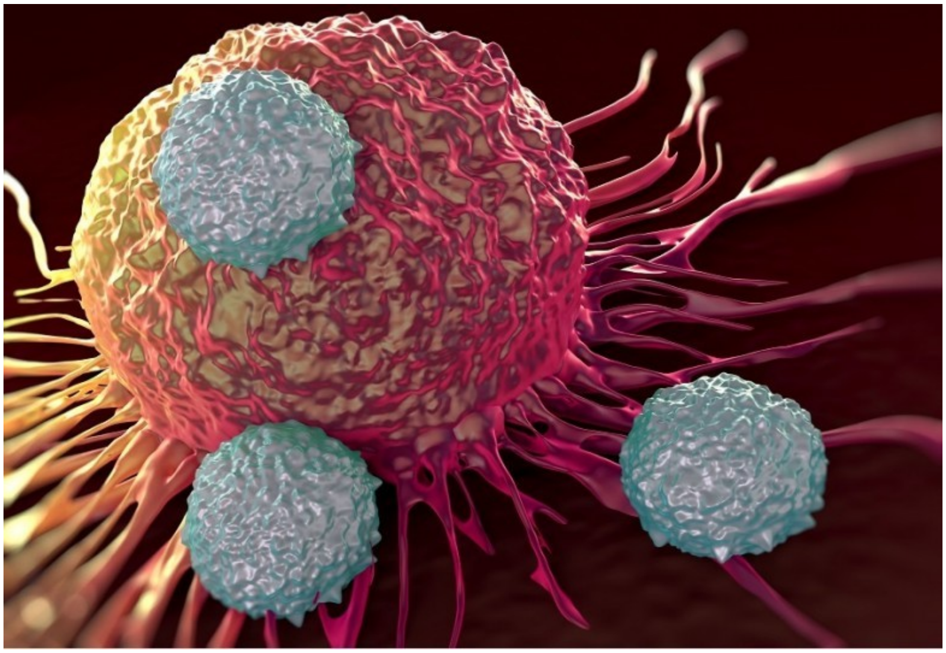
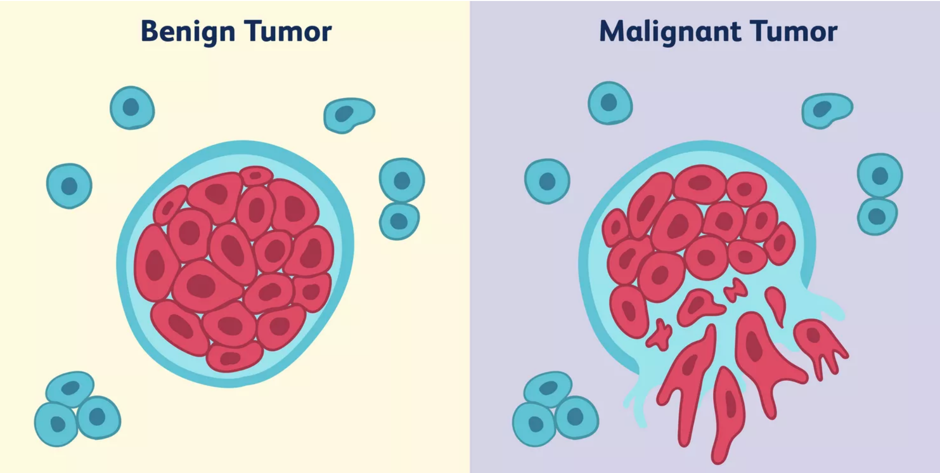
Technion - Israel Institute of Technology

Machine Learning in Healthcare (097248)

Wisconsin Breast Cancer Dataset





Daniel Engelsman @ August 16, 2020

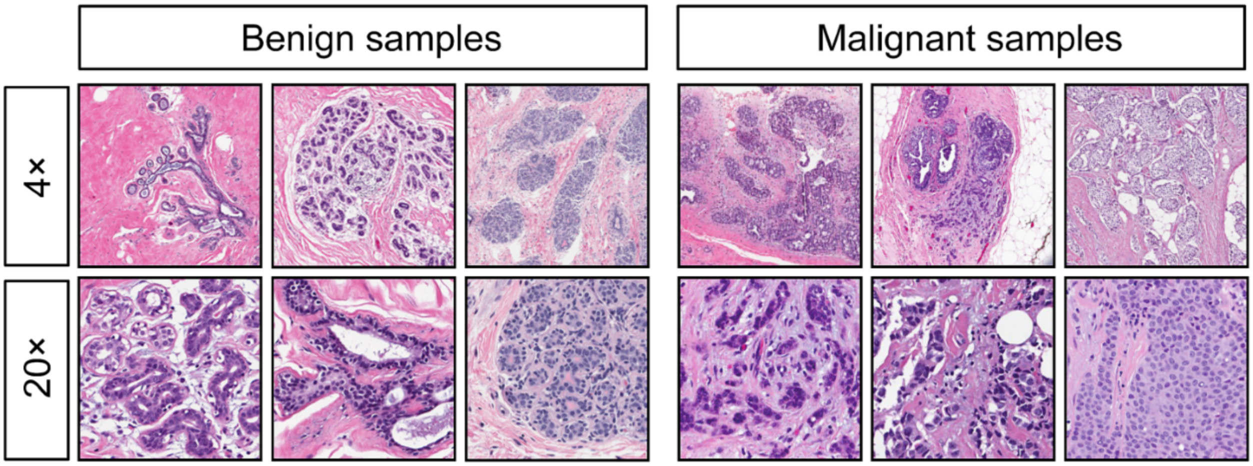
Contents

1 Description of the Problem

Breast cancer is a form of cancer that develops from breast tissue. The most common risk factors include : being female, obesity, a lack of physical exercise, alcoholism, an early age at first menstruation, having children late in life or not at all, older age and a family history of breast cancer [1]. Typically it presents as a lump that feels different from the rest of the breast tissue, often self detected by the person's fingertips.

Cancer grows by simple cell division. It begins as one malignant cell, which then divides and becomes two bad cells and so forth. Each division takes about two months [2], and detection by hand cannot be felt before the 30th division, hence pre-existence of 2-3 years.

The earliest breast detection are obtained by a mammogram : an X-rays screening method that can offer an approximate likelihood whether a lump is cancerous. When examinations are inconclusive, a healthcare provider can remove a sample of the lump fluid / biopsy, and then examine it deeper under a microscopic analysis (FNA) :



Visual differences between benign (non-cancerous) and malignant (cancerous) cells can be very elusive at early detection stages. Researchers rely on typical dissimilarities of each cell's characteristics (perimeter, area, texture etc.) to diagnose its status (ACR).

For this reason, the pre-diagnosis allows a narrow window of opportunity that can be lifesaving, before the cancer spreads to secondary sites (Metastasis). Thus, analysis is extremely important and should be handled by experts only.

1.1 Why AI ?

As seen from articles along the semester, clinicians, regardless their years of experience and expertise, eventually make mistakes. These wrong indications accumulate over time, resulting in high rates of false positives and false negatives (FPR / FNR) [3].

Contrarily, AI systems have proven to outperform radiologists when it comes to interpreting breast cancer mammograms, resulting in higher area under the operating characteristic curve (**AUC-ROC**) [4]. Exploiting the vast amounts of labeled data in the electronic health records (EHR) makes the AI frameworks helpful in fight against cancer [5].

2 Background and Related Work

The Wisconsin Breast Cancer Dataset (WBCD) was first released on 1992 by Wisconsin university, and since became a milestone in the AI progress. However, early publications utilized mainly classic machine learning methods due to primitive computational platforms.

Bennett et al. (1992) [6] proposed a minimization scheme which reduces the average sum of misclassified points belonging to two disjoint points sets in $\mathbb{R}^n$ space. Disjoint of the two convex subsets leads to a complete separation plane between them.

Street et al. (1993) [7] used ten-fold cross-validation on a single separating plane utilizing only three most important features. Mangasarian (1995) [8] showed how linear programming (LP) techniques are capable of minimizing a penalty over a misclassified points, thus converging to almost perfect classification.

Y-J Lee (2000) [9] utilized a linear support vector machine (SVM) to extract 6 features from a total 30. Then, using Gaussian kernel non-linear SVM, the reduced dataset underwent division into three prognostic groups, ending up with a high level of accuracy.

The progress in artificial neural networks (ANN) in the beginning of the 2000s opened the door for many novel approaches. Pantazi (2002) [10] showed how Kohonen model of self-organizing maps can be utilized for cluster analysis on the WBCD. Revett [11] et al. (2005) presented a medical decision system based on a probabilistic NN to perform dimensionality reduction to eliminate redundant attributes from the dataset.

Huang et al. (2007) [12] were first to achieved an outstanding classification accuracy of 98.14% by hybridizing a fuzzy-artificial immune system with k-nearest neighbor algorithm.

Belciug (2010) [13] assessed the effectiveness of three different clustering algorithms with respect to the WBCD, by comparing the performance of a classical k-means algorithm with a much more sophisticated methods : SOM-Kohonen network and a cluster network.

The ”big bang” of the deep learning (DL)

Since 2012, the ImageNet project is responsible for many technological breakthroughs. Nowadays, a simple deep learning (DL) model can easily obtain a precise classification. These models manage to automatically redundant tough procedures e.g dimensionality reduction, decimation and feature extraction, outputting eventually impressive results.

