Final Challenge: Alzheimer Patient Classification

KAOUTHER MOUHEB

STATISTICAL LEARNING AND DATA MINING

UNIVERSITY OF CASSINO AND SOUTHERN LAZIO – SPRING 2022

DATA ANALYSIS

The first step of the challenge is to analyze the datasets. The results are briefly summarized as follows:

- ***Dimensionality***: We inspect ***p*** the number of predictors and ***n*** the number of samples in each dataset:
  - Task 1: n = 164, p = 429 **=> *very high*** dimensionality (***p >>> n***)
  - Task 2: n = 172, p = 63 **=> *Low*** dimensionality ***(p < n)***
  - Task 3: n = 172, p = 593 **=> *very high*** dimensionality (***p >>> n***)
- ***Balance***: we compare the number of samples in each class:
  - Task 1: 81 AD vs 83 CTL **=> *Balanced*** dataset.
  - Task 2: 82 AD vs 90 MCI **=> *Balanced*** dataset.
  - Task 3: 82 CTL vs 90 MCI **=> *Balanced*** dataset.
- ***Correlation***: we calculate the correlations between each pair of predictors, and we draw the corresponding ***correlation matrices*** (Figure 1). We notice the presence of ***high correlations*** between some pairs of the variables (dark red) in all tasks.
- ***Range***: by printing the summary of each dataset, we notice a difference in the range of variables. E.g., ***E2F2∈ [8, 13]*** whereas ***background ∈ [0.003, 0.007].***
- ***Outliers***: the package "***rrcovHD***" is used to detect outliers based on the result of ***robust PCA***. The results show the existence of 5 outliers in task 1, 6 in task 2 and 4 in task 3. However, it is seen that the outliers in tasks 1 and 3 are farther in distance from the other points compared to the outliers in task 2.

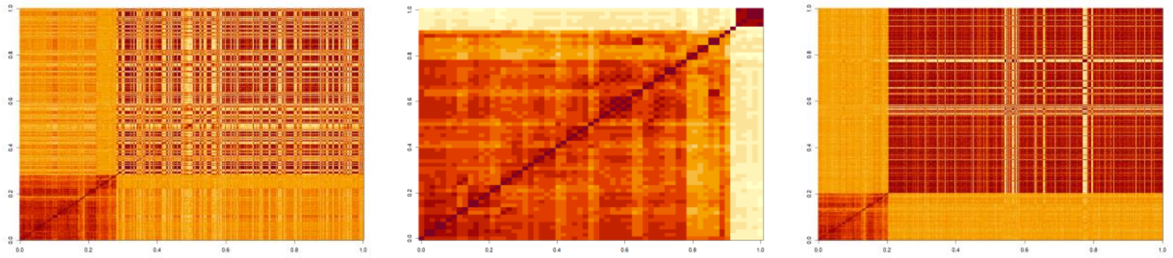

Fig 1: Correlation matrices of task 1, 2 and 3 respectively

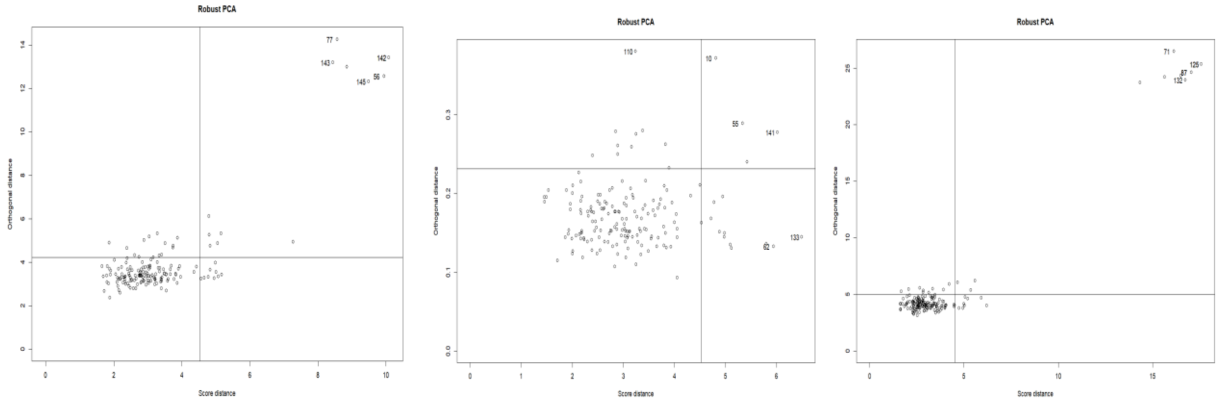

Fig 2: Robus PCA results of task 1, 2 and 3 respectively

METHODOLOGY

In order to solve the classification problem, for each task we will try ***different classification algorithms*** and use the one that provides the ***highest AUC/ROC score using k-fold Cross-Validation***. The following pipeline is applied for each model:

1

Data Preprocessing

To prevent the models from being biased toward predictors with larger values, for each model, the "***center***" and "***scale***" preprocessing steps provided by "***Caret***" are applied. ***Preprocessing is applied within each fold*** when applying k-fold cross validation to avoid positive bias towards the test samples.

