Zurich University of Applied Sciences
School of Management and Law

Department Banking, Finance, Insurance

Master of Science in Banking and Finance

HS 2021

Project 9
Biomedicine – Breast Cancer

submitted by:

Andreas Bittel
Heimstrasse 6a
9014, St. Gallen
von Turtmann-Unterems
Matriculation number: 16-605-289

Severin Marchetti
Rigastrasse 10
7000, Chur
von Horgen
Matriculation number: 15-535-222

Fabio Tanchis
Schulstrasse 46
8451, Kleinandelfingen
von Kirchberg SG
Matriculation number: 16-570-210

submitted to:

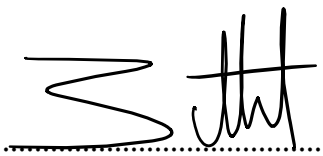
Dr. Bledar Fazlija
Department of Banking, Finance, Insurance

Winterthur, 08.01.2022

Statement of truth

"We hereby declare that we have written this thesis independently, without the assistance of third parties and using only the sources indicated, and that we will not hand out copies of this thesis to third parties without the written consent of the course director."
At the same time, all rights to the work are assigned to the Zurich University of Applied Sciences (ZHAW). The right to citation of authorship remains unaffected.

Student name
Andreas Bittel

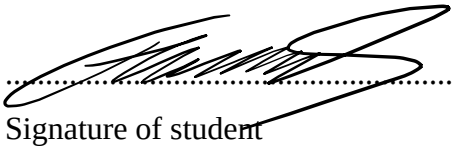


.....
Signature of student

Student name
Severin Marchetti

.....
Signature of student

Student name
Fabio Tanchis



.....
Signature of student

Abstract

With a Breast-cancer dataset, which contains 30 features of 569 patients, as a basis, there have already been several attempts to increase the accuracy of the cancer diagnosis with the support of machine learning tools. The aim of this project is to implement the Gaussian Naïve Bayes, K-Nearest Neighbours, Support Vector Machines (SVM) models and apply the Stacking method to determine the most precise and stable prediction. As the results reveal, the Linear Kernel SVM appears to perform best overall (Recall: 97.83 %, Precision: 97.83 %, F1: 97.83 %) whereas the Nearest Centroid Classifier shows the worst performance compared to the other models (Recall: 97.83 %, Precision: 83.33 %, F1: 90.00 %). In this project the ensembled learning method Stacking is used to demonstrate more robust results (Recall: 100.00 %, Precision, 93.88 %, F1: 96.84 %). Future works could include further research on ensembled learning methods.

Table of contents

List of figures

List of tables

List of abbreviations

BernoulliNB	Bernoulli Naïve Bayes
CategorialNB	Categorial Naïve Bayes
ComplementNB	Complement Naïve Bayes
et al.	et alia
f.	following(s)
FN	false negative
FNAs	fine needle aspirations
FP	false positive
GNB	Gaussian Naïve Bayes
GaussianNB	Gaussian Naïve Bayes
K	nearest neighbours
KNN	Nearest neighbour method
MMH	Maximal Margin Hyperplane
MultinomialNB	Multinomial Naïve Bayes
NBC	Naïve Bays Classifier
NCC	Nearest Centroid Classifier
NNC	Nearest Neighbors Classification
NNR	Nearest Neighbors Regression
p.	page(s)
se	standard error
SVC	Support Vector Classifier
SVM	Support Vector Machines
TN	true negative
TP	true positive
ZHAW	Zurich University of Applied Sciences

1 Introduction

1.1 Definition of the problem

The importance of a correct cancer diagnosis is enormous and even more so when it concerns the most common type of cancer in Switzerland, breast cancer (krebsliga.ch, 2021). With the base of the collected data of the Breast Cancer Wisconsin Data Set, the three machine learning models Gaussian Naïve Bayes (GNB), K nearest neighbours (KNN) and Support Vector Machines (SVMs) are used in this project. These models are to predict whether the tumors are benign or malignant. The data is obtained by fine needle aspirations in tumor cells. In a subsequent test, the trained models are tested for the relevant key figures of their diagnoses. In a final step the ensembled learning method stacking is applied to test wether the model is more robust.

1.2 Possible applications

This type of diagnosis can be used for any type of cancer. The large amount of available data collected by the medical examination would have to be limited by appropriate tests. The possible applications of these machine learning models are not limited to cancer research or even health care. In principle, these models can be used in every possible area. It is advised to apply several models and select the best-performing ones. In a further step beyond this project's scope, feature selection could be applied to the best performing models. Thus, clinics with fewer professionals and equipment do not need to conduct always every test. With a selected number of tests (features), a fairly confident diagnosis might be given.

1.3 Aim of the project

The aim of this project is to train the machine learning models GNB, KNN, SVM and Stacking in order to determine the most reliable model. The evaluation is then based on different key metrics. The results are always reflected on their meaningfulness.

2 Literature review

According to the creators of this dataset, this data was originally used with the goal to increase the correctness of the diagnosis of breast tumours where a cell sample has been taken using fine needle aspirations (FNAs) (Street et al., 1993). At the time, FNAs were a more accessible alternative to a full biopsy, but diagnoses using this procedure were too imprecise. Therefore Street et al. (1993) used image processing and machine learning techniques in their paper to get to higher accuracies.

In their first paper, they used a so-called MSM-Tree, a Multi-surface Method, to classify the data. In this method, a linear programming model is used to place a series of separating planes in the feature space. To get to a classifier that adapts better to new, unseen cases, they minimized the number of separating planes and the number of features used. They ended up by using the three feature values: mean texture, worst area and worst smoothness. With this single-plane classifier they could separate 97.3 % of the cases successfully. To see how the model performs on new data, they performed a ten-fold cross-validation and achieved an accuracy of 97.0 % (Street et al., 1993).

In the second paper of the same authors (Mangasarian et al., 1995), they even get to an accuracy of 97.5 % with the same model. Moreover, they claim that this model was used in practice and has achieved 100.0 % correctness in 131 consecutive new cases that have been diagnosed (Mangasarian et al., 1995, p. 4).

As the data provider itself clearly shows, the Breast Cancer Wisconsin (Diagnostic) Data Set has already been part of a large number of scientific research papers (University of California, School of Information and Computer Science, 2019). Most of the listed papers use either more complex or other models than the ones used in this paper. Nevertheless, the method's accuracies in the two papers of the data creators mentioned above can be used as a good benchmark.

3 Methodology

3.1 Data preparation

In a first step, the data set is imported and examined with the help of Python. Attention is paid to which data are present, how many and what type they are. The existing dataset contains the data of 569 patients and all attributes, except Diagnosis, are expressed by numerical values. The following ten features were measured for each cell:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Every patient then gets an ID number attributed and the result if the examined cell was malignant or benign. Since several measurements were taken, the scientists calculated the 1) mean, 2) standard error (se) and 3) the largest (= worst) measurement. Therefore, instead of ten, this project handles 30 features of each patient.

To be able to work with it better in the further course of the project, the values of the attribute "Diagnosis" are converted into numerical values. Accordingly, there are two possible values in this attribute. The values previously stored with M for malignant are now overwritten with 1 and those stored with B for benign are overwritten with 0. At the same time, the ID feature is deleted, as it does not contribute to further information.

The prevalence of positive values in the Diagnosis attribute is 37.26 % (see figure 1). This means that a malignant tumour was diagnosed in 37.26 % of the cases in the data set. The data of this dataset is not balanced.

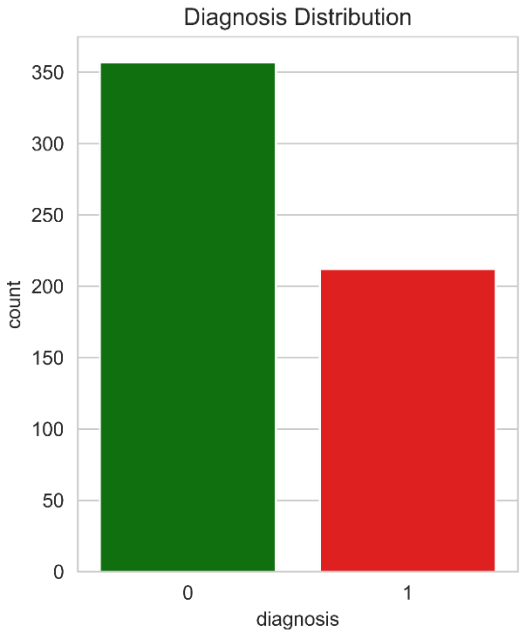


Figure 1: Diagnosis Distribution

Next, an overview of the data is created.

	radius_mean	textrue_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave_pts_mean	symmetry_mean	fractal_dim_mean	radius_se
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	0.405172
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	0.277313
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	0.111500
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	0.232400
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	0.324200
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	0.478900
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	2.873000

Table 1: Overview dataset mean (own representation).

	textrue_se	perimeter_se	area_se	smoothness_se	compactness_se	concavity_se	concave_pts_se	symmetry_se	fractal_dim_se	radius_worst
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	1.216853	2.866059	40.337079	0.007041	0.025478	0.031894	0.011796	0.020542	0.003795	16.269190
std	0.551648	2.021855	45.491006	0.003003	0.017908	0.030186	0.006170	0.008266	0.002646	4.833242
min	0.360200	0.757000	6.802000	0.001713	0.002252	0.000000	0.000000	0.007882	0.000895	7.930000
25%	0.833900	1.606000	17.850000	0.005169	0.013080	0.015090	0.007638	0.015160	0.002248	13.010000
50%	1.108000	2.287000	24.530000	0.006380	0.020450	0.025890	0.010930	0.018730	0.003187	14.970000
75%	1.474000	3.357000	45.190000	0.008146	0.032450	0.042050	0.014710	0.023480	0.004558	18.790000
max	4.885000	21.980000	542.200000	0.031130	0.135400	0.396000	0.052790	0.078950	0.029840	36.040000

Table 2: Overview dataset se (own representation).

	textrue_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave_pts_worst	symmetry_worst	fractal_dim_worst	diagnosis
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	25.677223	107.261213	880.583128	0.132369	0.254265	0.272188	0.114606	0.290076	0.083946	0.372583
std	6.146258	33.602542	569.356993	0.022832	0.157336	0.208624	0.065732	0.061867	0.018061	0.483918
min	12.020000	50.410000	185.200000	0.071170	0.027290	0.000000	0.000000	0.156500	0.055040	0.000000
25%	21.080000	84.110000	515.300000	0.116600	0.147200	0.114500	0.064930	0.250400	0.071460	0.000000
50%	25.410000	97.660000	686.500000	0.131300	0.211900	0.226700	0.099930	0.282200	0.080040	0.000000
75%	29.720000	125.400000	1084.000000	0.146000	0.339100	0.382900	0.161400	0.317900	0.092080	1.000000
max	49.540000	251.200000	4254.000000	0.222600	1.058000	1.252000	0.291000	0.663800	0.207500	1.000000

Table 3: Overview dataset worst (own representation).

The search for the so-called N/A values as well as the zero values did not lead to any hits. Accordingly, no adjustments have to be made to the data set and the results are not distorted.

In a next step, a visual representation of the dataset is conducted by separating the mean, se and worst data (the whole dataset is visualized in the appendix) . As in the figures below can be seen the split between malignant (red) and benign (green) is more clearly represented in the dataset of the mean and the worst. Finally, a heatmap with the correlations is displayed.

