Classification of Gene Expression Data

Project #6

Emanuele Alberti

Eugenio Emmolo

Dario Fontanel

Francesca Petrocchi

Data Retrieval

00

NATIONAL CANCER INSTITUTE GENOMIC DATA

#1 – Local Check

#2 – Get UUIDs

- Primary site: **Breast**
- Program: **TGCA** (The Cancer Genome Atlas)
- Project: **TGCA-BRCA**

01

#3 – Download

02

#4 – Data import into memory

- N x M matrix (#N samples ; #M features)
- Sample labels list
- Label mapping {label, numerical value [0,4]}
- Ensembl dictionary

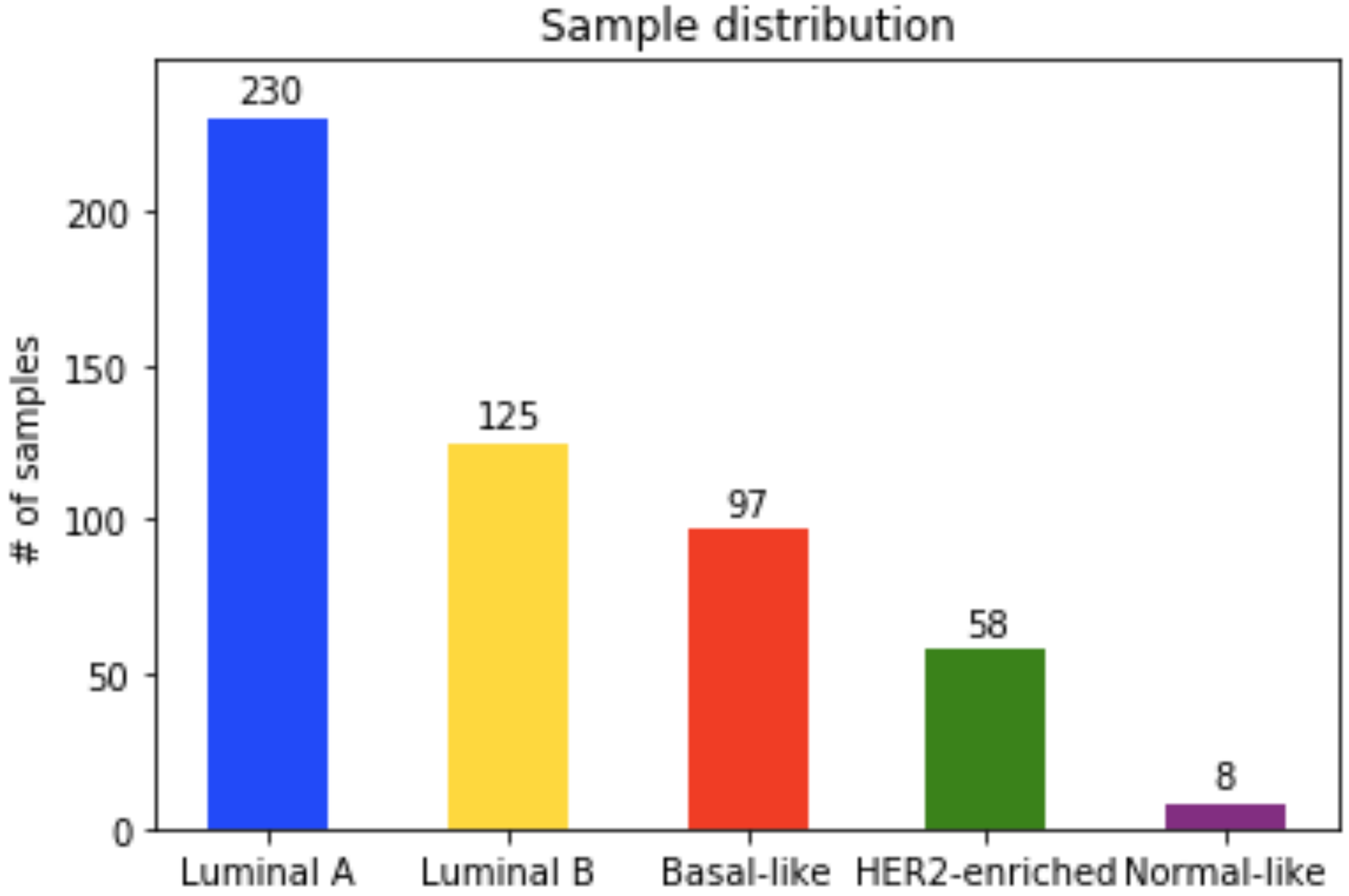
03

#5 – Build Associative Array

04

#6 – Convert Ensembl to HGNC Symbol nomenclature

Sample Distribution



***Normal-Like* CLASS**

Some choices about sample balancing attempt to deal with this outlier class (1.5% of the population).

Our classifiers might not be very good at classifying this class since we have not enough test data to check the goodness of our model.

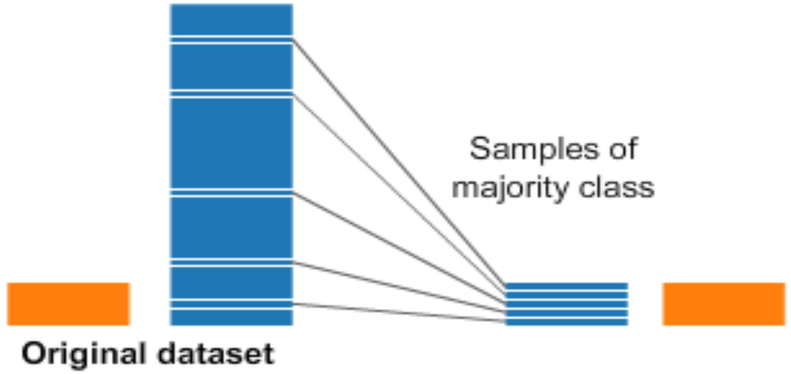
01

02

03

04

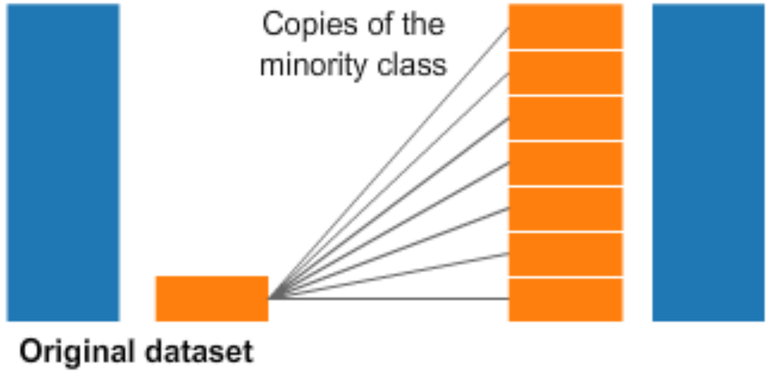
Undersampling



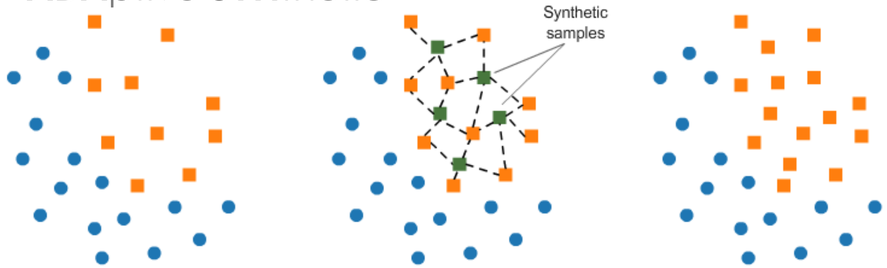
- **CONTROLLED**
  Random undersampling
- **CLEANINGSMOTE**
  Edited **Nearest-Neighbors**
- **TOMEK'S LINK**

Balancing with either of these methods means reducing each class to the *Normal-like* cardinality, being left with a total dataset of 40 samples: definitely a waste of information.