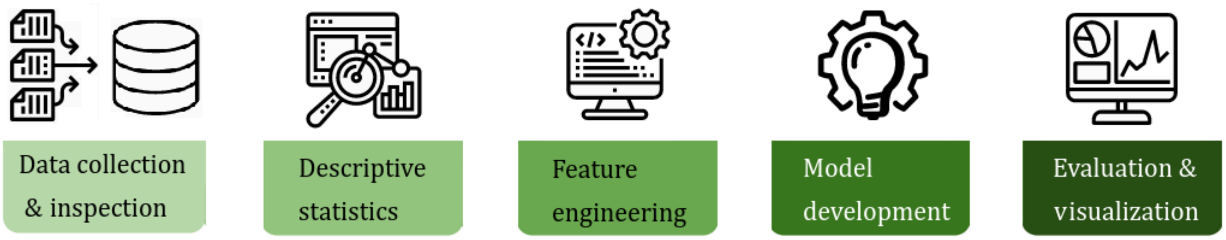
Karthik et al. (2016) [14] showed how Computer-Aided Diagnosis (CAD) can employ DNN as classifier model and recursive feature elimination (RFE) for feature selection. After optimizing the train-test split the model reached 98.62% accuracy.

Abdel-Zaher et al. (2018) [15] presented a CAD detection scheme using deep belief network unsupervised path (DBN-NN), achieving 99.68% accuracy rate and 100% sensitivity.

2.1 Intention Statement

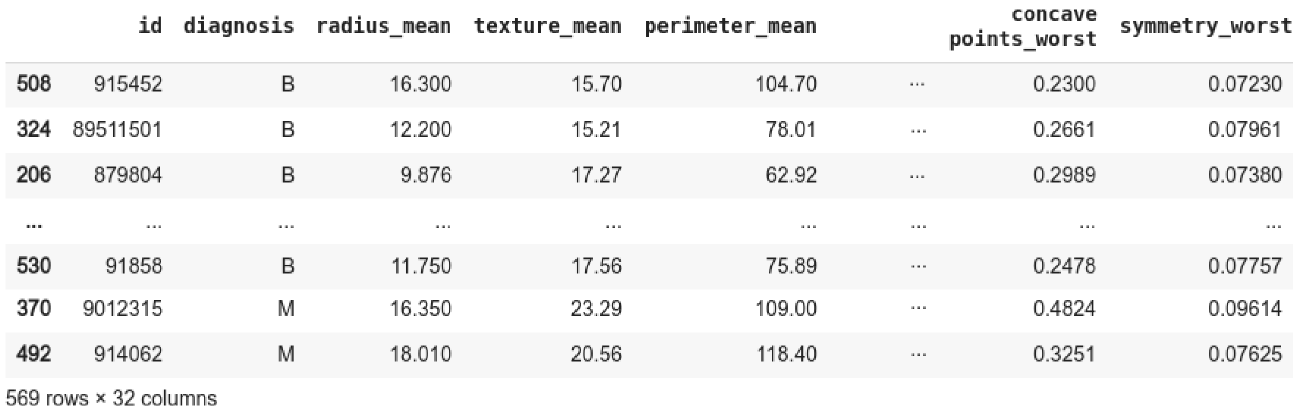
In this project I aim to explore the data thoroughly, in search of hidden patterns using non deep learning classification tools. The findings will be analyzed in context of the problem's clinical aspects using interactive data visualizations. This way I hope to establish a robust understanding of machine learning in context of healthcare.

2.2 The workflow



3 Description of the Data

The raw dataset (including indices and **id**) after applying random shuffling [↓] :



	id	diagnosis	radius_mean	texture_mean	perimeter_mean		concave points_worst	symmetry_worst
508	915452	B	16.300	15.70	104.70	...	0.2300	0.07230
324	89511501	B	12.200	15.21	78.01	...	0.2661	0.07961
206	879804	B	9.876	17.27	62.92	...	0.2989	0.07380
...	...	...	...	...	...	...	...	...
530	91858	B	11.750	17.56	75.89	...	0.2478	0.07757
370	9012315	M	16.350	23.29	109.00	...	0.4824	0.09614
492	914062	M	18.010	20.56	118.40	...	0.3251	0.07625

569 rows × 32 columns

Number of samples : $n = 569$ Full dataset : $D = \{X, Y\} = \{x_i, y_i\}_{i=1}^{n}$

Input space : $\dim(X) = (n \times 10) \underset{\text{case :}}{\times} \underset{\textbf{a, b, c}}{3}$ Output space : $\dim(Y) = n$

Dependent variable / Target $\big(\; y_i = \{B,\, M\} \; \big)$

Diagnosis - Binary indication whether a sample is Benign (0) or Malignant (1).

Feature space X consists of three different analysis cases (same features) :

a := Mean (μ) = $X_{[:,\,1:10]}$ **b** := SE ($\frac{\sigma}{\sqrt{n}}$) = $X_{[:,\,11:20]}$ **c** := Worst (3 max-values) = $X_{[:,\,21:30]}$

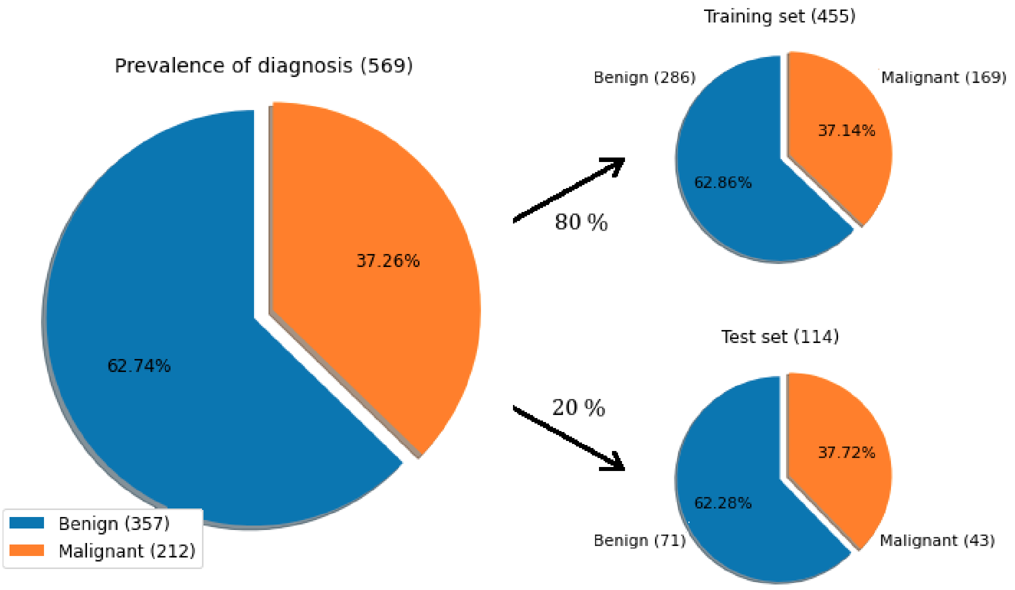
For example, let us examine the i-th sample from the mean case (**a**) :

Independent variable / Feature $\big(\; x_i \in \mathbb{R}^{10} \; \big)$

Radius - mean of distances from center to points on the perimeter.
Texture - standard deviation of gray-scale value.
Perimeter - $P = \sum_{i=1}^{n} l_i$ (Given n vertices)
Area - $A = \frac{1}{2}\big(\sum_j w_j z_{j+1} - z_j w_{j+1} \big)$ (Given n vertices on WZ plane)
Smoothness - Local variation in radius lengths.
Compactness - (P^2 / A - 1.0)
Concavity - Severity of concave portions of the contour.
Concave points - Number of concave portions of the contour.
Symmetry - Relative difference of two half-planes.
Fractal dimension - ("coastline approximation" - 1)

To sum up, the dataset is composed such that : $X = [X_{\text{Mean}}^{(a)},\, X_{\text{SE}}^{(b)},\, X_{\text{Worst}}^{(c)}] \in \mathbb{R}^{(n \times 10) \times 3}$

3.1 Data composition



Although division is not perfectly balanced, it is sufficiently good. The first manipulation will help us gaining some intuition by comparing the mean value of every feature :

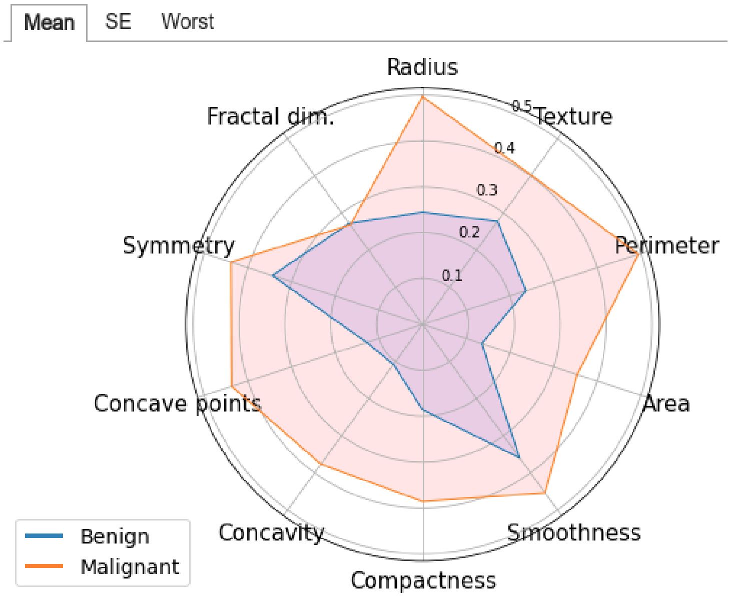
1. Rescaling (min-max normalization) :

$$\hat{X} = \frac{X - \min(X)}{\max(X) - \min(X)} \in [0,1]^n$$

2. Comparison between the labels :

$$\bar{\hat{X}}^a_{\{y=M\}} \text{ vs. } \bar{\hat{X}}^a_{\{y=B\}}$$

(!) Note that both are dimensionless



This presentation is useful as it compares the classes by measure of similarity $d(B, M)$.