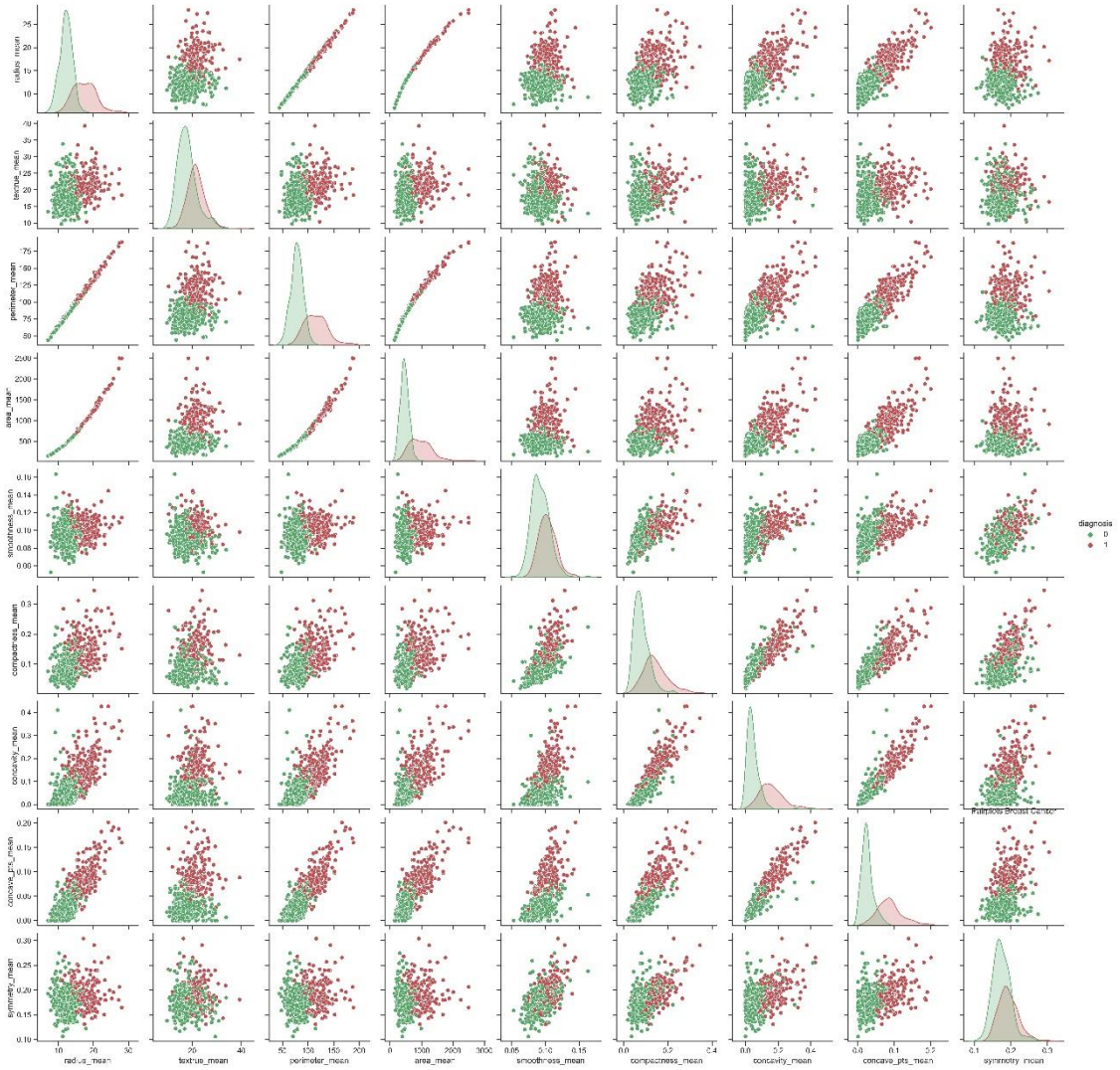


Figure 2: Scatterplot dataset mean (own representation).

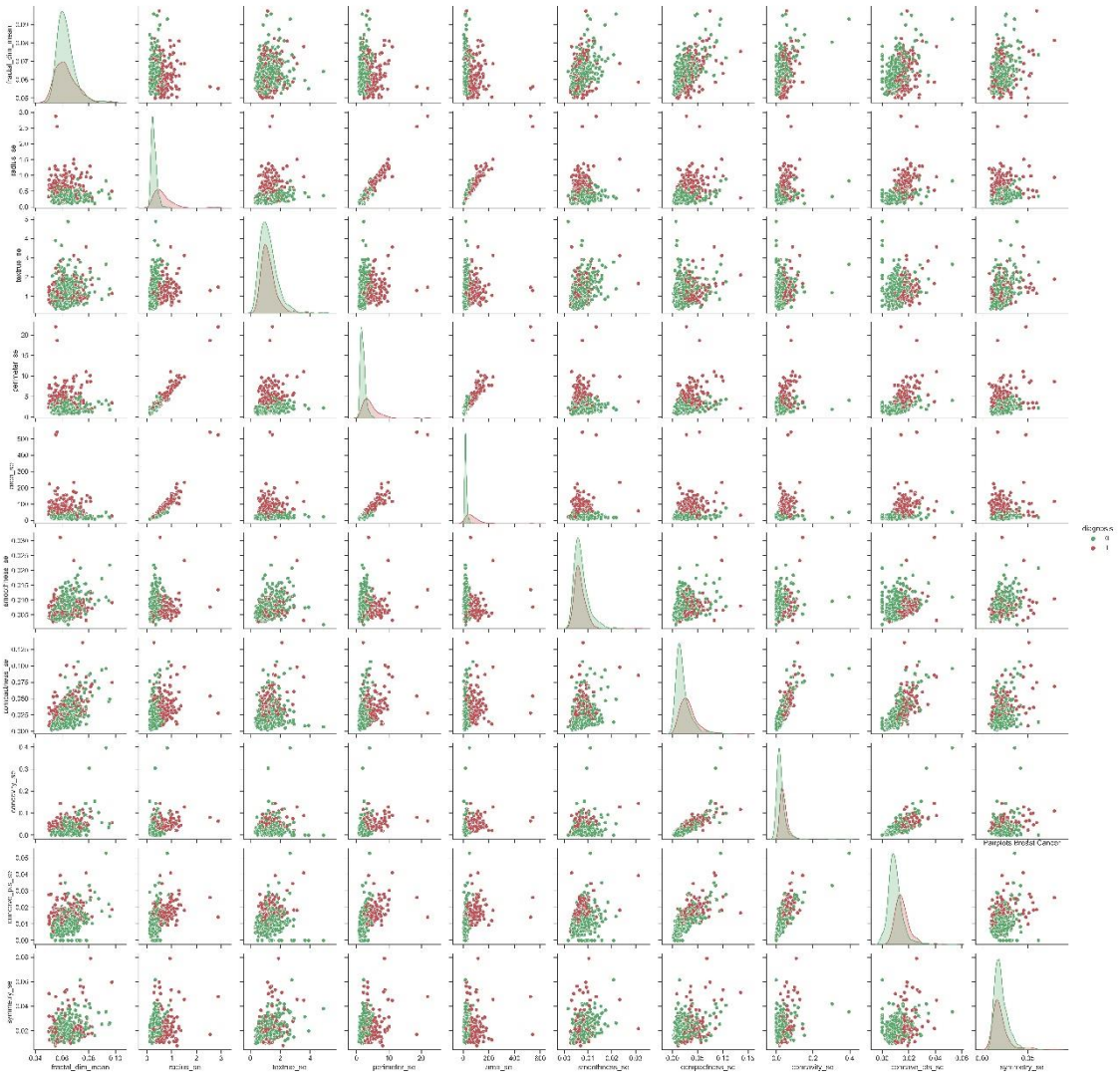


Figure 3: Scatterplot dataset se (own representation).

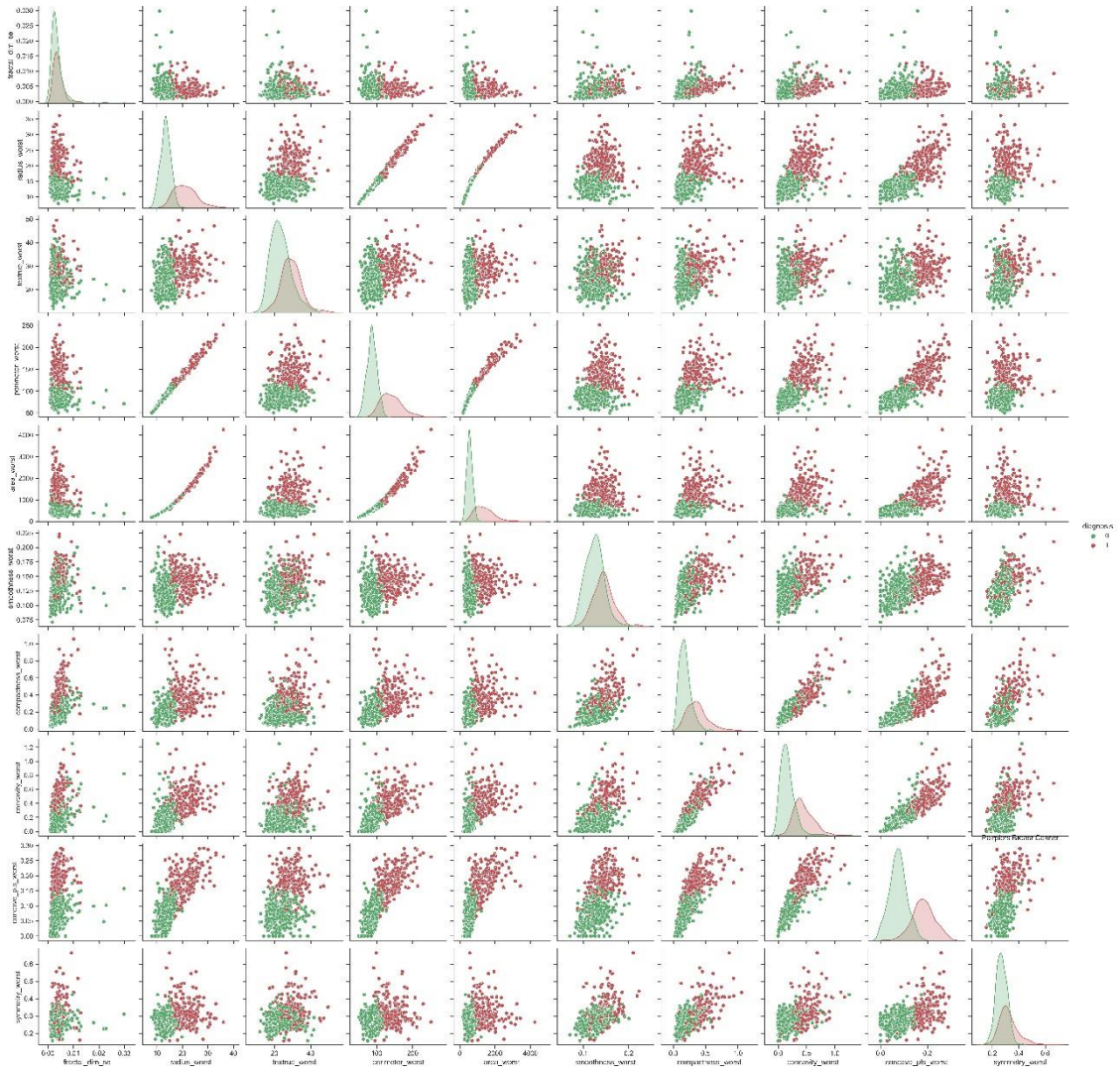


Figure 4: Scatterplot dataset worst (own representation).

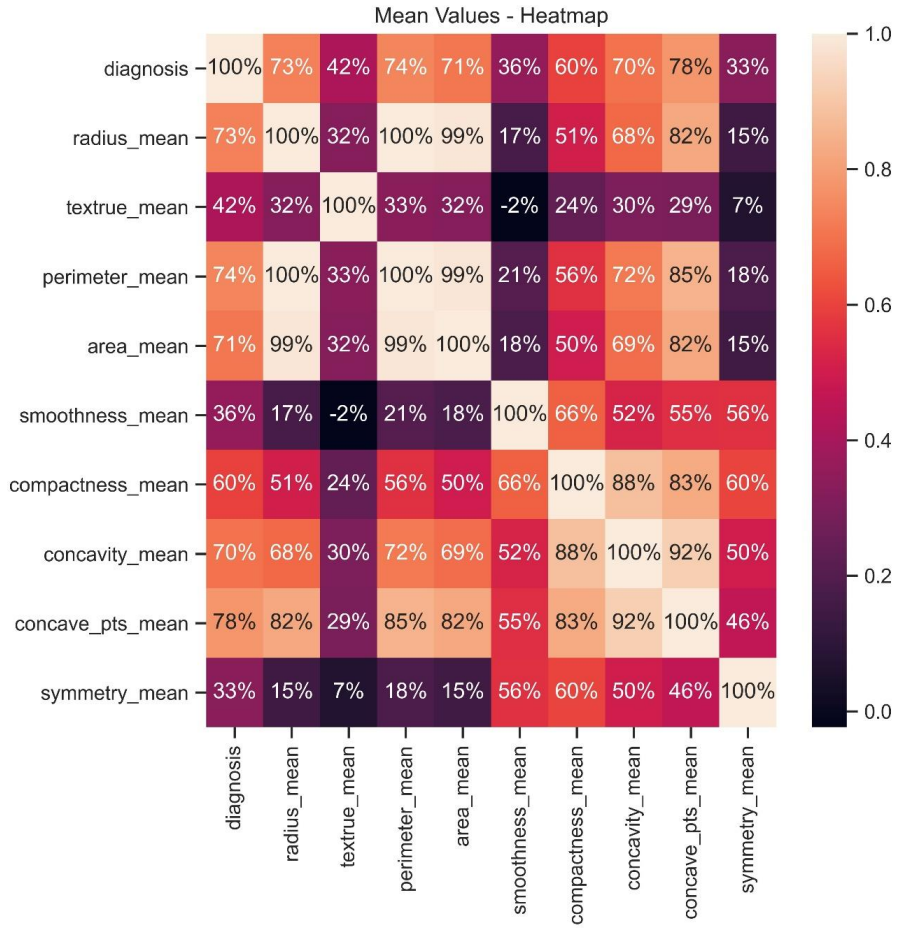


Figure 5: Heatmap dataset mean (own representation).