Oversampling

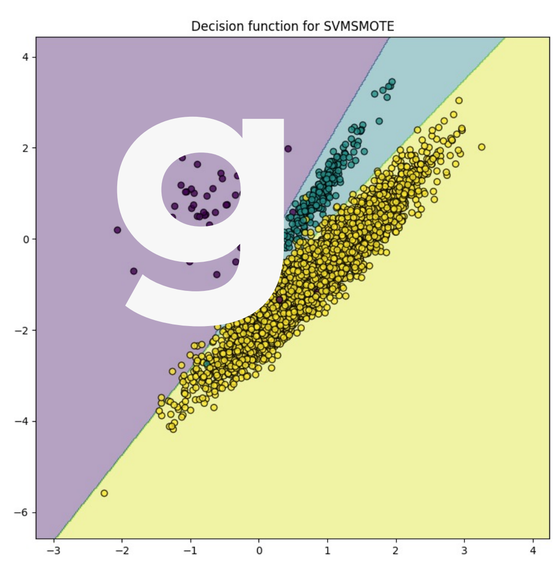


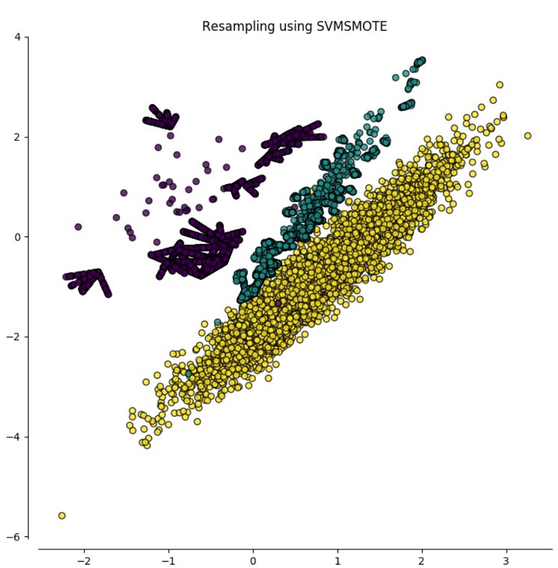
- **RANDOM**
- **SMOTE**
  Synthetic **M**inority **O**versampling **T**ecnique
- **ADASYN**
  ADAptive **SYN**thetic

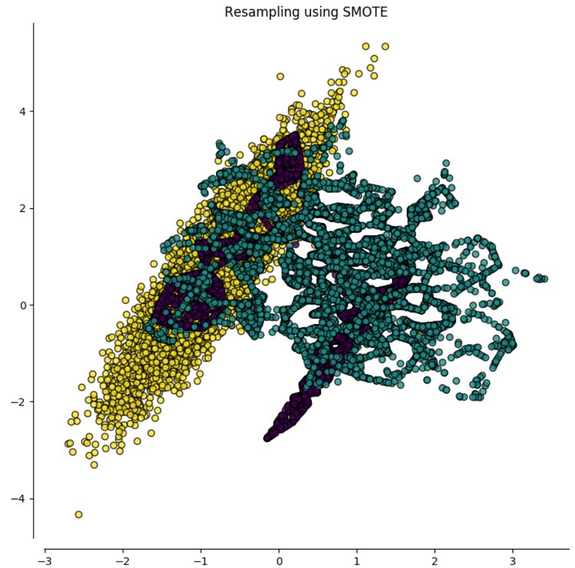


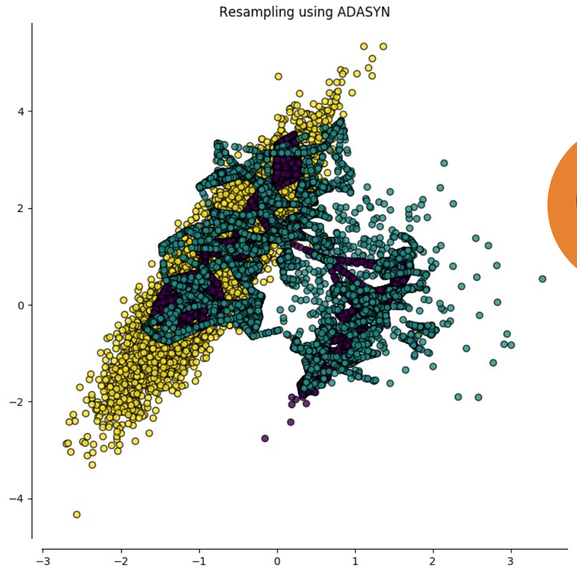
- **SVMSMOTE**
  SMOTE algorithm with the **SVM** variant

Oversampling









Proportional Oversampling

Partially **ignore** the problem by keeping the bias which actually reflects the real life samples **imbalance**, but imposing a minimum number of samples for the minority class.

The user can control the expansion of the **minority class** to reach a specific proportion towards the **majority class**.

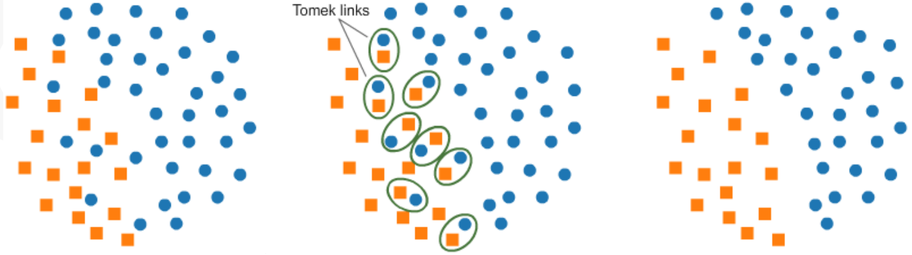
Mixed Approach

SMOTEEN

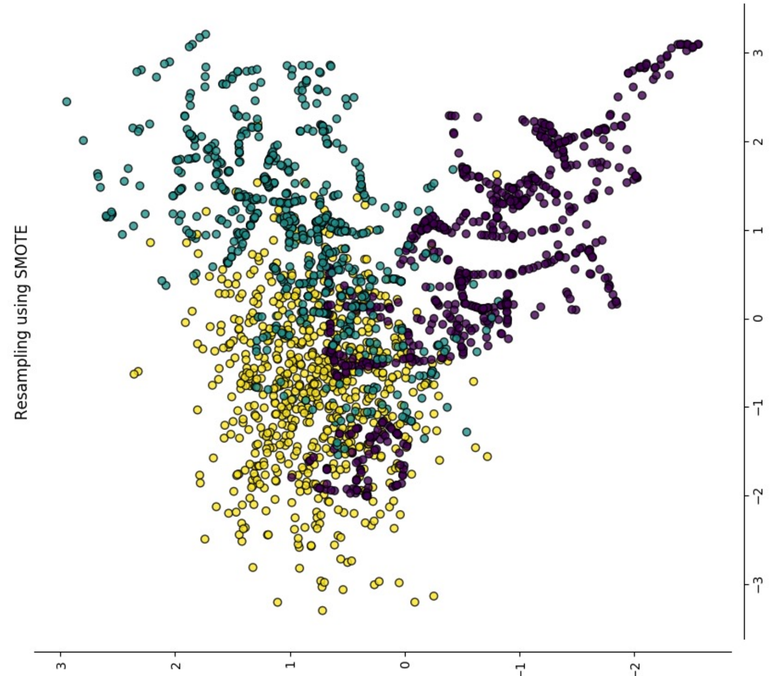
Many noizy samples removed
Real samples removed
Much importance to artificial samples
Information loss