Note : Interactive switching between the tabs (cases) is available via the *Colab* version. Anyway, for the sake of analysis, I will refer to all of the cases.

True for all : Almost all cases exhibited higher feature values in the malignant class. However, Fractal dimension seemed to be equal for both - $d(B, M) \approx 0$.

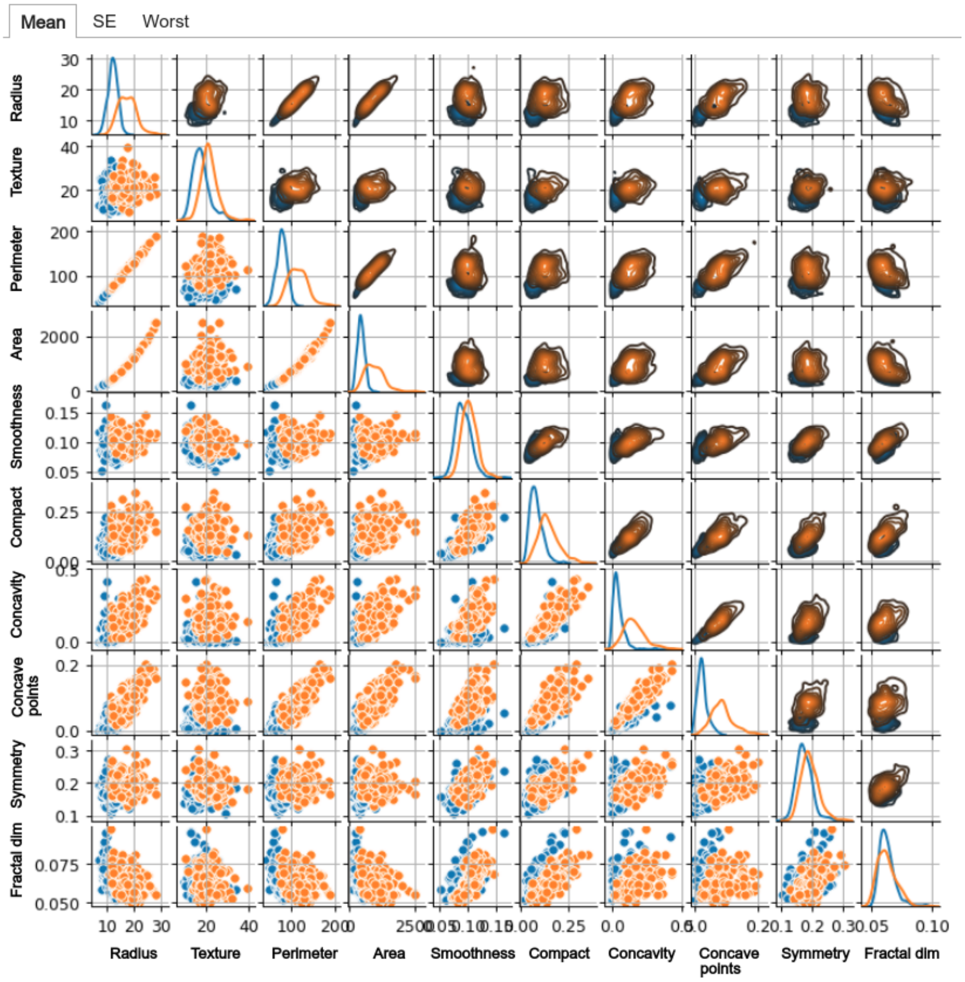
Mean case (a) : The most important comparison (above) as it manages to capture most of the feature dissimilarities between (B/M) :

- Proximity - $d(B, M) \leq 0.1$: Symmetry, Smoothness
- Dissimilarity - $d(B, M) > 0.1$: The remained features

Standatd Error case (b) : The differences in the SE comparison are significantly smaller, implying that the SE group distributes uniformly, and is useless for any inference.

Worst case (c) : Resembles **(a)** case, where the M features are much greater than B.

3.2 Statistical analysis

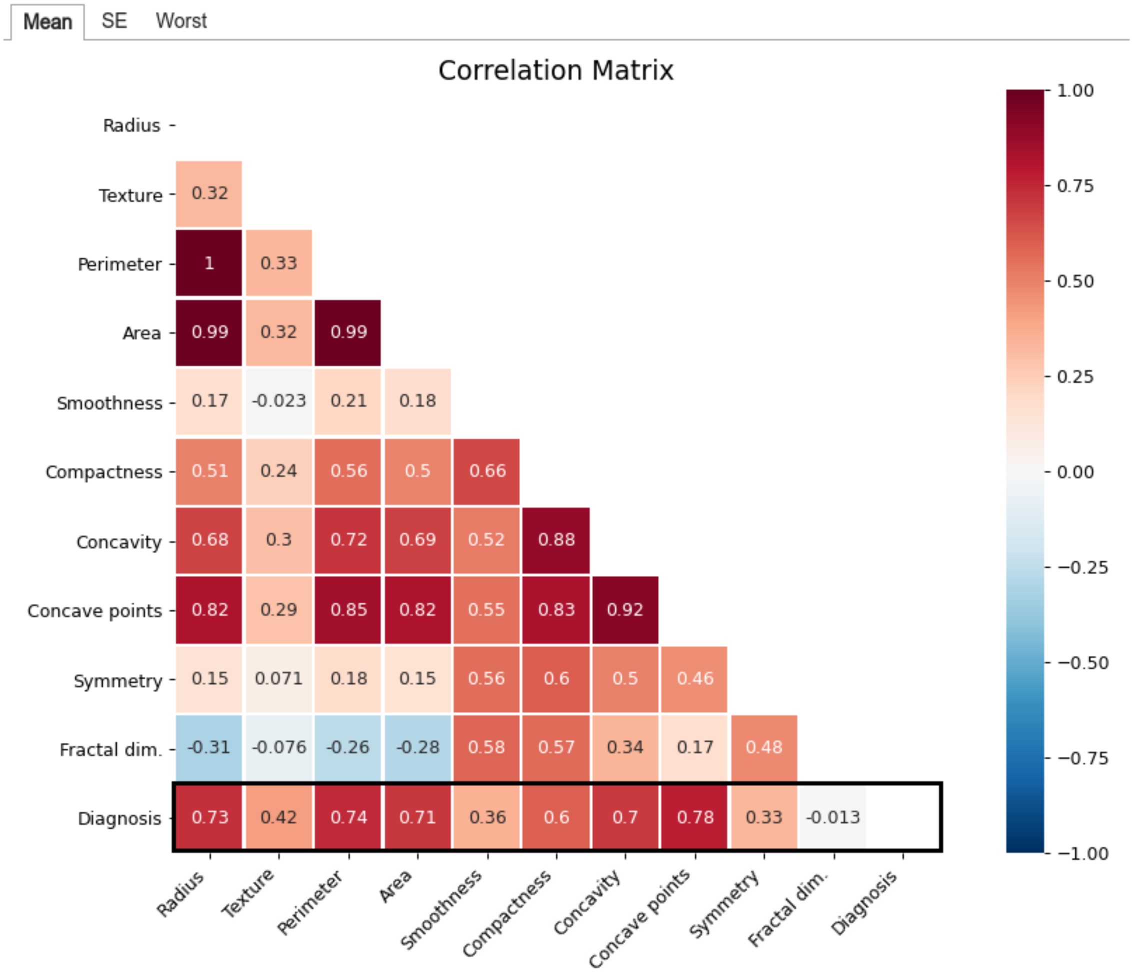


The above table maps each independent variable onto a row and column in a grid of multiple axes in order to explore bivariate relationships between every feature pair. The **lower** triangular presents a raw scatter plot of the joint probability distribution $P_{XY}(x,y)$. The **upper** triangular presents the same, using smoothed contour lines (KDE).
The **diagonal** exhibits the marginal distribution of both variables - $P_X(x)$, $P_Y(y)$.

This analysis shows not only the (B/M) differences but also the way each feature distributes. Smoothness and Fractal dimension distributes similarly, hence helpless in the classification attempt. In contrast, Concavity and Radius show significant difference.

In order to better understand which features contribute most, an efficient method can be calculating the Pearson correlation coefficients to obtain a correlation matrix. Thereby, the strength of linear relationships are obtained in two aspects (Appendix A) :

(i) **feature** vs. **feature** - Every pair of two independent variables.
(ii) **feature** vs. **target** - Every pair of feature-target (bottom rectangle).

