between the two classes easier through the reduced number of features between which this relationship needs to be found. The preprocessing is executed in the following order:

1. Feature Selection
2. Feature Scaling
3. Dimensionality Reduction

2.2.1 Feature Selection

Not all features are relevant for the classification, as some might not be informative. To prevent uninformative features creating noise and increasing computation time, a feature selection method would be wise to apply. This will not only result in a problem with a lower dimensionality but also extract features which show a significant difference between classes. Since we have a particular problem of two classes a 2-sampled t-test is performed to determine the relevant features.

2.2.2 Feature scaling

Variety in the ranges of features can cause issues within the decision boundary of classifiers (i.e. an SVM without scaling can have a smaller street width compared to one that uses scaled data). To prevent this a scaler is applied to all selected features. The standard, min-max and robust scaler are applied and evaluated. By plotting the means and standard deviations of the resulting scaled sets, an evaluation can be made which scaler fits the most in the desired pipeline.

2.2.3 Dimensionality reduction

Lastly, dimensionality reduction is attempted with PCA. To maintain a good balance between the reduction of features and the retention of feature information, the smallest number of components are selected which show a 95% variance. The output of the PCA will be used as input for training the classifiers. The t-SNE method is also used to reduce the dimensions with the goal being to visualize the data.

2.3 Classification

2.3.1 K-NearestNeighbor (kNN)

Generally k-Nearest Neighbour classifiers perform very well on low complexity problems with lower dimensionality. This method is tested on our high dimensional problem, so the results can be compared to the other classifiers. This might confirm that this problem is too complex for a more simple classifier to solve.

2.3.2 Logistic Regression (LR)

The Logistic Regression classifier is also used. Logistic regression is useful for explaining the relationship between dependent and independent variables. With the appropriate solver and hyperparameters logistic regression should be able to estimate the decision boundary accurately.

2.3.3 Random Forest (RF)

Next a Random Forest classifier is trained and tested. From data analysis the complexity of the dataset will become more clear. Just by looking at the feature space however, we can already see that this dataset is potentially highly complex. An ensemble model such as Random Forest allows for such complex classification tasks.

2.3.4 Support Vector Machine (SVM)

Lastly a Support Vector Machine is trained and evaluated. This classifier will usually perform well in high dimensional problems because the solution is not phrased in terms of features, but in terms of objects. The kernel used for the SVM classifier is the RBF kernel. This is used due to the high dimensionality of the data from which a non-linear relationship can be expected.

2.4 Optimization & Evaluation/validation

For optimization of the hyperparameters the cross-validation methods GridSearch and RandomSearch are implemented. For all classification methods an ROC curve will be plotted. The performance on the train, test and validation sets will be expressed with the ROC AUC score, confusion matrix and classification report. Each model is run on a 5 fold split of the test data with for each fold an ROC AUC score. Per model, these scores are then used to define the ROC AUC mean and the corresponding standard deviation.

3 Results

3.1 Data analysis

All samples are included in the analysis and all features are used for feature selection. A heatmap visualization (Figure 7.1) shows that the data is clean with no unavailable values or nulls.

3.2 Data visualization

Due to the huge set of dimensions it is impossible to visualize the data in it's original form, but to get a better understanding of how the data is spread out the t-SNE dimensionality reduction approach is used to reduce the number of features to 3 (Figure 1).

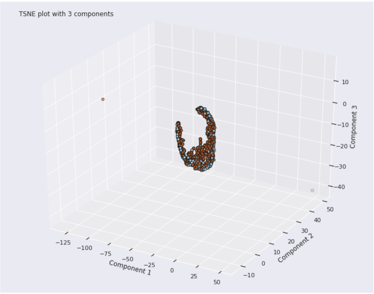


Figure 1: Manifold learning with t-SNE and 3 components visualized

3.3 Preprocessing

3.3.1 Feature selection

Two-sampled t-test is used for selecting informative features from the training data. The 2-sampled t-tested is performed on all features and resulted in 136 remaining features. A visualization of the results of the feature selection is show in Figure 2.