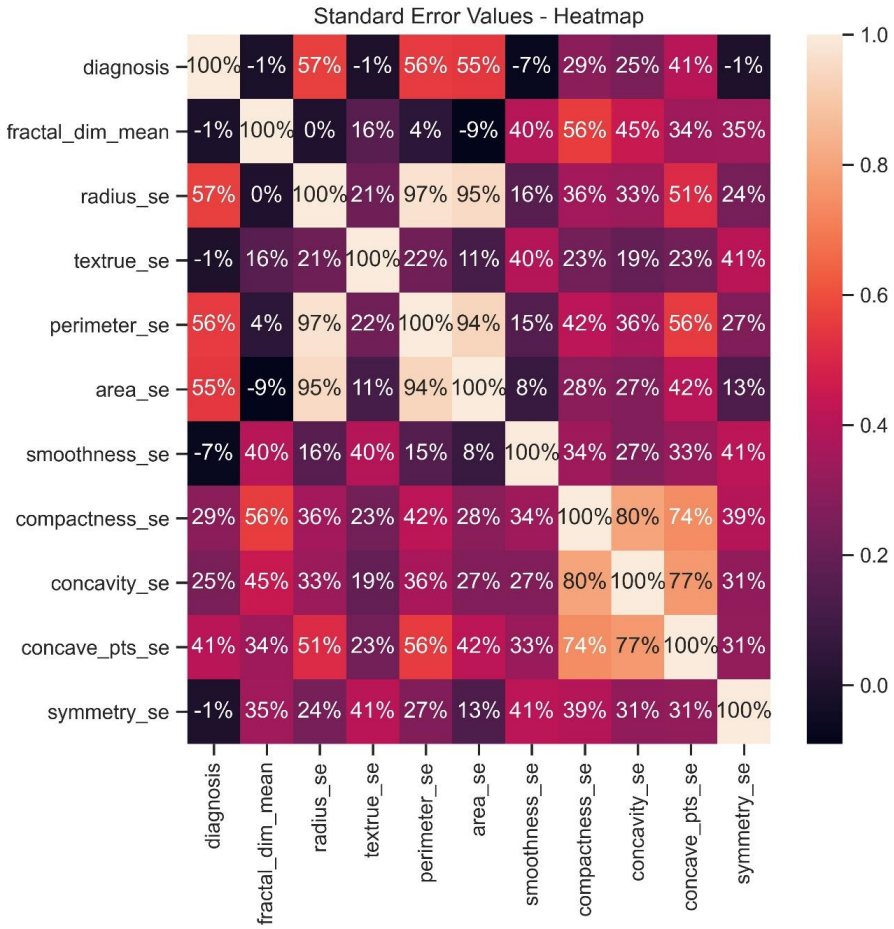


Figure 6: Heatmap dataset se (own representation).

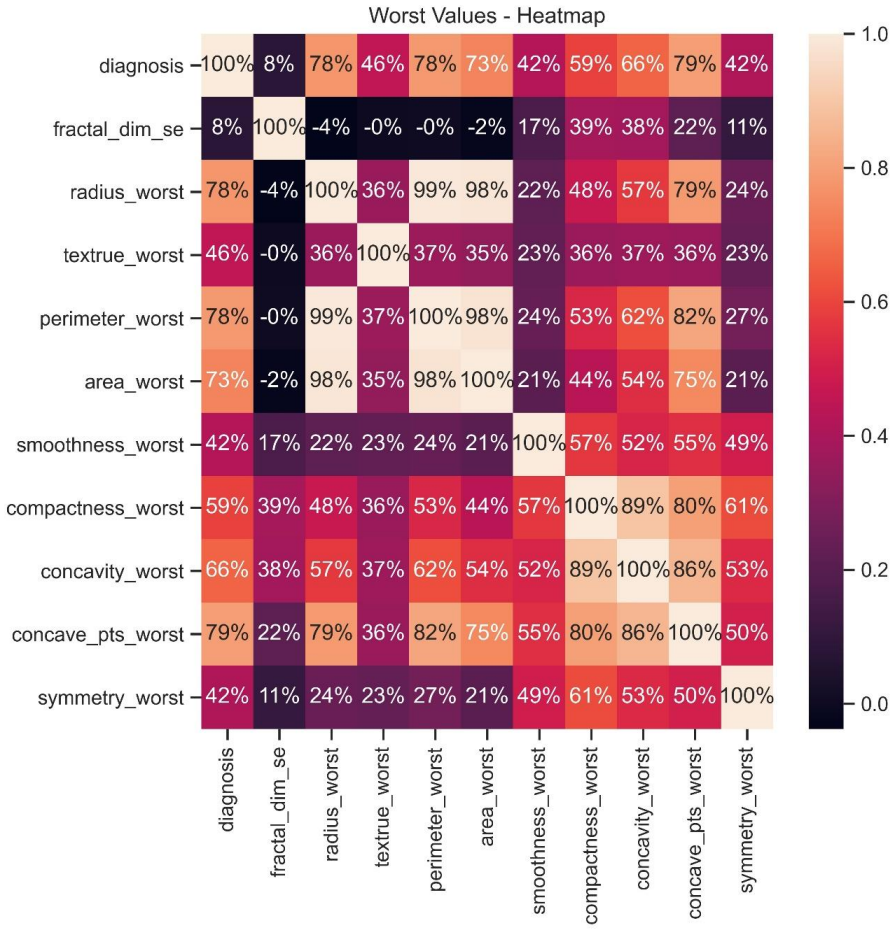


Figure 7: Heatmap dataset worst (own representation).

Then, the original data set is divided into different proportions. In this project, the training set contains 70 % of the data, the validation set 10 % and the test set 20 %.

Set	**Proportion**
Training	70 %
Validation	10 %
Test	20 %

Table 4: Dataset split (own representation).

The validation set is needed when a model contains a tuning parameter. In this case, the optimal tuning parameter is defined with the help of the training set. The model is then fitted with the optimal tuning parameter on the validation set and finally applied to the test set, to prevent overfitting.

To increase the efficiency of the algorithms, the data is normalized with the standardization technique, which is operated on each column.

$$x' = \frac{x - \mu_x}{\sigma_x}$$

Where

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)^2}$$

In a final step the diagnosis column is separated into a new dataset to enable the models to be evaluated by comparing them with the predictions.

3.2 Data evaluation

3.2.1 Confusion matrix

A confusion matrix provides a visual representation in form of a table for evaluating the performance of a predictive analysis. The number of true positives (TP), true negatives (TN), false positives (FP = type 1 error) and false negatives (FN = type 2 error) is displayed (Narkhede, 2018).

		Predicted Values	
		Positive (1)	Negative (0)
Actual Values	Positive (1)	TP	FN
	Negative (0)	FP	TN

Figure 8: Confusion matrix (based on Narkhede, 2018).

3.2.2 Classification Measures

In addition to the confusion matrix, other evaluation measures can contribute to the understanding of the results achieved (see table 5).

Name	**Formula**	**Deskription**
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Accuracy measures the correct predictions (independent of positive or negative) in relation to all predictions.
Recall	$\frac{TP}{TP + FN}$	Recall measures the accuracy of correct positive predictions in relation to all actual positive values.
Precision	$\frac{TP}{TP + FP}$	Precision measures the accuracy of correct positive predictions in relation to all positive predictions.
Specificity	$\frac{TN}{TN + FP}$	Sensitivity measures the accuracy of correct negative predictions in relation to all actual negative values
F1-Score	$2 * \frac{Recall * Precision}{Recall + Precision}$	F1 is the harmonic mean between precision and recall. This ratio is especially useful, when the data is not balanced and where off-diagonals are more relevant than diagonals.

Table 5: Classification Measures (own representation based on Suresh, 2020 & Sharp, 2021).

3.2.3 Predefined Functions

To simplify this project, a few functions were predefined since they are repeatedly applied to every model. The following definitions are predefined in the code:

- Prevalence calculation
- Calculate specificity
- Print report of results (calculation of scores)
- Create empty table
- Create confusion matrix
- Create lists and save calculation results

4 Naïve Bayes Classifier

4.1 Theoretical foundations

4.1.1 Introduction

Bayes' theorem originated with the mathematician Reverend Thomas Bayes (1702-1761) and was first published after his death. Mr Price refers to Mr Bayes' ideas in a letter to Mr Canton(Bayes, 1763). His theorem deals with conditional probabilities by working with existing or known information. Additional information also helps how the probabilities should be adjusted (Newbold et al., 2013, p. 132). The theorem is mathematically expressed as follows:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}, when\ P(B) \neq 0$$

	Term	**Meaning**
$P(A\|B)$	Posterior Probability	Probability of event A to happen, given event B has already happened.
$P(B\|A)$	Likelihood	Probability of event B to happen, given event A has already happened.
$P(A)$	Class Prior	Probability of event A to happen.
$P(B)$	Predictor Prior	Probability of event B to happen.

Table 6: Terminology Bayes' Theorem (Jason, 2017, p. 83 f.)

Bayes' theorem lays the foundations for a Naïve Bayes Classifier (NBC). The classifier is "naive" in that the conditional probabilities are calculated independently (Murphy, 2012, p. 82).

Bayes' theorem can also be applied to multiple features if the features are mutually exclusive and collectively exhaustive (Newbold et al., 2013, p. 135). The general formula for several features is expressed as follows:

$$P(A|x_1, \dots x_n) = \frac{P(x_1, \dots x_n|A) * P(A)}{P(x_1, \dots x_n)}, if\ P(x_1, \dots x_n) \neq 0$$

If the probability of $P(A|B)$ (Posterior Probability) is calculated for different hypotheses, the one with the highest probability can be selected. This hypothesis is also called «Maximum A Posteriori» (MAP) hypothesis and can be expressed as follows (Jason, 2017, p. 84):

$$MAP(A) = \max\big(P(A|B)\big) = \max\left(\frac{P(A|B) * P(A)}{P(B)}\right)$$

4.1.2 Areas of application & possible Classifier (scikit)

The libraries of Scikit-learn.org (n.d.a) are applied in this project. There exist several modules employing Naïve Bayes:

- Gaussian Naïve Bayes (GaussianNB)
- Multinomial Naïve Bayes (MultinomialNB)
- Complement Naïve Bayes (ComplementNB)
- Bernoulli Naïve Bayes (BernoulliNB)
- Categorial Naïve Bayes (CategorialNB)

Through the data preparation in chapter 3.1, it is evident that the data for this project is numerical in nature and not categorical, especially after preparation. This property excludes the following classifiers for this project: MultinomialNB, ComplementNB und CategorialNB. Solely the diagnosis is binary distributed with benign (0) or malignant (1). All other features are not binary distributed, so the BernoulliNB classifier is not suitable for this project either. This leaves the GaussianNB Classifier, which is able to handle numerical and non-binary distributed data. This classifier assumes that the data of each feature follows a normal distribution with the mean μ_y and variance σ_y^2:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}$$

Figure 9 shows how the GaussianNB classifier works. It calculates the probabilities of two features at each data point and compares its normalized values afterwards. The data point to be assessed is assigned to the feature with the smaller value (here in red respectively class A).