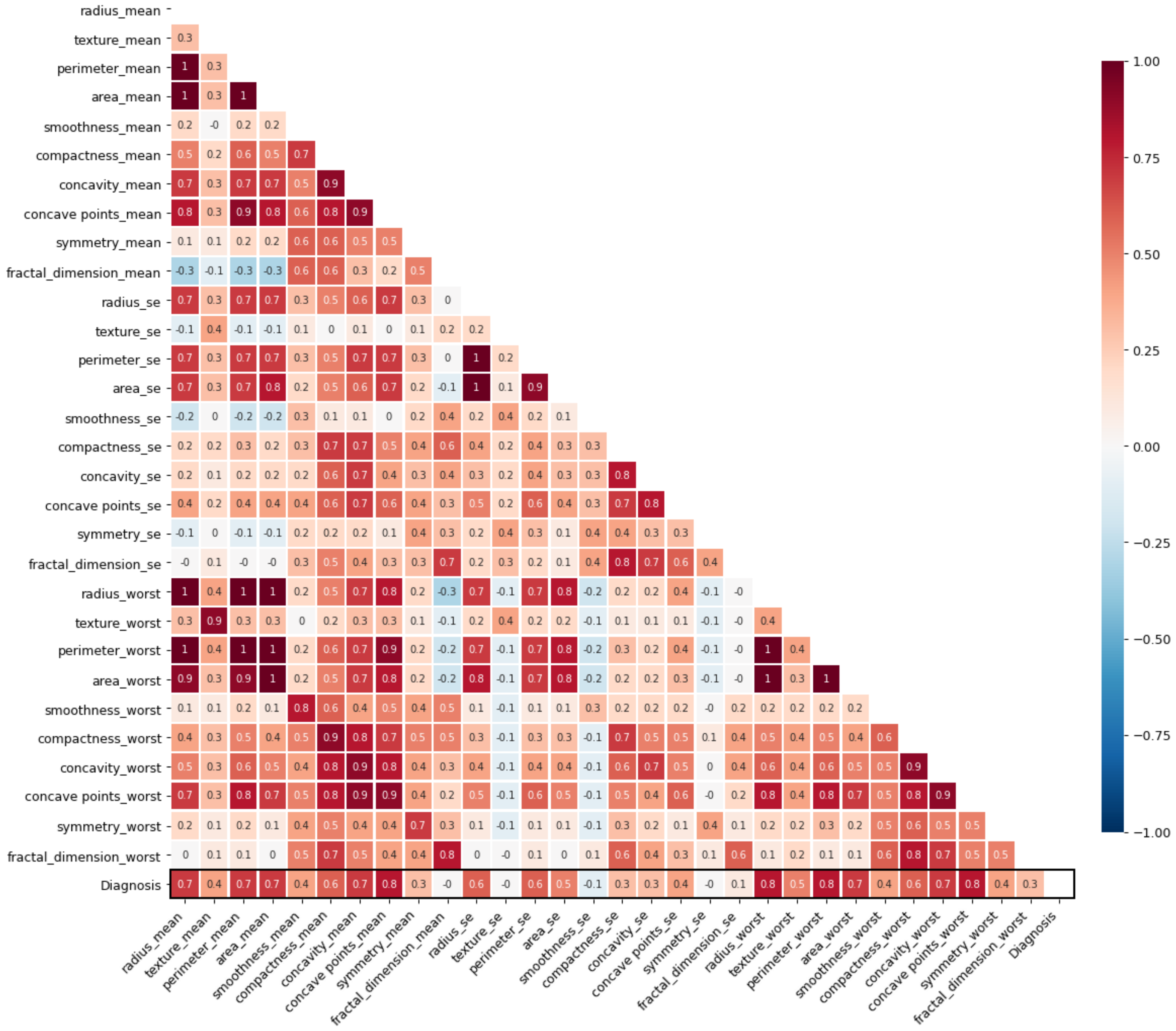


	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	Concave points	Symmetry	Fractal dim.	Diagnosis
Radius											
Texture	0.32										
Perimeter	1	0.33									
Area	0.99	0.32	0.99								
Smoothness	0.17	-0.023	0.21	0.18							
Compactness	0.51	0.24	0.56	0.5	0.66						
Concavity	0.68	0.3	0.72	0.69	0.52	0.88					
Concave points	0.82	0.29	0.85	0.82	0.55	0.83	0.92				
Symmetry	0.15	0.071	0.18	0.15	0.56	0.6	0.5	0.46			
Fractal dim.	-0.31	-0.076	-0.26	-0.28	0.58	0.57	0.34	0.17	0.48		
Diagnosis	0.73	0.42	0.74	0.71	0.36	0.6	0.7	0.78	0.33	-0.013	

(i) Rows $1:(n-1):$ Based on the linear regression assumptions, independent variables should not be correlated with each other (Multicollinearity). As can be seen, the Radius (r) and the Perimeter (P) have almost 100% correlation score, apparently since most samples are circular and $P = f(r)$. Therefore, one of the feature is redundant to another (provide same information) and can be removed to ease calculations and reduce overfitting.

(ii) n-th Row : Contrarily to (i), here the correlation expresses the feature relevance to the target (**diagnosis**). Higher values mean higher contribution / relevance. For example, the Radius, Texture, Area, Concavity exhibit ($\geq 70\%$), while Fractal dimension ($\sim 1\%$).

In order to extract maximum relevance from entire X, it can be applied on all 30 features :



	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	radius_se	texture_se	perimeter_se	area_se	smoothness_se	compactness_se	concavity_se	concave points_se	symmetry_se	fractal_dimension_se	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst	Diagnosis
radius_mean																															
texture_mean	0.3																														
perimeter_mean	1	0.3																													
area_mean	1	0.3	1																												
smoothness_mean	0.2	-0	0.2	0.2																											
compactness_mean	0.5	0.2	0.6	0.5	0.7																										
concavity_mean	0.7	0.3	0.7	0.7	0.5	0.9																									
concave points_mean	0.8	0.3	0.9	0.8	0.6	0.8	0.9																								
symmetry_mean	0.1	0.1	0.2	0.2	0.6	0.6	0.5	0.5																							
fractal_dimension_mean	-0.3	-0.1	-0.3	-0.3	0.6	0.6	0.3	0.2	0.5																						
radius_se	0.7	0.3	0.7	0.7	0.3	0.5	0.6	0.7	0.3	0																					
texture_se	-0.1	0.4	-0.1	-0.1	0.1	0	0.1	0	0.1	0.2	0.2																				
perimeter_se	0.7	0.3	0.7	0.7	0.3	0.5	0.7	0.7	0.3	0	1	0.2																			
area_se	0.7	0.3	0.7	0.8	0.2	0.5	0.6	0.7	0.2	-0.1	1	0.1	0.9																		
smoothness_se	-0.2	0	-0.2	-0.2	0.3	0.1	0.1	0	0.2	0.4	0.2	0.4	0.2	0.1																	
compactness_se	0.2	0.2	0.3	0.2	0.3	0.7	0.7	0.5	0.4	0.6	0.4	0.2	0.4	0.3	0.3																
concavity_se	0.2	0.1	0.2	0.2	0.2	0.6	0.7	0.4	0.3	0.4	0.3	0.2	0.4	0.3	0.3	0.8															
concave points_se	0.4	0.2	0.4	0.4	0.4	0.6	0.7	0.6	0.4	0.3	0.5	0.2	0.6	0.4	0.3	0.7	0.8														
symmetry_se	-0.1	0	-0.1	-0.1	0.2	0.2	0.2	0.1	0.4	0.3	0.2	0.4	0.3	0.1	0.4	0.4	0.3	0.3													
fractal_dimension_se	-0	0.1	-0	-0	0.3	0.5	0.4	0.3	0.3	0.7	0.2	0.3	0.2	0.1	0.4	0.8	0.7	0.6	0.4												
radius_worst	1	0.4	1	1	0.2	0.5	0.7	0.8	0.2	-0.3	0.7	-0.1	0.7	0.8	-0.2	0.2	0.2	0.4	-0.1	-0											
texture_worst	0.3	0.9	0.3	0.3	0	0.2	0.3	0.3	0.1	-0.1	0.2	0.4	0.2	0.2	-0.1	0.1	0.1	0.1	-0.1	-0	0.4										
perimeter_worst	1	0.4	1	1	0.2	0.6	0.7	0.9	0.2	-0.2	0.7	-0.1	0.7	0.8	-0.2	0.3	0.2	0.4	-0.1	-0	1	0.4									
area_worst	0.9	0.3	0.9	1	0.2	0.5	0.7	0.8	0.2	-0.2	0.8	-0.1	0.7	0.8	-0.2	0.2	0.2	0.3	-0.1	-0	1	0.3	1								
smoothness_worst	0.1	0.1	0.2	0.1	0.8	0.6	0.4	0.5	0.4	0.5	0.1	-0.1	0.1	0.1	0.3	0.2	0.2	0.2	-0	0.2	0.2	0.2	0.2	0.2							
compactness_worst	0.4	0.3	0.5	0.4	0.5	0.9	0.8	0.7	0.5	0.5	0.3	-0.1	0.3	0.3	-0.1	0.7	0.5	0.5	0.1	0.4	0.5	0.4	0.5	0.4	0.6						
concavity_worst	0.5	0.3	0.6	0.5	0.4	0.8	0.9	0.8	0.4	0.3	0.4	-0.1	0.4	0.4	-0.1	0.6	0.7	0.5	0	0.4	0.6	0.4	0.6	0.5	0.5	0.9					
concave points_worst	0.7	0.3	0.8	0.7	0.5	0.8	0.9	0.9	0.4	0.2	0.5	-0.1	0.6	0.5	-0.1	0.5	0.4	0.6	-0	0.2	0.8	0.4	0.8	0.7	0.5	0.8	0.9				
symmetry_worst	0.2	0.1	0.2	0.1	0.4	0.5	0.4	0.4	0.7	0.3	0.1	-0.1	0.1	0.1	-0.1	0.3	0.2	0.1	0.4	0.1	0.2	0.2	0.3	0.2	0.5	0.6	0.5	0.5			
fractal_dimension_worst	0	0.1	0.1	0	0.5	0.7	0.5	0.4	0.4	0.8	0	-0	0.1	0	0.1	0.6	0.4	0.3	0.1	0.6	0.1	0.2	0.1	0.1	0.6	0.8	0.7	0.5	0.5		
Diagnosis	0.7	0.4	0.7	0.7	0.4	0.6	0.7	0.8	0.3	-0	0.6	-0	0.6	0.5	-0.1	0.3	0.3	0.4	-0	0.1	0.8	0.5	0.8	0.7	0.4	0.6	0.7	0.8	0.4	0.3	

4 The Method

(i) Extract the most significant features and reduce their dimensionality $(d_0 > d_1 > d_2)$:

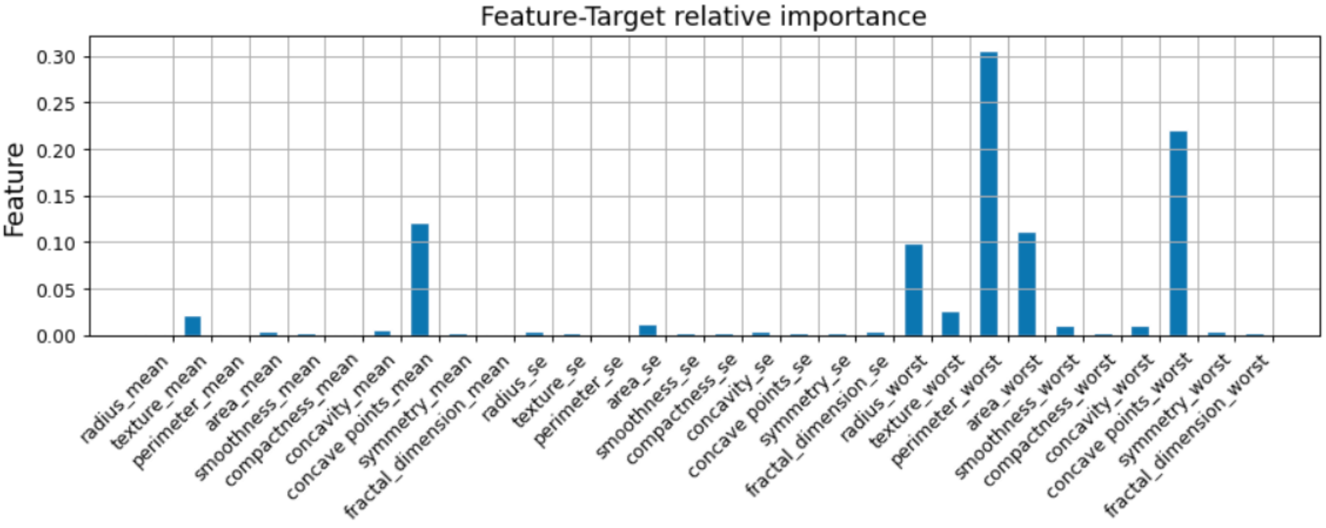
$$X_0 \in \mathbb{R}^{n\times d_0} \underset{\text{Feature selection}}{\Rightarrow} X_1 \in \mathbb{R}^{n\times d_1} \underset{\text{Dim. compression}}{\Rightarrow} X_2 \in \mathbb{R}^{n\times d_2}$$

(ii) Compare performances between different classifiers on the data : $f^* : X_2 \to Y$

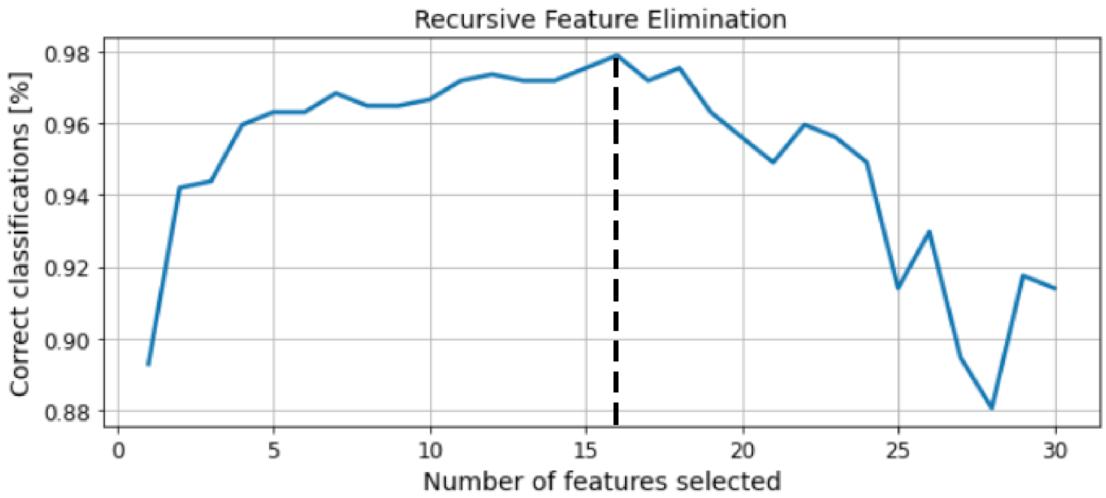
4.1 Feature engineering

The values in the bottom rectangle (previous figure) denote the correlation strength of features to the target. While most featrues exhibit weak connections - $|corr| \leq 0.3$, some are strongly related $|corr| \approx 0.8$. Therefore the constraint is simple : features should maximize the correlation to the target but minimize correlation with each other.

To that end, a reliable tool is the random forest regressor, based on GINI importance [16]. The method fits a number of classifying decision trees on different subgroups, and calculates each feature importance as the sum over the number of splits (across all tress), proportionally to the number of samples it splits :

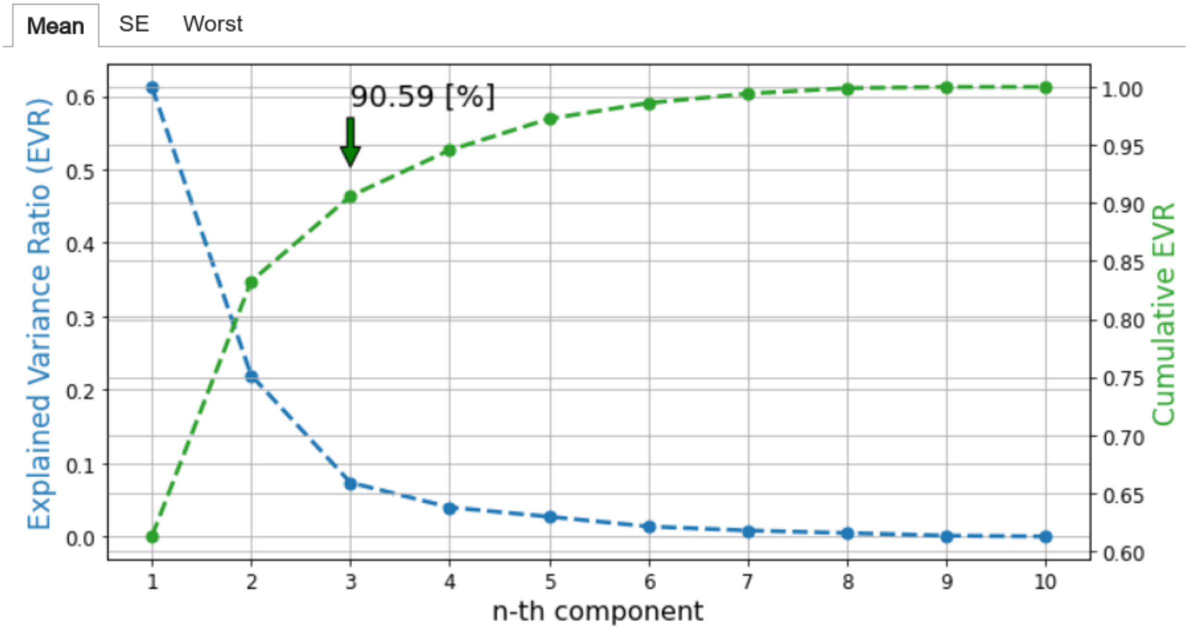


As seen before, the Concave points and the perimeter contribute by far more than any other feature. Note that five features manage to contribute up to 80% of the importance. Using recursive feature elimination (RFE) [17], the features are sorted such that high-ranked (informative) features increase the model accuracy ($d \leq 16$). But contrastingly, further addition of low-ranked features, would cause a subsequent decrease in the accuracy :



Principal component analysis (PCA)

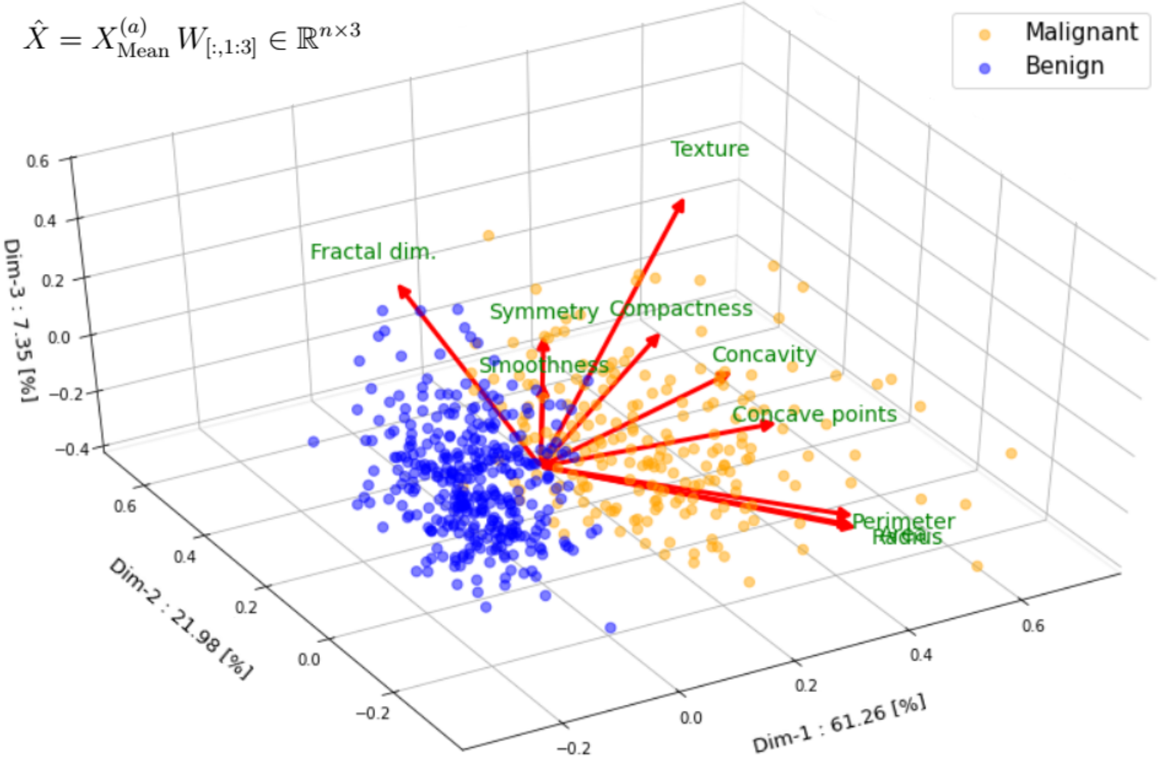
PCA is a practical step in dimensionality reduction, by projecting the data from a high-dimensional space onto a lower one. The transformation manages to reveal the internal structure of the data that explains most the data variance (Appendix B) :



The figure above presents PCA on case (**a**) ($X^{(a)}_{\text{Mean}} \in \mathbb{R}^{n\times 10}$). From left is the PCA's ability to capture proportional variance (EVR), and from right is the cumulative sum.

Let $\mathbf{W} \in \mathbb{R}^{10\times 10}$ be a weights matrix whose columns are the eigenvectors of $X^T X$. By choosing $\mathbf{W}$'s top 3 eigenvectors, the observations can be projected onto a 3D space :

$$\hat{X} = X^{(a)}_{\text{Mean}} \, W_{[:,1:3]} \in \mathbb{R}^{n\times 3}$$



Each variable (feature) that went into PCA has an associated red arrow (after scaling factor), in the directions that maximize each of the PC's variance [18]. Here, the Concave points feature (strong correlation), maximizes the 1st PC. Contrarily, Fractal dimension and Symmetry, contribute poorly to the 3rd PC. Therefore, the full dataset will undergo :

$$X_0 \in \mathbb{R}^{n\times 30} \quad \underset{\text{Feature selection}}{\Rightarrow} \quad X_1 \in \mathbb{R}^{n\times 16} \quad \underset{\text{Dim. compression}}{\Rightarrow} \quad X_2 \in \mathbb{R}^{n\times 3}$$

Such that all that is left to do, is to train a prediction function $f : X_2 \to Y$ that will be able to classify the compressed data correctly, in terms of selected metrics.

4.2 Model selection

The learning procedure defines a model which associates the correct label for each input sample - $y_i = f(x_i)$. The performance metric on the output space denotes the cost of wrong labeling. Here, for binary (0-1) loss : $l(\hat{y},\, y) = \mathbb{I}\{\hat{y} \neq y\}$.

4.2.1 Metrics

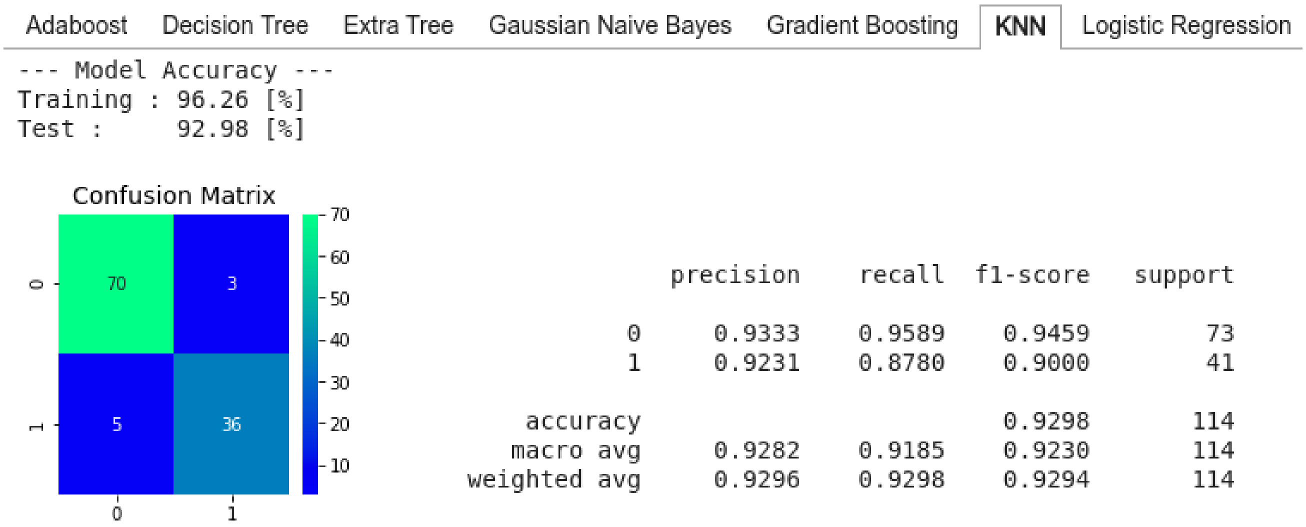
An evaluation metric is a function that measures a classifier's performance, thus allows comparison between several models. From left is a confusion matrix, which defines different combinations for each indication (**T** - True, **F** - False, **P** - Positive, **N** - Negative) :

Confusion matrix		**Predicted Class**	
		P	**N**
Actual Class	**P**	**TP**	**FN**
	N	**FP**	**TN**

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad \text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP+FN}} \qquad \text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \qquad F_1 = 2 \cdot \frac{\text{Pr} \cdot \text{Re}}{\text{Pr+Re}}$$

Combining the metrics improves understanding of the big picture, and enables to overcome data imbalance. Optimality is obtained by minimizing false indications (**FP, FN**) :



```
--- Model Accuracy ---
Training : 96.26 [%]
Test :     92.98 [%]
```

```
              precision    recall  f1-score   support

           0     0.9333    0.9589    0.9459        73
           1     0.9231    0.8780    0.9000        41

    accuracy                         0.9298       114
   macro avg     0.9282    0.9185    0.9230       114
weighted avg     0.9296    0.9298    0.9294       114
```

Note : Interactive switching between the tabs is available via the *Colab* version.

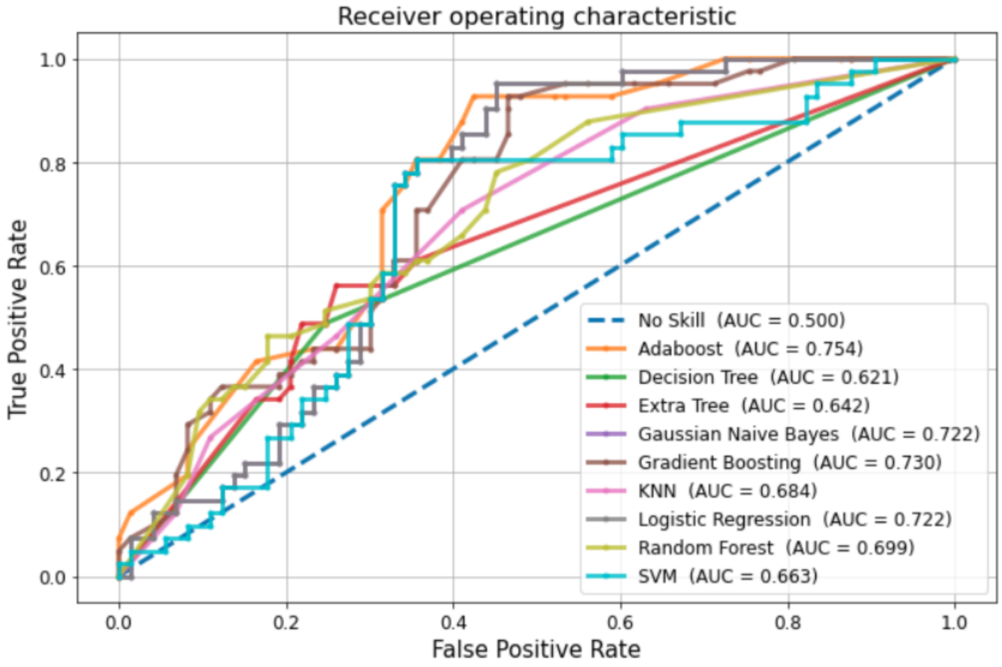
4.2.2 Classifiers

The following list of classifiers was utilized in search of an optimal candidate :

- AdaBoost
- Decision Trees
- Extra Trees
- Gaussian Naive Bayes
- Gradient Boosting
- Logistic Regression
- KNN
- SVM
- Random Forest

An **ROC** reflects a binary classifier ability to discriminate classes, using a probabilistic analysis. Each threshold is a point on the **ROC** graph, denoting the TPR/FPR tradeoff.