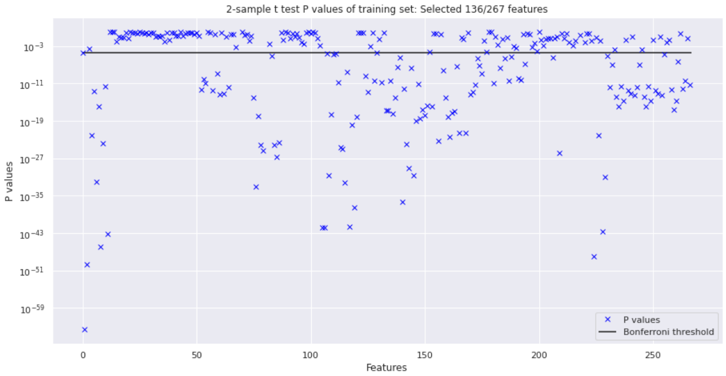


Figure 2: The T2-test probabilities plotted along with the Bonferroni threshold

3.3.2 Feature scaling

The mean and standard deviation of the selected features have been scaled using the MinMaxScaler from sklearn. The min-max scaler is known to be more sensitive to outliers. However after comparing the three methods the min-max scaler seemed to perform better in achieving the goal of a comparable scaled feature set. The result of feature scaling is displayed in Figure 3.

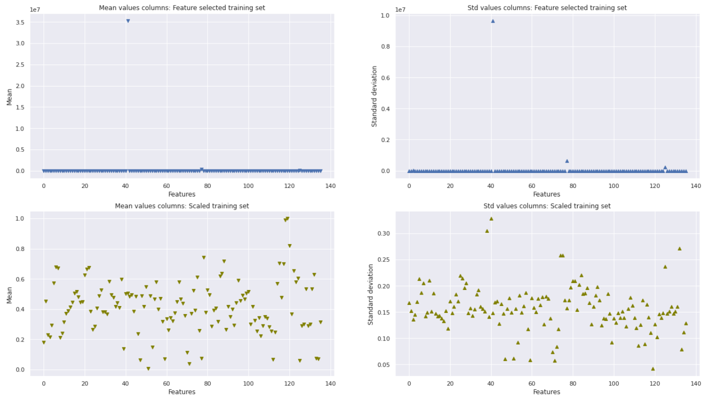
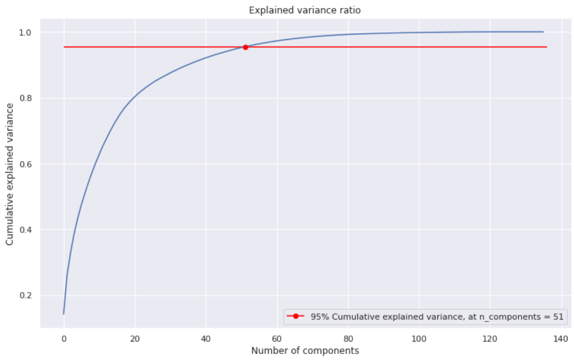


Figure 3: The results of feature scaling in comparison with the original dataset

3.3.3 Dimensionality reduction

With the PCA it was found that 95% of the cumulative explained variance was reached with 51 components (Figure 4). The scatter plot of the first two principal components shows the overlap of both label groups in the feature-space of the first two components. (Figure 5).



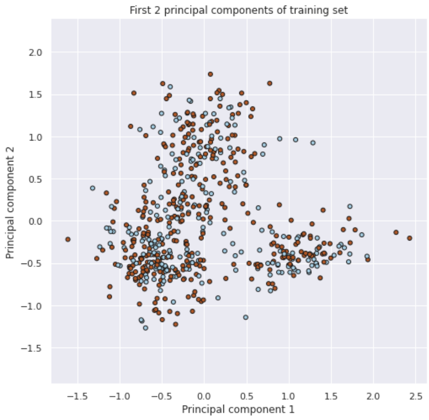


Figure 4: 95 Percentile line with the number of PCA components

Figure 5: PCA with 2 components to visualize the data complexity

3.4 Classifiers

Using the GridSearchCV for kNN and RandomSearchCV for the LR, SVM and RF classifiers a few optimal hyperparameters are retrieved from the train set. Testing the classifiers with these hyperparameters on the test set, with the other parameters left to the default of Sklearn. The kNN classifier had an AUC score of 0.73 using 15 nearest neighbors; the logistic regression model performed with an AUC of 0.85 using L2 regularization with slack being 0.43 and the solver set to sag; the support vector machine had a score of 0.84 using scale for the gamma parameter and a slack of 0.995; the random forest classifier scored 0.81 using 91 trees, a maximum of 35 nodes and a maximum depth of 7. These scores are calculated upon the entire test set. A detailed description of the performance scores can be found in Table 7.2. In Figure 6 the ROC curves for each model can be seen along with the mean and 95% CI of the ROC AUC score. These mean scores and 95% CI, or double the standard deviation, are found by testing the classifiers on 5 folds of the test set.

