INTERNATIONAL HELLENIC UNIVERSITY

Breast cancer classification with machine learning methods

Elaboration:
Panitsidis Lazaros

Supervision:
Papadopoulou Foteini

February 2023

Contents

Theoretical introduction

1. Description of the problem
2. Data
3. Algorithms
4. Performance metrics
5. Generalised evaluation
6. Hyper-parameter selection methods

Implementation

1. Data pre-processing
2. Visualization
3. Feature selection
4. Principal component analysis
5. Results
6. Conclusions

Description of the problem

- Real data from biopsies.
- Machine learning makes more accurate predictions than humans.
- The goal is to achieve the best possible result, with the fewest possible computational resources.
- Investigate the effect of reducing the number of features on the performance of the algorithms.

Data

- 30 features
- 569 samples

University of Wisconsin
Date: 1995-11-01
Number of web visits: 1.913.905

1. Radius
2. Perimeter
3. Area
4. Smoothness
5. Compactness
6. Concavity
7. Concave points
8. Symmetry
9. Fractal dimension
10. Texture

For each feature, the following have been calculated:

1. Mean value (mean)
2. Worst (largest) value (worst)
3. Standard error (se)

Algorithms

1. Gaussian Naive Bayes (GNB)
2. Linear Discriminant Analysis (LDA)
3. Quadratic Discriminant Analysis (QDA)
4. Ridge Classifier
5. k-Nearest Neighbors (KNN)
6. Support Vector Machines (SVM)
7. Decision Tree
8. Random Forest
9. Gradient Tree Boosting (LGBM)
10. Adaptive Boosting (AdaBoost)
11. Extreme Gradient Boosting (XGBoost)
12. Stochastic Gradient Descent (SGD)
13. Multi-Layer Perceptron (MLP)

Performance metrics

- TP, True Positive
- TN, True Negative
- FP, False Positive
- FN, False Negative
- P, all positives
- N, all negatives

- Accuracy : $\frac{TP + TN}{Sample\ Size}$
- Precision : $\frac{TP}{TP+FP}$
- Recall : $\frac{TP}{TP + FN} = \frac{TP}{P}$
- F1-score : $2 \cdot \frac{precision \cdot recall}{precision + recall}$

Generalised evaluation

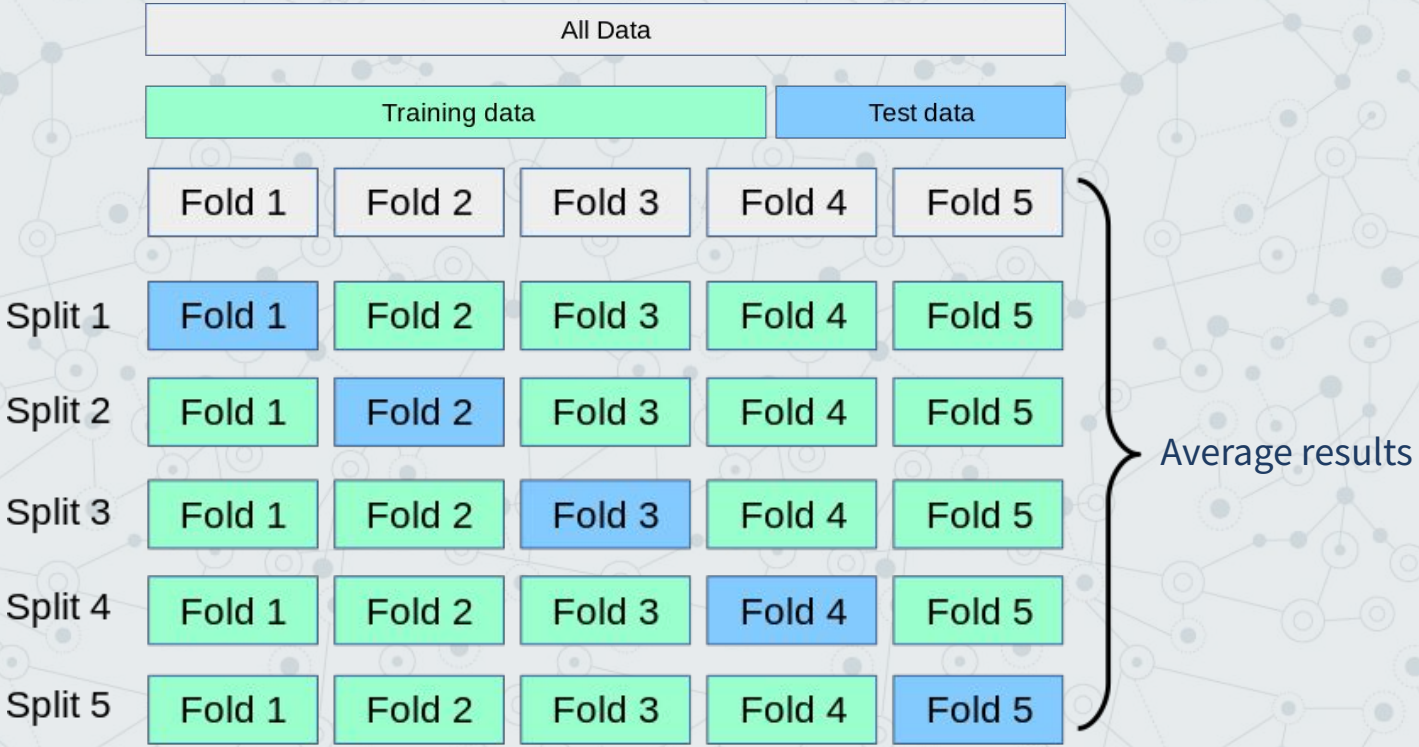
Cross Validation

- No wasted data
- All available data are used for training and testing
- Statistically more reliable assessment of performance