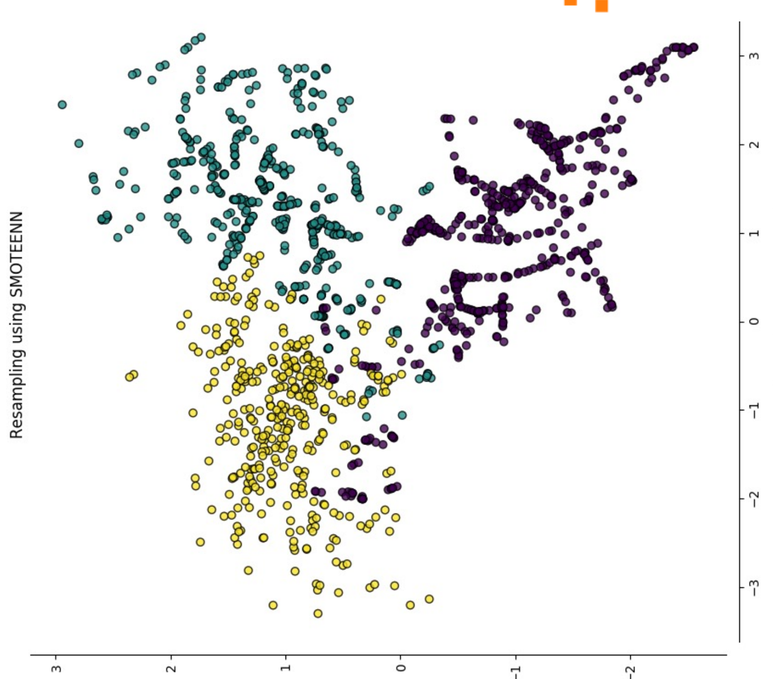
SMOTETomek

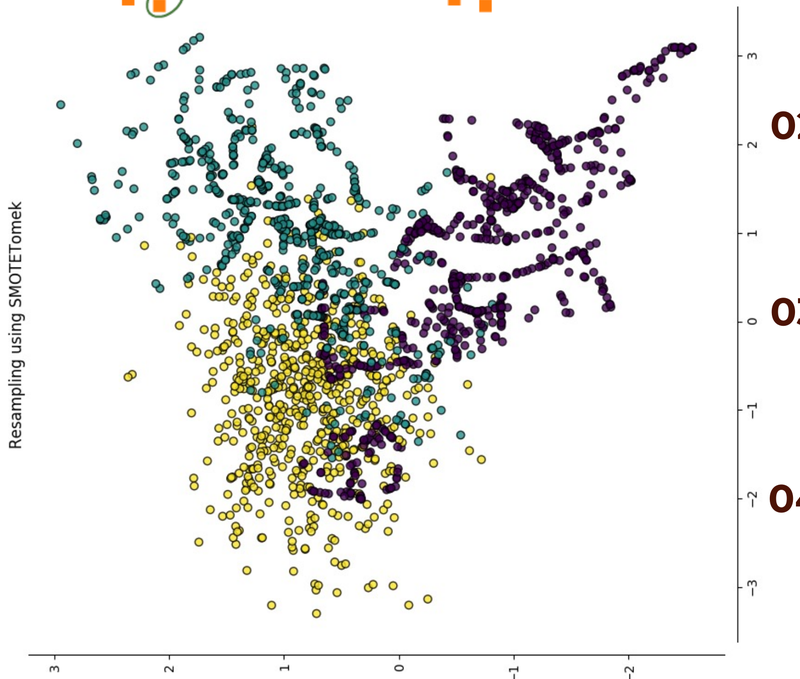
✓ Increase of space between two nearby classes
✓ Easier classification



01





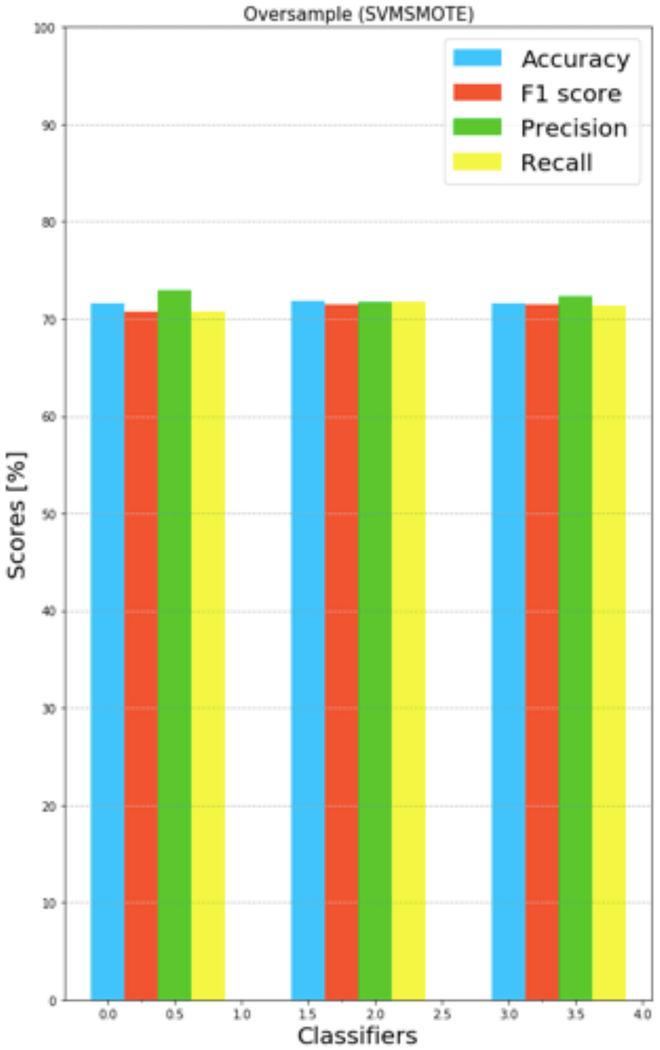


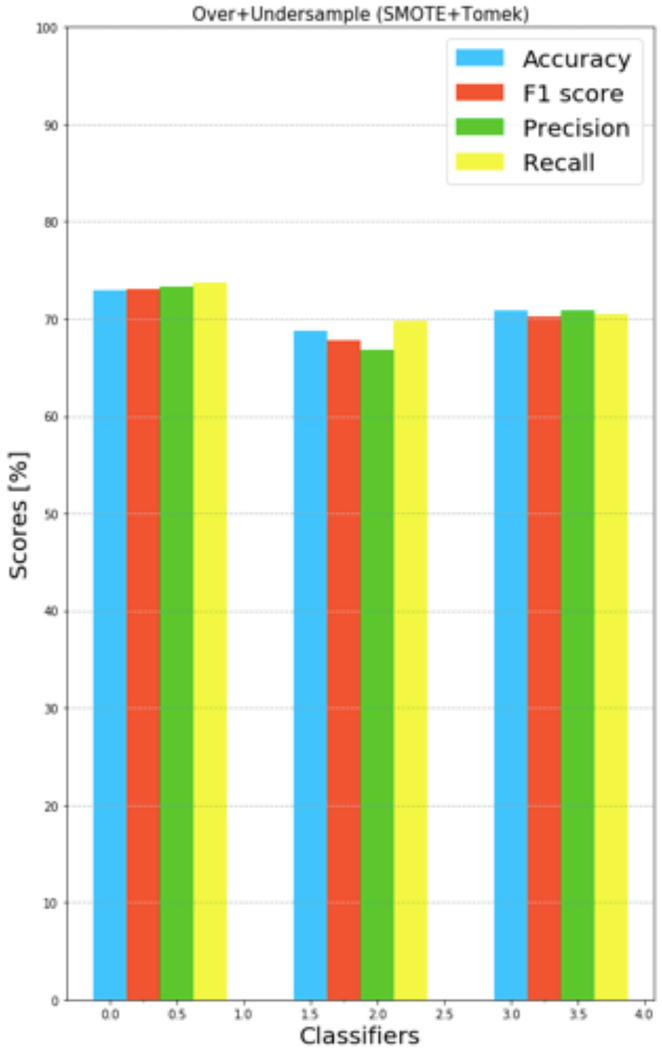
02

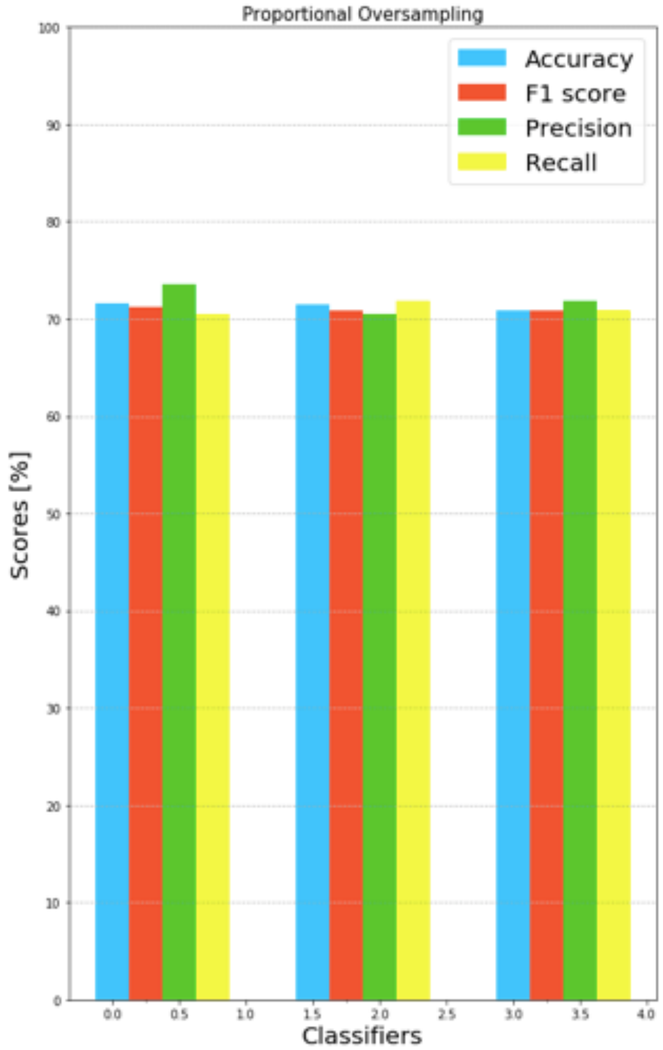
03

04

Performance







Filter Approach

STEP #01

Discard of **useless features**

No significant information that helps in the classification
Program slow down

0,01	Variance Threshold
60.485	Initial number of features
57.680	Number of features after the filter
2.803	Features removed