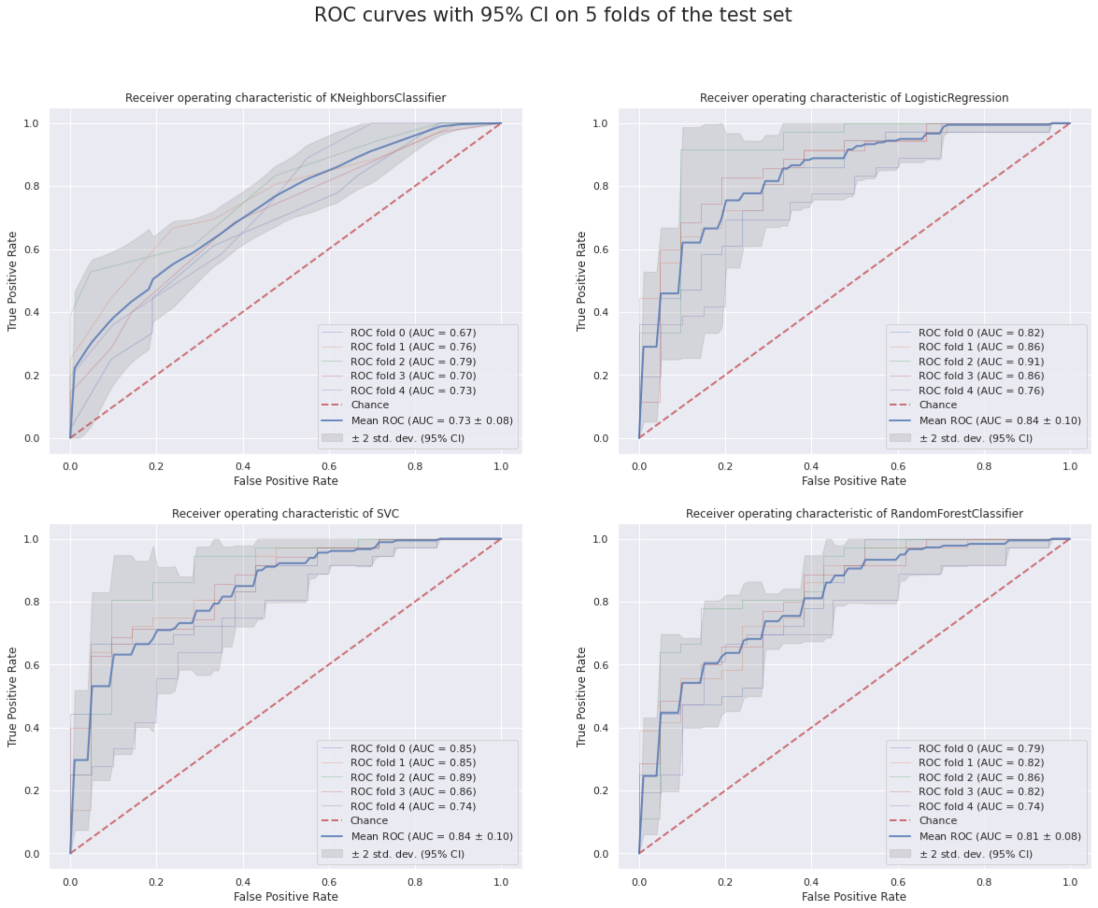


Figure 6: ROC-curves for the different classifiers with a 5-fold cross validation

4 Discussion

With an aging population the early diagnosis of AD is becoming increasingly important. A machine learning classifier has the potential to detect AD at an early stage. In this study, the SVM and logistic regression models showed comparable results. When calculating the test scores on the 5 folds of the test set, they result in very similar mean and standard deviation of the ROC AUC scores (0.84 +- 0.10 and 0.84 +- 0.10). The AUC on the train set, however, is 0.975 for SVM and 0.827 for logistic regression, implying that SVM is showing more overfitting signs of the decision boundary on the train set. The KNN has the lowest ROC AUC score on the test data (mean AUC = 0.73) and the performance on the train data was also relatively low (AUC = 0.769), which is likely caused by this classification method not being capable of classifying for this complex problem at hand. Meanwhile the random forest scored in between these previous results (mean AUC = 0.809) on the test data but it shows overfitting signs on the training data (AUC = 1.000). Thus it will probably not generalize well to other (unseen) data.