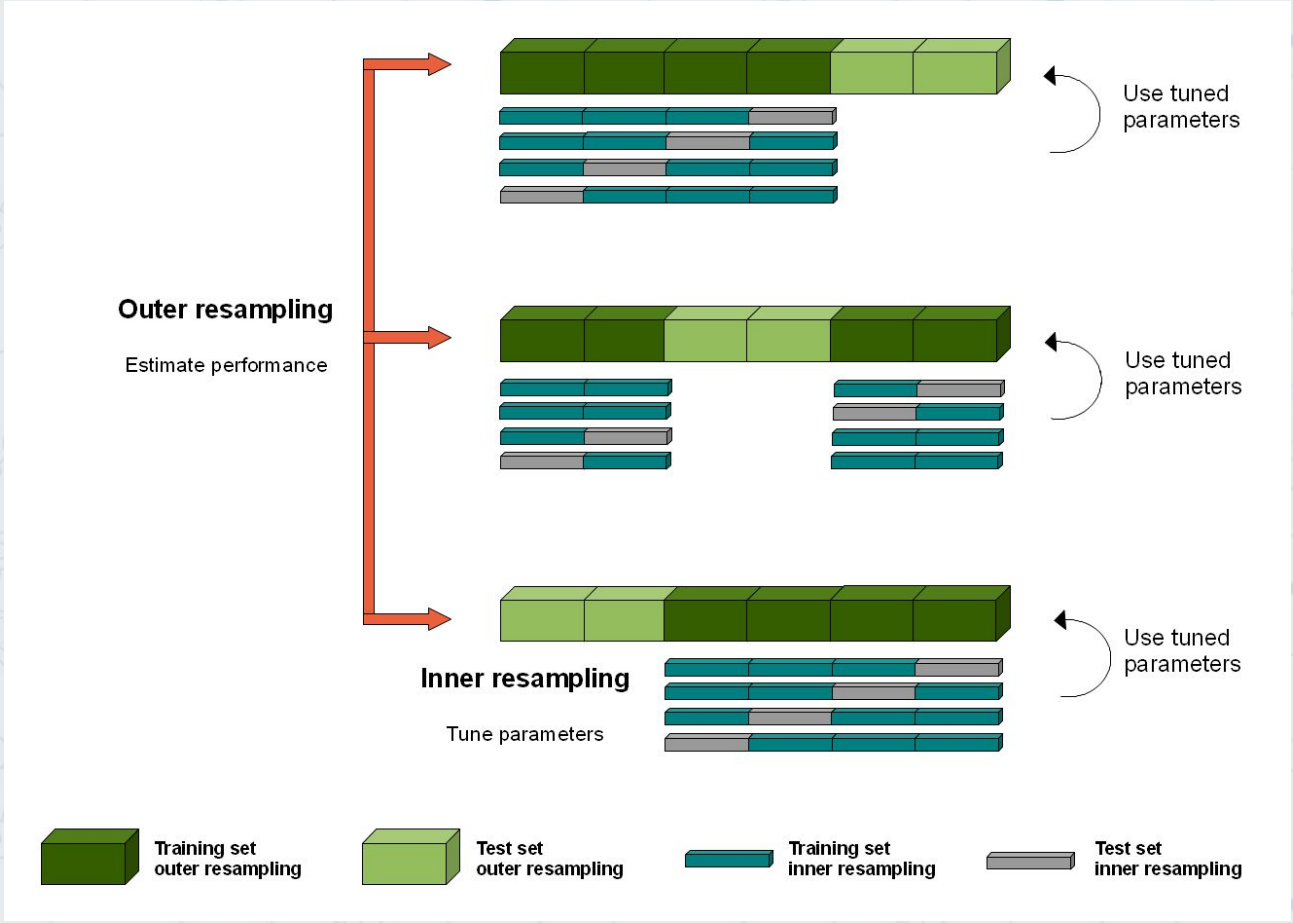
Nested Cross Validation

- Use for hyper-parameter optimization
  - hyper-parameter : tuning settings of an algorithm that affect its performance
- No information leakage in the model
- Avoiding overfitting
  - overfitting : performs well on training data but not on new/unknown data

Cross Validation



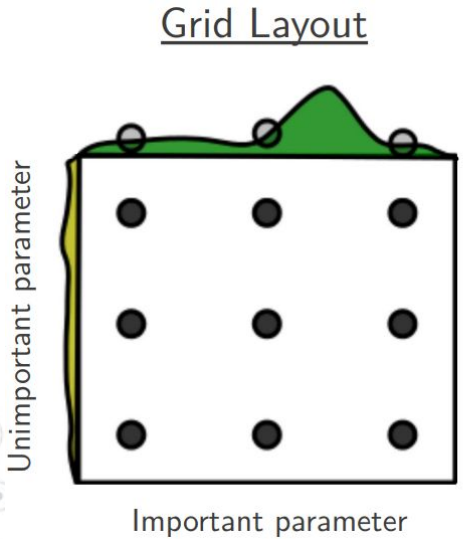
Nested Cross Validation



Hyper-parameter selection methods

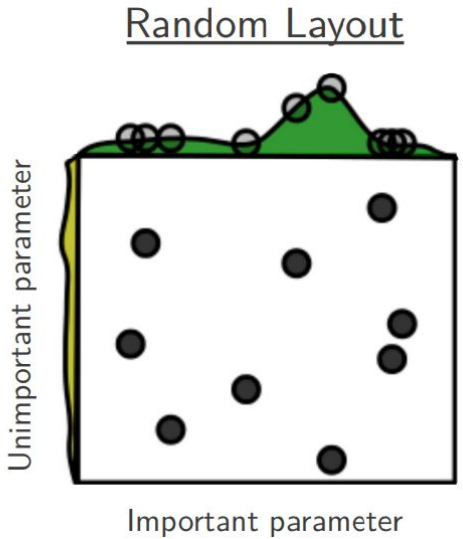
Grid Search

- Exhaustive search in the huper-parameter value grid
- Very time consuming process



Randomized Search

- Random combinations from the hyper-parameter value grid
- Saves time



Implementation

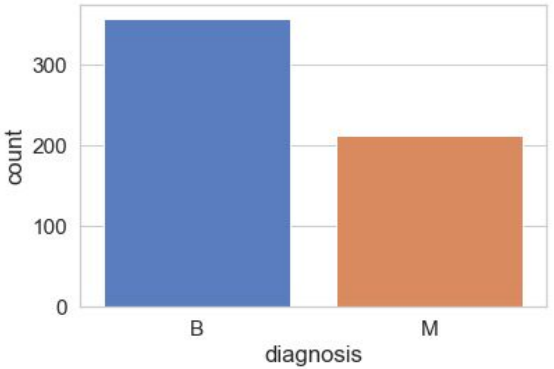
Data pre-processing

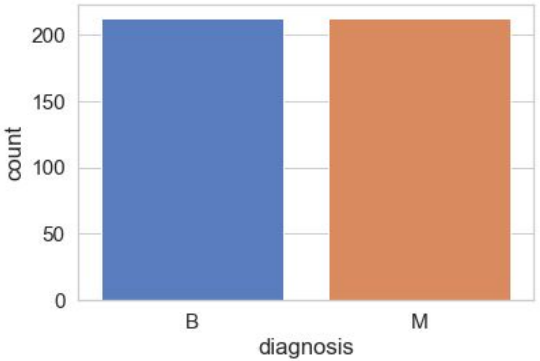
1. Read data

2. “Cleaning” the data
   a. Remove unwanted observations
   b. Correction of structural errors
   c. Management of unwanted ectopic values
   d. Handling missing data

3. Undersampling

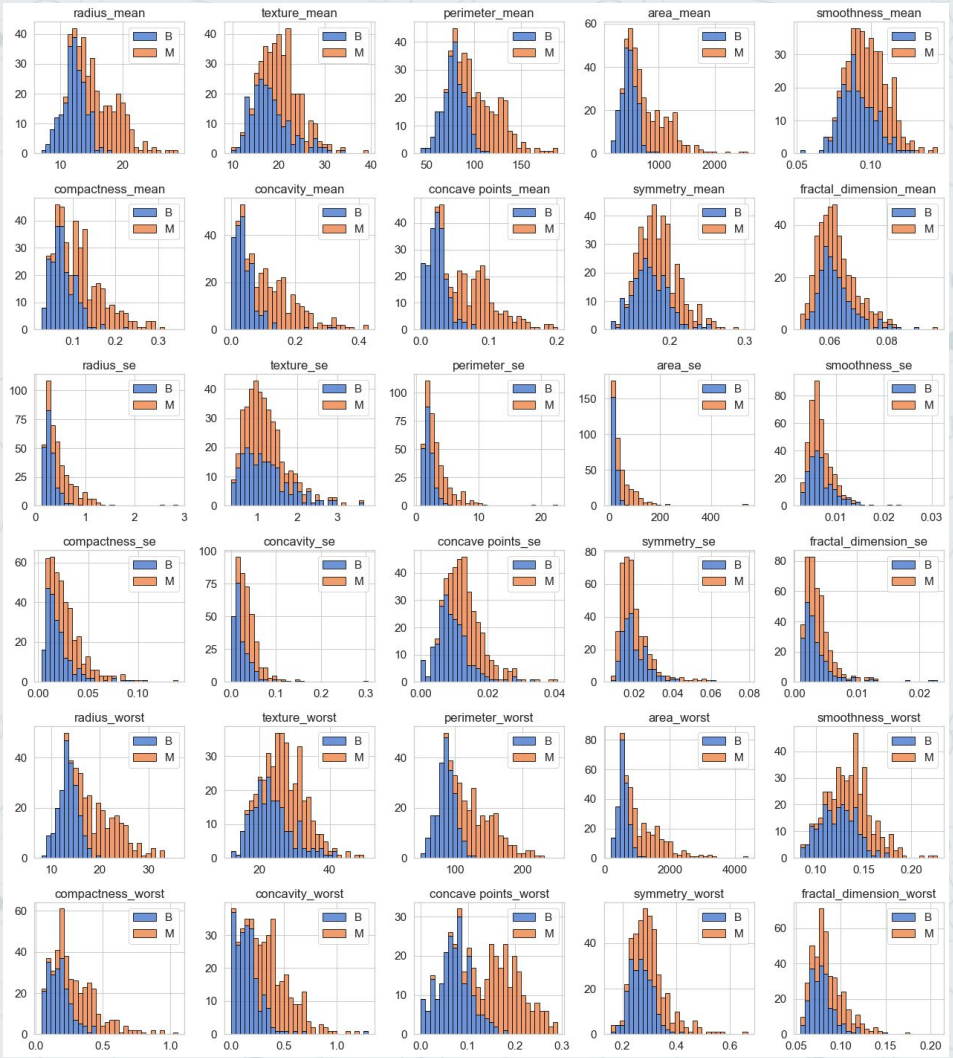
- Benign → Blue color
- Malignant → Orange color