STEP #02 – OPTION #1

PCA

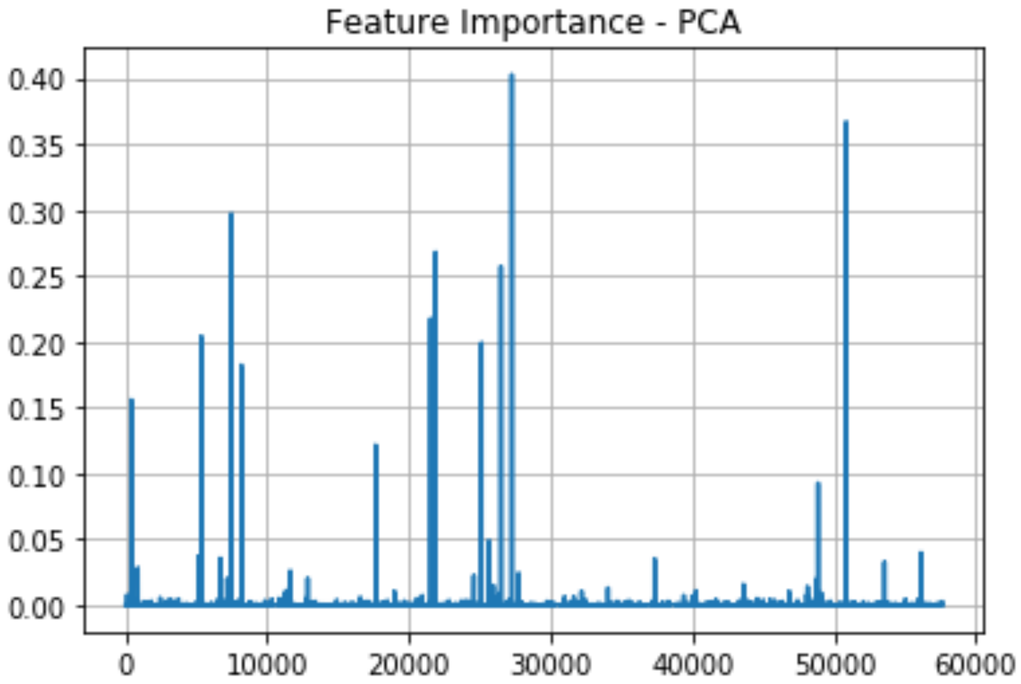
DIMENSIONALITY REDUCTION

Insufficient results
Not suitable for the **practical use** of our study

CUSTOM APPROACH

Features ranking based on a value of importance that represents how much each feature is influential in determining the first 10 Principal Components

Search for the best **threshold** value to reject features below that fixed limit



57.680	Initial number of dimensions
518	Number of dimensions after PCA
92,05 %	Variance Threshold
10	Most important PCs
67	Selected Features

FEATURES 0,12%

STEP #02 – OPTION #2

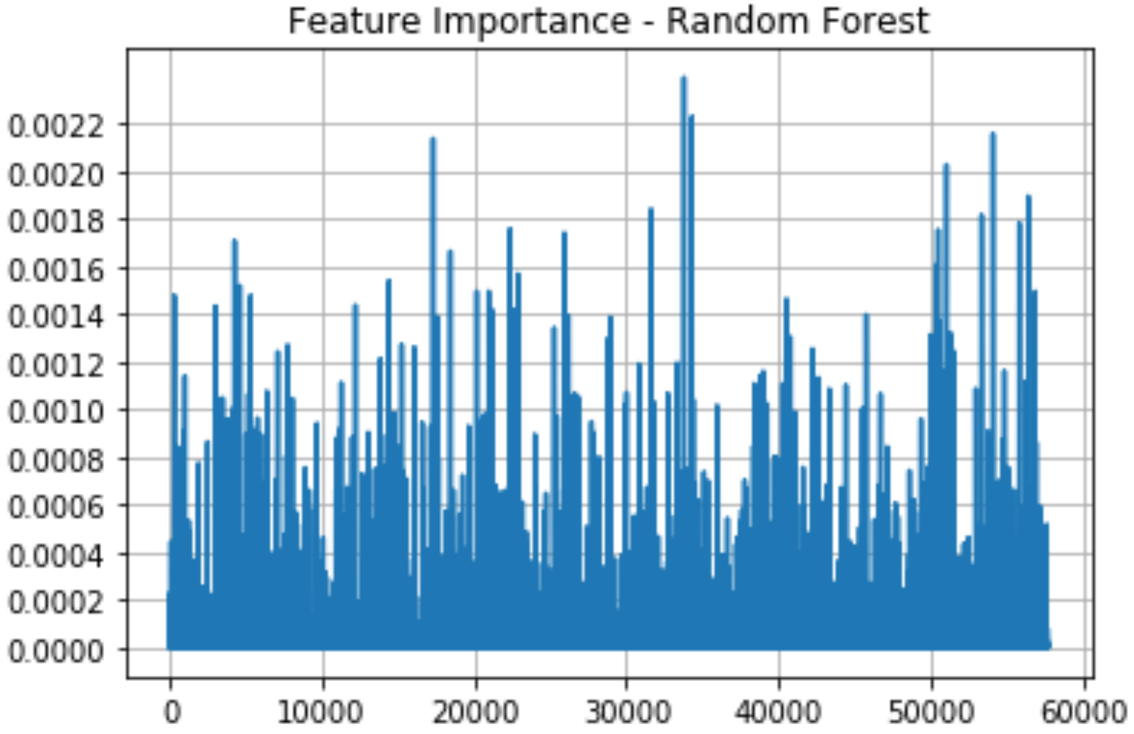
Random Forest

Classification algorithm well suitable for:

- ✓ Multiclass problems
- ✓ Number of features much larger than the number of samples

In order split the data into subsets which most heavily belong to one class, every node in each Decision Tree of the Forest is a condition on a single feature evaluated by means of the **GINI** Index level of impurity.

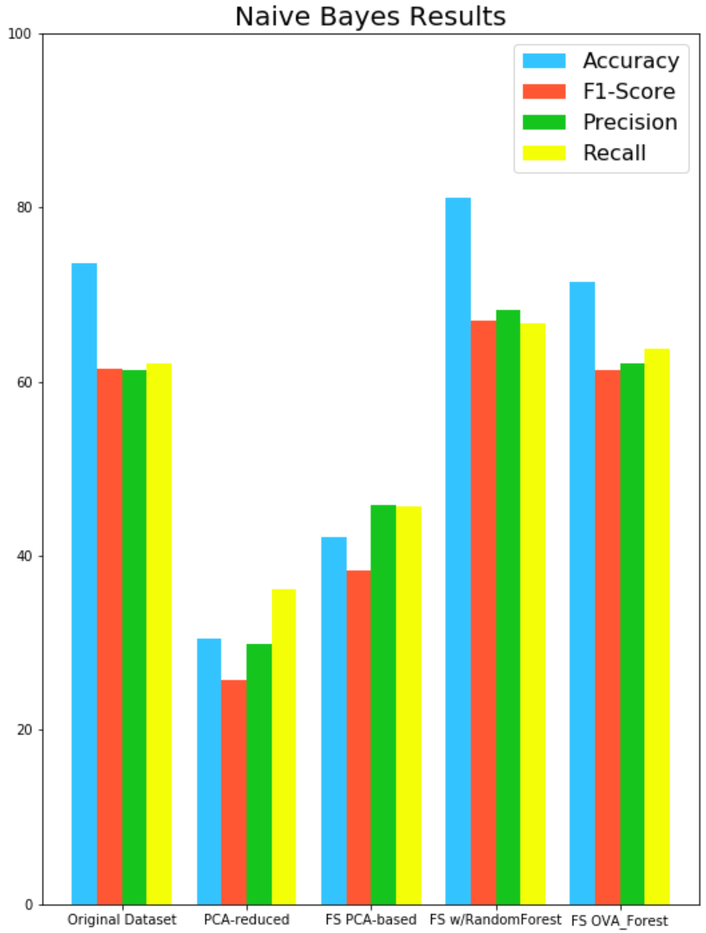
The best **threshold** value to select a subset of features is a trade off between the number of features and the performance.