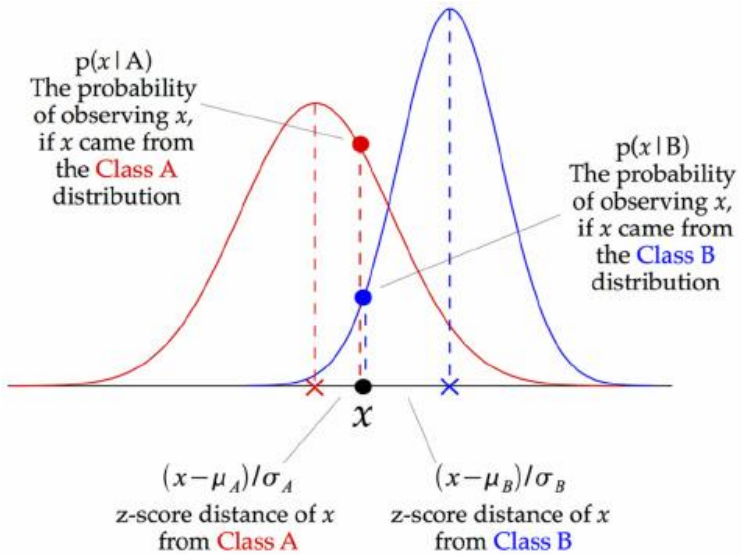


Figure 9: Exemplary representation of the GaussianNB classifier (Majumder, 2020).

4.1.3 Advantages & Disadvantages

A key advantage of the NBC is that good predictions can be achieved in an uncomplicated way with little training data (Zhang, 2004, p. 1). The already mentioned assumption that the variables are normally distributed is very strong.

4.2 Implementation & Adjustments

The GaussianNB module is imported from scikit-learn.org. In a first step, the model is trained with the training set. In this first implementation the original data and not the transformed (standardized) data is used. Afterwards, the model makes a prediction about the test set, calculates and saves the predefined scoring measurements and finally displays the confusion matrix.

Unlike this model, the other models need standardized data. The whole process is repeated with the standardized data.

4.3 Results / Interpretation

Since the confusion matrix and the key figures have already been explained in chapter 3.2.1 and 3.2.2, the focus here is on the interpretation of the results. Because the GaussianNB classifier does not require a tuning parameter, the validation set is not needed.

The GNB classifier correctly classified all but seven predictions and thus achieved a recall of 91.30 %, precision of 93.33 % and an F1 score of 92.31 % (see table 7). Even though the validation set is not used with this classifier and no tuning parameter has to be set,

this does not mean that the other models are necessarily better. When setting the tuning parameter, there is a higher risk of overfitting than with Bayes.

Compared with the standardized data (see table 8 and figure 11), the first data scored always higher. Since both models have four FN and 42 TP, they get the same recall. To conclude, it is best to use the non-pre-processed data using GNB for this project.

Model	**Accuracy**	**Recall**	**Precision**	**Specificity**	**F1**
GNB	0.9386	0.9130	0.9333	0.9559	0.9231

Table 7: Key figures Gaussian Naïve Bayes (own representation).

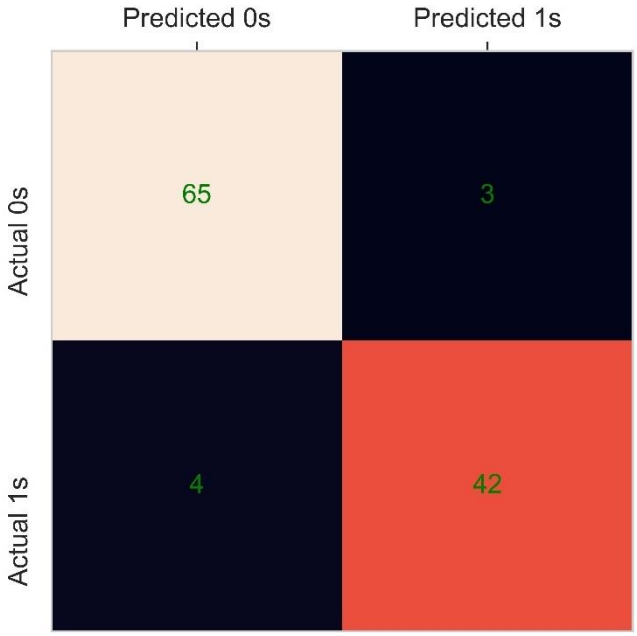


Figure 10: Confusion Matrix Gaussian Naïve Bayes (own representation).

Model	Accuracy	Recall	Precision	Specificity	F1
GNB_tf	0.9211	0.9130	0.8936	0.9265	0.9032

Table 8: Key figures Gaussian Naïve Bayes with standardised dataset (own representation).

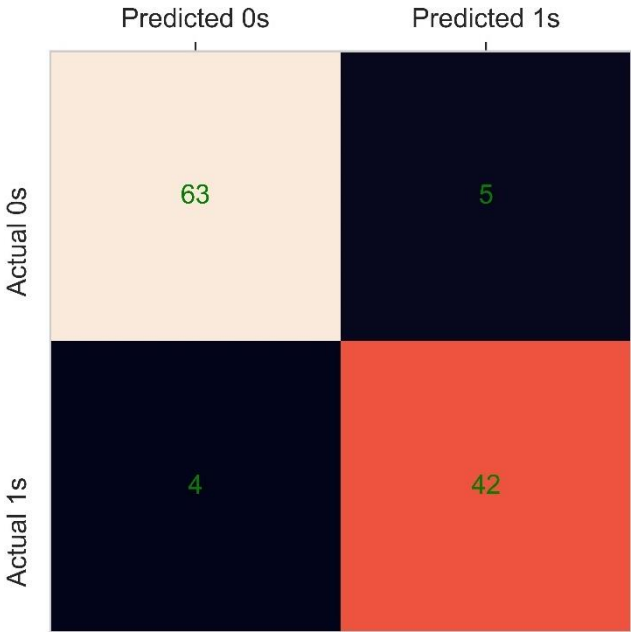


Figure 11: Confusion Matrix Guassian Naïve Bayes with standardised data (own representation).

5 Nearest neighbor method (KNN)

5.1 Theoretical foundations

5.1.1 Introduction

The nearest neighbour method (KNN) is used as a local method for data sets with a large amount of data. Various variants make it possible to implement applications to a wide range of problems, which also enables multi-label classifications and regressions (Kramer, 2013, p. 13).

The KNN method is subject to the basic assumption that regions are created when looking at the data. In these regions, data points are closer to each other and show a greater similarity (see figure 12) (Rajaguru & Prabhakar, 2017, p. 31).

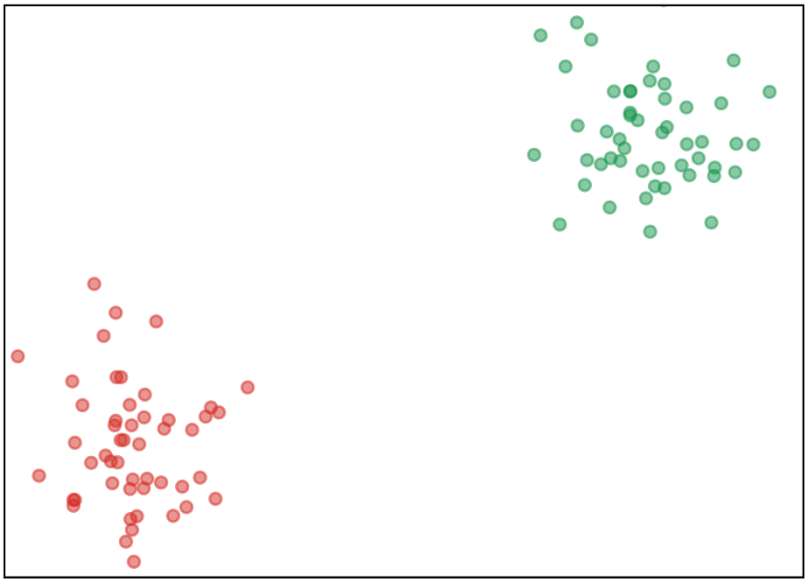


Figure 12: Density-based data (own representation).

For the classification of new data points (see figure 13, star), respectively the allocation of a region, the distances of the new data point is taken into account employing the Euclidean distances to the nearest neighbours (K) from the existing data points (von der Hude, 2020, p. 99).

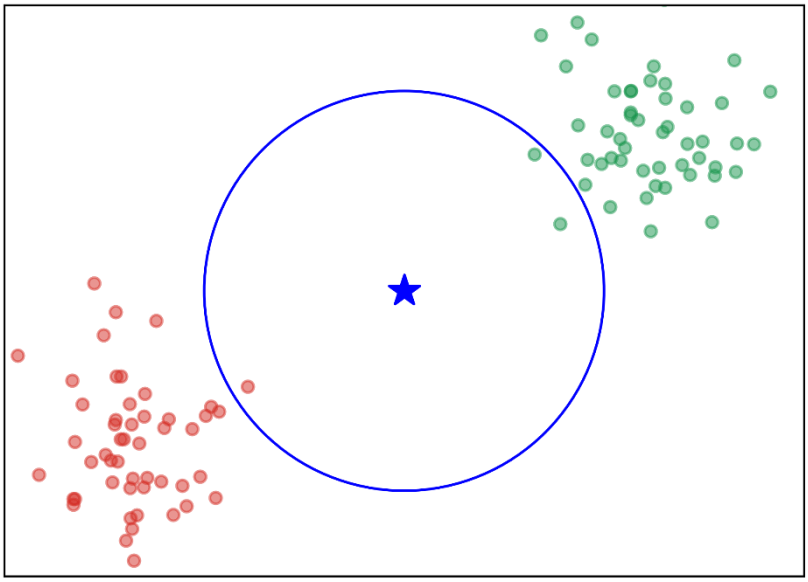


Figure 13: Classification of new data point by means of KNN (own representation).

The Euclidean distance measures the distances between the point under consideration and the existing data points in a Cartesian coordinate system and extends the Pythagorean theorem (von der Hude, 2020, p. 44).

$$L_1 - metric\text{: } L_1(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

$$L_2 - \text{metric: } L_2(\text{x},\text{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

How many neighbours this should be done with is determined by K and a simple majority vote. When determining K, it should be considered that a lower K is associated with high sensitivity to random fluctuations. Large values for K bear the risk of overfitting (von der Hude, 2020, p. 100).

5.1.2 Areas of application & possible Classifier (scikit)

For the Application of the KNN, various modules are available on scikit-learn.org:

- Nearest Neighbors Classification (NNC)
- Nearest Neighbors Regression (NNR)
- Nearest Centroid Classifier (NCC)

A closer look at the individual libraries shows that the NNC is the classic KNN method. The NNR method is particularly suitable for the application of continuous and not discrete variables, which is why it is not considered further in this project. In contrast to the

previous KNN methods, the NCC application does not compare the direct neighbours of the new data set, but considers the distance of the new point to the centre of the respective regions (see figure 14, triangle), with the target to prevent possible misclassifications due to outliers (Scikit-learn.org, n.d.b).

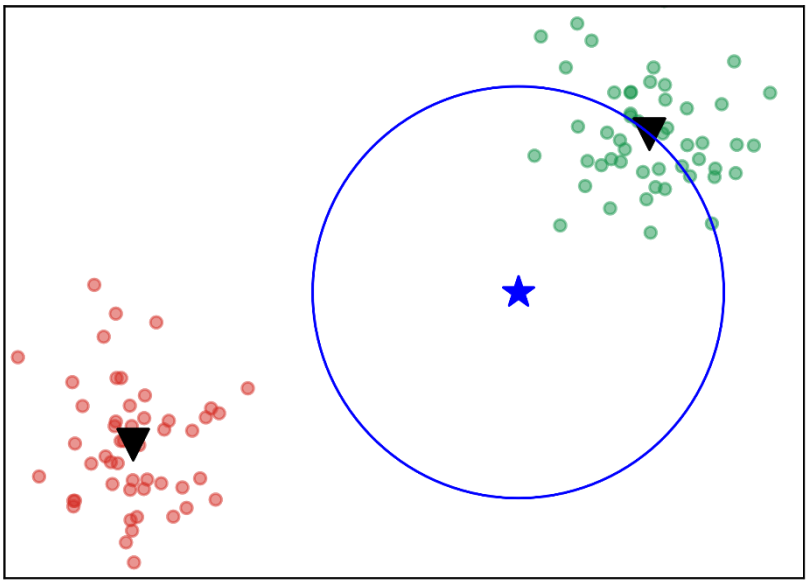


Figure 14: Classification of new data point by NCC (own representation).