The method proposed here is to reduce the dimensionality of the problem through 2-sample t-testing and PCA to find features which retained 95% of the original feature data variance. We conclude that the SVM and logistic regression models have promising performance scores. Although the performance is not high enough for clinical diagnosis of AD, these models might help to objectify T1-weighted MRI scan features as a measure for MCI.

A limitation of the current study is the uncertainties in finding optimal hyperparameters using RandomSearchCV. With every iteration and train/test split slightly different values are retrieved. But not all values within the hyperparameter space are tested for. This can be accomplished through the use of GridSearchCV but the computational resources are not available for this task.

The ADNI database is broad enough for various future studies. This well documented dataset could be used to further tune the methods described here (such that they do not overfit on the training data) or try other methods. It seems a limit has been reached in performance scores using the described methods separately and using the given data. Future research may benefit from using combinations of classifiers (ensembles), as well as using more data which would in turn allow for utilization of more advanced models such as neural networks.

5 Reflection

Group
Firstly we would like to mention the unusual circumstances for this assignment. Having spend no time together physically means time planning and communication was more important than usual. To assist with this we used WhatsApp and Discord as a means of communication. Discord allowed us to have a long extended 'conference call' setting, in which we could both check up on each other, as well as work together in real time on a problem (using screen sharing). Although it took some getting used to, it has proven to be a very valuable tool for this situation. Another worrying aspect was the lack of face to face guidance from course instructors. Through the use of Slack we were able to get all our questions answered. With response time being as quick as it was, this caused no trouble and allowed for a smooth workflow, even when a problem was met. Because all of the course material was crucial for the end result, the bulk of the programming had to be done at a later stage. This was incorporated into our planning. Crucial infrastructure for the rest of the assignment was setup and discussed in the first week. E.g. of how to use the git fork, branches and versions of the notebook(s). After this, we agreed to mostly focus on the weekly material and supporting exercises to gain some proficiency. Then, in the final two weeks the majority of the code was written while also reporting our progress. To tie up all the loose ends we worked together in real time for better coordination. For the significant discussion and conclusions this was crucial.

Karan Ramsodit
During the course of this project I mainly kept busy with the implementation side. On the implementation side I mainly worked on the data analysis, preprocessing with t2 tests, t-SNE testing and visualization and logistic regression. Because of my experience in machine learning I also provided assistance to colleagues whenever possible. For the report I mainly worked on the structuring and review of the final report. I experienced the overall teamwork as pleasant and motivating. Everyone tried to the best of their abilities to apply the techniques learned through the lectures and notebooks. There were times when this was tougher for some then others which did cause some delay during implementation but not so significant that it had a negative effect on the entirety of the project.

Enzo Kerkhof
The start of the course was in a new interesting environment for me. But I like coding and the topics are something I am interested in so I took the lead in reading the course material that came available. This resulted in starting the Whatsapp group and the git fork. In this way I thought that everybody could start at their own pace and time when motivated. Unfortunately I become ill the second week of the course. The combination of this and, I suppose, the other group members also adapting to the new situation resulted in limited communication during the middle of the course. This was slightly demotivating to work on the project. I decided to involve everybody more by starting to adapt the exercises to our dataset. Karan has/had a lot of prior knowledge of the topic and after implementing some of the data analysis this motivated me to continue exploring how certain simple classifiers would perform. This led to me creating a testing notebook which I eventually kept cleaning up and implementing the working code in. Due to this I kept a very good overview of the pipeline we were making and I tried to keep everybody involved the decision making processes. To make decisions on your own in this topic is difficult and I am positive how in the later stages of the assignment everybody got re-involved and we delivered something we all can stand behind.

Saskia Bijl
Personally, I thought this assignment went really well. At first we did not spend a lot of time on the assignment, but concentrated on watching the lectures and understanding the exercises. Then when we had some more knowledge we were able to decide on were we wanted to go with the assignment. Despite being unable to physically sit together, we made arrangements to talk over discord and work on the assignment simultaneously. As coding is not my strongest suit, I learned a lot about applying the theories discussed in the lectures and my teammates, who seemed to be way ahead of me, were very helpful. We all contributed ideas when discussing what steps to take to building our algorithm and then divided the tasks at hand. When someone completed a task or ran into a problem, they suggested to the group to have another meeting. Towards the end of the assignment Enzo and Karan continued optimizing the code while Bram and I started writing the paper. Finally, when the deadline was nearing, we had to make some choices on what we still wanted to implement in the code. We all checked the paper to ensure it satisfied the grading criteria and that it contained the latest numbers and graphs.