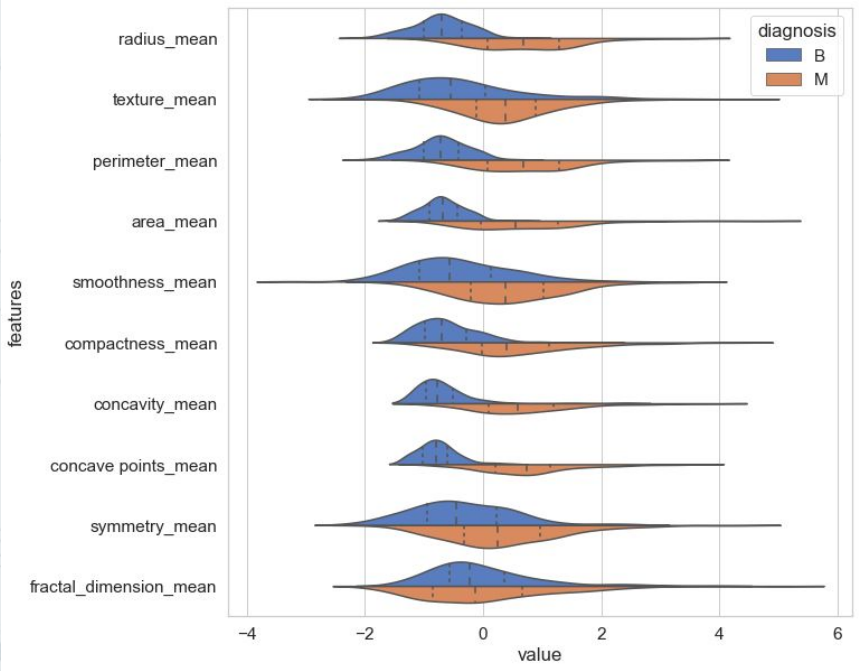


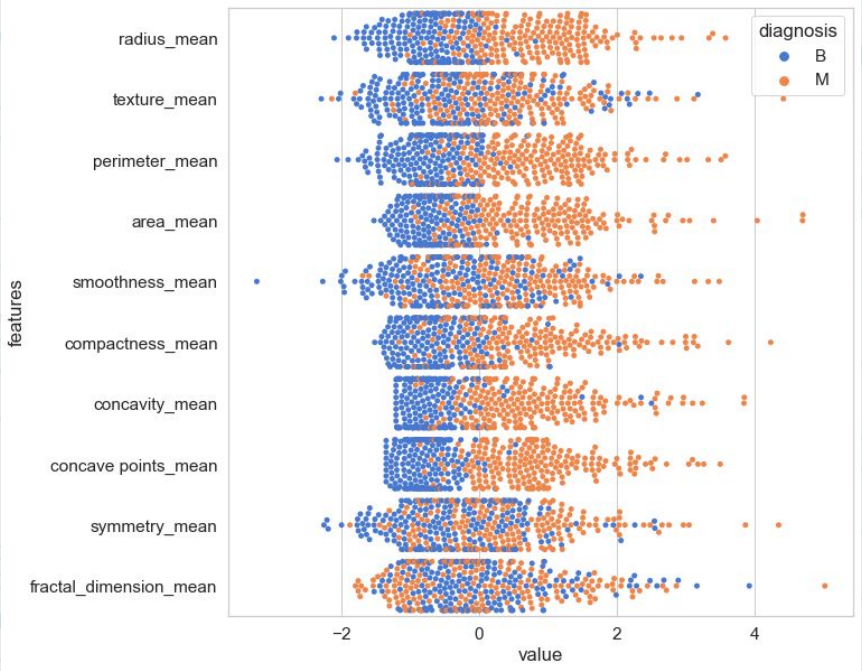
Visualisation

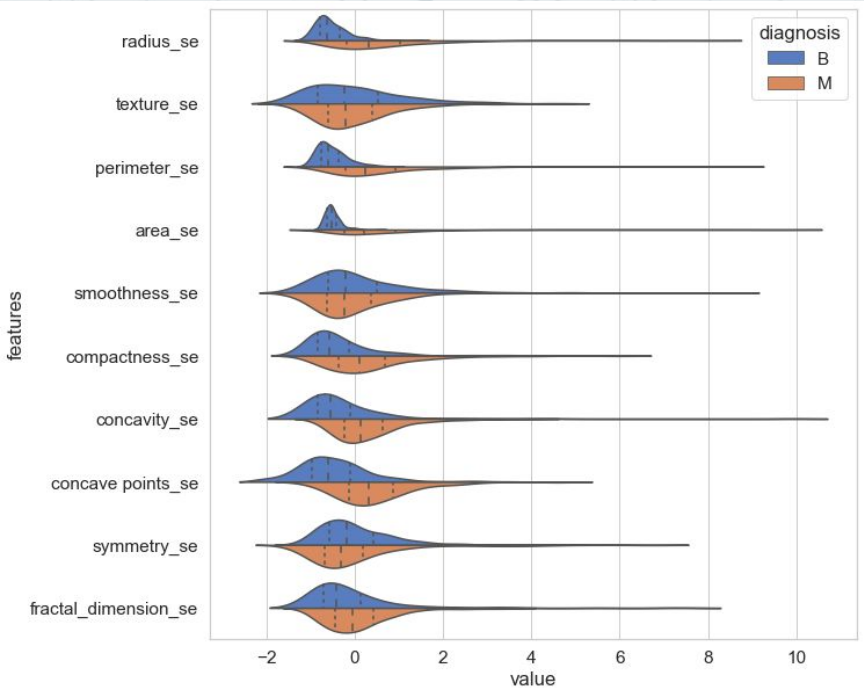
- Separation between benign/malignant cancer in many features
- Malignant samples usually have higher values
- High separability is desirable for effective classification
- An initial assessment can be made for the most indicative features
- Sample values have a large variation between features
- Normalisation has been done to have a mean of 0 and a variance of 1