5.1.3 Advantages & Disadvantages

The KNN method, as the Naïve Bayes Classifier presented earlier, has some advantages and disadvantages. KNN makes it possible to deliver meaningful results with large amounts of data and many data dimensions, even if these do not have linear correlations. Furthermore, the method is easy to understand and intuitive. Among the disadvantages, however, it becomes apparent that although the KNN method can process a large amount of data, this is also required in order to have enough training data, which means that it quickly needs a lot of computing power and storage capacity. Furthermore, at no point can it be determined whether the solution found corresponds to the optimal result or whether an improvement can still be realised (Hezel, 2015).

5.2 Implementation & Adjustments

5.2.1 Nearest Neighbors Classification (NNC)

The KNeighborsClassifier module is imported from scikit.org, the default K is set at five and the model is trained with the training data to make a prediction on the test set.
As in chapter 4.3, the key figures are calculated and a confusion matrix is created (see table 9 and figure 15).

In a next step the optimal parameter K is determined by looping the predictions with values between two and 60 for K. These calculations are performed on the training and validation sets. The results are saved in a table and displayed in figure 16.
With the optimal K, the calculation is repeated using the test set and the result is recorded in a new table of key figures and a confusion matrix (see table 10 and figure 17).

5.2.2 Nearest Centroid Classifier (NCC)

The NearestCentroid module is, as before, imported from scikit.org. The default model is fitted to the training data and the prediction is made for the test set, the key figures are calculated and the confusion matrix is created (see table 11 and figure 18). The optimal shrinking threshold is determined by looping the predictions with values between -60 and 15 and also performed on the validation set, saved and displayed (see figure 19) and used for a new calculation on the test set. Finally, the key figures and the confusion matrix are recalculated and displayed (see table 12 and figure 20).

5.3 Results / Interpretation

5.3.1 Nearest Neighbors Classification (NNC)

A recall of 89.13 %, a precision of 95.35 % and F1 of 92.13 % was already achieved in the first run. If we look at the confusions matrix (see figure 15), we see that 107 from 115 classifications were correctly predicted.

Model	Accuracy	Recall	Precision	Specificity	F1
KNN_NNC	0.9386	0.8913	0.9535	0.9706	0.9213

Table 9: Key figures KNN (NNC) (own representation).

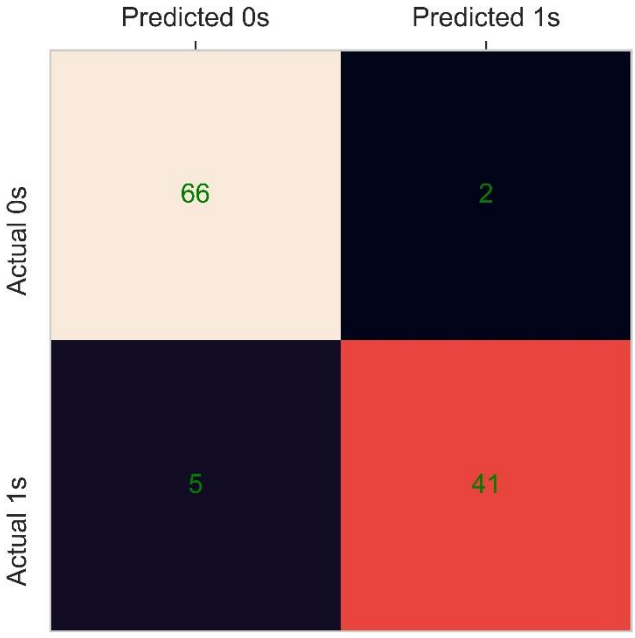


Figure 15: Results KNN (NNC) (own representation).

The plot below shows that K = 3 is the best choice since this is one value where the prediction on the validation set is the highest.

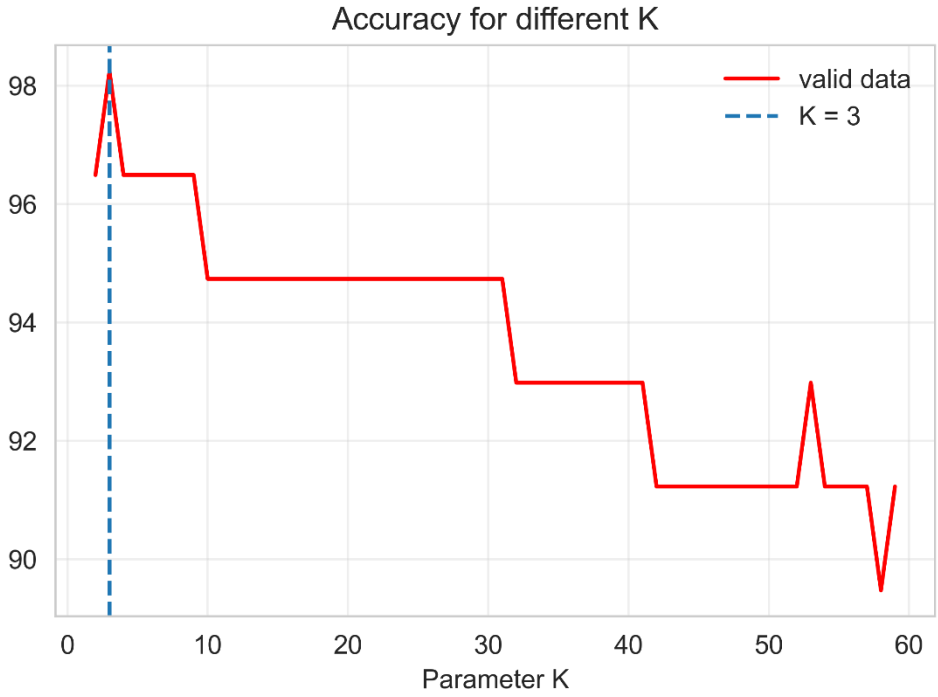


Figure 16: Accuracy for different K (own representation).

The determination of the optimal K value shows that the F1 could be increased to a value of 95.65 %. Furthermore, it can be observed that recall has improved significantly to 95.65 % as well as a slightly better precision score of 95.65 % has been achieved too. With a look at the confusion matrix (see figure 17), which was determined with the optimal K value, it can be seen that the slightly increased accuracy is due to the fact that three people who are ill were classified accordingly.

Model	Accuracy	Recall	Precision	Specificity	F1
KNN_NNC	0.9649	0.9565	0.9565	0.9706	0.9565

Table 10: Key figures KNN (NNC) with optimal K (own representation).

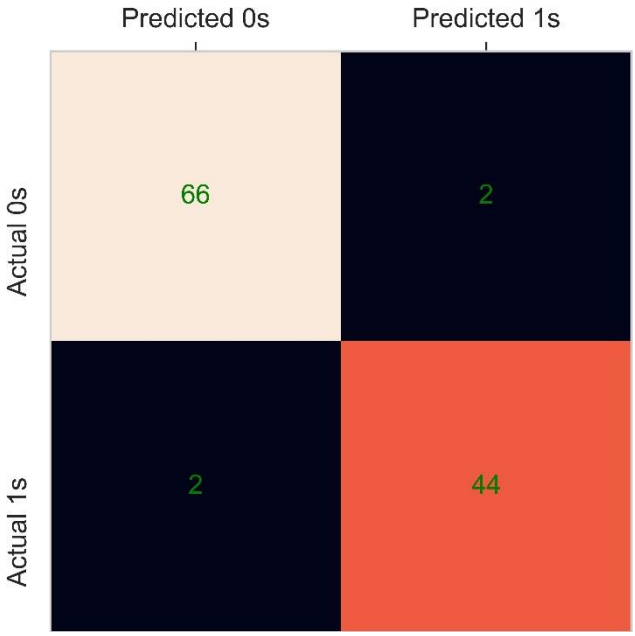


Figure 17: Results KNN (NNC) with optimal K (own representation).

5.3.2 Nearest Centroid Classifier (NCC)

With the standard NCC model, the F1 started lower compared to the NNC model with 88.89 %. On the one hand, there is a much higher recall score of 91.23 %, but on the other hand, a lower precision score of 90.91 %.

Model	Accuracy	Recall	Precision	Specificity	F1
KNN_NCC	0.9123	0.8696	0.9091	0.9412	0.8889

Table 11: Key figures KNN (NCC) (own representation).

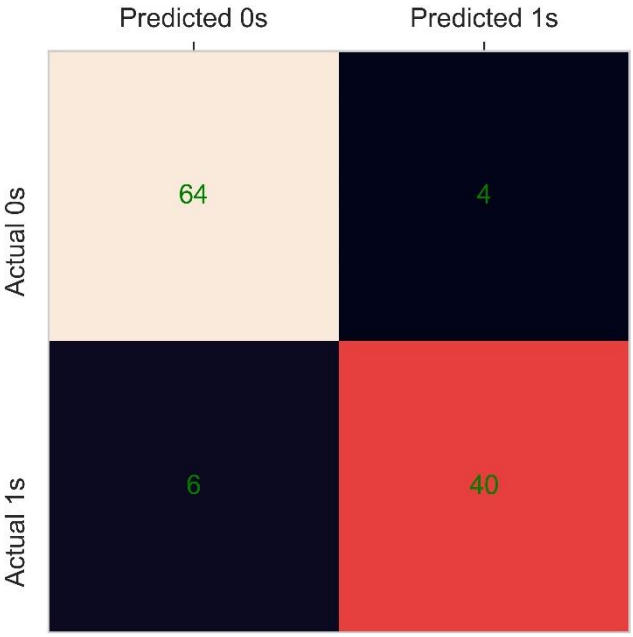


Figure 18: Results KNN (NCC) (own representation).

It is observable that the optimal shrinking threshold is getting better until it reaches the peak at K = 10 with a major drop afterwards.

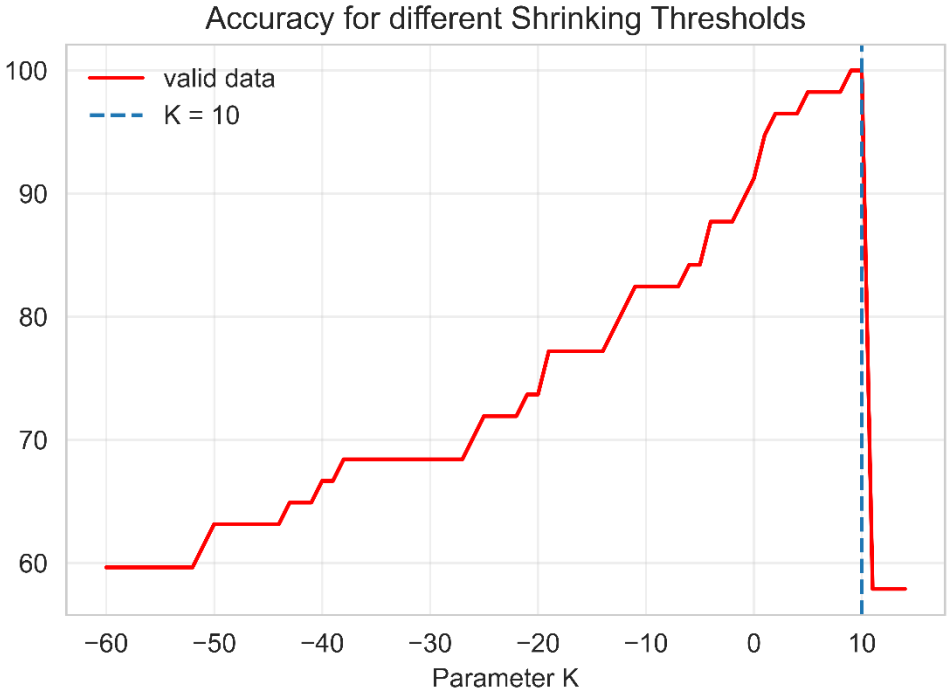


Figure 19: Accuracy for different Shrinking Thresholds (own representation).

After the determination of the optimal shrinking threshold, there is an improvement of the F1 score (90.00 %), a positive development in the recall score (97.83 %) but a negative development in the precision score (83.33 %). Even though the accuracy is lower compared with the NNC method, it is favourable to observe a higher recall due to the fact that with the NCC method, only one prediction has been detected as a FN.