2

Feature Selection

To avoid ***the curse of dimensionality***, we apply ***feature selection*** and ***dimensionality reduction*** techniques to obtain ***the most discriminative subset of predictors*** or ***principal components***. The methods are validated using ***k-fold cross-validation*** and evaluated using the ***AUC metric***. We set the seed to a certain fixed value in each task to test on the same split for all models. 3 Main methods will be tried:

Recursive Feature Elimination

The **rfe** function of "**Caret**" is used to perform ***recursive feature elimination*** with ***multiple subset sizes. 5-fold cross validation*** is used to validate the results. If the model has parameters, ***grid search*** is used to test a few values for each subset of features. The subset suggested by RFE is then compared with the other feature selection methods (below) using the same sampling splits (by cross-validation).

Removing Correlated Predictors

A ***correlation filter*** is used to remove predictors with ***correlation higher than a threshold*** with any of the other predictors. ***Multiple thresholds*** (0.6, 0.7, 0.8) are tested for each classifier. If the model has parameters, ***grid search*** is used to fine-tune the parameters for each correlation filter and the parameter that gives the ***highest AUC*** is retained.

Principal Component Analysis

PCA is applied with ***multiple thresholds (***cutoffs for the cumulative percent of variance to be retained**)**. If the model has parameters, ***grid search*** is used to fine-tune the parameters and find the ***best ones for each PCA threshold***. ***K-fold Cross-validation*** is used to validate the results using the ***AUC metric***.

3

Test Outlier Removal

The best model and features obtained from step 2 are retained. We use the ***validation set approach*** to test the impact of ***removing outliers*** on the results of the optimal model on the validation set. If the results improve in terms of ***accuracy,*** outliers are removed from the training set.

Example: Task 1 AD vs CTL

To have a closer look at the methodology we will take Task 1 as an example. Similar analysis is performed for the other tasks.

- In task 1, we need to classify patients to 2 classes:
  - AD: Alzheimer Disease
  - CTL: Control
- To achieve this, we will train the following models:
  - Logistic Regression
  - Linear Discriminant Analysis
  - Quadratic Discriminant Analysis
  - K-Nearest Neighbors
  - Support Vector Machine with Linear kernel
  - Support Vector Machine with Radial Basis Function kernel
  - Random Forest.
- We will take ***k-NN*** as an example for demonstration. The same analysis is applied to the other models.
  - First, RFE is applied with subsets of sizes 1, 5, 10, 25, 50, 100 and 250. The results are validated using 5-fold repeated cross validation.
  - The result shows that RFE suggests using ***10 predictors***. The results of RFE are summarized in Figure 3. The variable importance is shown in Figure 4.
  - The resulting 10 predictors are saved to be tested later with the same 10-fold cross-validation split with the other k-NN models.