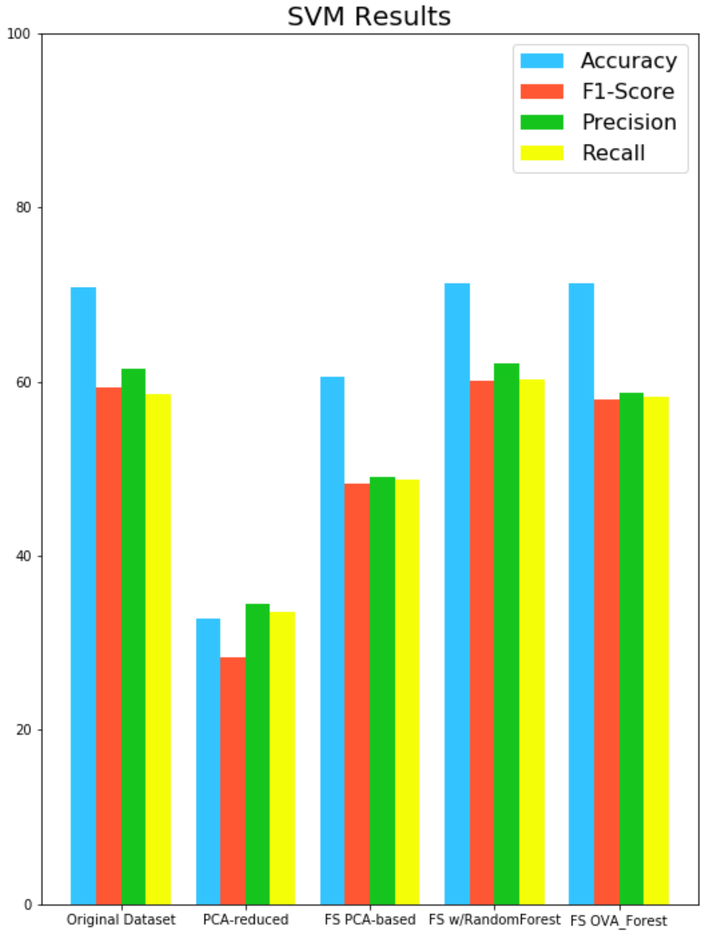


FEATURES

0,2%

Performance





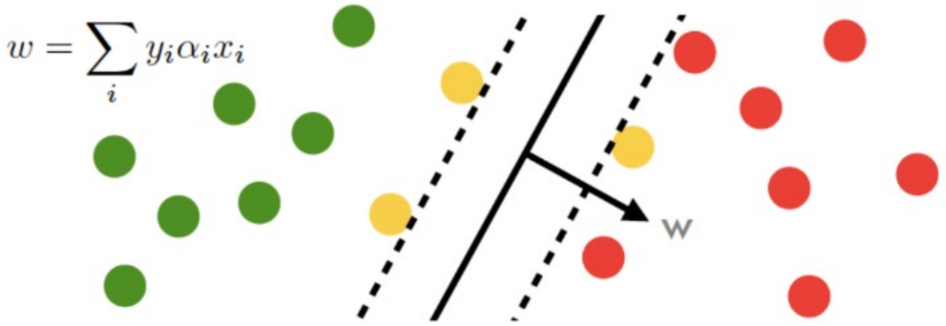
Support Vector Classifier

1. **Dataset Split**
   - 70% **Training** set
   - 30% **Test** set

2. **Oversampling**
   SMOTETomek is applied to the Training set only, leaving test set composed only by real data

3. **Cross Validation**
   Stratified 5-Fold is performed on the amplified Training set

3. **Grid Search**
   To avoid overfitting, the best SVC parameters has been chosen among a range of values
   - **Kernel** { **linear**, poly, rbf, sigmoid }
   - **C** { **0.001**, 0.01, 0.1, 1, 10, 100, 1000 }
   - **Gamma** { 0.001, 0.01, 0.1, 1, 10, 100, 1000 }



SVC Kernel

Performance evaluation on test set

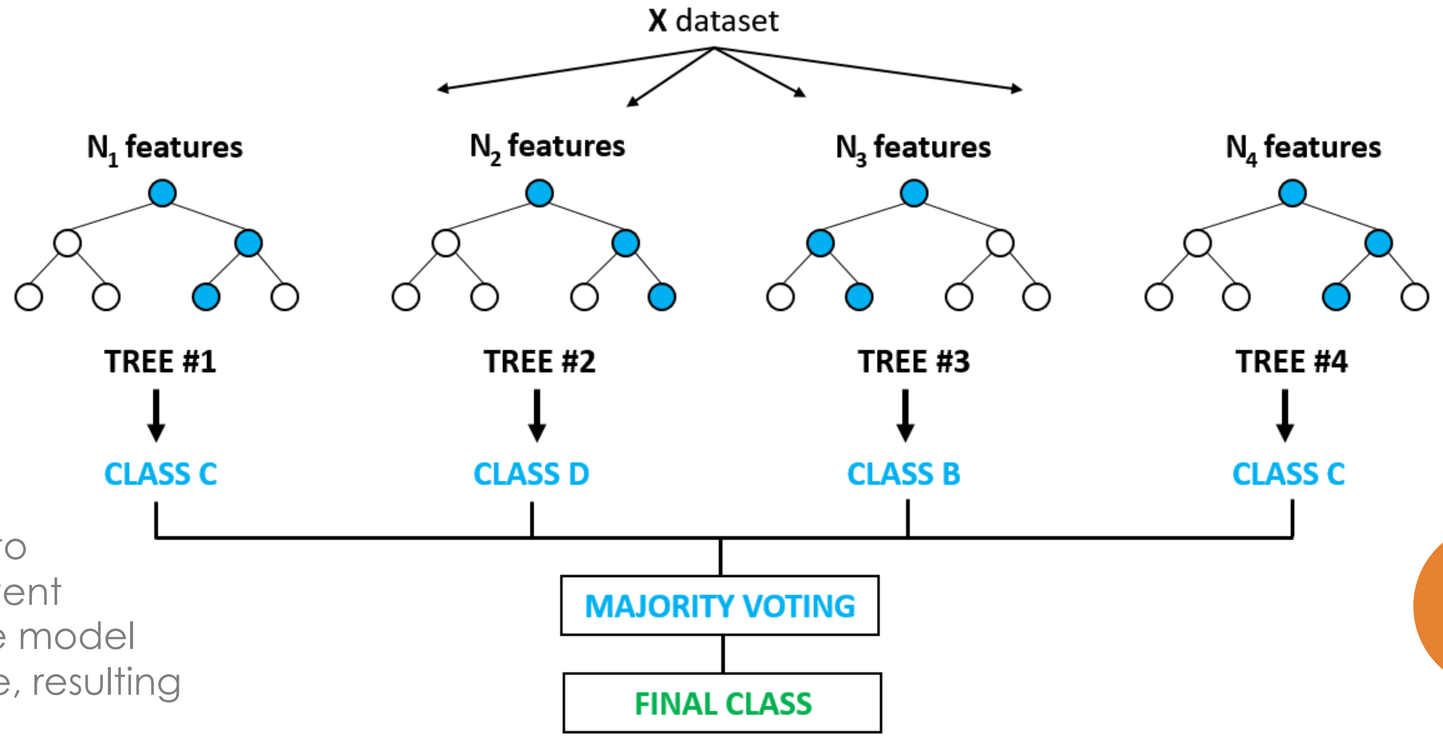
Kernel	linear
C	0.001
Accuracy	0.6730769230769231

Random Forest

The "forest" is an ensemble of **Decision Trees**, features to be evaluated are **randomly** spread among them in order to increase the **diversity** and get more accurate predictions.
Every node split is performed on the locally best attribute.

A higher number of trees increase the computation time but is a way to improve the performance and prevent **overfitting**: while the variance of the model decreases, the bias doesn't change, resulting in more stable predictions.

Random Forest shows excellent performance even when most predictive variables are **noise**.