Model	**Accuracy**	**Recall**	**Precision**	**Specificity**	**F1**
KNN_NCC	0.9123	0.9783	0.8333	0.8676	0.9000

Table 12: Key figures KNN (NNC) with optimal shrinking threshold (own representation).

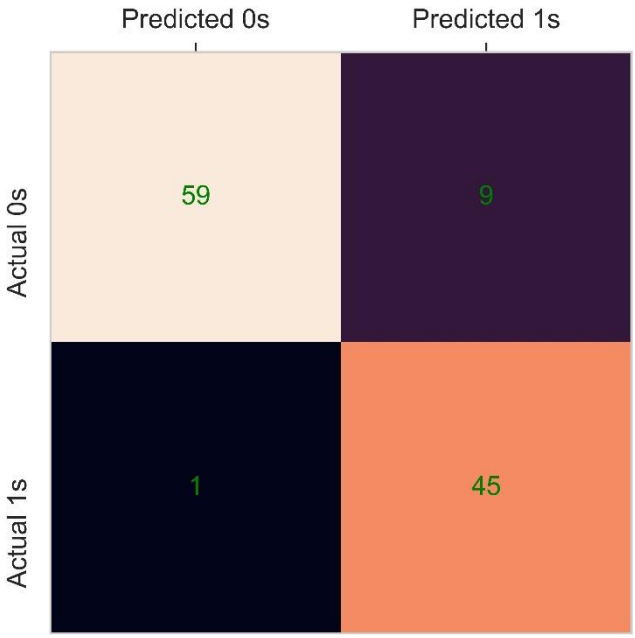


Figure 20: Results KNN (NNC) with optimal shrinking threshold (own representation).

6 Support Vector Machines (SVMs)

6.1 Theoretical foundations

6.1.1 Introduction

Support Vector Machines (SVMs) are a powerful type of supervised machine learning models that can be used to solve classification as well as regression tasks (quantstart, n.d.). The focus of this paper lies on the supervised binary classification. This means that the SVMs should be able to categorize new patients into two separate groups (e.g., benign or malignant) based on the existing data.

A SVM creates a so-called “feature space” for such a situation. The feature space is a finite-dimensional vector space in which each dimension is represented by a feature of a particular object (quantstart, n.d.). In this project, there are 30 features and therefore, the feature space has 30 dimensions. Through linear partition of this feature space, which is called a linear separating hyperplane, the SVM creates the two categories benign and malignant. If the model gets a new input of patient data, it will categorize the data according to the existing model and so decide whether the cell is benign or malignant.

A good SVM model, therefore, can be found with an optimal linear separating hyperplane. The linear hyperplane is always one dimension lower than the feature space which can be best explained by an illustration like figure 21.

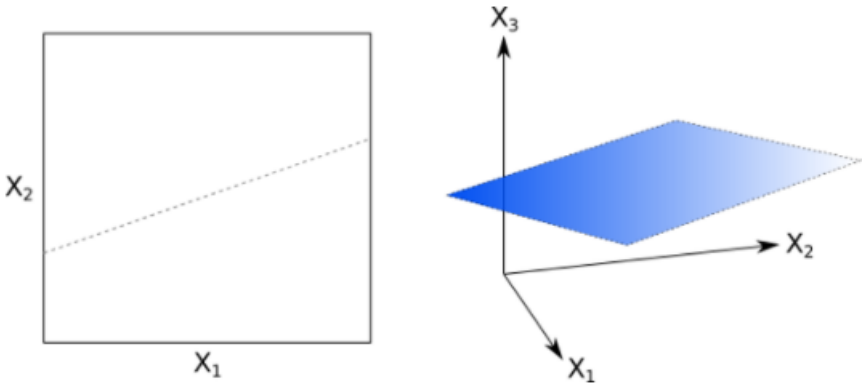


Figure 21: One- and two-dimensional hyperplanes (quantstart, n.d.).

On the left side, the feature space has two dimensions, while the hyperplane is a one-dimensional line. On the right side, the feature space has three dimensions, and the hyperplane, therefore, has two.

But the SVM does not have to separate spaces as in figure 21. It has to construct a hyperplane that separates the data points perfectly in order to be able to categorize them correctly. As illustrated in figure 22, there are several ways to separate data points from each other (left), but there is only one perfect separation (right). Unfortunately, it is not

possible to illustrate the separation of the 30 dimensions of this project. Therefore, the following figures with fewer dimensions are used to demonstrate the issue.

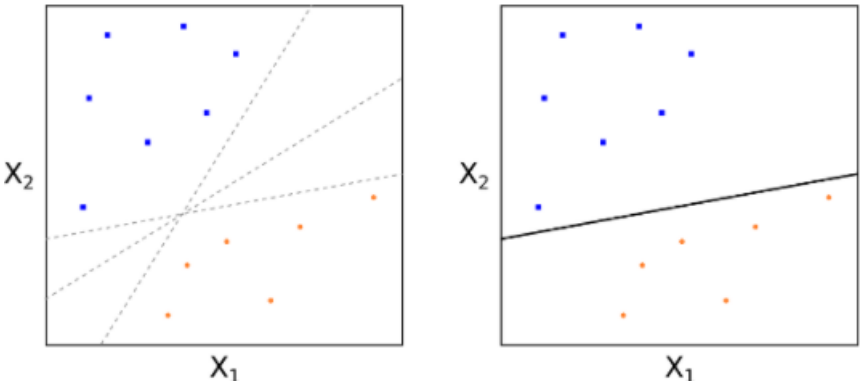


Figure 22: Finding the perfect hyperplane (quantstart, n.d.).

To determine the most optimal hyperplane, SVMs use the concept of the maximal margin hyperplane (MMH), which is the hyperplane that is farthest from any data point, resulting in the optimal separation. To find the MMH, the perpendicular distance from each data point to an existing hyperplane has to be computed (quantstart, n.d.). This distance is given by the following formula:

$$\frac{t_n(w^T x_n + b)}{\|w\|}$$

The smallest distance of a point to the hyperplane is called margin and the MMH is the one with the greatest margin. The location of the MMH only depends on the data points that are nearest to it and therefore, directly lie on the margin.

$$\arg\max_{w,b}\left\{\frac{1}{\|w\|}\min_n[t_n(w^T x_n + b)]\right\}$$

These points are called support vectors and can be seen as points A, B and C in figure 23 (quantstart, n.d.).

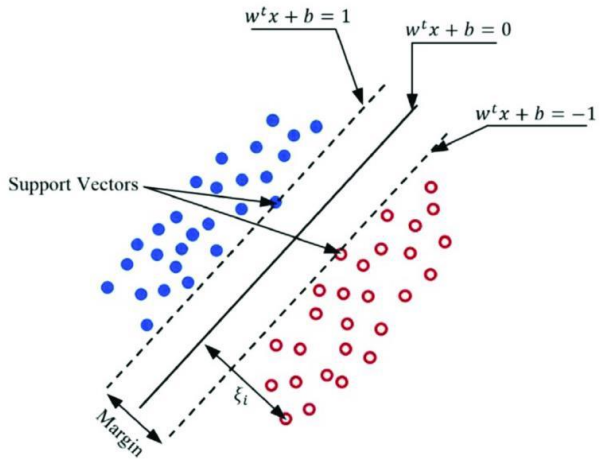


Figure 23: MMH with support vectors (Oluwaseun & Desmond, 2019).

6.1.2 Kernel Trick

If a satisfactory separation of the two-dimensional data is not possible (see figure 24, left) the so called kernel trick can be applied. The kernel trick increases the dimensionality of the feature space to enable a better separation (see figure 24, right). For example, the new dimension can be achieved by combining the existing features x_1 and x_2 (Fazlija, 2021):

$$x_3 = e^{-(x_1^2+x_2^2)}$$

$$\varphi(x_1, x_2) = \begin{bmatrix} x_1 \\ x_2 \\ e^{-(x_1^2+x_2^2)} \end{bmatrix} \in \mathbb{R}^3$$

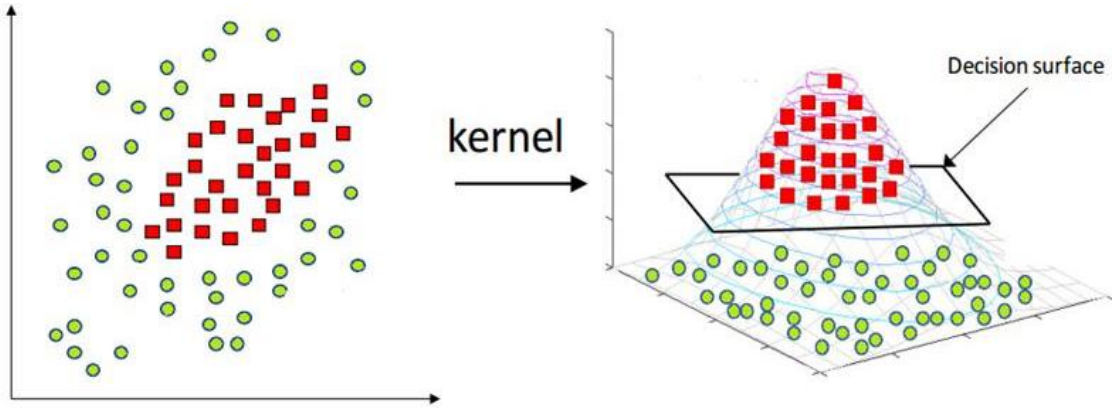


Figure 24: Kernel trick (Zhang, 2018).

6.1.3 Areas of application & possible Kernel (scikit)

SVM is a popular tool and therefore has many areas of application. The major applications are face detection, text categorization, image classification, handwriting recognition. There are also more specific applications in the medical area, such as protein or cancer classification (= bioinformatics) (Data-Flair, n.d.).

There are several parameters in a SVM that can be adjusted. The most prominent parameter is the choice of kernel. In this project, the following three kernel functions of Scikit-learn.org (n.d.c) are implemented:

- Linear: $\mathrm{K}(x, z) = x^T z$
- Polynomial: $\mathrm{K}(x, z) = (x^T z + 1)^d$
- Gaussian radial basis function (RBF): $\mathrm{K}(\mathrm{x}, \mathrm{z}) = \exp(-\gamma \|x - z\|^2)$

The regularization can be adjusted with the penalty parameter C. There is a trade-off between correct classifications of training values against the maximized margin. Simpler decision functions with larger margins are achieved by increasing C. The cost then is a lower accuracy. By decreasing C, the opposite is achieved. The classification will be more

correctly (Scikit-Learn.org, n.d.d). The hypertuning of C helps to find an ideal value that does not risk overfitting. Another trade-off is involved between the fitting time and the prediction time. A lower value for C generated more support vectors and may thus increase the prediction time (Scikit-Learn.org, n.d.d). The parameters are chosen as follows in a first step: C_param = [0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000]. To cover a large range of values and to save computing power, the grid is set largely.

Gamma is another parameter that needs to be tuned for non-linear models. With gamma the influence of one single training sample can be defined. The lower gamma is, the farther other samples can be to be affected. Increasing gamma means that the model tries to exactly fit the training data (Mothadi, 2018). The model is sensitive to the choice of gamma. If gamma is set too high, only the support vector itself is considered for the model. Adjusting the penalty C does then not have any influence to prevent the model of overfitting (Data-Flair, n.d.). The parameters are chosen as follows: gamma_param = [0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100].

The degree of the polynomial is a parameter that needs to be tuned when using the polynomial kernel. These parameters are also predefined as follows: degree_param = [2, 5, 10, 20, 30].

6.1.4 GridSearchCV

Scikit-Learn.org (n.d.e) offers the GridSearchCV to simplify the hypertuning of the previously mentioned parameters. In a first step, the following parameters are defined for the grid: C, gamma, degree and kernel. In a next step, these parameters are included into the GridSearchCV. This tool usually conducts the cross validation on predefined folds. For this project the splitting has already been done (see chapter 3.1). Therefore, this predefined split needs to be integrated with the help of the tool PredefinedSplit. To accomplish this, all validation values are attributed the value 0 and all training values are attributed the value -1. Like this the GridSearchCV can differentiate which values to use for training and which ones for the validation process. With this additional step we avoid an additional cross validation in our training data. Moreover, if this was not precised, the validation set would not be used in this model.