Bram Gerritse
As mentioned in the group reflection, the circumstances during this project required a lot of adaptation. To add on to that, the adaptations had to be made in a relatively short time, giving us little room for error. That being said I believe we managed to get on the same page early on in to the project. As for the programming part of the assignment, I feel I mostly served an assisting role. I was personally responsible for testing several different feature selection methods, as well as the Random Forest classifier, along with pairing up with my colleagues for troubleshooting. In most cases I relied heavily on the weekly exercises supporting the course material to complete my parts of the assignment. That was different from Enzo and Karan, who were more proficient. This meant that, as the course progressed, I feel i was able to contribute more. In the report I took a more leading role. I was responsible for the Introduction as well as the majority of the Method section. In conclusion I believe the assignment went well. The generous timeline as a result of only following this course, meant we could give each other the space and time to adapt and learn about the material. The only negative that stands out for me personally, was a lack of communication in the middle of the project. I should have acted more on this by communicating to my colleagues what I was working on.

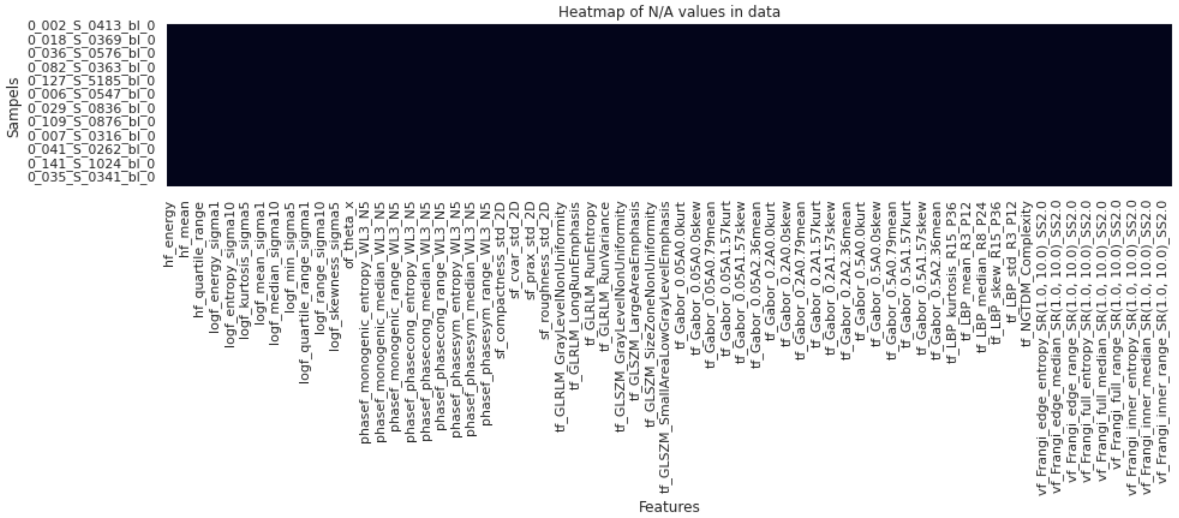
6 Bibliography

[1] Petersen RC, Aisen PS, Beckett LA, Donohue MC, Gamst AC, Harvey DJ, et al. Alzheimer's Disease Neuroimaging Initiative (ADNI): Clinical characterization. Neurology. 2010 Jan;74(3):201–209.

[2] ADNI: About.(n.d.). 2020;Available from: http://adni.loni.usc.edu/about/.