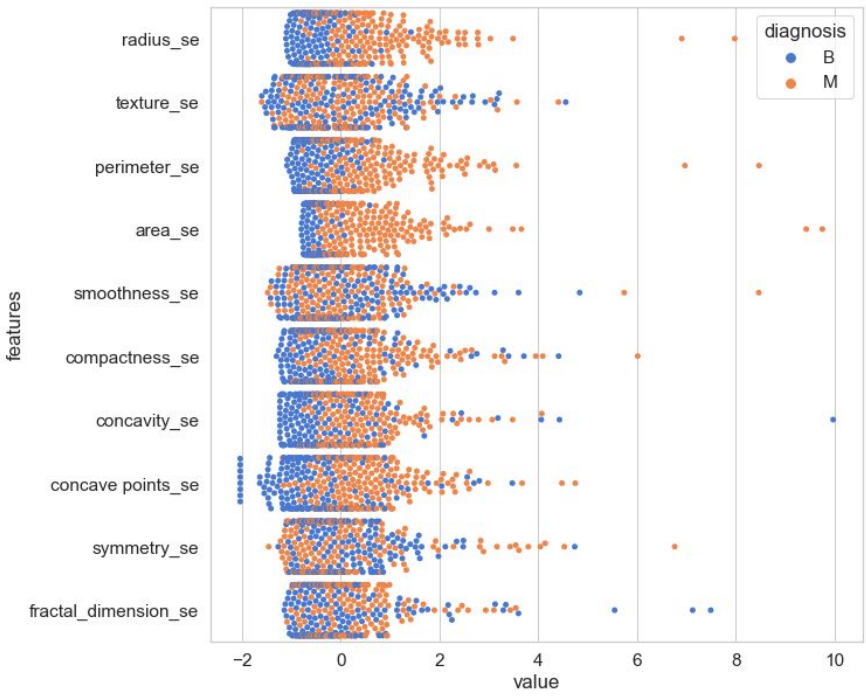


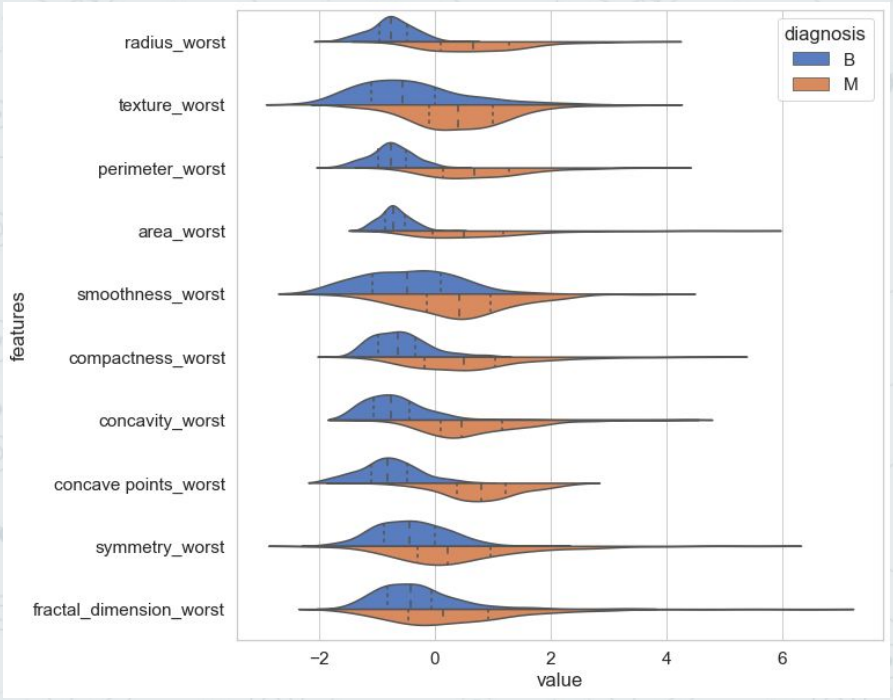
Violin and Swarm plots

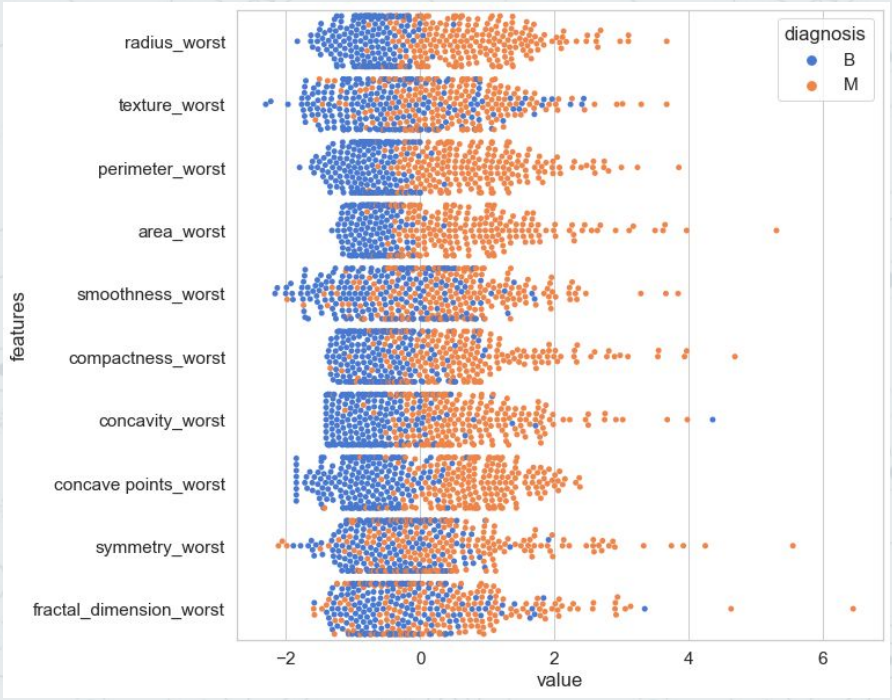





diagnosis
B
M
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
features
−2
0
2
4
6
8
10
value

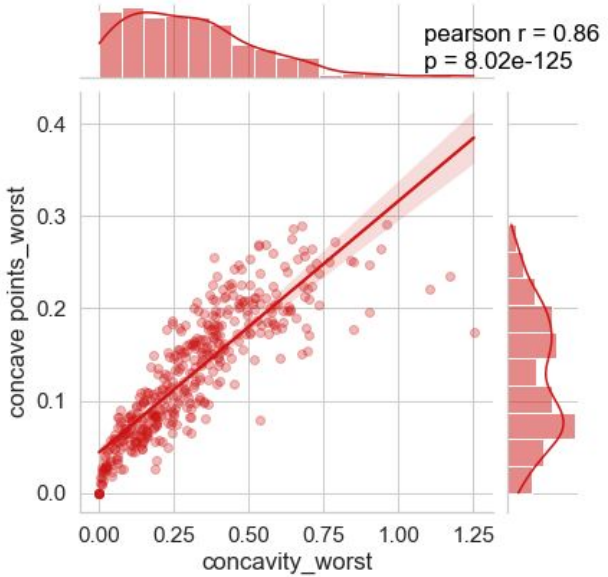

diagnosis
B
M
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
features
−2
0
2
4
6
8
10
value


diagnosis
B
M
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
features
−2
0
2
4
6
value