6.1.5 Advantages & Disadvantages

SVM comes in handy if the data is not only linear but also non-linear. Furthermore, the number of dimensions can easily be adjusted and even be larger than the number of samples (Scikit-learn.org, n.d.c). There is a large variety of applications as already mentioned above. Since many parameters such as the kernel function, penalty C or gamma can be adjusted, the SVM is very versatile (Data-Flair, n.d.).

The SVM is not able to handle text structures and therefore, sequential information (Data-Flair, n.d.). The abundance of parameter choices becomes a challenge (Scikit-learn.org, n.d.c). It bears a higher risk that one crucial parameter is not tuned appropriately.

6.2 Implementation & Adjustments

To conduct these classifications, SVC is imported. To have a benchmark, we first run each kernel on its own. The process is always the same. First, the kernel is defined, then fitted with the training data and finally, a prediction about the test set is made. The results are displayed in the same manner as for the other models. This process is repeated for the polynomial and RBF kernel.

In a second step, the models are tuned and GridSearchCV displays the best parameters based on the validation process. These parameters are then finally used to make predictions about the test set.

6.3 Results / Interpretation

The results of the program described above can be found in the figures 25, 26 and 27.

Model	Accuracy	Recall	Precision	Specificity	F1
SVM_Lin	0.9561	0.9783	0.9184	0.9412	0.9474

Table 13: Key figures Linear kernel (own representation).

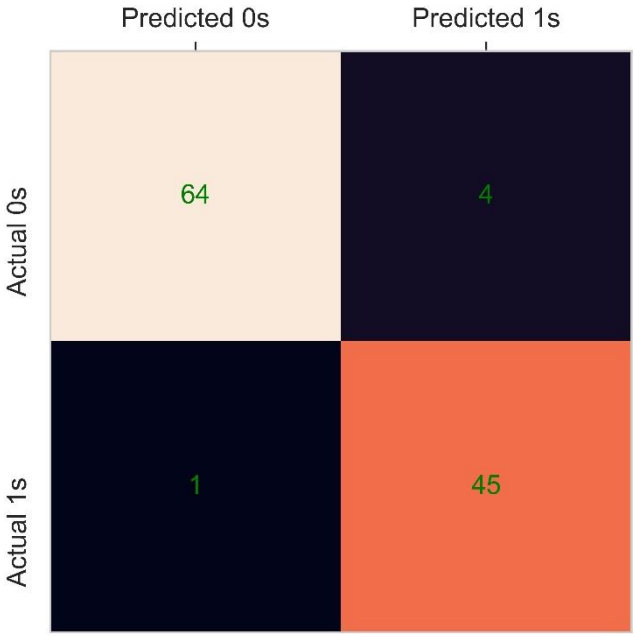


Figure 25 Results Linear kernel (own representation).

Model	Accuracy	Recall	Precision	Specificity	F1
SVM_Poly	0.7544	0.4565	0.8750	0.9559	0.6000

Table 14: Key figures Polynomial kernel (own representation).

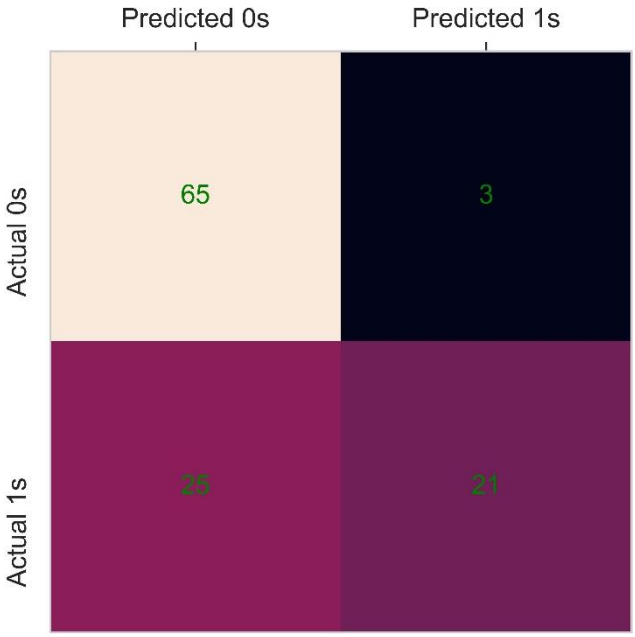


Figure 26 Results Polynomial kernel (own representation).

Model	Accuracy	Recall	Precision	Specificity	F1
SVM_RBF	0.9561	0.9265	0.9362	0.9559	0.9462

Table 15: Key figures RBF kernel (own representation).

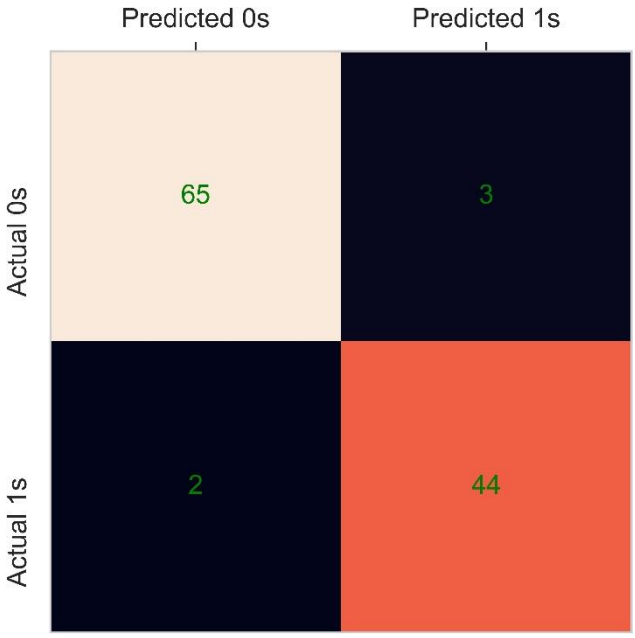


Figure 27 Results RBF kernel (own representation).

The untuned results already show that the polynomial kernel falls back drastically when compared to the other two kernels. While the linear and the RBF kernel have the same accuracy (95.61 %), the linear kernel has a slightly better recall value with 97.83 % compared to 93.62 %. On the other hand, the RBF kernel has a higher precision value (93.62 %) than the linear kernel (91.84 %).

Through the application of the above mentioned GridSearchCV, based on the validation process, the best parameters for the SVC are:

C	0.1
Kernel	Linear
F1 Score	100 %

Table 16: Best parameters for SVC (own representation).

To prevent overfitting, no fine gridsearch is conducted. In a final step, the SVC is applied again to the test data with the best parameters (see table 16). The results of this application are as follows:

Model	Accuracy	Recall	Precision	Specificity	F1
SVM_tuned	0.9825	0.9783	0.9783	0.9853	0.9783

Table 17: Key figures SVC with best parameters (own representation).

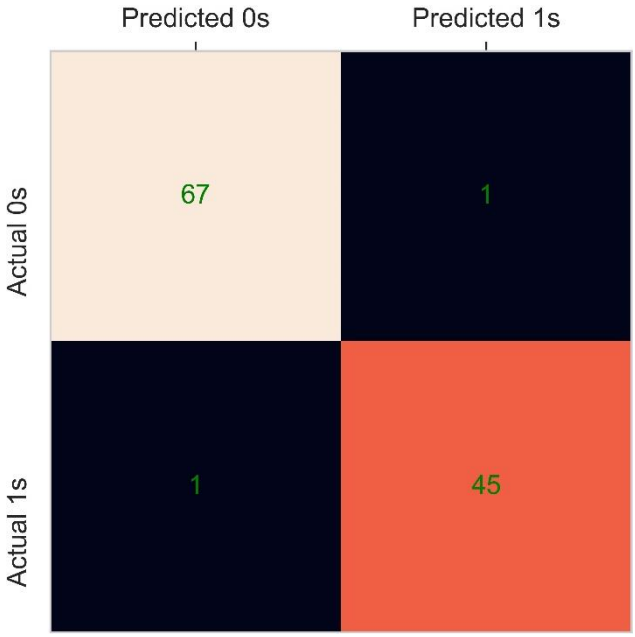


Figure 28 Results SVC with best parameters (own representation).

A clear improvement can be observed compared to the first results. The F1 increased from 94.74 % to 97.83 %, whereas the precision score rose about 7 % from 91.84 % to 97.83 %. The recall value remained stable at a high level of 97.83 %, which is delightful. Even if the cross validation F1 score was at 100 %, no overfitting was conducted since the final F1 value of 97.83 % is close to 100 %.

7 Stacking

Ensemble learning is not a new statistical model but a method in which the advantages of different models are combined to eliminate the individual weaknesses (Kurama, 2019). The individual models are regarded as so-called weak learners. The aim is to combine them in ensemble learning so that the bias and/or variance are reduced as much as possible and thus create a strong learner. Three well-established methods are widely used in practice, namely bagging, boosting and stacking (Rocca, 2019). Since stacking is used in this report as an extension to the other models, only the theoretical basis of this method will be examined in more detail.

The main idea is to combine different weak learners such as the previously mentioned and used models and learn a neural network as a meta-model. The results of the individual three weak learners will be aggregated together and get a final prediction which is based on them (Rocca, 2019). In the upcoming section there will be shown the implementation which will be based on the optimal models from the previous sections. This makes it possible to compare the individual models with the aggregation directly and to determine if stacking is a value adding process in this case.

7.1 Implementation

The StackingClassifier module is imported from scikit-learn.org. Then the best tuned model of each classifier is added as follows:

- GNB: no tuning possible
- KNN: K = 3
- SVM: Kernel = Linear, C = 0.1

It is important to mention that the used data is the same for every model. Since the GNB did perform worse with pre-processed data, the StackingClassifier is run a second time without GNB to check if the result improves. Finally, the report is printed, a confusion matrix is plotted and the values are saved.

7.2 Results / Interpretation

With the stacking of all three models, an F1 score of 93.75 %, recall score of 97.83 % and a precision score of 90.00 % could be achieved (see table 18 and figure 29). Leaving the GNB out for the second run has improved the results. The F1 score increased to 96.84 %, the recall to 100.00 % and the precision score to 93.88 %. Since this dataset is small (n = 569), it is advised that the recall score of 100.00 % is appreciated with caution. These stacking scores are higher than the ones of the KNN model but lower than the SVM. The principal advantage of these scores is their robustness.

Model	**Accuracy**	**Recall**	**Precision**	**Specificity**	**F1**
Stacking_gnb	0.9474	0.9783	0.9000	0.9265	0.9375

Table 18: Key figures Stacking including GNB (own representation).

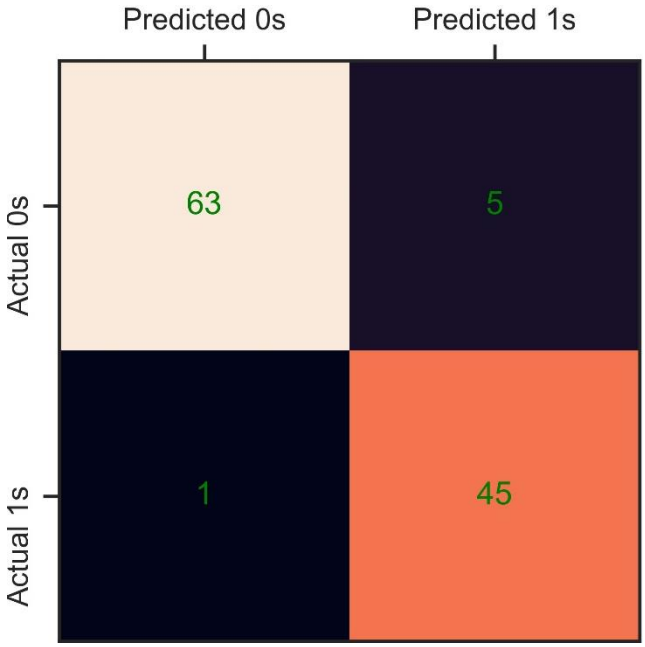


Figure 29: Results Stacking including GNB (own representation).

Model	Accuracy	Recall	Precision	Specificity	F1
Stacking	0.9737	1.0000	0.9388	0.9559	0.9684

Table 19: Key figures Stacking KNN & SVM (own representation).