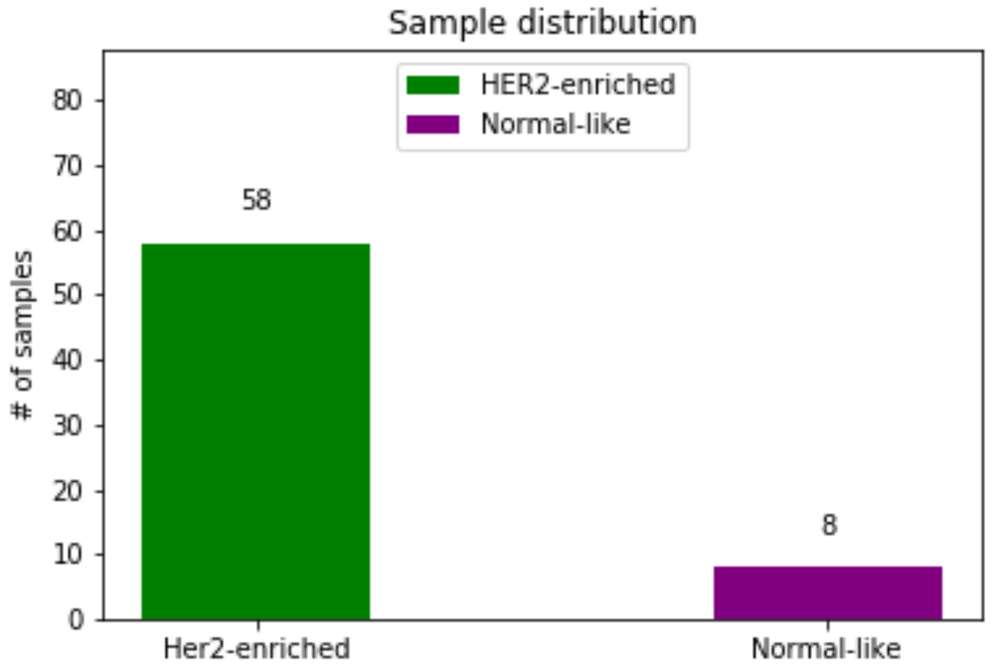
One vs. All Tree

This model comes from the application of **Decision Trees** and **One vs. All SVM**.
The idea is to allow the data to be imbalanced, but classifying using more decision trees in cascade in a specific order.

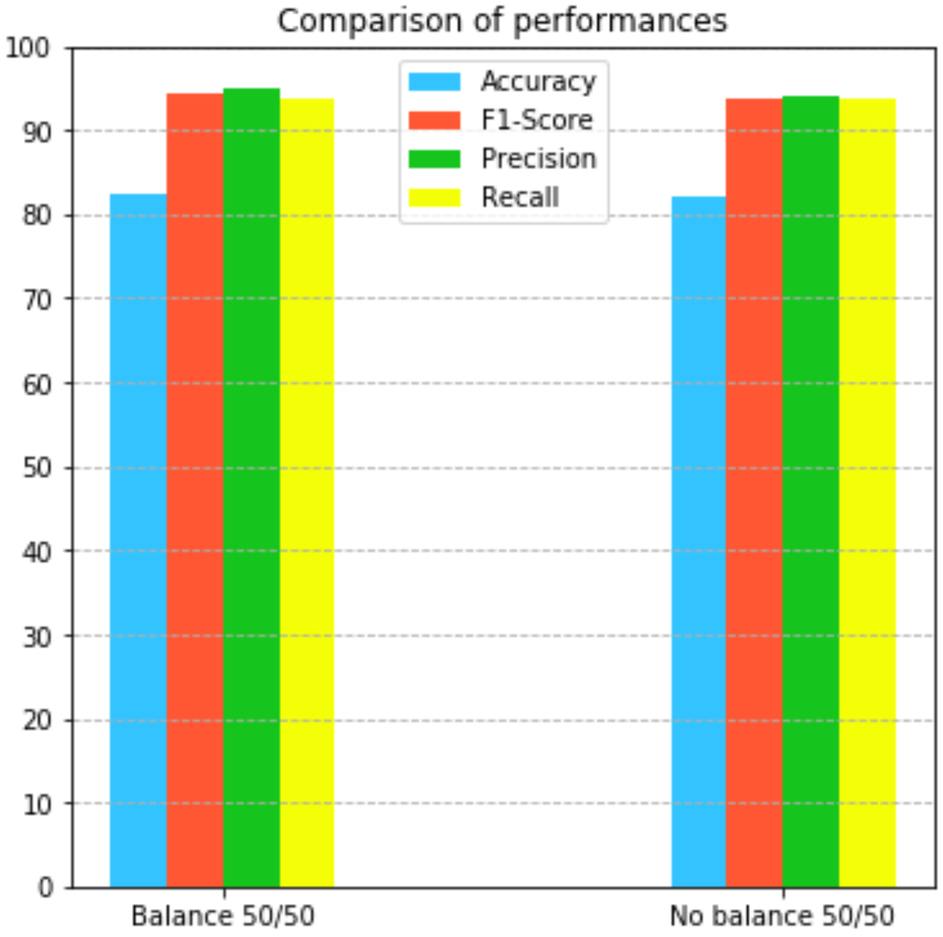
ONE-VS-ALL APPROACH

We create N−1 trees (**N** = # of classes), each of which solves a **binary classification** problem. We ask our model whether the sample belongs to class Y or not, and we keep iterating over the different models until one label for the data is found.

The training of the models leaves an overall **class imbalance**, but in each single binary problem we have a balanced situation. In order to achieve this, we start training our models from the minority classes and we always make sure that the distribution is not heavily in favour of one of the two.



One vs. All Tree



AVERAGE SCORES with 50/50 balancing

ACCURACY 82.43%

	PRECISION	RECALL	F1-SCORE	SUPPORT
Luminal A	87.34%	85.22%	86.13%	46
Basal-like	95.09%	93.84%	94.37%	19
Luminal B	68.82%	77.60%	72.86%	25
HER2-enriched	77.51%	67.12%	71.45%	11
Normal-like	70.00%	50.00%	53.33%	1

AVERAGE SCORES without 50/50 balancing

ACCURACY 82.24%

	PRECISION	RECALL	F1-SCORE	SUPPORT
Luminal A	87.76%	84.35%	85.86%	46
Basal-like	94.04%	93.84%	93.86%	19
Luminal B	70.03%	78.40%	73.83%	25
HER2-enriched	72.76%	74.09%	73.26%	11
Normal-like	0.00%	0.00%	0.00%	1

One vs. All Tree

FEATURE SELECTION

The models that we have trained can be used also for feature selection.

Random Forest combines different groups of features and tell us which attributes have been the most informative. As we did with that feature selection mode, we keep exploiting this information but coming from the different forests that have been created for each binary classification problem.

In fact, with the first model we get the features that allow us to discriminate among *HER2-enriched* and *Normal-like*, then get the best features of the problem *Basal-like* vs. *HER2-enriched + Normal-like*, and so on.

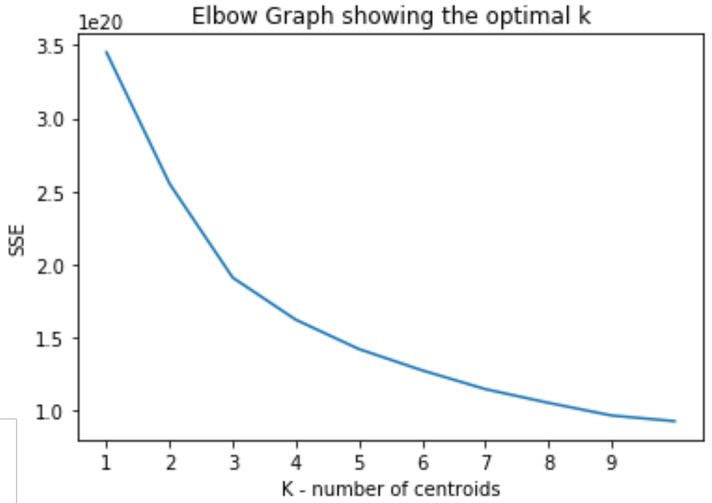
The algorithm that extracts the features removes the duplicates the more it adds features to the best set.