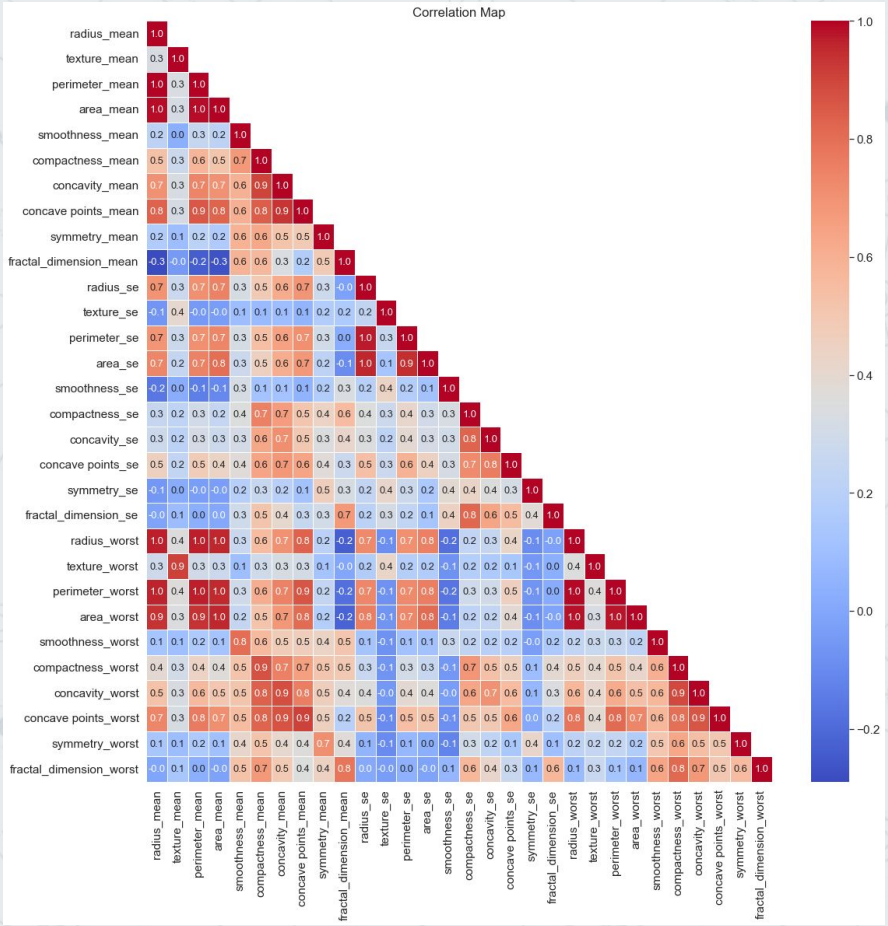

diagnosis
B
M
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
features
−2
0
2
4
6
value

Feature Selection

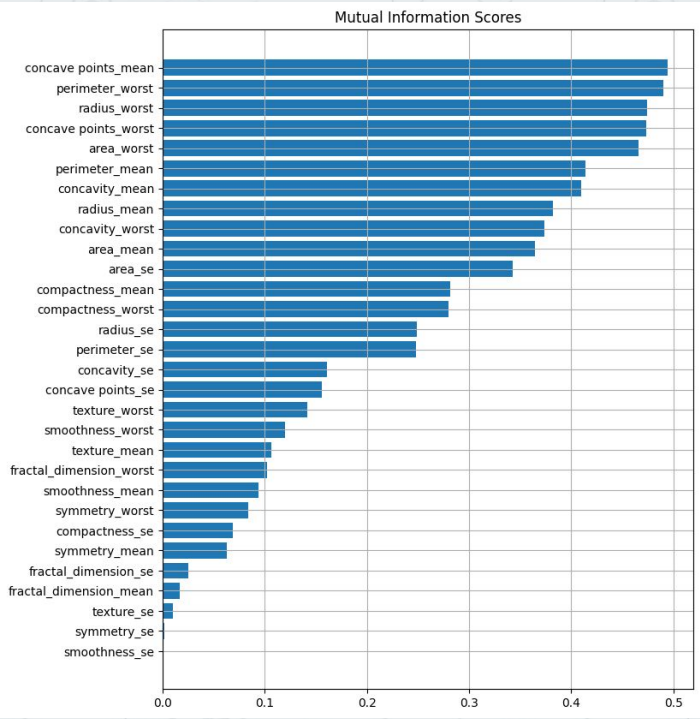
- Linear correlation test with *Pearson's r coefficient*
- 0 → no correlation
- -1 ή +1 → exact linear correlation
- Example between 2 features



Heat map and Mutual Information

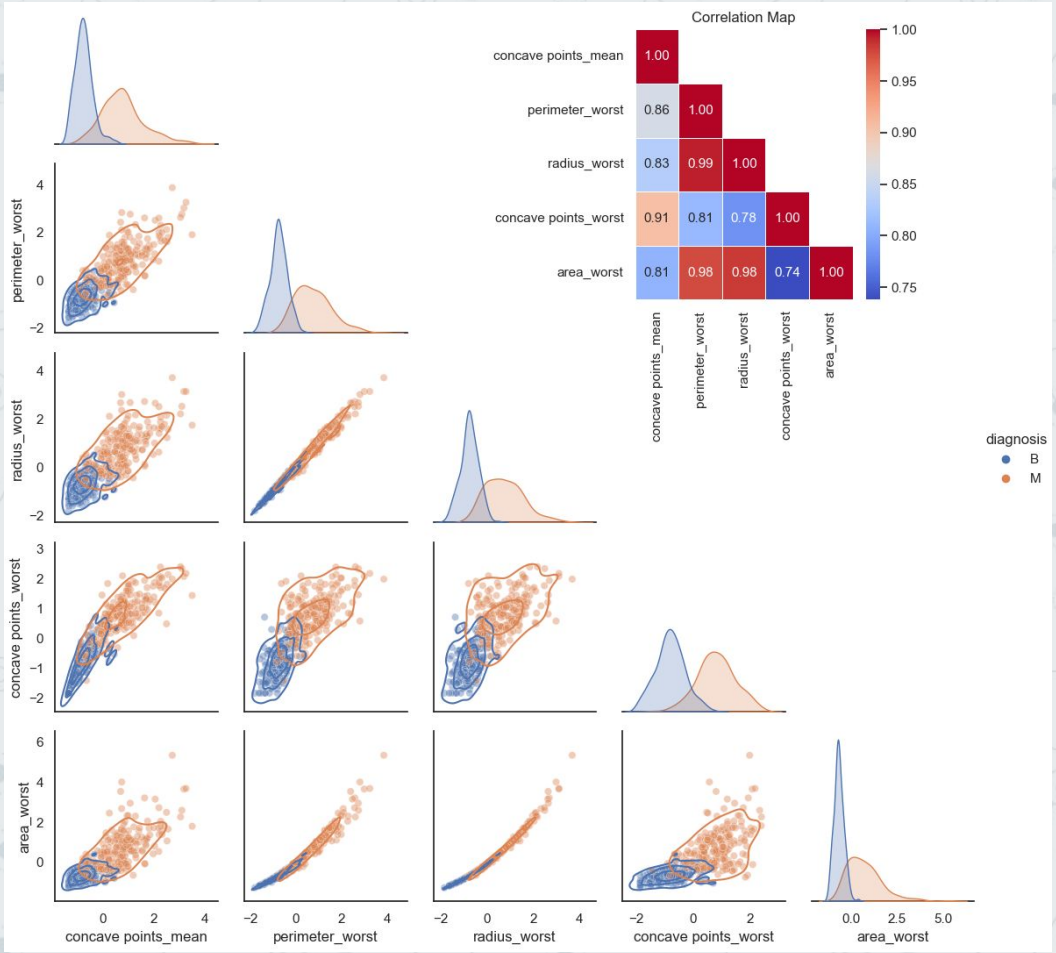


- MI can detect any kind of correlation.
- MI between the features and the target (diagnosis).
- It shows how much information can be obtained from one random variable by observing another.