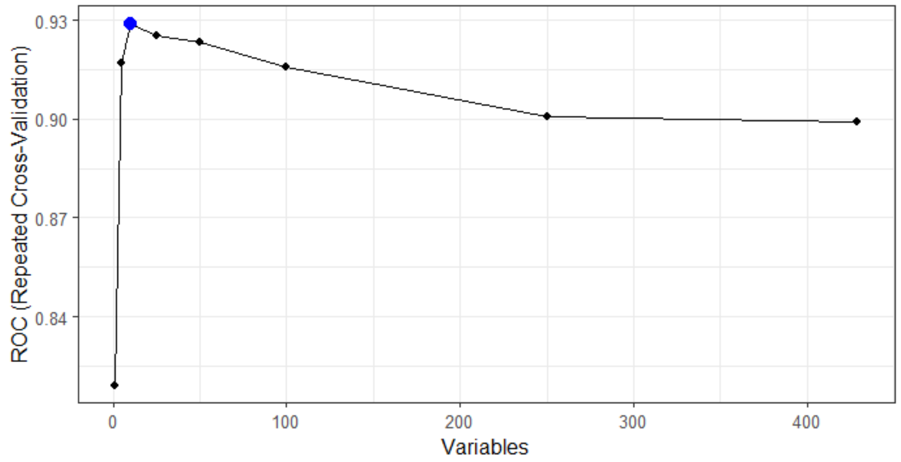

Figure 3: k-NN RFE results

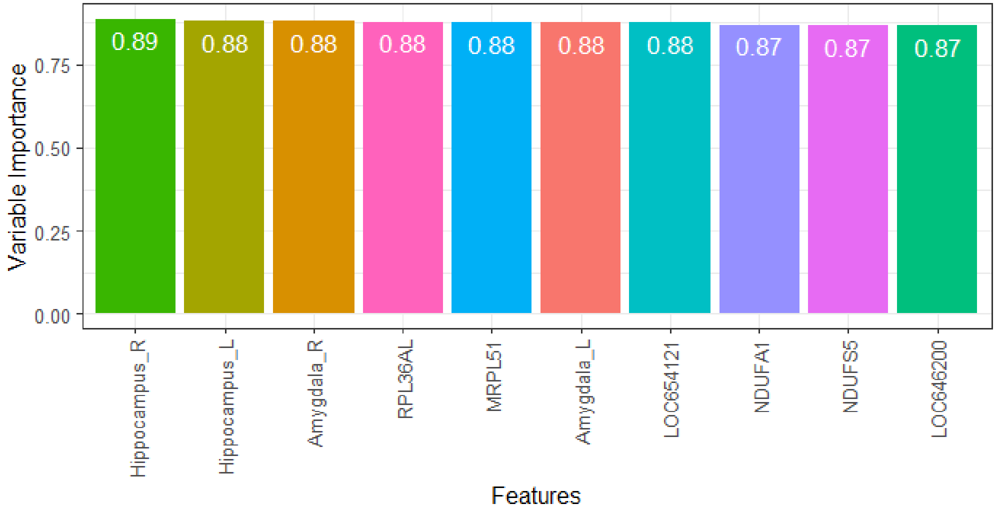

Figure 4: k-NN variable importance

Example: Task 1 (cont'd)

- Next, "***recipes***" library is used to create a ***grid of models*** with different preprocessing steps. We will add the following models to the grid:
  - A ***baseline*** with no feature selection (it uses all predictors).
  - A model that uses the predictors suggested by ***RFE*** (previous slide).
  - Models that use ***correlation filters*** with different thresholds (0.6, 0.7, 0.8).
  - Models that apply ***PCA*** with different thresholds (0.75, 0.8, 0.85, 0.9, 0.95).

- The models in the grid are then trained using ***10-fold repeated cross validation with 5 repeats***. A ***grid search*** with length 10 is applied to find the ***optimal parameter k*** for each model. In all experiments, the data is preprocessed using "***scale***" and "***center***" within each fold according to the formula:

$$x'_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$$

Where *x'* is the new value of the data, *i* is the sample index, *j* is the predictor index

- The models are compared with respect to ***the area under the ROC***.
- Figure 5 summarizes the results in term of AUC/ROC.
- Table 1 shows the optimal number of neighbors k found by the grid search for each model. As well as the number of predictors/principal components.
- From the results, it is seen that the k-NN model that gives the best median Area Under the ROC is the one that uses the ***predictors suggested by RFE*** and a k number of neighbors equal to ***21***. The model achieves ***AUC/ROC = 0.9531250 and MCC = 0.7745967.***
- This model is then saved to be compared with the other classifiers.

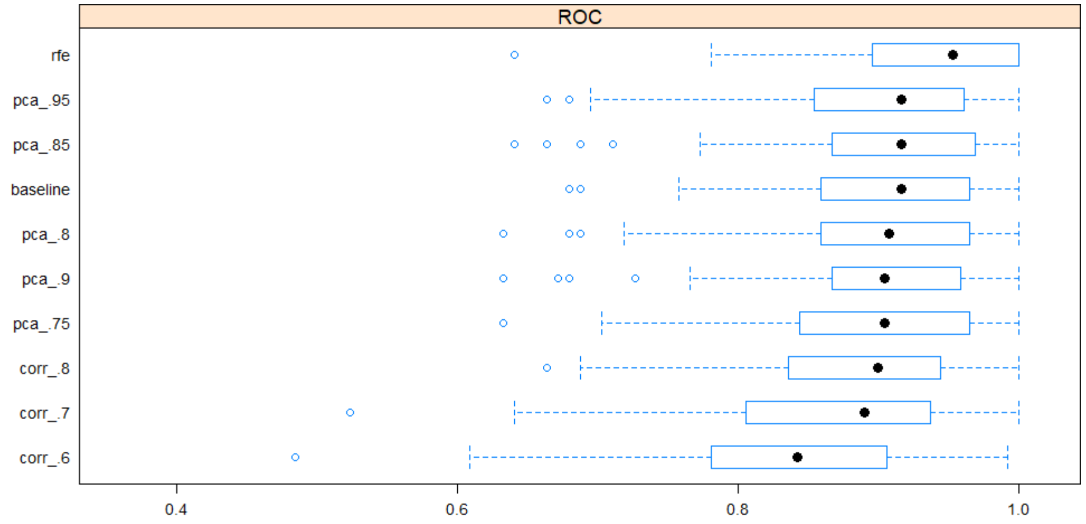

Figure 5: k-NN results

Model	Optimal k	# Variables
Baseline	23	429
RFE	21	10
Correlation Filter 0.6	23	56
Correlation Filter 0.7	13	102
Correlation Filter 0.8	21	188
PCA 0.75	11	9
PCA 0.8	13	15
PCA 0.85	11	25
PCA 0.9	15	41
PCA 0.95	15	70

Table 1 : k-NN optimal k parameters and variable numbers

Example: Task 1 (cont'd)

The final analysis results of all models trained on Task 1's dataset are summarized in Table 2.