- What would happen if we took features that scored poorly in the correlation matrix ?



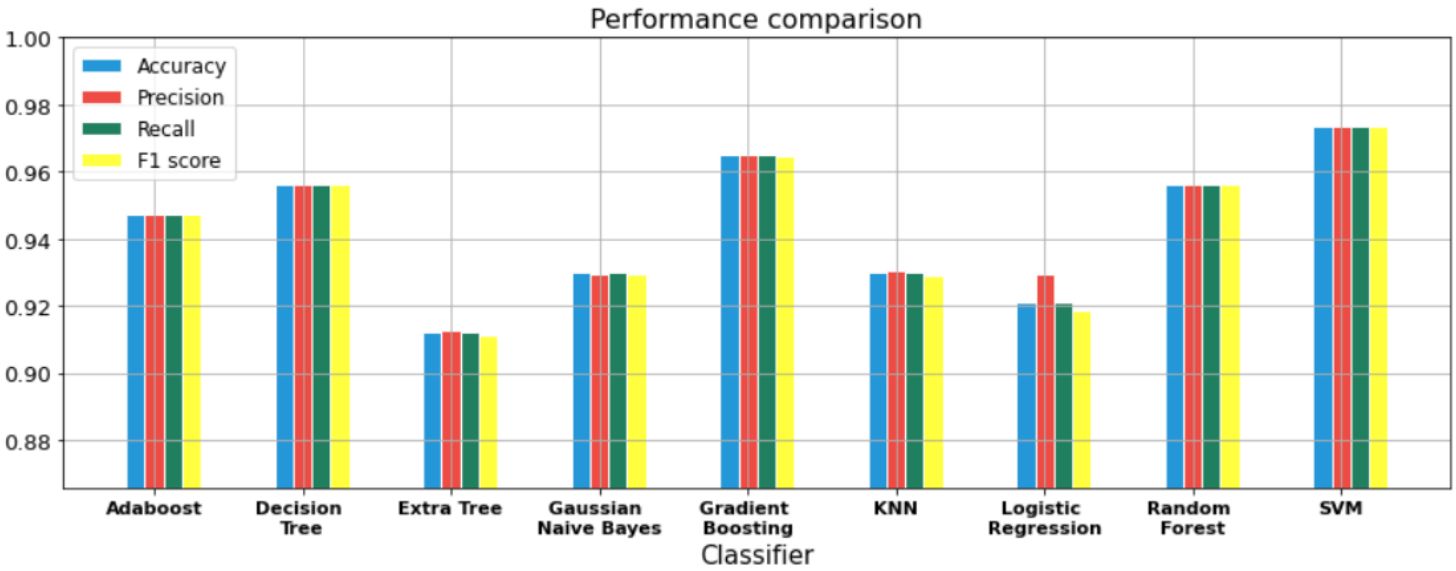
The **AUC** is the area under the **ROC**, which expresses the prediction success rate from 0-1. Features with low-ranked contribution to the explained variable, lead to poor performances, slightly above a random decision / ”No skill” (**AUC**=0.5). Ideally, the perfect classification will exhibit a Γ-shape that crosses the (0, 1) point in the FPR-TPR plane. Meaning that there exists a threshold with 100% correct indications.

Next is the result section, implementing the pipeline on the full dataset :

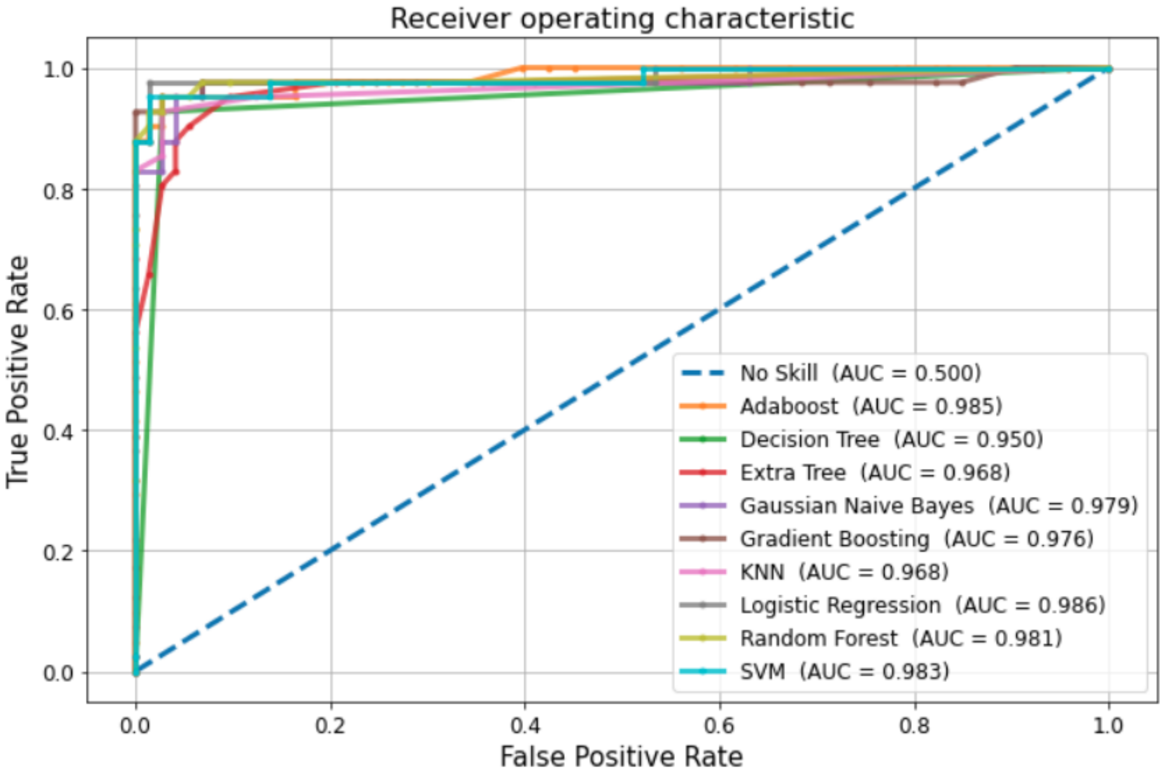
$$X_0 \in \mathbb{R}^{n\times 30} \quad \underset{\text{Feature selection}}{\Rightarrow} \quad X_1 \in \mathbb{R}^{n\times 16} \quad \underset{\text{Dim. compression}}{\Rightarrow} \quad X_2 \in \mathbb{R}^{n\times 3}$$

5 Results

Below is a comparative analysis of the chosen classifiers :



Similarly, using the FPR/TPR values for calculating the **AUROC** :