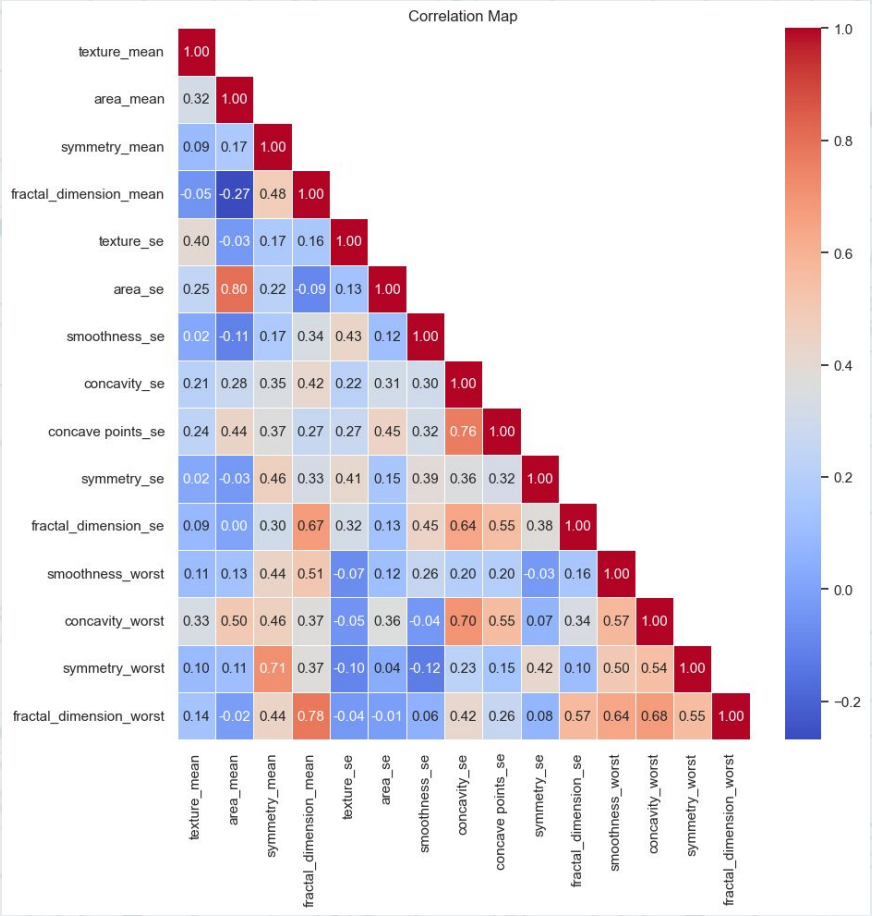


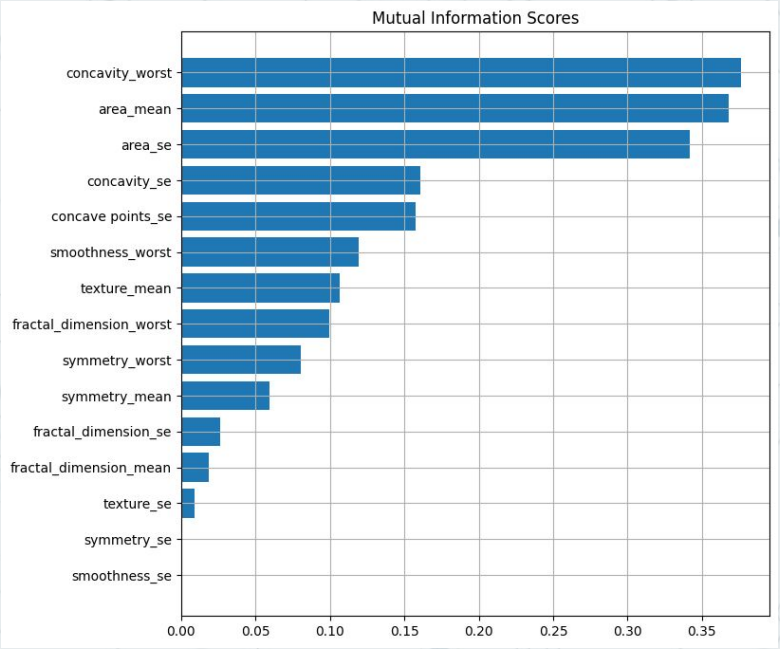
Further investigation of the correlation for the 5 features with the highest MI score

- Large MI score due to high separability
- High correlation between them



Features with linear correlation ≤ 0.8





- Remove the last 2 features with an MI score equal to 0.
- There are 13 features left, which will be further reduced by the feature selection methods.

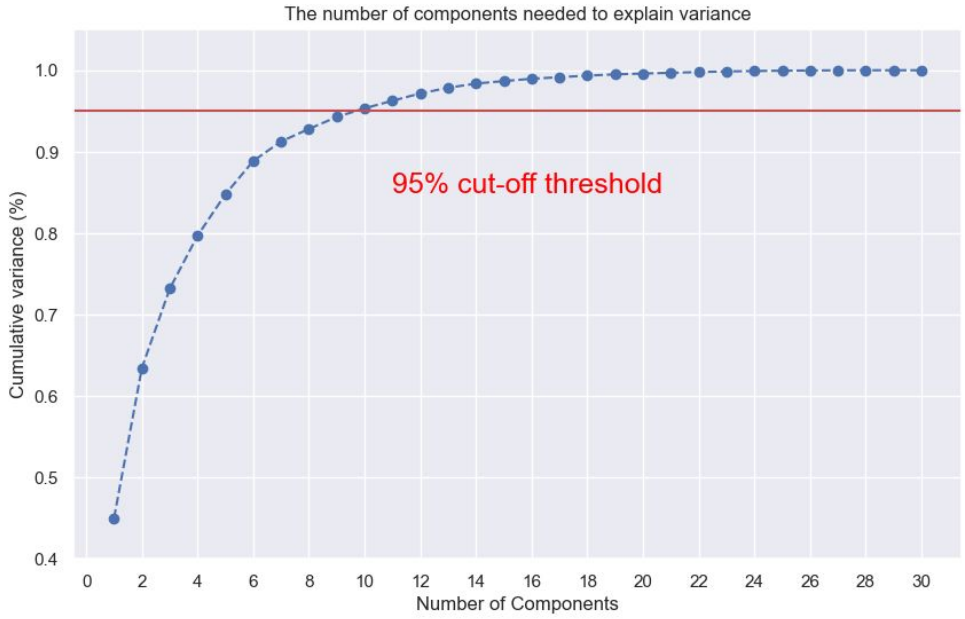
Summary of feature selection

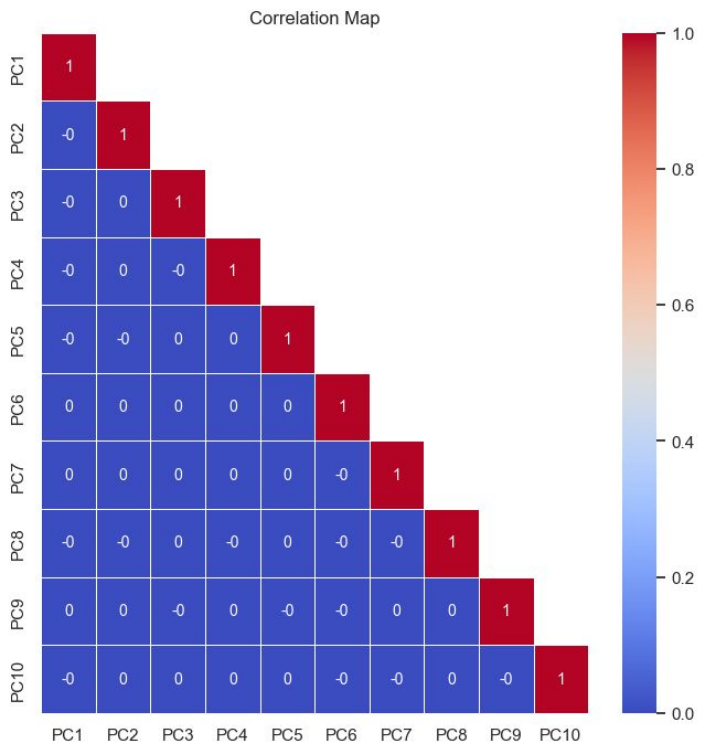
- Features selected by each feature selection method.
- Those voted by all methods will be used.