Model	The best Feature selection method	The number of predictors/principal components used	Optimal model parameters	Area under the ROC	MCC
Logistic Regression	PCA with threshold = 0.8	15	No parameters	0.9531250	0.7638889
Linear Discriminant Analysis	PCA with threshold = 0.9	41	No parameters	0.9557292	0.7692428
Quadratic Discriminant Analysis	PCA with threshold = 0.75	9	No parameters	0.9218750	0.7692428
K-Nearest Neighbors	Recursive Feature Elimination	10	k = 21	0.9531250	0.7745967
SVM with Linear Kernel	***No Feature Selection (baseline)***	***429***	***C = 1***	***0.9722222***	***0.7745967***
SVM with Radial Basis Function Kernel	No Feature Selection (baseline)	429	Sigma = 0.001794168 , C = 2	0.9583333	0.7638889
Random Forest	Recursive Feature Elimination	10	mtry = 2	0.9414062	0.7789731

- It is seen from the results that the model that achieved the best Area under the ROC is ***Support Vector Machine*** with a ***linear kernel*** that uses ***all the predictors*** in the dataset.

- We can notice from the results that, unlike SVM, simpler methods such us ***k-NN*** and ***Logistic regression*** work better with a ***small number of predictors***. In the case of these models, the least complex models gave the best results. It can also be seen that ***PCA*** and ***RFE*** perform better than removing correlated predictors for feature selection and dimensionality reduction.

- Finally, We split the dataset to 150 training samples and 14 samples for validation. We test the ***accuracy*** of the best model (linear SVM) using ***all the samples in the training set*** from one hand and ***after removing outliers*** from another. The results show that in Task 1, removing outliers does not change the results of the model. Therefore, they will not be removed.

- The best model, SVM with Linear kernel is then retrained using the ***whole training set*** and used to predict the classes for the test set.

RESULTS FOR TASK 2 AND TASK 3

Task	Model	The best Feature selection method	The number of predictors	Optimal model parameters	AUC/ROC	MCC
-2-	Logistic Regression	PCA with threshold = 0.85	12	None	0.8055556	0.4166667
AD	Linear Discriminant Analysis	PCA with threshold = 0.90	18	None	0.8040123	0.4084912
	Quadratic Discriminant Analysis	PCA with threshold = 0.75	6	None	0.7638889	0.4260064
VS	K-Nearest Neighbors	No feature selection (Baseline)	63	K=15	0.7916667	0.5093840
MCI	SVM with Linear Kernel	PCA with threshold = 0.85	12	C = 1	0.8055556	0.4366100
	SVM with RBF Kernel	**PCA with threshold = 0.85**	**12**	**Sigma = 0.05678335, C = 0.25**	**0.8194444**	**0.4166667**
	Random Forest	Correlation Filter with th = 0.80	28	mtry = 19	0.7847222	0.4084912
-3-	Logistic Regression	PCA with threshold = 0.85	16	None	0.8888889	0.6017536
MCI	Linear Discriminant Analysis	Recursive Feature Elimination	1	None	0.8750000	0.5493503
	Quadratic Discriminant Analysis	PCA with threshold = 0.85	16	None	0.8472222	0.5277778
VS	K-Nearest Neighbors	Recursive Feature Elimination	5	k = 9	0.8726852	0.5493503
CTL	*SVM with Linear Kernel*	PCA with threshold = 0.85	16	C = 1	0.8750000	0.6042610
	SVM with RBF Kernel	**Recursive Feature Elimination**	**250**	**Sigma = 0.005432391 C = 4**	**0.9027778**	**06527778**
	Random Forest	Recursive Feature Elimination	10	mtry = 5	0.8750000	0.5493503

- We can see from the table that for both tasks **SVM with RBF kernel** gave the best result in terms of AUC/ROC. This model is used to obtain the results for the test set.
- We can see that in Task 2 the MCC is low which reflects the difficulty of separating the two classes AD and MCI and thus deciding whether a patient has Alzheimer's disease, or a mild cognitive impairment based on the predictors provided in the dataset.
- We can also see that in task 2, the curse of dimensionality is less problematic than the other tasks. This is seen in the result of k-NN that usually suffers from this problem, but it performed the best without feature selection in this task.

Thank you!