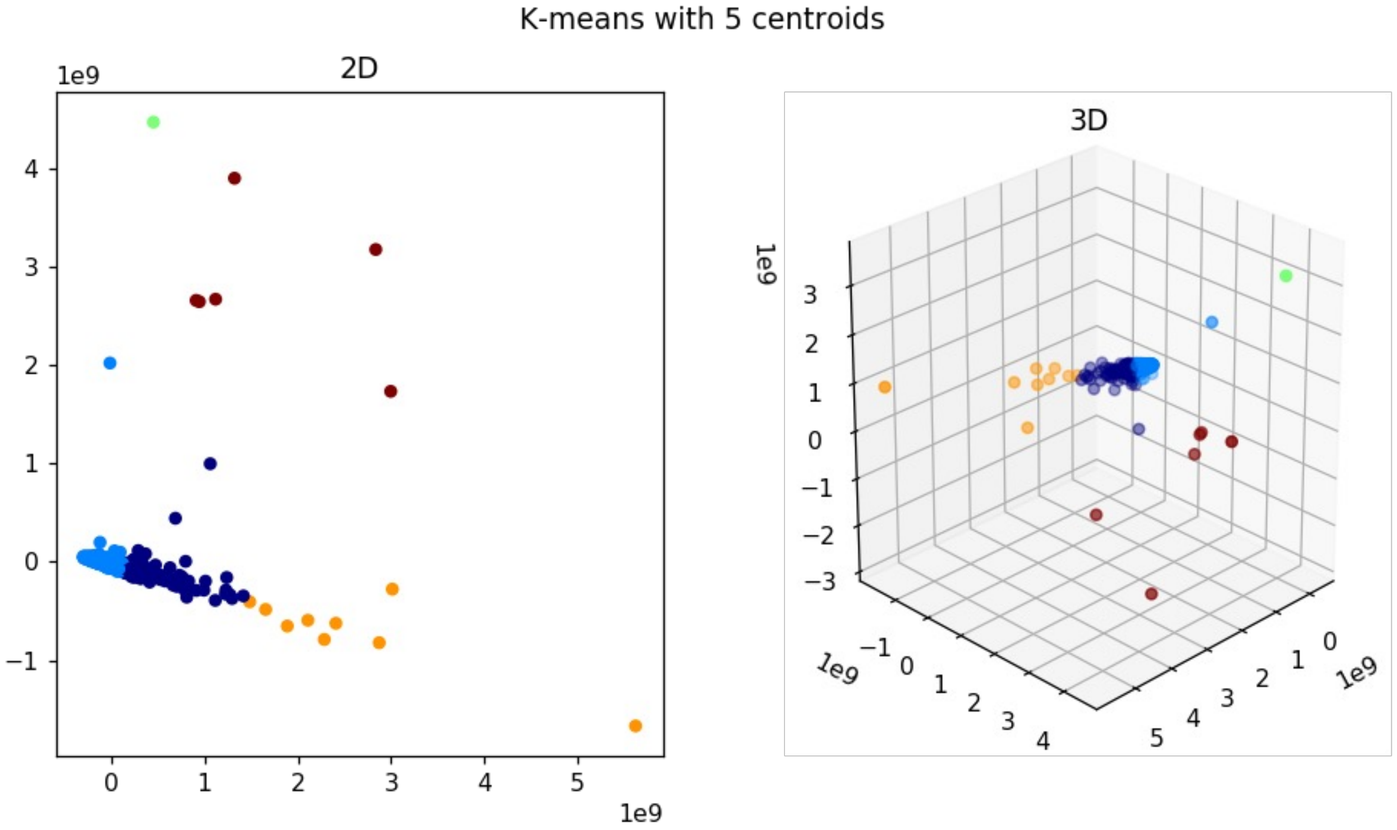
FEATURES

0,13%

K-means

Total intra-cluster variance minimization
Knee Approach to select the best **K** number of clusters to be created





- ✓ Quick convergence
- ✓ Good performance in case of globular shapes
- No guarantee to reach the global optimum
- Choose of the number of clusters to be found

01

02

03

Hierarchical Clustering

STRATEGY

- Agglomerative
- Divisive

MEASURE OF DISSIMILARITY

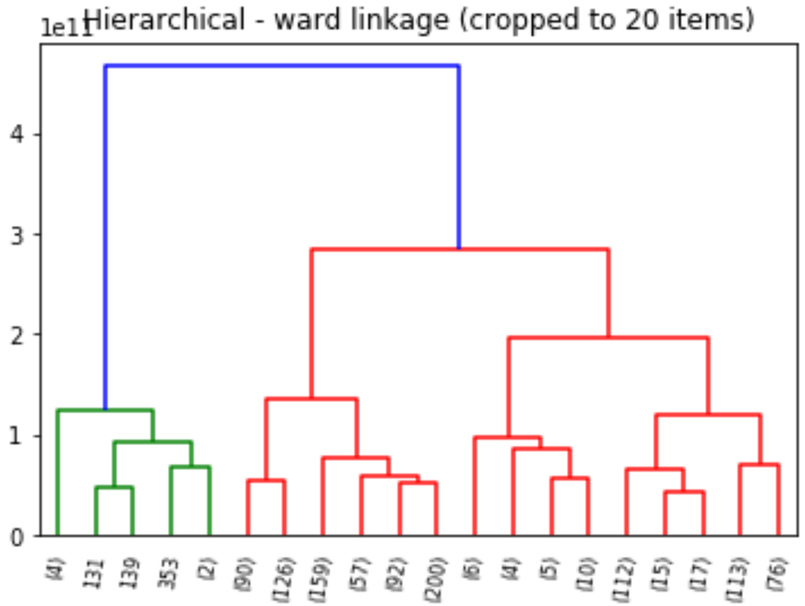
A **Grid Search** has been performed to find the best parameters for our purpose.

◆ AFFINITY [Measure of distance]

- **Euclidean**
- Manhattan
- Cosine

◆ LINKAGE [Dissimilarity of sets]

- **Ward**
- Complete
- Single
- Average



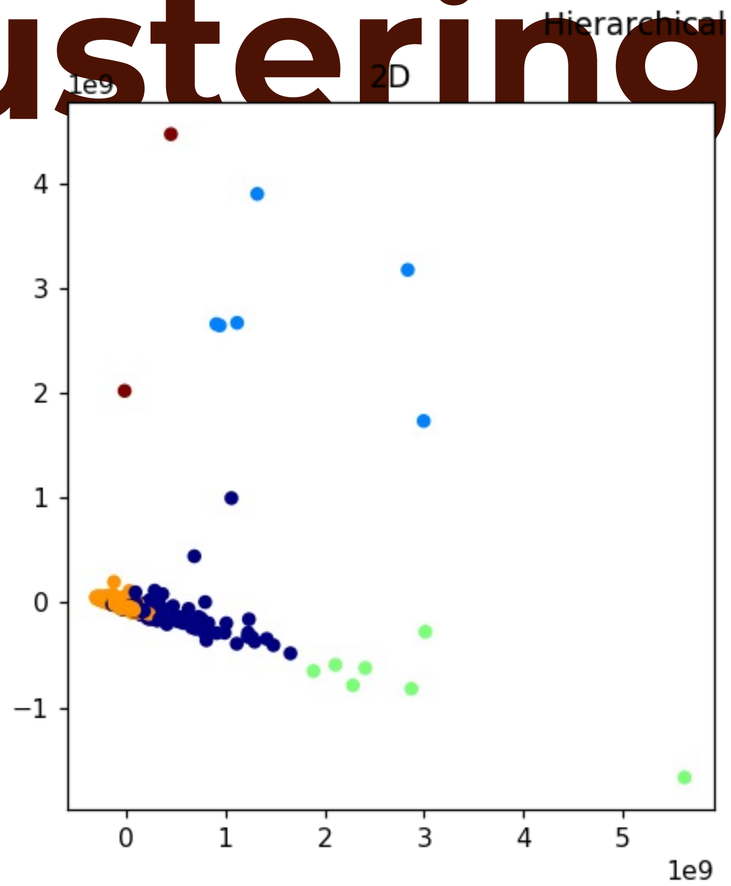
✓ Simple
✓ Informative output structure
At least O(N^2)
Outliers sensitivity