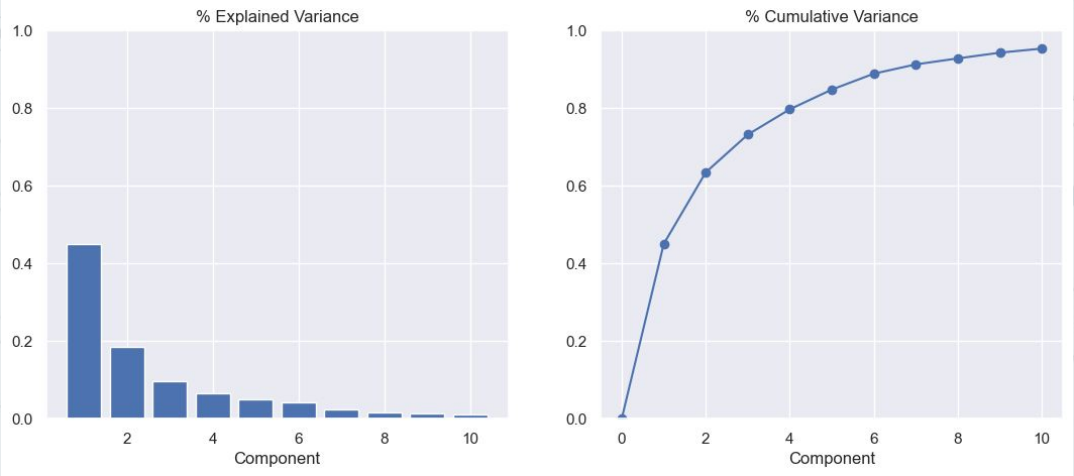
Χαρακτηριστικά / Μέθοδοι επιλογής							
MI (φθίνουσα)	K-Best	RFE	RFECV	RF FI	XGB FI	MRMR	Σύνολο
concavity worst	✓	✓	✓	✓	✓	✓	6/6
area mean	✓	✓	✓	✓	✓	✓	6/6
area se	✓	✓	✓	✓	✓	✓	6/6
concavity se	-	✓	✓	✓	-	-	3/6
concave points se	✓	✓	✓	✓	✓	✓	6/6
smoothness worst	✓	✓	✓	✓	✓	✓	6/6
texture mean	✓	✓	✓	✓	✓	✓	6/6
fractal dimension worst	-	-	✓	-	-	✓	2/6
symmetry worst	✓	✓	✓	✓	✓	✓	6/6
symmetry mean	✓	-	✓	-	-	-	2/6
fractal dimension se	-	-	✓	-	✓	-	2/6
fractal dimension mean	-	-	✓	-	-	-	1/6
texture se	-	-	-	-	-	-	0/6

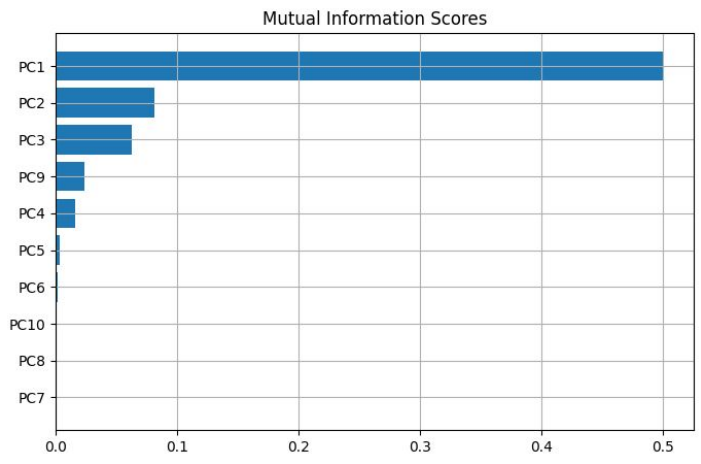
Principal Component Analysis

- Dimensional reduction and decorrelation technique.
- Transforms a correlated multivariate distribution into orthogonal linear combinations of the original variables.
- Threshold 95% of the information corresponding to 10 principal components (dimensions).



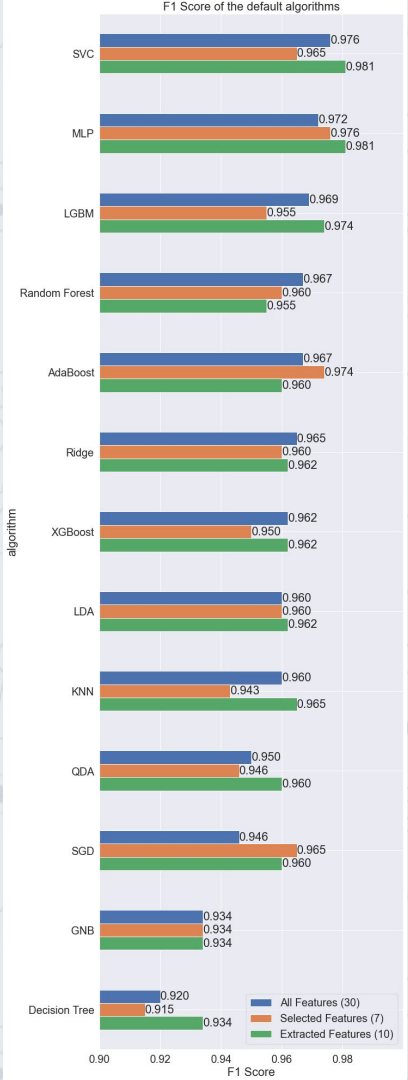

Correlation Map
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
1.0
0.8
0.6
0.4
0.2
0.0

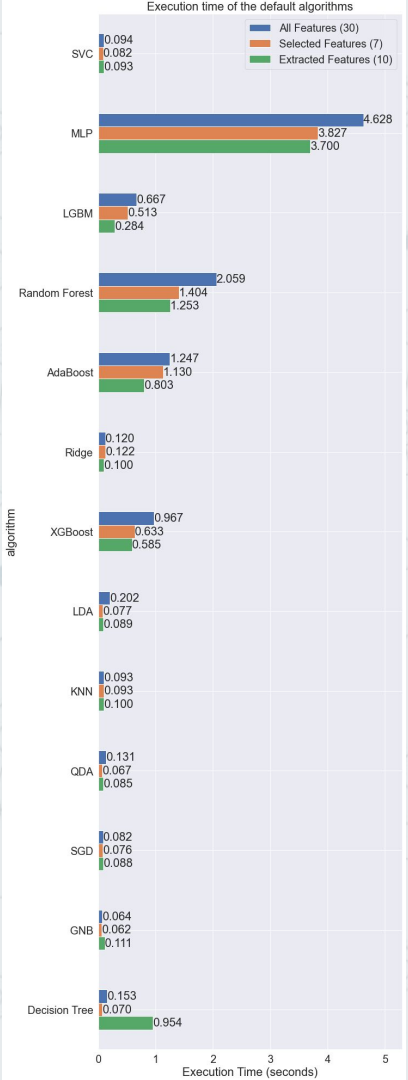

% Explained Variance
Component
% Cumulative Variance
Component


Mutual Information Scores
PC1
PC2
PC3
PC9
PC4
PC5
PC6
PC10
PC8
PC7

Results

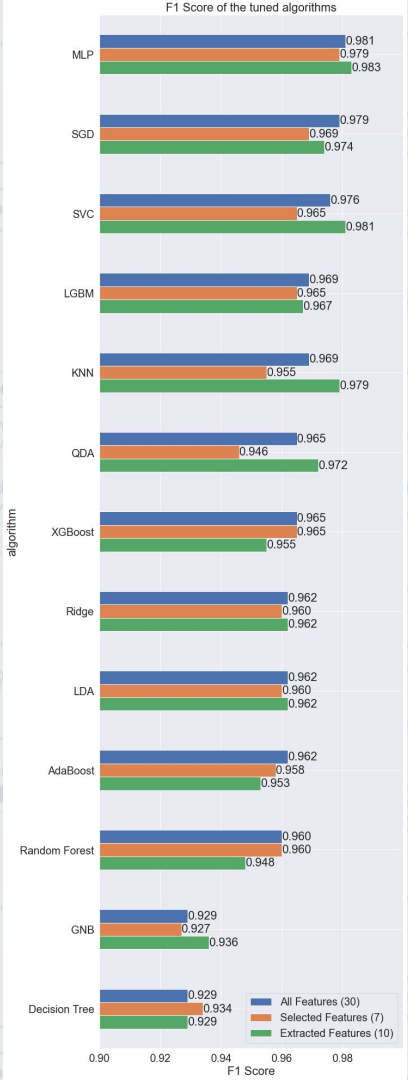
- The same hyperparameter grid was used in each algorithm in all 3 feature sets.
- MLP was optimised by trial and error, so there is no time measurement.
- The execution time of the algorithms is the time of the cross validation process.
- In the algorithms with tuned parameters, the time of the nested cross validation loop is included (so it increases quite a bit).

