**Method 1 : Logistic regression with penalty Lasso**

We cannot use directly the Lasso method because it is made for regression problem – but however we can use the logistic regression with the penalty L1 (Lasso penality)

**L1 regularization** adds a penalty term proportional to the absolute values of the coefficients to the model's cost function

Cost Function=Loss Function+α∑∣βi​∣

It shrinks the coefficient to 0 – the non important ones.

So we use this function :

logistic = LogisticRegression(penalty='l1', solver='saga', max\_iter=5000, random\_state=42)

solver saga : it determines the best algorithm to minimize the the cost function

random\_state is a seed

The cost function is

A math equation with a plus and a positive symbol

Description automatically generated

Loss fct for log :

A black text on a white background

Description automatically generated

Loss fct for log + penalty

A math equation with black text

Description automatically generated with medium confidence

C is the inverse of alpha – we want a small c to have a stronger regularixation

Cv = 5 – cross validation

We find c that have the max accuracy with the cross validation test

We use this c to predict the features in the logistic regression function

We evaluate the model with this c

OUTPUT : ACCURACY/CONFUSION MATRIC/NUMBER OF FEATURES SELECTED

**TO IMPROVE : GRIDSEARCH + LOGISTIC REGRESSION PARAMETER + GRAPH WITH THE IMPORTANT GENES**

**TO ADD : ROC CURVE + MAYBE THE DIFFERENCE BETWEEN MSE TRAIN/TEST DATA**

A black background with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

Lasso considered all the variables to be independant

RECALL + F1 + PRECISION

**Method 2 : Random Forest**

1 step

We need to use all the features in order to rank their importance after, so we built a tree with all the features.

2step

We rank the features by importance – we find their indice and we sort them in a descending order

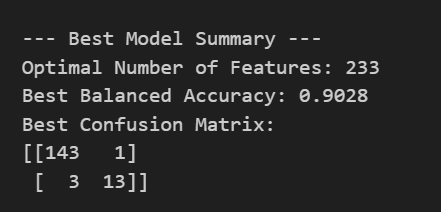
3 step

Now we select the number of features that maximixes the accuracy of our data. We create a loop that add a each step a new feature from the most important to the least and iterate over the entire tree

4 step

We find the best tree

Find the confusion matrix and the number of features



**TO IMPROVE : PARAMETER FOR THE RANDOM FOREST CLASSIFIER + DO WE CREATE A GRAPH WITH THE MPORTANT GENES ?**

**TO ADD : ROC CURVE + MAYBE THE DIFFERENCE BETWEEN MSE TRAIN/TEST DATA**

**Method with elastinet penalty**

**T2.3: Elastic Net Logistic Regression for Feature Selection**

The **elastic net** is a hybrid of LASSO and ridge regression that combines the L1 (sparsity-inducing) and L2 (grouping effect) penalties. For feature selection in high-dimensional data, this method is particularly useful as it can handle correlated features better than LASSO alone.

A math equation with black text

Description automatically generated

α: Mixing parameter (0≤α≤10 \leq \alpha \leq 10≤α≤1):

* α=1\alpha = 1α=1: LASSO regression (L1 penalty).
* α=0\alpha = 0α=0: Ridge regression (L2 penalty).

A math equation with black text

Description automatically generated

1st : we create a regression model with alpha 0,.1 and the penalty equal to 0.5

We need to set a cross validation to find the best alpha and lambda with the grid seacrch cv

It can handle correlated features better than LASSO alone.

It avoids overfitting by including a ridge-like penalty.

Often the genes are correlated between each other’s so this is why it is interesting to take that into consideration

A black background with white text

Description automatically generated

A graph with lines and dots

Description automatically generated

**A black screen with white text

Description automatically generated**

****

**AVEC LE GRAPH**

**A computer screen with white text

Description automatically generated**

**Method nearest Shrunken Centroid :**

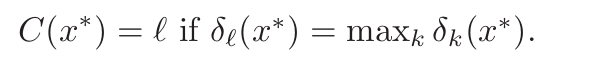
Features are independent – means theh covariance matrix is diagonal - we don’t have enough data to prove their dependency

By selecting only the most relevant features, NSC provides a sparse model

Diagonal covariance LDA rule :

A mathematical equation with black text

Description automatically generated

Classification rule :  


We call the procedure nearest shrunken centroids (NSC). The shrinkage procedure is defined as follows. -- to shrink the coefficient to 0.

1st step A math equations with a white background

Description automatically generated with medium confidence

We prefer the soft thresholding it is a smoother operation and typically works better.

2nd step =

A black text on a white background

Description automatically generated

3rd step =

A black text on a white background

Description automatically generated

Delta need to be determined