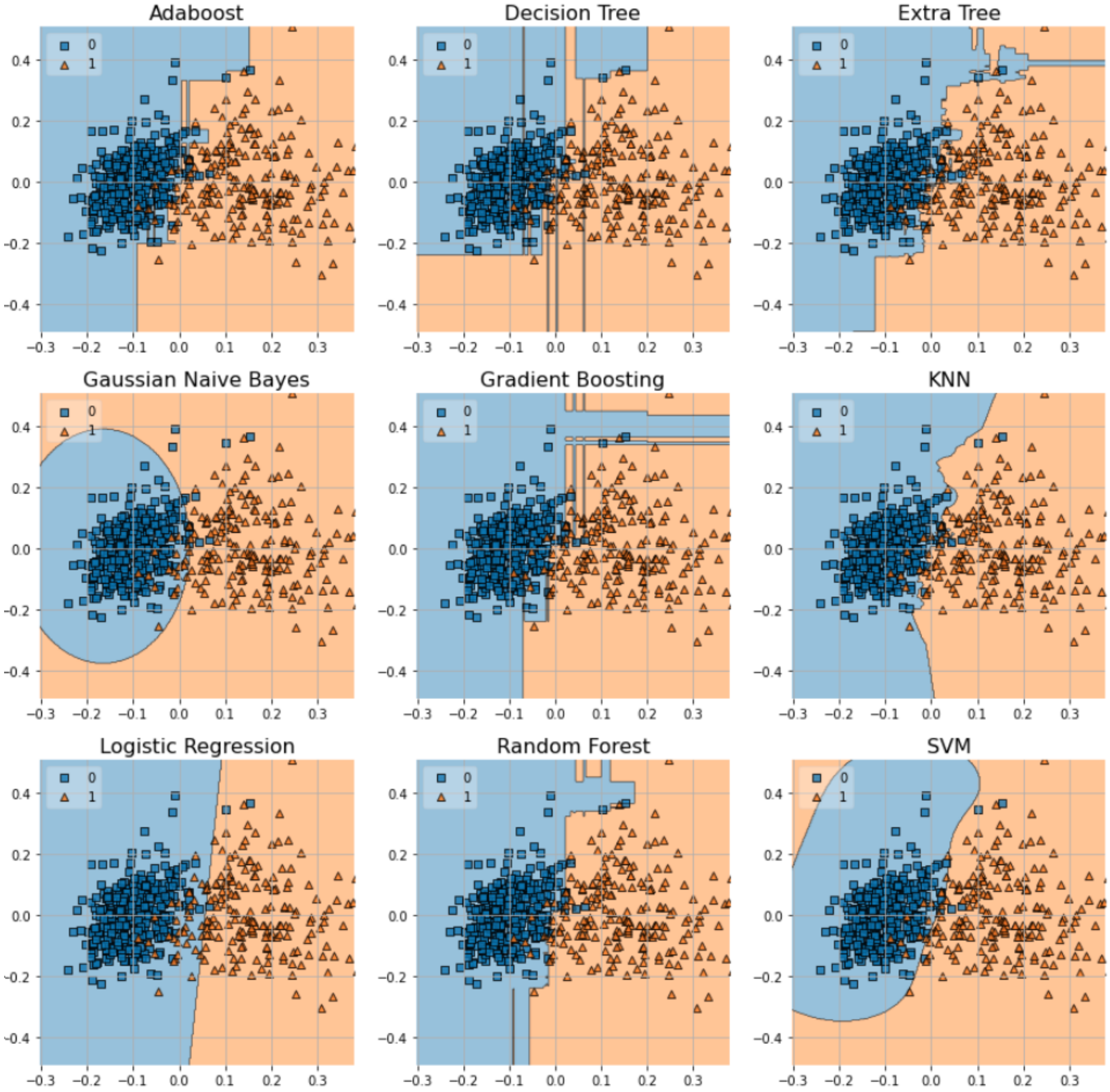


Algorithm	Accuracy	F_1 score	AUC
SVM	97.38%	97.28%	0.9833
Gradient Boosting	96.57%	96.51%	0.9761
Random Forest	95.66%	95.60%	0.9811

Top-3 classifiers

5.1 Bonus : decision boundary

Consider an additional PCA, this time to a 2D plane. Using the amazing [mlxtend] library, the decision boundary obtained by each one of the classifiers, can be demonstrated :



6 Discussion

In this project I examined the use of several classification techniques for breast cancer diagnosis, after reducing the raw data to a lower representation form. All ML algorithms exhibited high performance on the binary classification, as measured by the chosen metrics.

As a project course, the scope of work is eventually limited to an amount of 15 pages long. Therefore, hyperparameter optimization and other useful heuristics were not included.

6.1 Limitation of the study

Prior engineering : The dataset as it is publicly available, is already after analysis with specific features, chosen by the researchers. Thereby, the user has no access to the raw image scans at full dimensionality. Thus leaving fewer options for action.

Low-dimensional representation : The process of extraction (by correlation), and then projection (by PCA), ends up with 10 times smaller feature space. Both operations rely on existence of linear relationships between the features. This hidden presumption may not always be true, and it might ignore important features that simply fail to score well.

Dataset aspect : The dataset contains a modest amount of samples, distributed at an imbalanced (B/M) ratio of 1.684. During training, most classifiers reached an accuracy of 100% (unlike test results), probably due to overfitting. However, rebalance technique that involved up-sampling of the malignant class, did not show any dramatic improvement.

6.2 Conclusions

Prediction model for classifying a target variable is statistical by nature. The results range crossed the 90% accuracy bar. Future work should refer to the limitation of the current project, and can be further challenged by unsupervised learning techniques.

To sum up, I find the project very instructive as I saw by myself how machine learning tools can be implemented wisely in context of healthcare, and yield satisfactory results.

References

[1] NCI. January 1980. Archived from the original on 25 June 2014. Retrieved 29 June 2014. National Cancer Institute

[2] Saunders C, Jassal S (2009). Breast cancer (1. ed.). Oxford: Oxford University Press. p. Chapter 13. ISBN 978-0-19-955869-8. Archived from the original on 25 October 2015.

[3] McKinney, S. M., Sieniek, M., Godbole, & Etemadi, M. (2020). Google DeepMind. International evaluation of an AI system for breast cancer screening. Nature, 577(7788).

[4] Nagendran, M., Chen, Y., Lovejoy, C. A., Gordon, A. C. & Maruthappu, M. (2020). IBM research group. Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies. bmj, 368.

[5] Jeter, R., Josef, C., Shashikumar, S., & Nemati, S. (2019). Emory University. Does the" Artificial Intelligence Clinician" learn optimal treatment strategies for sepsis in intensive care?. arXiv preprint arXiv:1902.03271.

[6] Bennett, K. P., & Mangasarian, O. L. (1992). Optimization Methods and Softwares Robust linear programming discrimination of two linearly inseparable sets. Optimization methods and software, 1(1), 23-34.

[7] Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993, July). International Society for Optics and Photonics. Nuclear feature extraction for breast tumor diagnosis. In Biomedical image processing and biomedical visualization (Vol. 1905, pp. 861-870).

[8] Mangasarian, O. L., Street, W. N., & Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), 570-577.

[9] Mangasarian, Y. J., & Wolberg, W. H. (2000). Breast cancer survival and chemotherapy: a support vector machine analysis. Discret Math Probl with Med Appl Work Discret (DIMACS) December 8–10, 1999, Volume 55 (p. 1).

[10] Pantazi, S., Kagolovsky, Y., & Moehr, J. R. (2002). Cluster analysis of wisconsin breast cancer dataset using self-organizing maps. Studies in health technology and informatics (SHTI), 431-436.

[11] Revett, K., Gorunescu, F., Gorunescu, M., El-Darzi, E., & Ene, M. (2005, November). A breast cancer diagnosis system: a combined approach using rough sets and probabilistic neural networks. In EUROCON 2005-The International Conference on" Computer as a Tool" (Vol. 2, pp. 1124-1127). IEEE.

[12] Huang, M. L., Hung, Y. H., & Chen, W. Y. (2010). Neural network classifier with entropy based feature selection on breast cancer diagnosis. Journal of medical systems.

[13] Belciug, S., Salem, A. B., Gorunescu, F., & Gorunescu, M. (2010). Clustering-based approach for detecting breast cancer recurrence. In 2010 10th International Conference on Intelligent Systems Design and Applications (pp. 533-538). IEEE.

[14] Karthik, S., Perumal, R. S., & Mouli, P. C. (2018). Breast cancer classification using deep neural networks. In Knowledge Computing and Its Applications. Springer.

[15] Abdel-Zaher, A. M., & Eldeib, A. M. (2016). Breast cancer classification using deep belief networks. Expert Systems with Applications, 46, 139-144.

[16] Nembrini, S., König, I. R., & Wright, M. N. (2018). The revival of the Gini importance? Bioinformatics, 34(21), 3711-3718.

[17] Lin, X., Yang, F., Zhou, L., Yin, P. & Xu, G. (2012). A support vector machine-recursive feature elimination feature selection method based on artificial contrast variables and mutual information. Journal of chromatography B, 910, 149-155.

[18] Holland, S. M. (2008). Principal components analysis (PCA). Department of Geology, University of Georgia, Athens, GA, 30602-2501.

Images sources

- Page 1 : British scientists develop a genetically modified virus. [dailymail.co.uk]
- Page 1 : Differences Between a Malignant and Benign Tumor. [pinterest.com]
- Page 2 : Benign and Malignant Breast Lumps. [verywell health.com]

Appendices

These part contains developments of the methods used along the work. Many of the explanations were taken from Wikipedia as it delivers clear and instructive details.

Appendix A - Correlation matrix

A correlation matrix is a table showing correlation coefficients between every two variables. Each cell in the table shows the Pearson correlation coefficient [ref] :

$+1$: perfect **positive** linear correlation
0 : **no** linear correlation
-1 : perfect **negative** linear correlation

(i) **feature** vs. **feature** - Every pair of two independent variables :

$$\operatorname{corr}(X) = \begin{bmatrix} 1 & \frac{\mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)]}{\sigma(X_1)\sigma(X_2)} & \cdots & \frac{\mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)]}{\sigma(X_1)\sigma(X_n)} \\ \frac{\mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)]}{\sigma(X_2)\sigma(X_1)} & 1 & \cdots & \frac{\mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)]}{\sigma(X_2)\sigma(X_n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)]}{\sigma(X_n)\sigma(X_1)} & \frac{\mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)]}{\sigma(X_n)\sigma(X_2)} & \cdots & 1 \end{bmatrix}$$

Unlike the off-diagonal entries, the principal diagonal denotes the correlation of each random variable with itself (= 1).

(ii) **feature** vs. **target** - Every pair of independent variable with respect to the target.

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

Due to the symmetry, it is suffice to present the lower triangular.

Appendix B - Principal component analysis

"PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on" [ref].

Consider data matrix $X \in \mathbb{R}^{n \times p}$. Its k-th component variance can be iteratively maximized, by subtracting the first $k-1$ principal components (PC) from $\hat{X}_k$:

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^{\mathrm{T}}$$

Then finding the weight vector which maximizes the variance of the new data matrix :

$$\mathbf{w}_{(k)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\} = \arg\max \left\{ \frac{\mathbf{w}^T \hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

After k iterations, the procedure gives the remaining eigenvectors of $X^T X$ with the maximum values for the quantity in brackets given by their eigenvalues - $\mathbf{T} = \mathbf{X}\mathbf{W}$. $\mathbf{W}$ is a p-by-p weights matrix whose columns are the eigenvectors of $X^T X$, and the corresponding eigenvalues are sorted in a descending order.

Note : PCA can be also obtained by calculating the covariance matrix or by **SVD**.

Appendix C - Code Access

The project and the code are completely accessible at for anyone who is interested :

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.9343&rep=rep1&type=pdf

- *fin* -