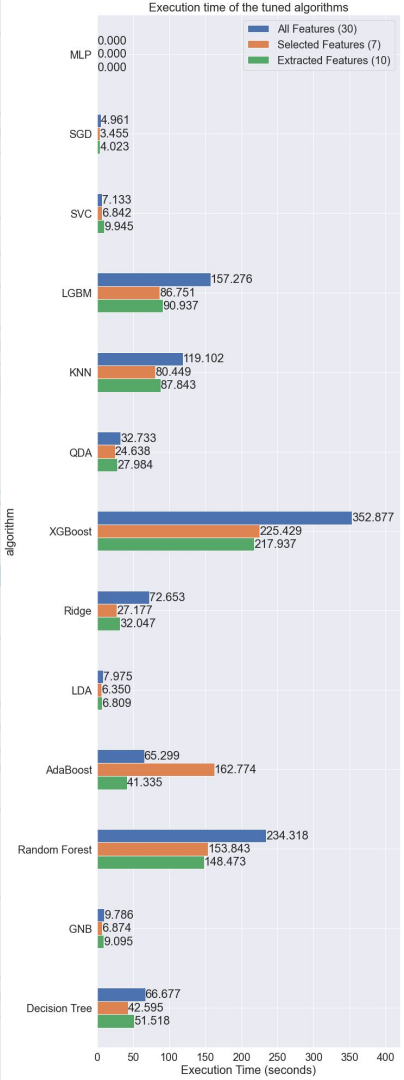




Algorithms with default hyperparameters

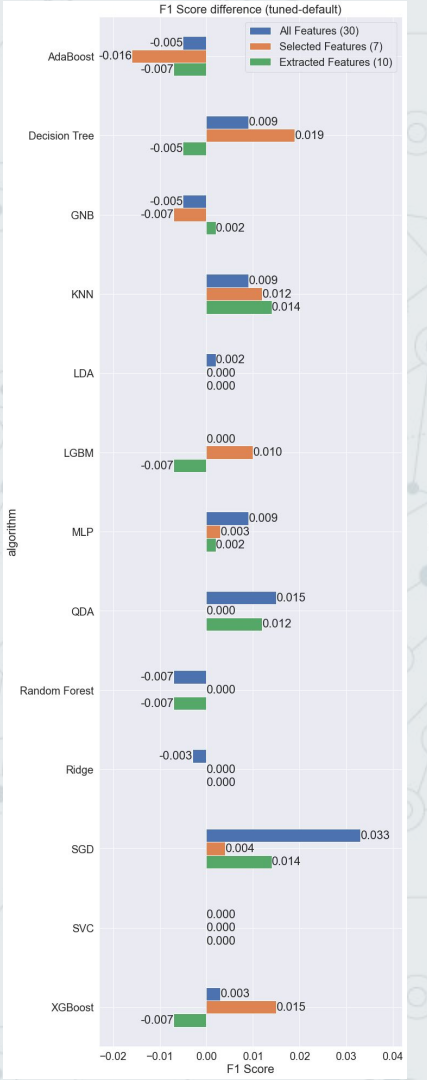
- **F1** scores from **0.915 to 0.981**
- Execution **times** from **0,062 to 4,628 sec**
- Ridge, LDA, GNB keep their performance constant with reduced features
- **Decrease** of **F1** score in **4/5** decision **tree** algorithms with **7** features (LGBM, XGBoost, Random Forest, Decision Tree)
- **AdaBoost** performs **better with weak decision trees** (boosts them better)
- **7 and 10** features lead to **faster execution** times
- The **10** features from **PCA** tend to lead to **better performance**, with the best being from **SVC and MLP**

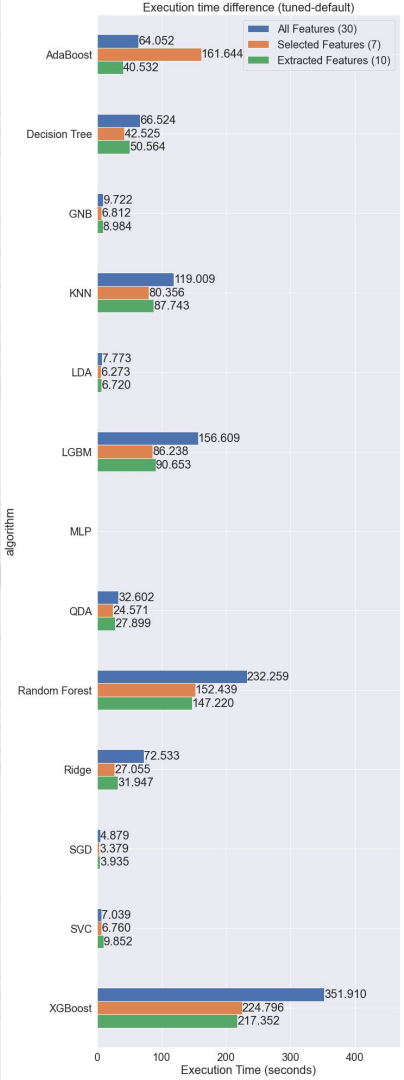




Algorithms with tuned hyperparameters

- **F1** scores from **0.927 to 0.983**
- Execution **times** from **3,455 to 352,877 sec** (almost 6 minutes!!)
- **Ridge** and **LDA** keep their **performance stable** with **reduced features**, while **GNB improved** with those from **PCA**
- Same scores for Ridge and LDA, LDA significantly faster
- **F1** scores **improved** for the **tree** algorithms with the **7** features, while they **decreased** with those from **PCA**
- **7 and 10** features lead to **faster** execution **times**
- The **10** features from **PCA** tend to lead to **better performance**, with the best being from **SVC, MLP, SGD and KNN**





Difference of results (tuned-default)

- **Some** algorithms **reduced** their **performance**, **AdaBoost on all** feature sets, while **Random Forests** and **GNB** on **2** of the 3
- The **increase** in **Decision Tree** performance affects the **decrease** in **AdaBoost** score
- Execution **times increased** too much for AdaBoost, Decision Trees, KNN, Random Forests, XGBoost
- **KNN, MLP, SGD improved** on **all** feature sets, while **Decision Tree, QDA, XGBoost improved** in **2** out of 3
- **SGD improved** significantly with **30** features and **Decision Tree** with **7** selected features
- **LDA** and **Ridge barely** changed with all features, while **SVC** remained **unchanged**

Conclusions

- **Best performance** in **F1** score with features from **PCA** (95,33% of total information).
- The algorithms with **7 selected** features showed **significant results** in terms of **execution time**.
- The **non linear classifiers** (MLP, SVC, KNN, SGD, QDA) showed the **best performance**.
- The **linear classifiers** (Ridge, LDA, GNB) had **low performance** but with **short execution time**.
- The **algorithms related** to **decision trees** (LGBM, XGBoost, AdaBoost, Random Forest, Decision Tree) had **moderate results** with **large execution time**.
- **MLP** had the **highest F1** score (0.983), but it took **a long time** to **optimize**.
- **SVC** had **slightly lower F1** score (0.981), but was **more efficient** in terms of computational **time**(0,093 seconds).
- In **real world applications** (breast cancer diagnosis), **accuracy is crucial** and **MLP** is the **preferred choice** due to its higher F1 score.

Future work

- Deepening the algorithms that had the best results
- Exploring different dimensionality reduction techniques
- Integration of more complex machine learning methods
- Exploring the use of synthetic data generation techniques
- Experimenting with other datasets

Thank you for your attention!!

Source code :
https://github.com/LazarosPan/Breast-Cancer-Classification-with-Machine-Learning-Methods