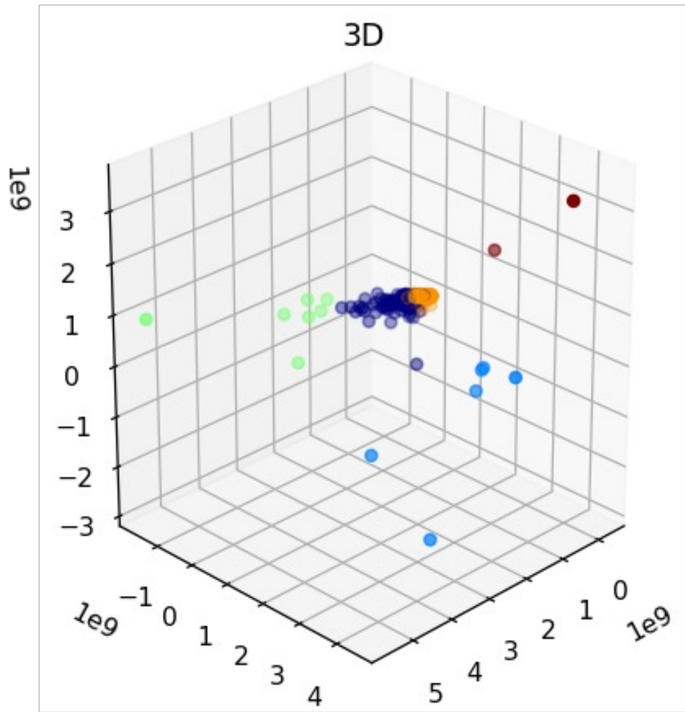
01

02

03

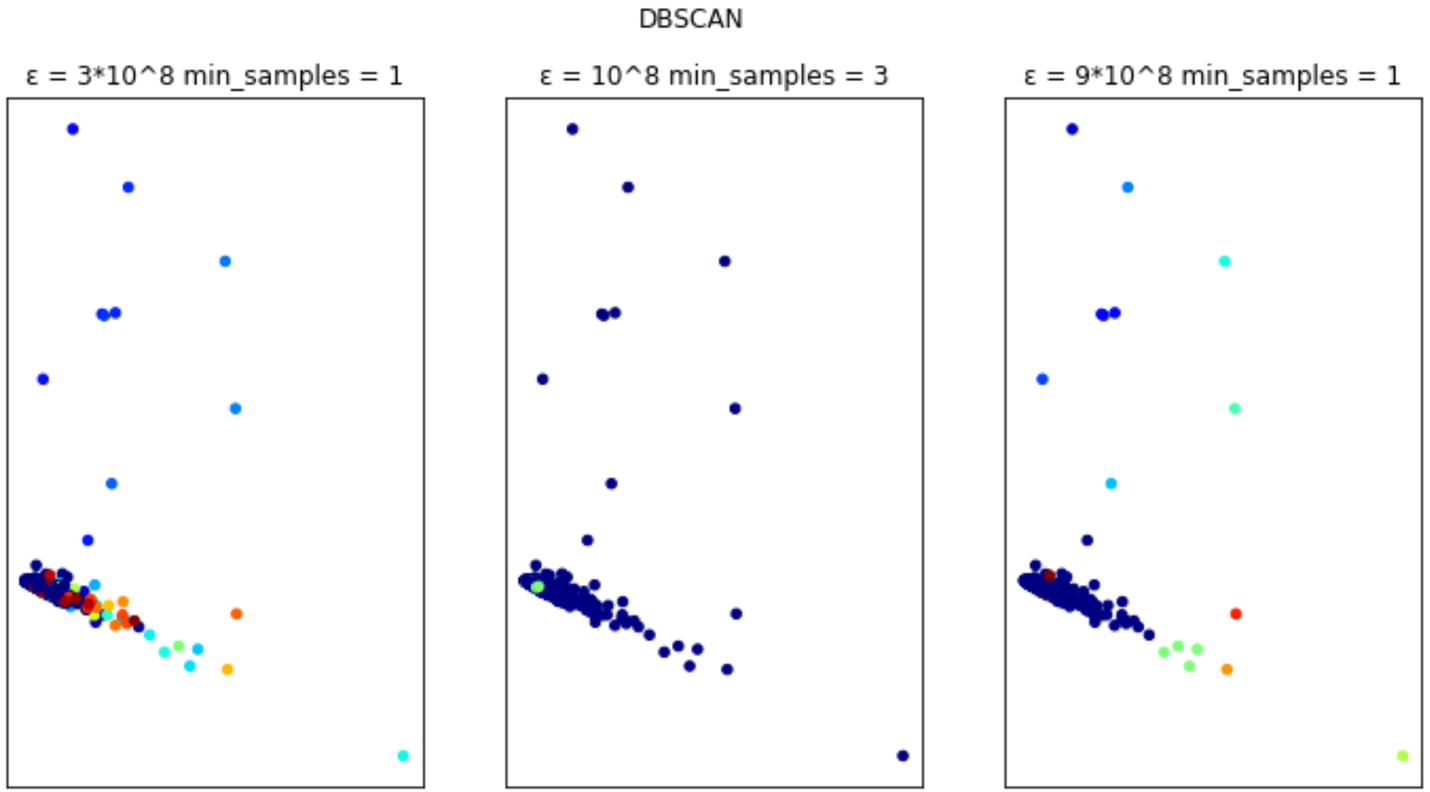
Hierarchical Clustering





DBSCAN

Density-based clustering algorithm.
Euclidean distance has been found to be the best measure of distance between two points
Curse of dimensionality makes hard to find the maximum distance threshold



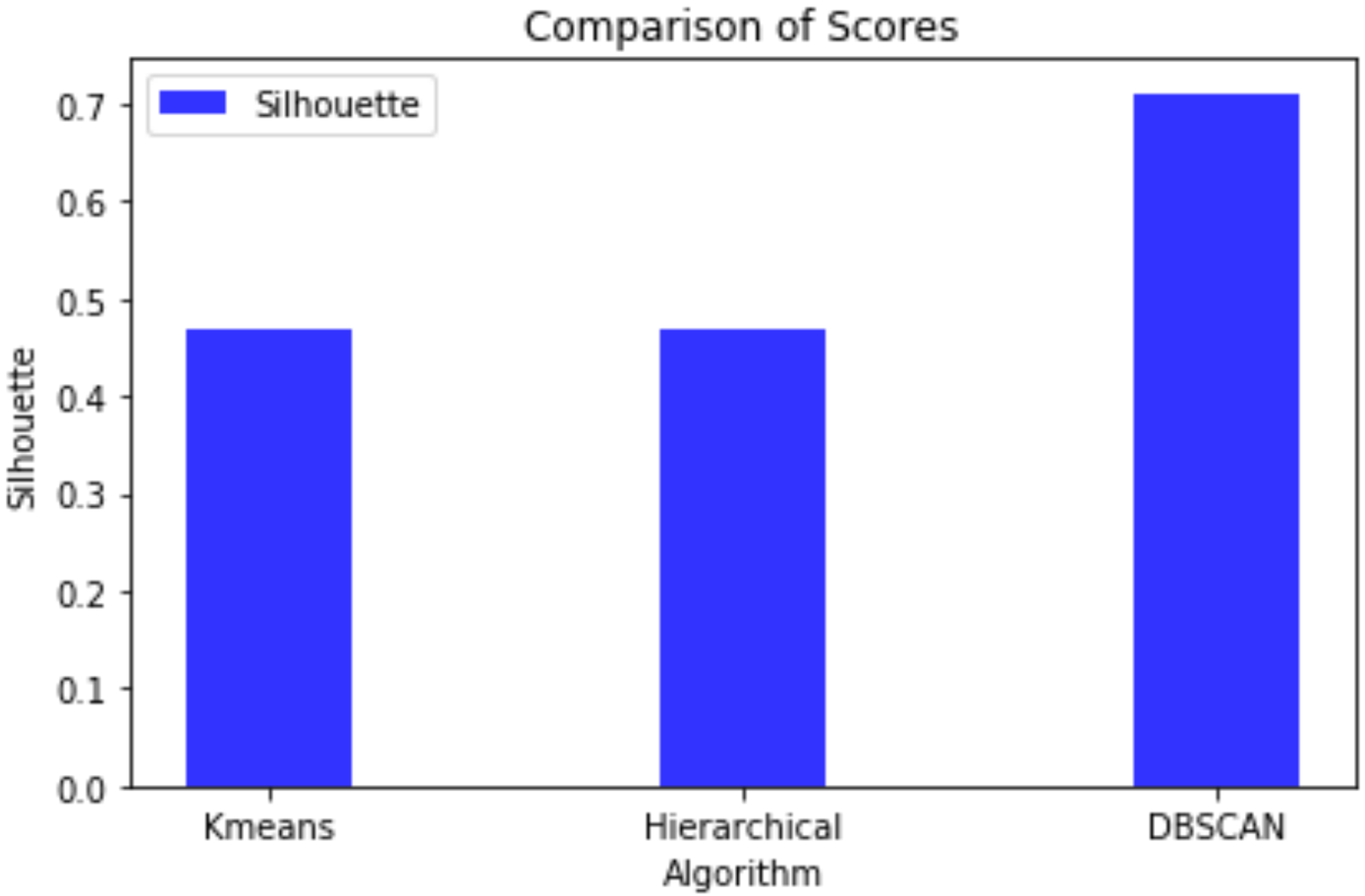
- ✓ No need to pre-set the number of clusters
- ✓ Can find arbitrarily shaped clusters
- ✓ Robust to noise and outliers

01 Not entirely deterministic

02 Quality depends on the Distance Measure used

03 Can't handle datasets with large differences in densities

Performance



K-means

Number of Centroids	5
Silhouette	0.457994144719

01

Hierarchical

Linkage	Ward
Affinity	Euclidean
Silhouette	0.468835582222

02

03

DBSCAN

eps	9*10^8
min_samples	1
Silhouette	0.7111