7 Appendix

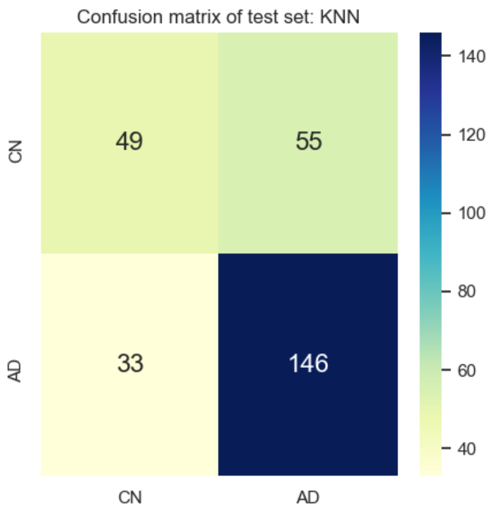
7.1 Heatmap of N/A values in ADNI dataset

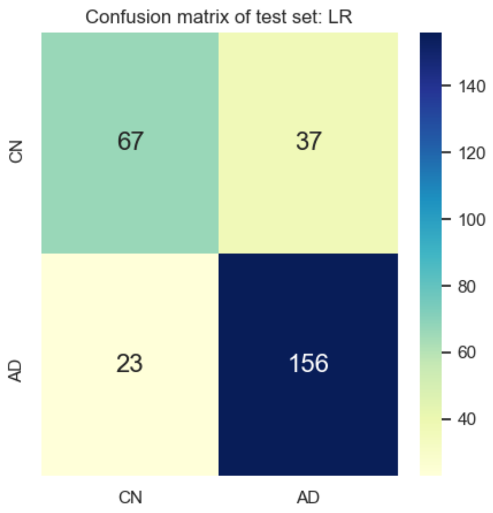


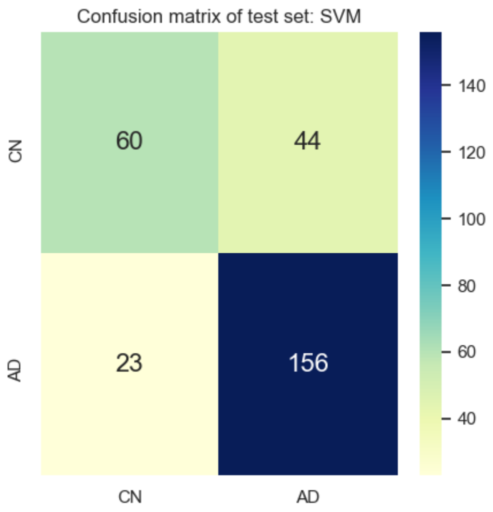
7.2 Classification reports

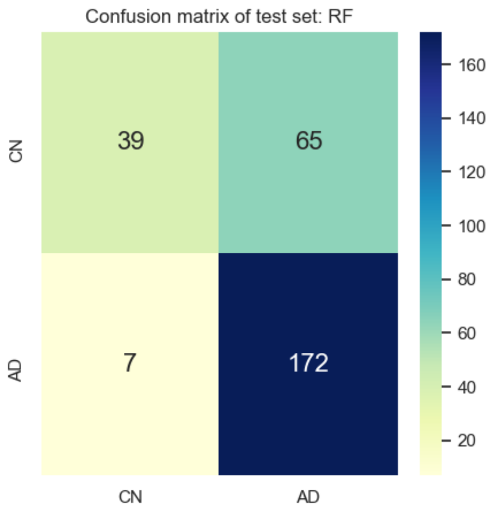
Classifier		**Precision**	**Recall**	**F1-score**	**Support**
KNN	0	0.6	0.47	0.53	104
	1	0.74	0.82	0.77	179
	Accuracy			0.69	283
	Macro avg	0.66	0.64	0.65	283
	Weighted avg	0.68	0.69	0.68	283
LR	0	0.74	0.64	0.69	104
	1	0.81	0.87	0.84	179
	Accuracy			0.79	283
	Macro avg	0.78	0.76	0.76	283
	Weighted avg	0.78	0.78	0.78	283
SVM	0	0.72	0.58	0.64	104
	1	0.78	0.87	0.82	179
	Accuracy			0.76	283
	Macro avg	0.75	0.72	0.73	283
	Weighted avg	0.76	0.76	0.76	283
RF	0	0.85	0.38	0.52	104
	1	0.73	0.96	0.83	179
	Accuracy			0.75	283
	Macro avg	0.79	0.67	0.67	283
	Weighted avg	0.77	0.75	0.71	283

7.3 Confusion matrices


Confusion matrix of test set: KNN
CN
AD
CN
AD


Confusion matrix of test set: LR
CN
AD
CN
AD


Confusion matrix of test set: SVM
CN
AD
CN
AD


Confusion matrix of test set: RF
CN
AD
CN
AD