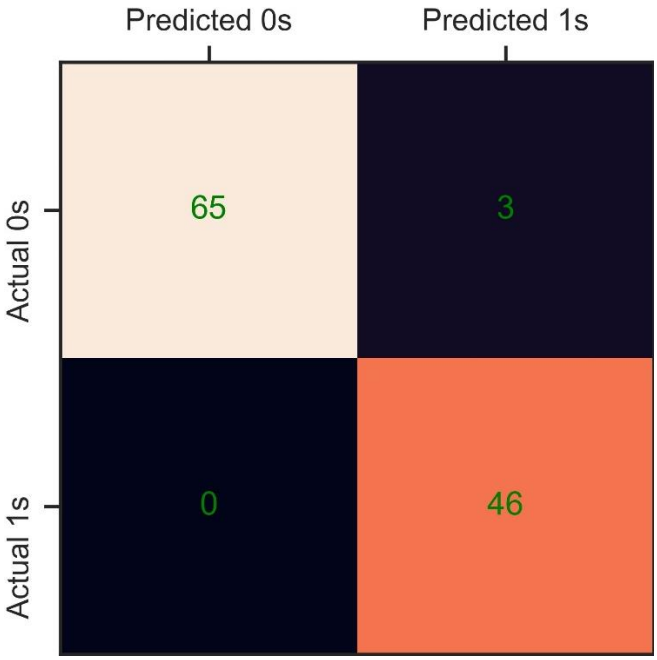


Figure 30: Results Stacking KNN & SVM (own representation).

8 Conclusion

Model		Accuracy	Recall	Precision	Specificity	F1
GNB	normal data	0.9386	0.9130	0.9333	0.9559	0.9231
KNN	NNC_tuned	0.9649	0.9565	0.9565	0.9706	0.9565
	NCC_tuned	0.9123	0.9783	0.8333	0.8676	0.9000
SVM	SVM_lin_best	0.9825	0.9783	0.9783	0.9853	0.9783
Ens. L.	Stacking	0.9737	1.0000	0.9388	0.9559	0.9684

Table 20: Results overview (own representation).

The individual results have shown that the models could be improved each time adjusted tuning parameters. Furthermore, it is important to notice that the data has never been overfitted. The differences between the validation and test results are not worrying.

If the best results of each model are compared (see table 20), an obvious pattern appears. For almost every scoring measurement except the recall, the NCC of the KNN model has the worst performance. The worst recall score exhibits the normal data of the GNB. On the other hand, the linear SVM has the best performance for almost every scoring measurement. Stacking registers best recall score of 100.00 %. The recall score focuses on the people who have cancer. To calculate this score, the TP are divided by the sum of TP and FN. This score is especially then important when FN are not tolerated. Predicting if a person has cancer or not is such a case. For the patient, it is better to get a FP (predict cancer when they do not have cancer) than a FN (predict no cancer when they have cancer). Therefore, having as few FN as possible is crucial. The denominator would then approach the value of TP and thus raise the recall score. It is delightful to see that four of the five models have a recall score higher 95 %.

The precision score answers how many of those predicted with cancer did have cancer. This score focuses on cancer prediction and highlights the TP share. The previously mentioned “false alarm” (FP = predict cancer when they do not have cancer) is summed up with the TP in the denominator. It is best to minimize these false alarms as with the recall score. It is preferred to have a lower precision score than a lower recall score in the medical area. If a patient gets diagnosed with cancer, further tests are likely conducted.

These two scores combined result in the F1 score. In this project, it is more important to focus on FN and FP than TN and TP. Moreover, the data of this project is not balanced. These two facts amplify the importance of the F1 score compared to the accuracy

(Huligol, 2019). Since the specificity focuses on predicting not having cancer and, therefore, focuses on true negatives, this score is of minor relevance.

The stacking of KNN NNC and SVM results reveals the second-highest F1 score (96.84 %). Not only the scores are appealing, but this model appears to be more robust than a single model. Considering a prevalence of positive values (= malignant) of 37.26 % (see chapter 3.1), it can be said that all scores except for the polynomial SVM are very convincing. Even the lowest accuracy score of 75.44 % for the polynomial SVM is still a acceptable result.

This project can be continued by comparing more machine learning models. Since robust models are welcomed in the medical area, the part of ensembled learning can also be more elaborated. To achieve even more stable models, more data would be beneficial. The achieved scores in this project confirm the findings of the mentioned literature (see chapter 2). Feature selection would then be an additional benefit for predictions. Time of professionals and patients would be saved and not as much computing power would be needed. Nevertheless, stable and reliable predictions can be made.

9 Bibliography

Bayes, T. (1763, January 01). LII. An essay towards solving a problem in the doctrine of chances. *Royal Society*, 370.

Chelliah, I. (2020). *An Introduction to K-Nearest Neighbors Algorithm*. https://towardsdatascience.com/an-introduction-to-k-nearest-neighbours-algorithm-3ddc99883acd

Data-flair.training (n.d.) *Support Vector Machines Tutorial – Learn to implement SVM in Python.* https://data-flair.training/blogs/svm-support-vector-machine-tutorial/

Downey, A. B. (2021). *Think Bayes Bayesian Statistics in Python* (2nd). O'Reilly Media.

Fazlija, B. (2021). *Support Vector Machines [Jupyter Notebook]*. Zurich University of Applied Sciences, Department Quantitative Finance.

Hezel, N. (2015). *KNN: Vorteile und Nachteile.* https://data-science-blog.com/blog/2015/09/12/knn-vorteile-und-nacheile/

Huligol, P. (2019). *Accuracy vs. F1-Score.* https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2

Jason, B. (2017). *Master Machine Learning Algorithms - Discover how they work*. https://datageneralist.files.wordpress.com/2018/03/master_machine_learning_algo_from_scratch.pdf

Kramer, O. (2013). *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Springer Berlin Heidelberg.

Kurama, V. (2019). *Introduction to Bagging and Ensemble Methods*. https://blog.paperspace.com/bagging-ensemble-methods/

Mangasarian, O.L., Street, W.N., & Wolberg, W.H. (1995). *Breast Cancer Diagnosis and Prognosis Via Linear Programming*. Oper. Res., 43, 570-577.

Mothadi, B. F. (2018). *In Depth: Parameter tuning for SVC*. https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT Press.

Narkhede, S. (2018). *Understanding Confusion Matrix*. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Newbold, P., Carlson, W. L., & Thorne, B. (2013). *Statistics for business and economics* (8th). Pearson.

Oluwaseun, O. & Desmond, I. (2019). *Urban Water Demand Forecasting: A Comparative Evaluation of Conventional and Soft Computing Techniques.* https://www.researchgate.net/publication/335911736_Urban_Water_Demand_

Forecasting_A_Comparative_Evaluation_of_Conventional_and_Soft_Computing_Techniques

Quantstart.com (n.d.). *Support Vector Machines: A Guide for Beginners.* https://www.quantstart.com/articles/Support-Vector-Machines-A-Guide-for-Beginners/

Rajaguru, H., & Prabhakar, S. K. (2017). *KNN Classifier and K-Means Clustering for Robust Classification of Epilepsy from EEG Signals*. Anchor Academic Publishing, Imprint der Diplomica Verlag GmbH.

Rocca, J. (2019). *Ensemble methods: bagging, boosting and stacking*. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Scikit-learn.org (n.d.a). *Naive Bayes.* https://scikit-learn.org/stable/modules/naive_bayes.html

Scikit-learn.org (n.d.b). *Nearest Neighbors*. https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-transformer

Scikit-learn.org (n.d.c). *Support Vector Machines*. https://scikit-learn.org/stable/modules/svm.html

Scikit-learn.org (n.d.d). *RBF SVM parameters*. https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html#sphx-glr-auto-examples-svm-plot-rbf-parameters-py

Scikit-learn.org (n.d.e). *Support Vector Machines*. https://scikit-learn.org/stable/modules/svm.html#kernel-functions

Sharp, E. (2021). *Sensitivity, Specificity, PPV and NPV*. https://geekymedics.com/sensitivity-specificity-ppv-and-npv/

Suresh, A. (2020). *What is a confusion matrix?*. https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

University of California, School of Information and Computer Science (2019). *UCI Machine Learning Repository*. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic %29

Von der Hude, M. (2020). *Predictive Analytics und Data Mining*. Springer Fachmedien Wiesbaden GmbH.

Zhang, G. (2018). What is the kernel trick? Why is it important?. https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d

Zhang, H. (2004). The Optimality of Naive Bayes. *Proc. FLAIRS.*, 6.

10 Appendix

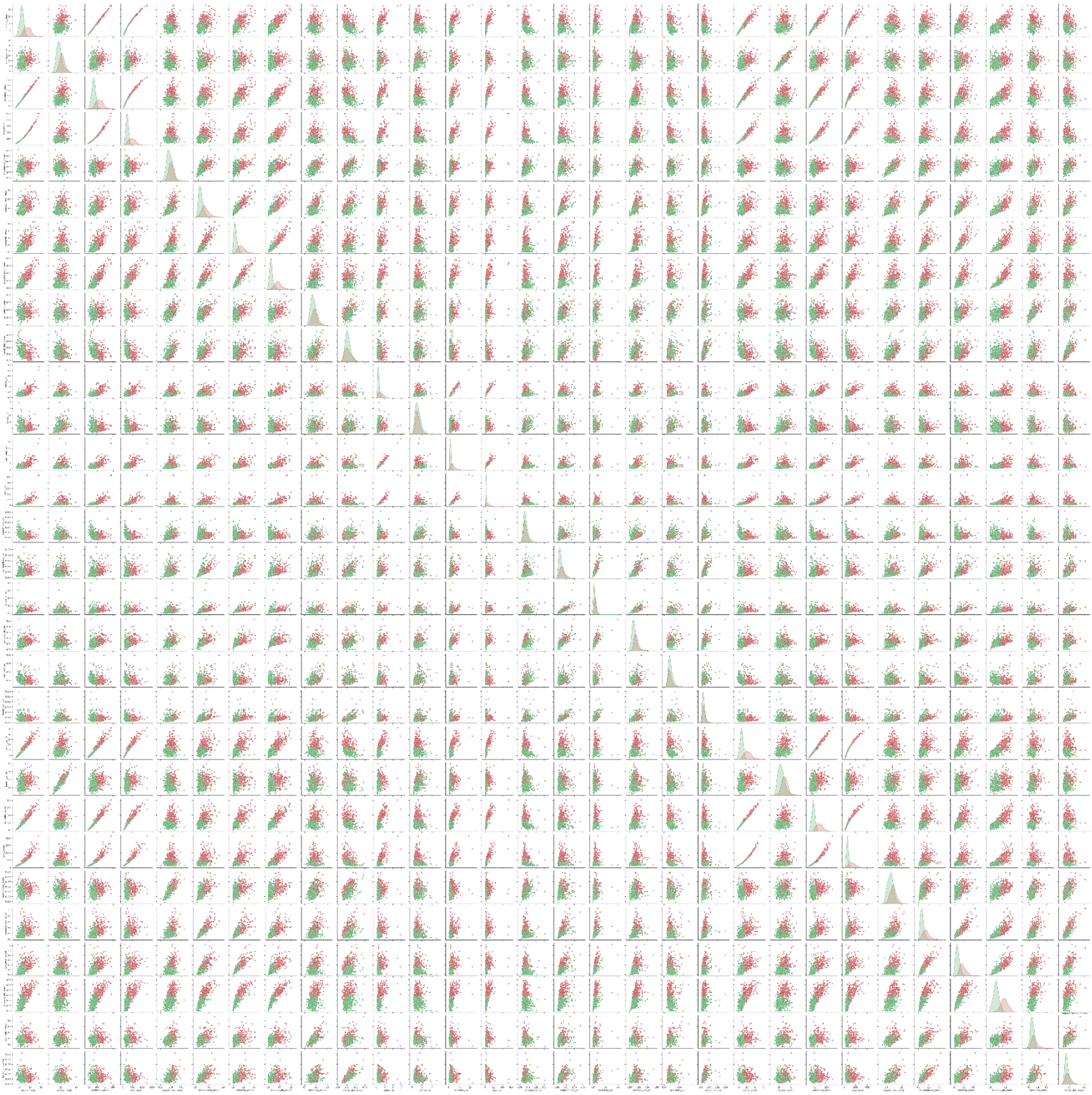
10.1 Github

All our data and jupyter notebooks can be found on github:

https://github.com/bittelandreas/BreastCancer

10.2 Scatterplots

The scatterplots of all 30 features look as follows:



10.3 Heatmaps

The heatmap of all 30 features look as follows: