SILESIAN UNIVERSITY OF TECHNOLOGY

IN GLIWICE

Faculty of Automatic Control, Electronics

and Computer Science

Computer Science

Masters degree, full-time, semester 1

Computer Vision and Pattern Recognition

Final report of the project
Classification of cancer data

Samir Abu Safieh
Artur Stalmach
Rafał Gomola
Maksymilian Kisiel

Table of contents

1. Introduction

The aim of the project was to make a classifier to analyse medical data on cancer. For this, it was necessary to compare different feature selection methods, such as ranked, packed and embedded, and find the optimal one.

2. Simple Classifiers

2.1. What are classifiers?

Classifiers are used in machine learning, as the name suggests, for classification. The way they work is that they classify input data into appropriate classes, based on the characteristics of that data. They learn to recognise patterns from the training data. Classifiers are used, for example, in text analysis or image recognition.

2.2. The SVM classifier

A Support Vector Machine (SVM) is an algorithm used for linear and non-linear classification, regression, or outlier detection. It works by mapping data onto a multidimensional space in such a way that separation is possible.

2.3. The KNN classifier

K-Nearest Neighbours (KNN) is a simple classifier that classifies a selected point based on its nearest neighbour points in the feature space. To determine which data points are closest to the selected query point, the distance between the query point and other data points must be calculated.

2.4. The Random Forest classifier

The Random Forest classifier is a machine learning algorithm based on an ensemble learning technique that creates a set of decision trees from a random subset of the training data. Each decision tree is trained on a different random subset of data and features, and the final classification result is determined by the majority vote of all trees (for classification) or the average of the results (for regression).

2.5. The Bayes classifier

Bayes‘ classifier is based on Bayes’ theorem (it is a theorem of probability theory, binding the conditional probabilities of two events conditioning each other), which gives it its name. The theorem is used to predict the class of a given data point based on its characteristics. This classifier calculates the probability of a point belonging to each of the possible classes and assigns it to the class with the highest probability.

Bayes' theorem:

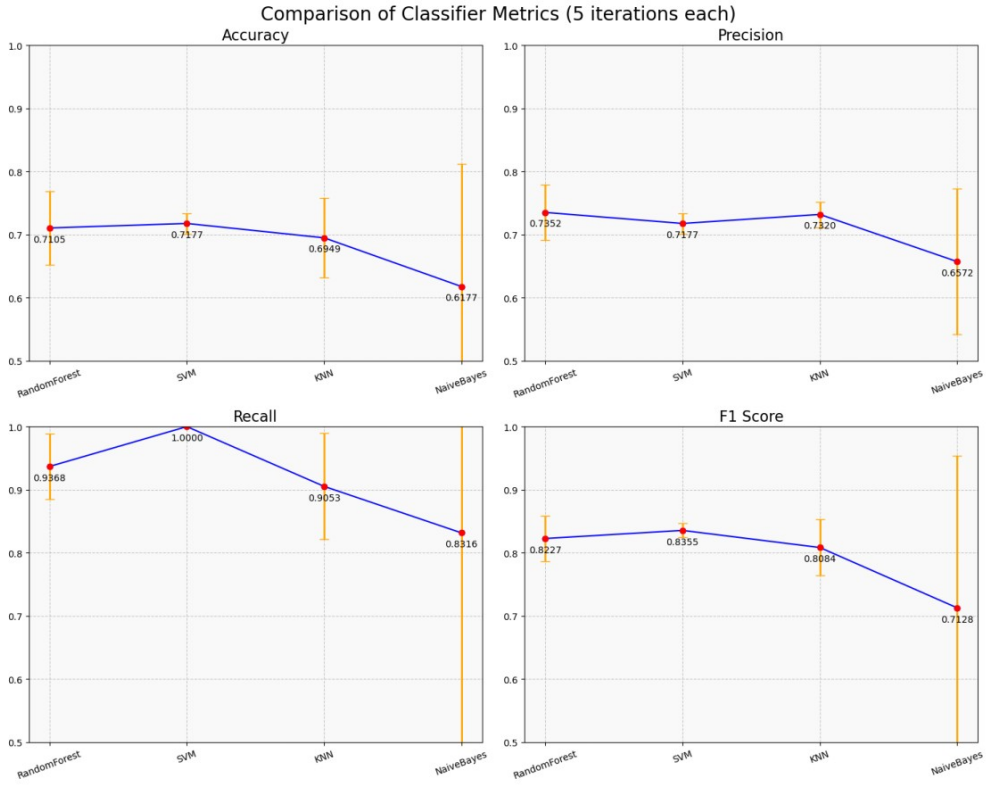
$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(B \mid A)\,\mathrm{P}(A)}{\mathrm{P}(B)}$$

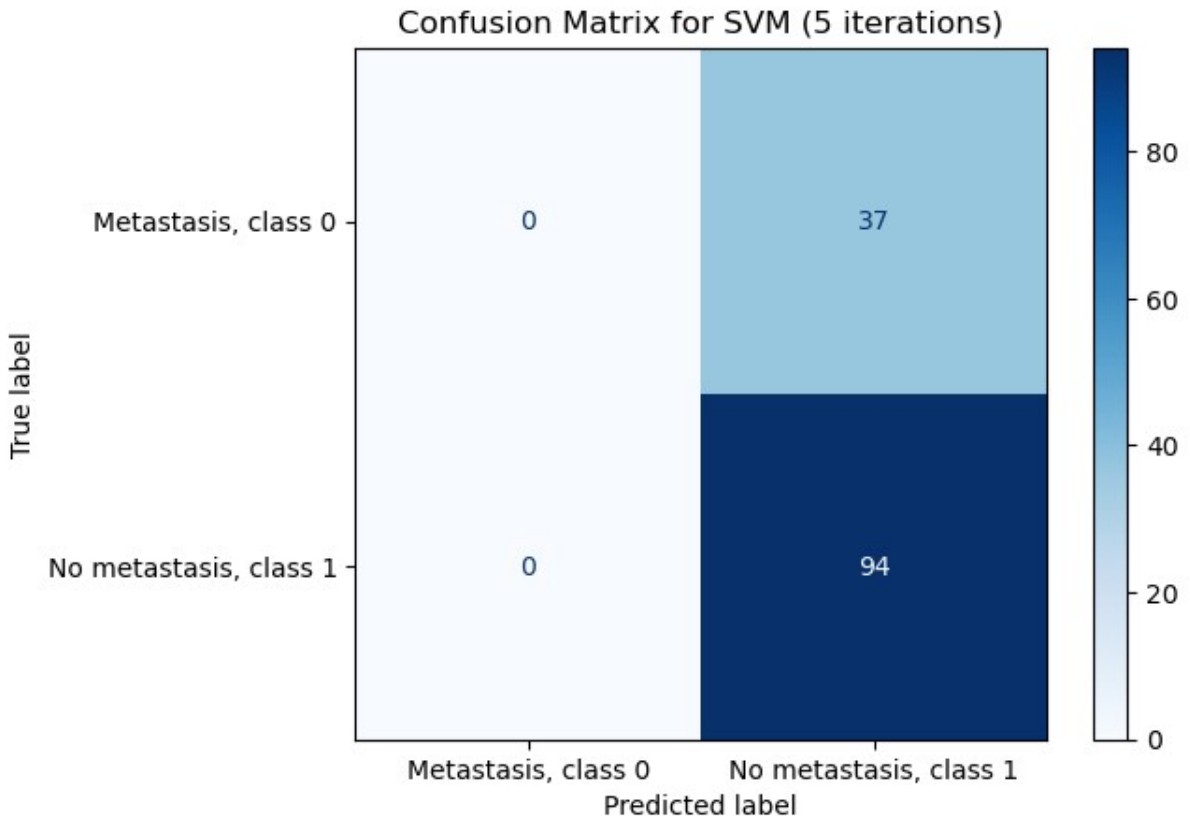
Where A and B are points, where:

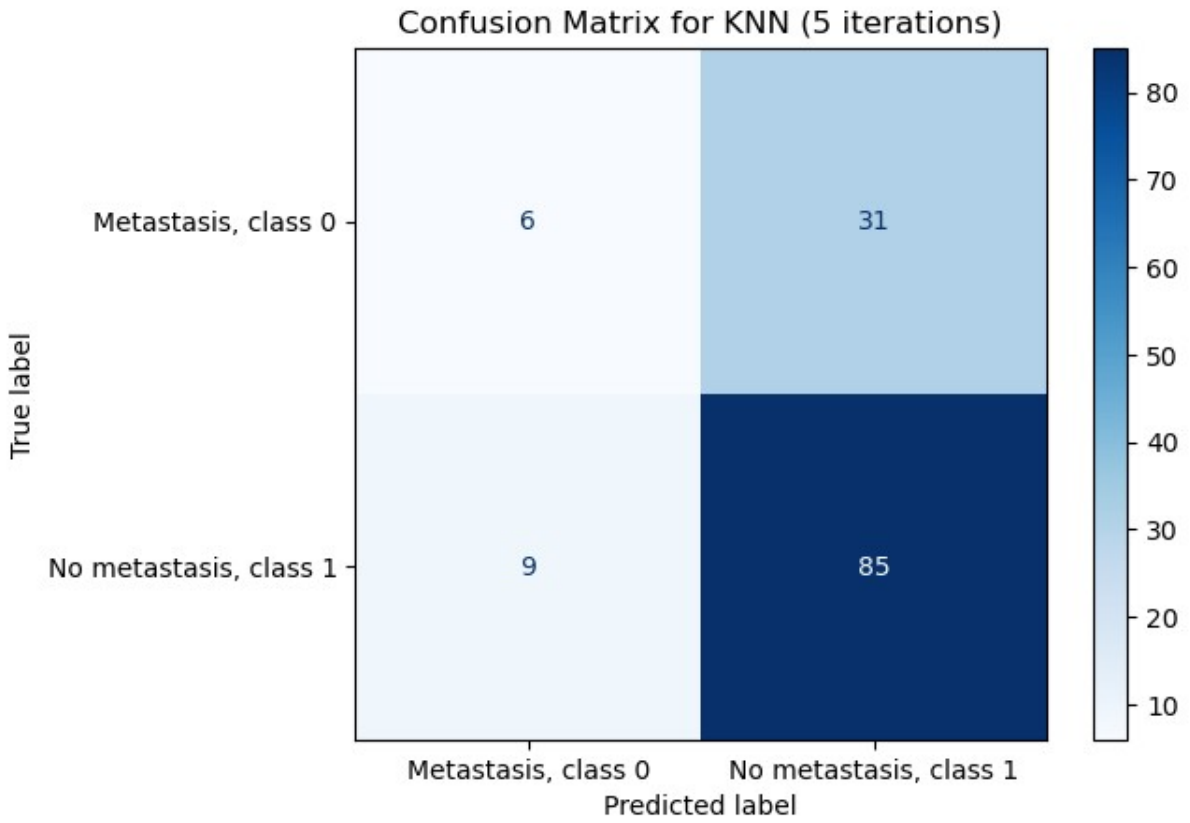
- P(A|B) denotes the conditional probability, i.e. the probability of event A occurring as long as event B occurs.

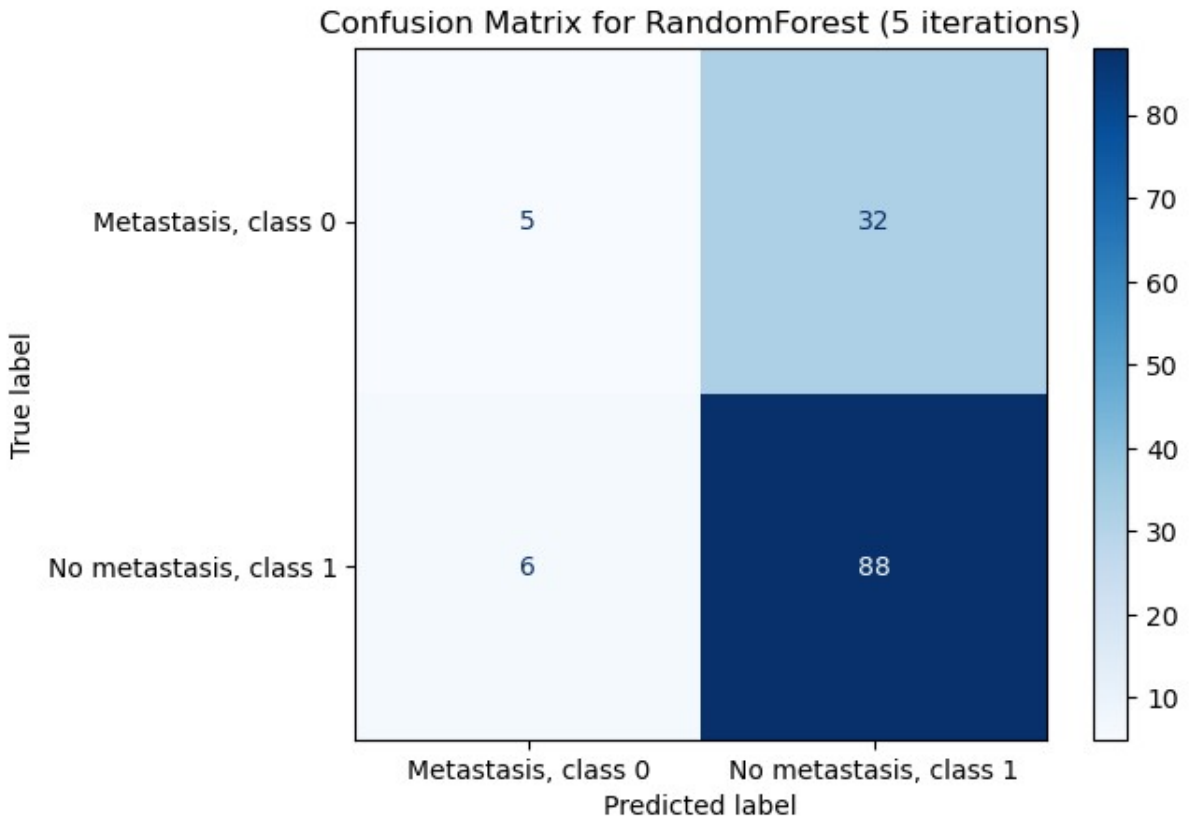
- P(A|B) denotes the probability of event B occurring as long as event A occurs.

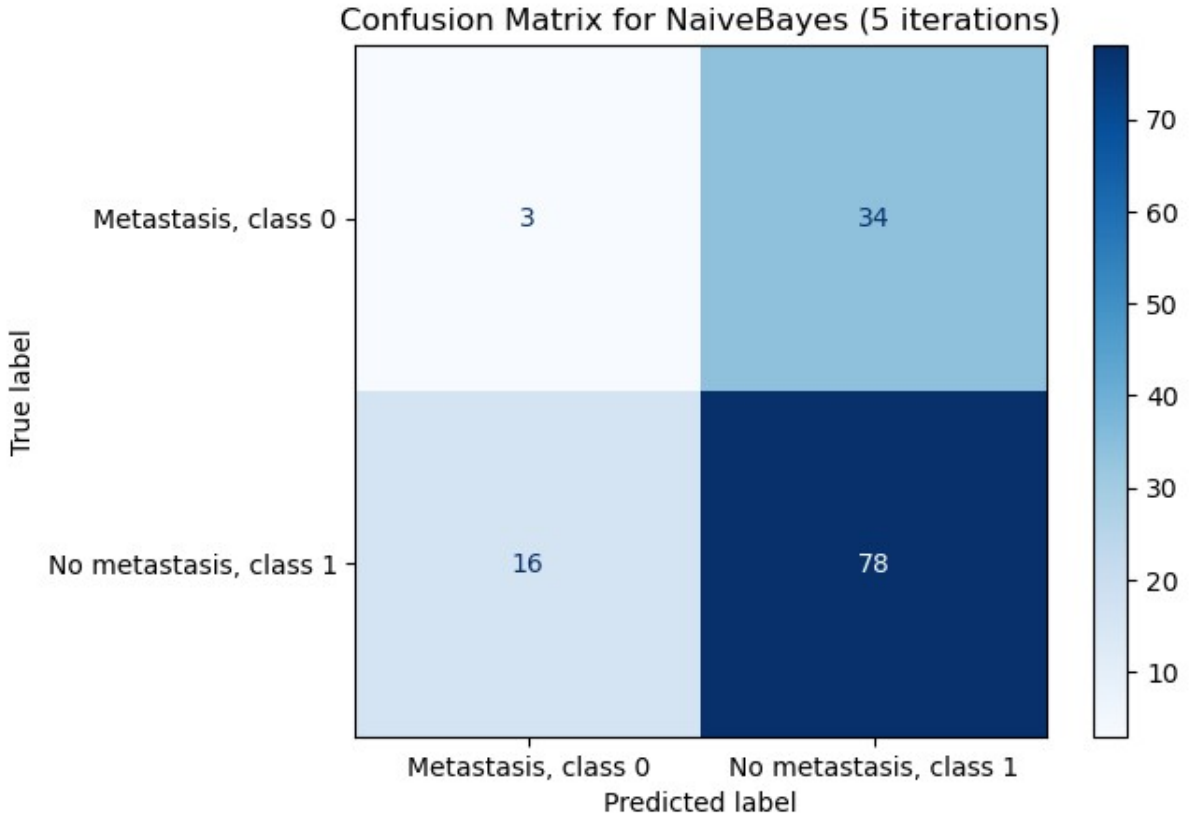
2.6. Results




Confusion Matrix for SVM (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
0
37
0
94
Metastasis, class 0
No metastasis, class 1
Predicted label
80
60
40
20
0


Confusion Matrix for KNN (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
6
31
9
85
Metastasis, class 0
No metastasis, class 1
Predicted label
80
70
60
50
40
30
20
10


Confusion Matrix for RandomForest (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
5
32
6
88
Metastasis, class 0
No metastasis, class 1
Predicted label
80
70
60
50
40
30
20
10

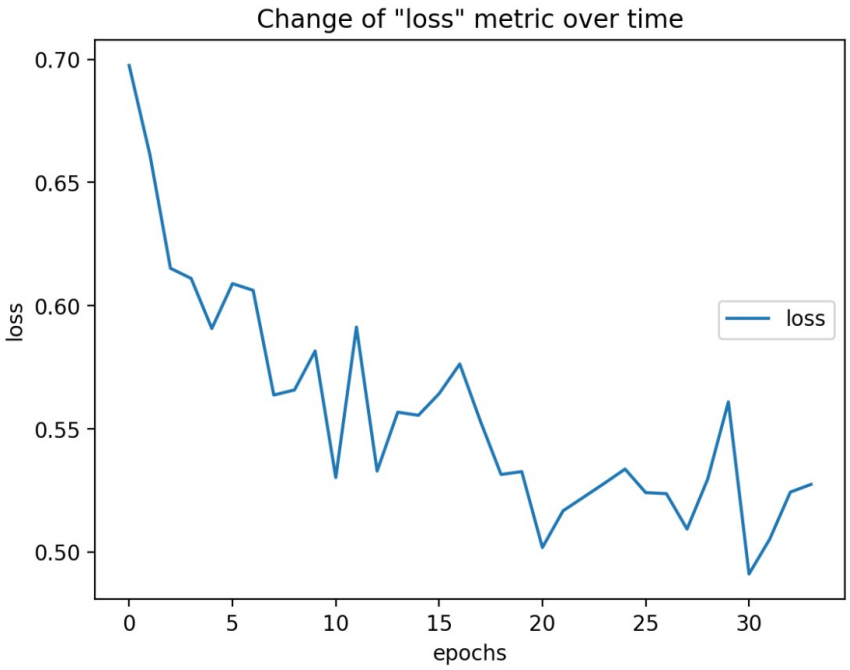

Confusion Matrix for NaiveBayes (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
3
34
16
78
Metastasis, class 0
No metastasis, class 1
Predicted label
70
60
50
40
30
20
10

3. Convolutional neural network (advanced classifier)

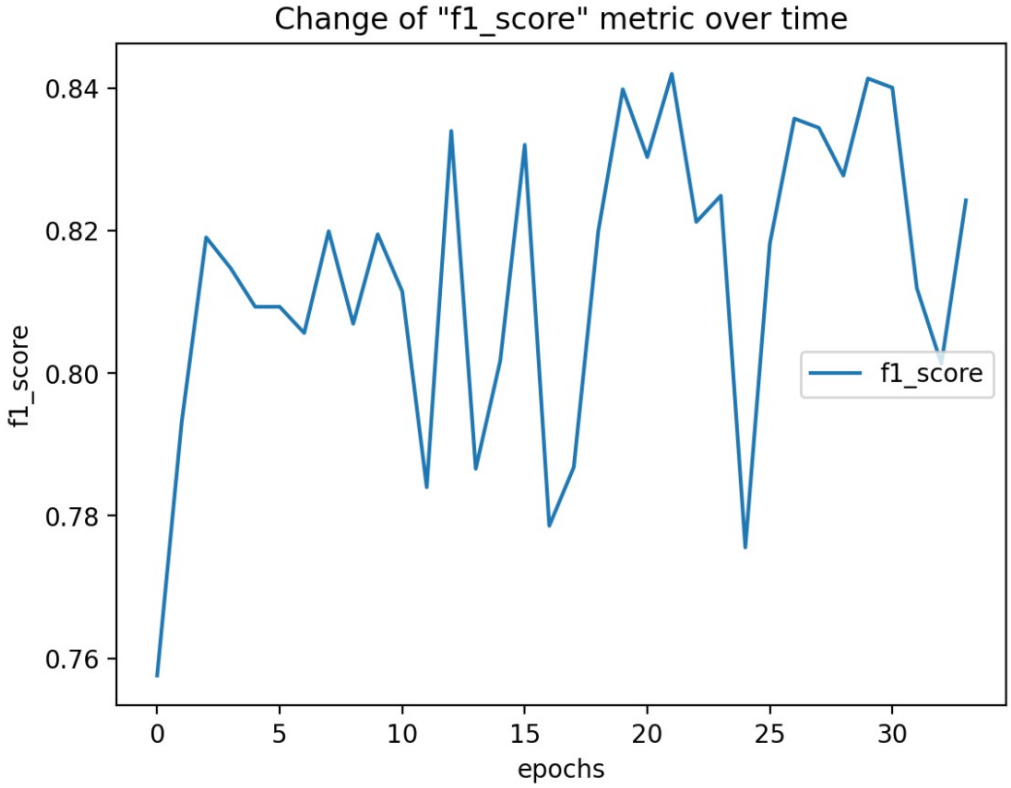
3.1. What is a convolutional neural network?

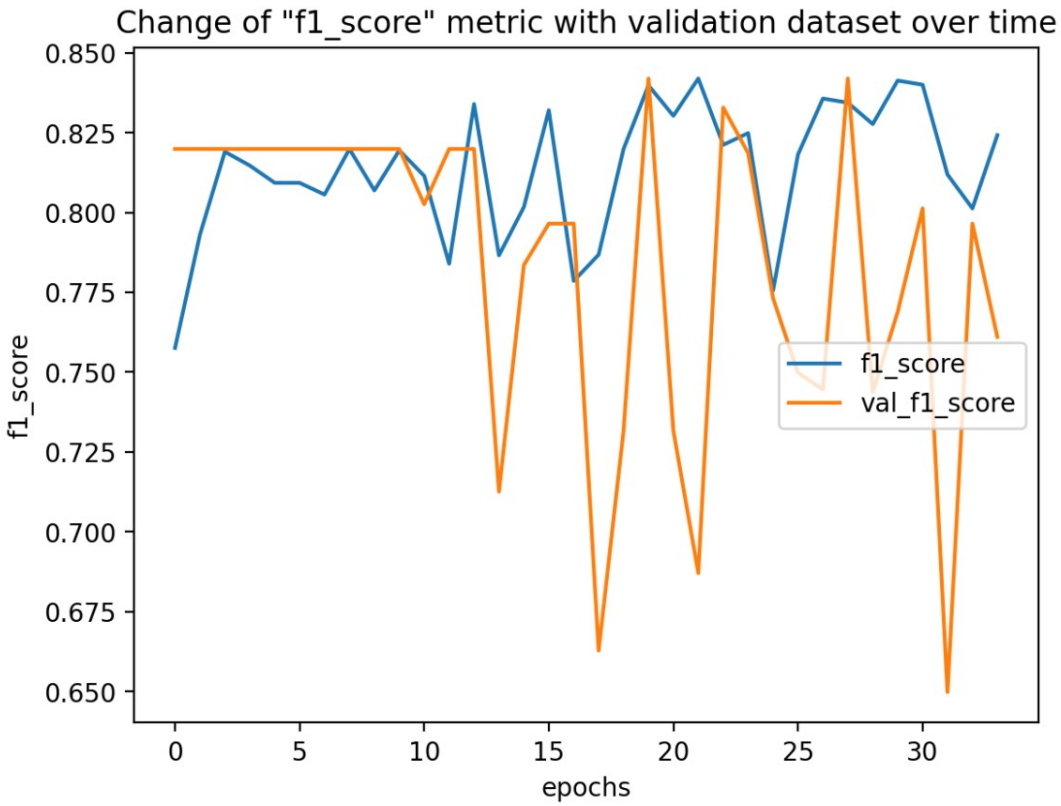
A Convolutional Neural Network (CNN) is a type of neural network, designed mainly for processing data with a grid structure, such as images. A CNN consists of convolutional layers, which use small filters to detect local patterns such as edges and textures, and ReLU layers, which introduce non-linearity. Pooling layers reduce the dimensions of the data while retaining key information. After processing through several such layers, the data is flattened and processed by fully connected layers that classify the data. CNNs are effective in image recognition, video processing and text analysis tasks due to their ability to automatically extract complex features.

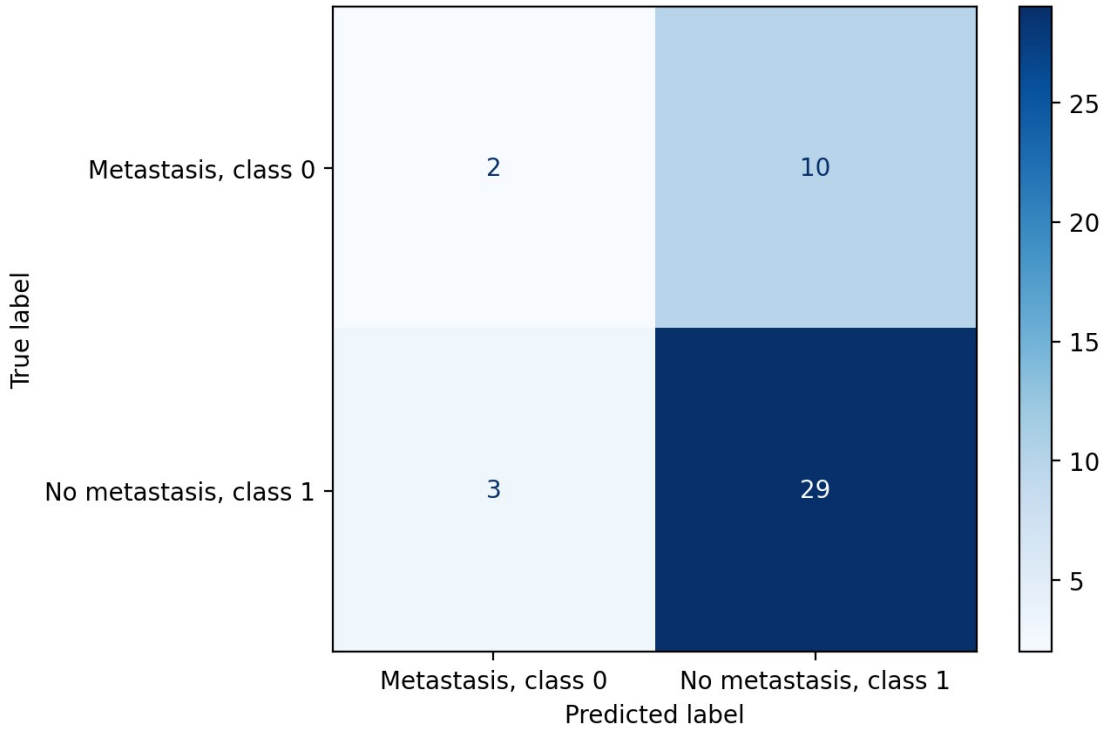
3.2. Results




Change of "loss" metric with validation dataset over time
loss
epochs
loss
val_loss
0.70
0.65
0.60
0.55
0.50
0
5
10
15
20
25
30


Change of "f1_score" metric over time
f1_score
epochs
f1_score
0.84
0.82
0.80
0.78
0.76
0
5
10
15
20
25
30


Change of "f1_score" metric with validation dataset over time
f1_score
epochs
f1_score
val_f1_score
0.850
0.825
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0
5
10
15
20
25
30


Metastasis, class 0
No metastasis, class 1
True label
2
10
3
29
Metastasis, class 0
No metastasis, class 1
Predicted label
25
20
15
10
5

4. Feature Selection Methods

4.1. What are they used for?

The selection of an appropriate selection method has a major impact on the performance of a classification algorithm. It is used to reduce complexity and computational requirements. Eliminating irrelevant features can improve model accuracy by focusing on more meaningful data. Reducing the number of features speeds up the model training process and reduces memory requirements.

4.2. Ranking methods

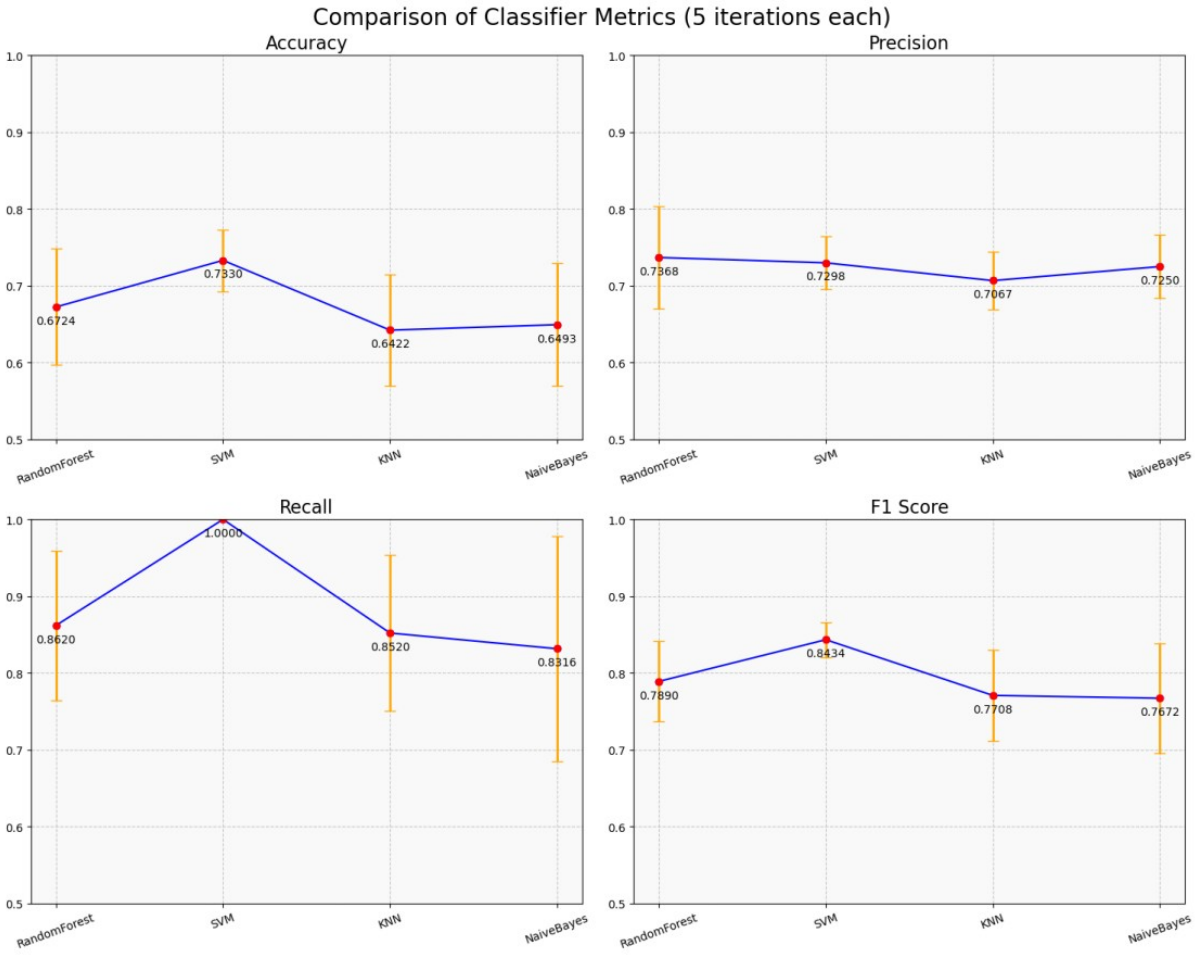
4.2.1. Description

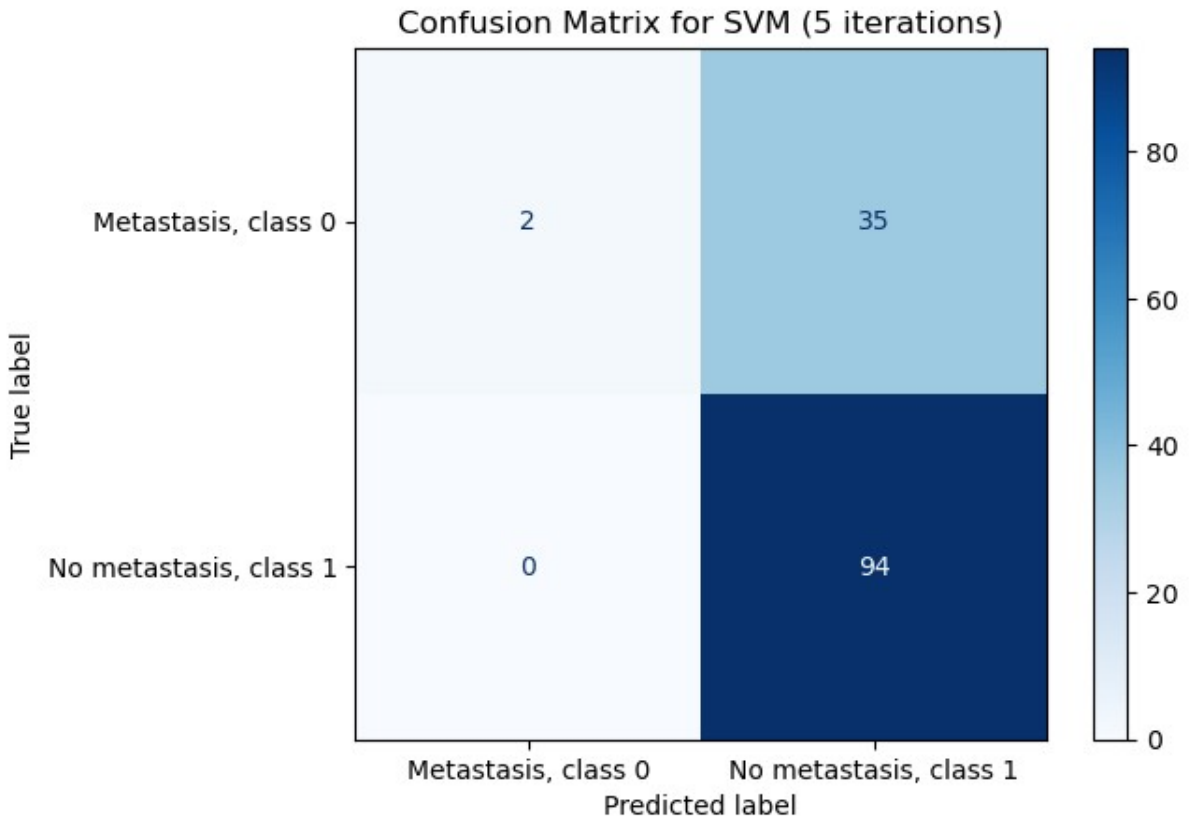
Ranking methods, also known as filter methods, are one of the simplest data selection methods. Their most important feature is their low computational complexity.

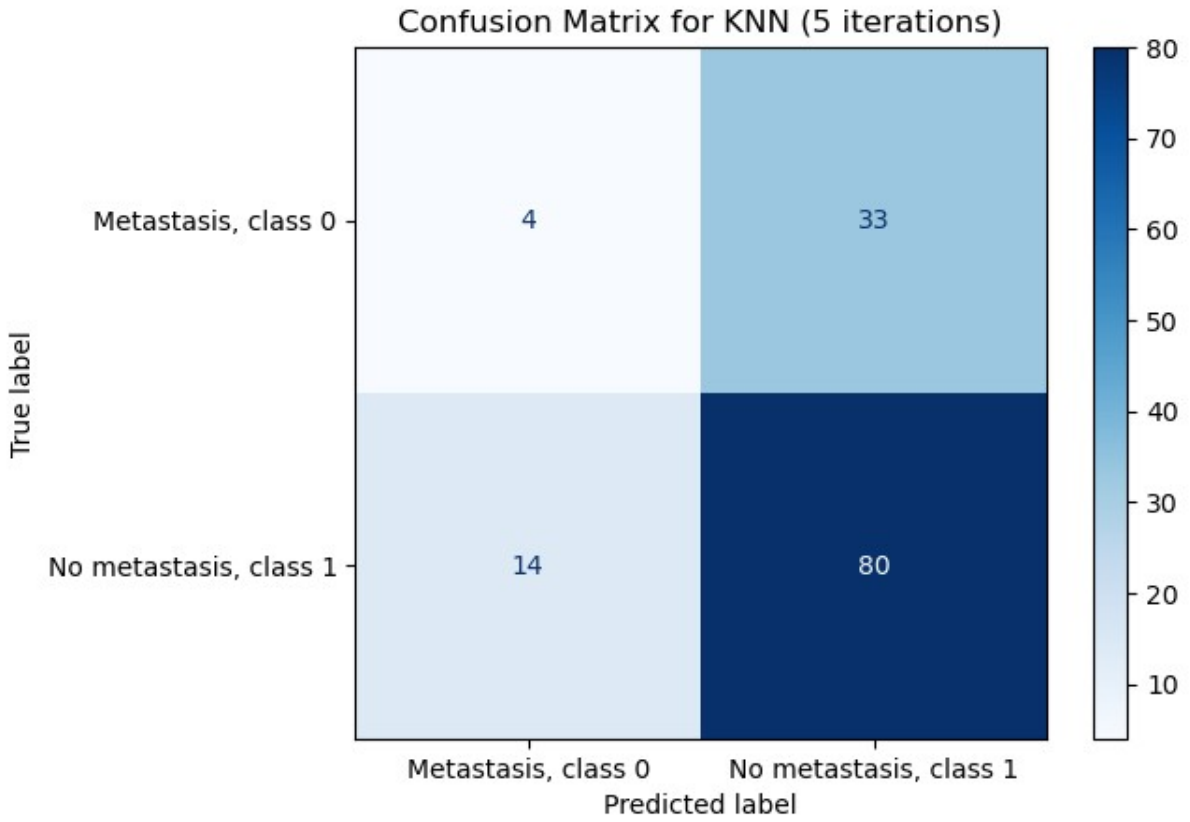
In order to create such a method, an index must be assigned to each feature to determine its quality based on the criteria adopted. The index value is used to sort the features. With this method, the best features (those with the highest ranking) are selected and the worst ones discarded. The threshold determining the features to be discarded can be set as required and based on a specific index value, or based on a specific number of best features to be selected.

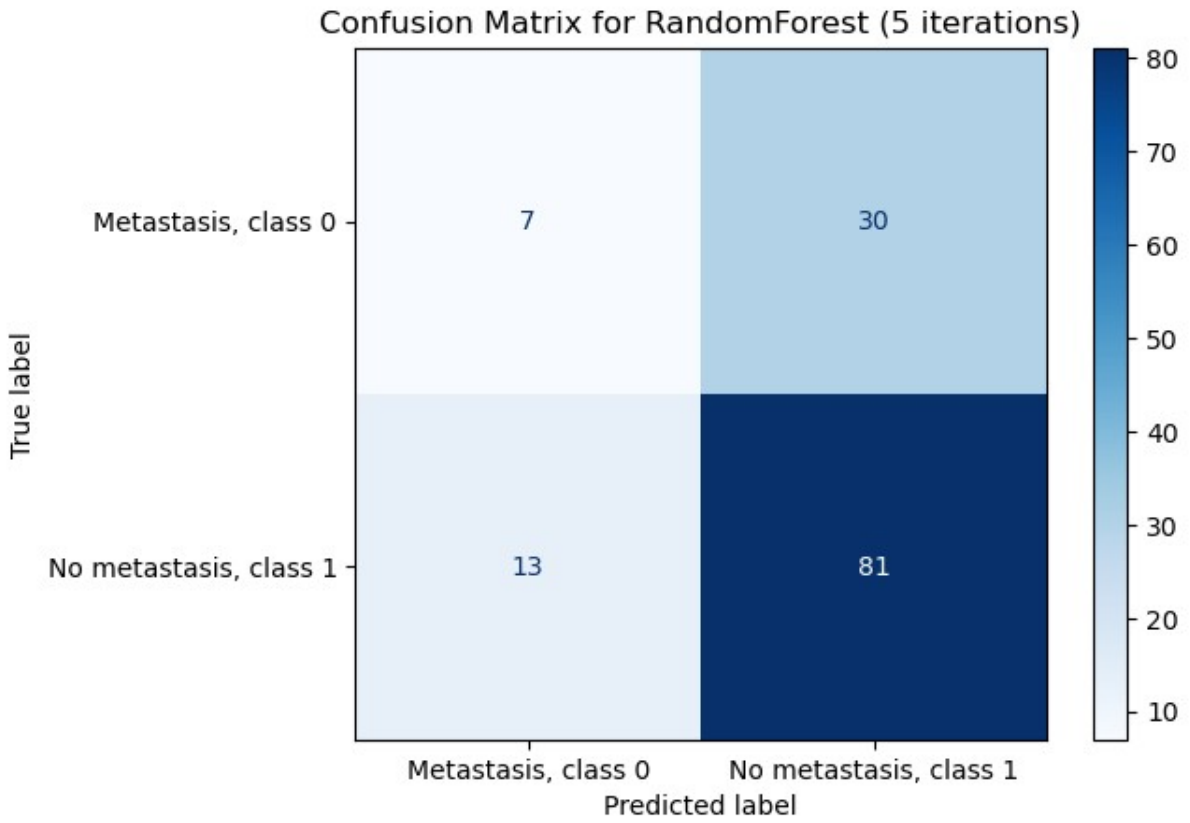
4.2.2. Results for simple classifiers

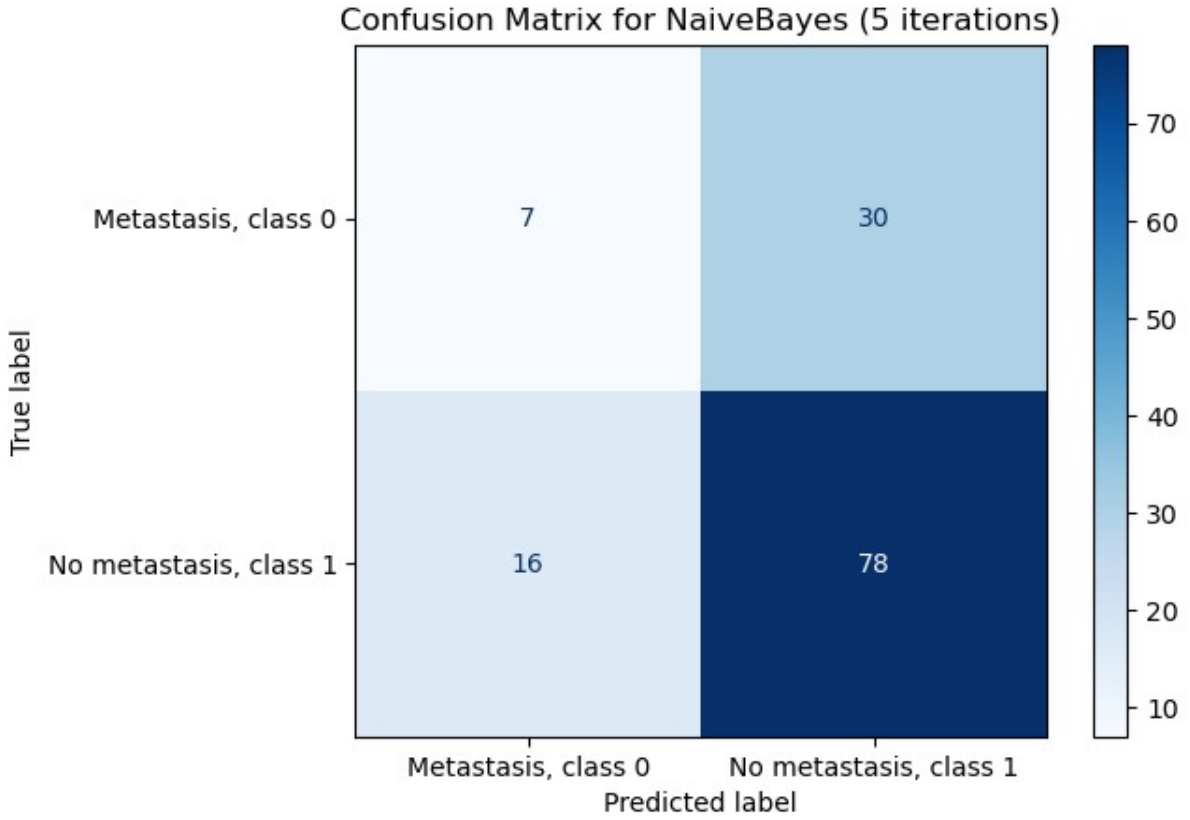
The ranking method was also applied to simple classifiers.



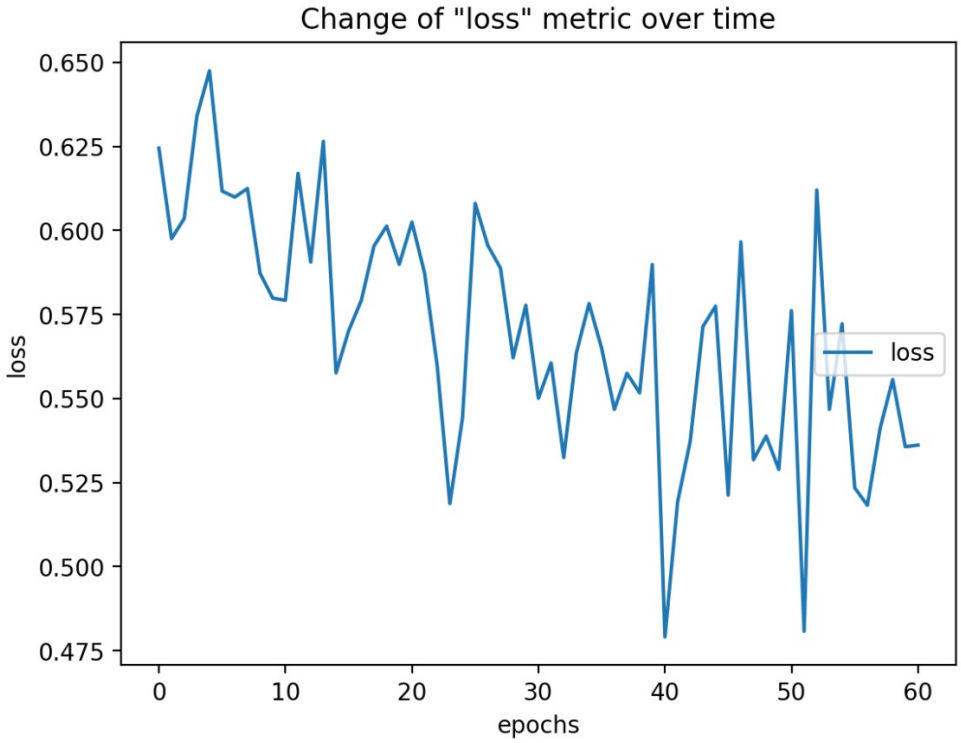

Confusion Matrix for SVM (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
2
35
0
94
Metastasis, class 0
No metastasis, class 1
Predicted label
80
60
40
20
0

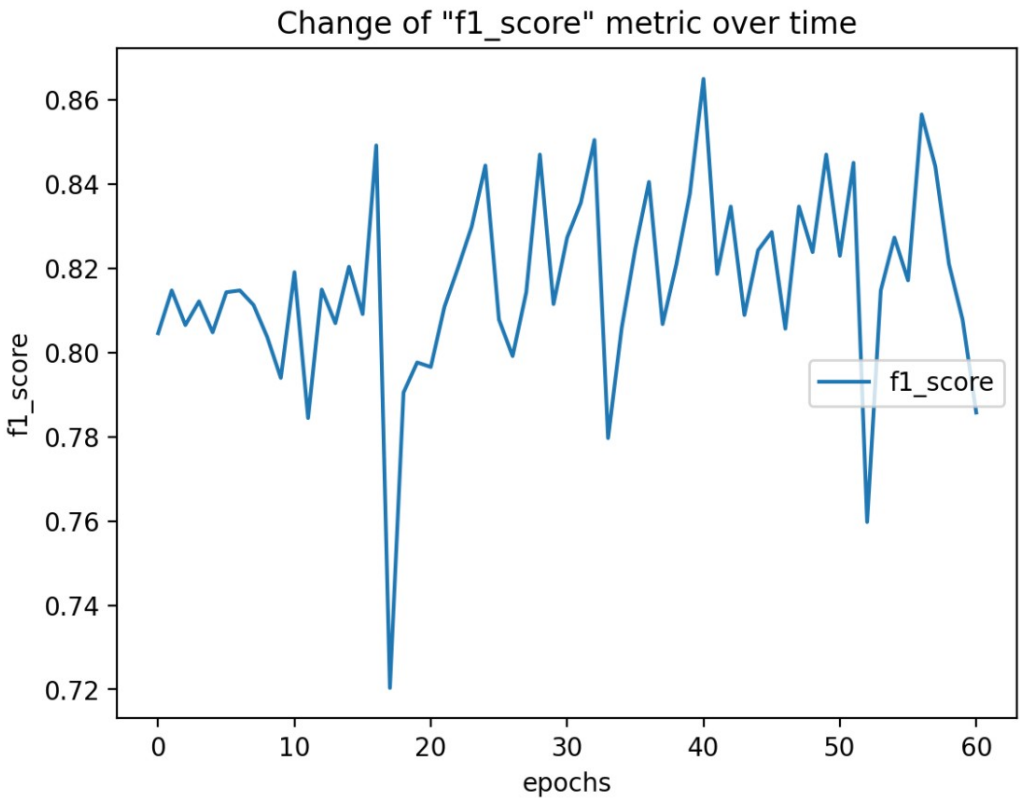

Confusion Matrix for KNN (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
4
33
14
80
Metastasis, class 0
No metastasis, class 1
Predicted label
80
70
60
50
40
30
20
10

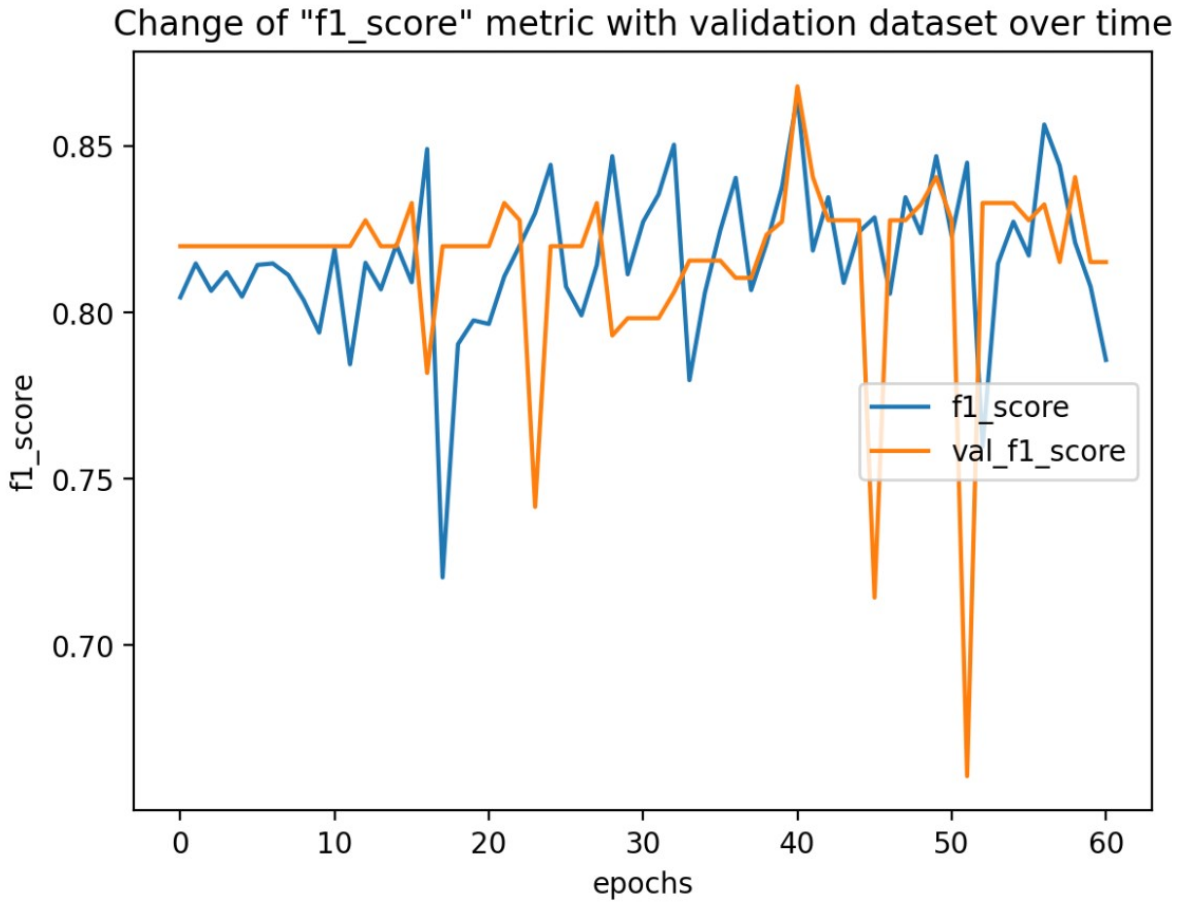

Confusion Matrix for RandomForest (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
7
30
13
81
Metastasis, class 0
No metastasis, class 1
Predicted label
80
70
60
50
40
30
20
10

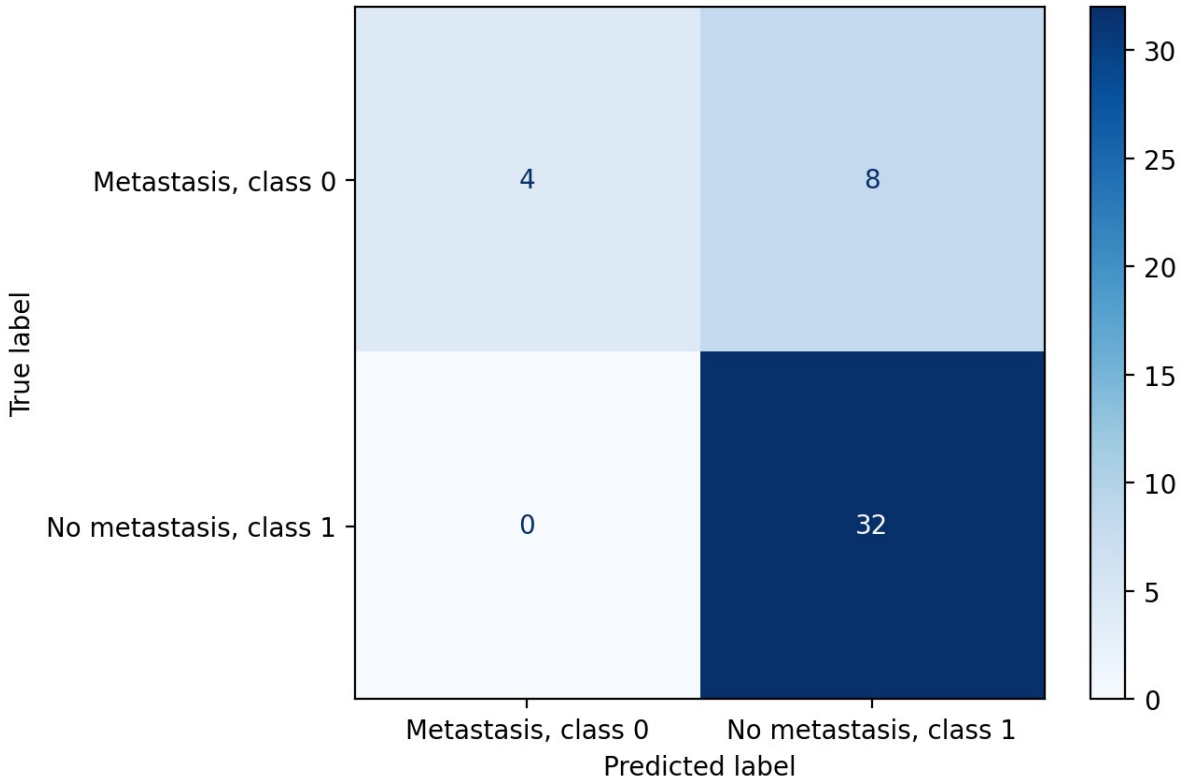

Confusion Matrix for NaiveBayes (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
7
30
16
78
Metastasis, class 0
No metastasis, class 1
Predicted label
70
60
50
40
30
20
10

4.2.3. Results for a convolutional network




Change of "f1_score" metric over time
f1_score
epochs
f1_score
0.86
0.84
0.82
0.80
0.78
0.76
0.74
0.72
0
10
20
30
40
50
60


Change of "f1_score" metric with validation dataset over time
f1_score
epochs
f1_score
val_f1_score
0.85
0.80
0.75
0.70
0
10
20
30
40
50
60


Metastasis, class 0
No metastasis, class 1
True label
4
8
0
32
Metastasis, class 0
No metastasis, class 1
Predicted label
30
25
20
15
10
5
0

4.3. Wrapped methods

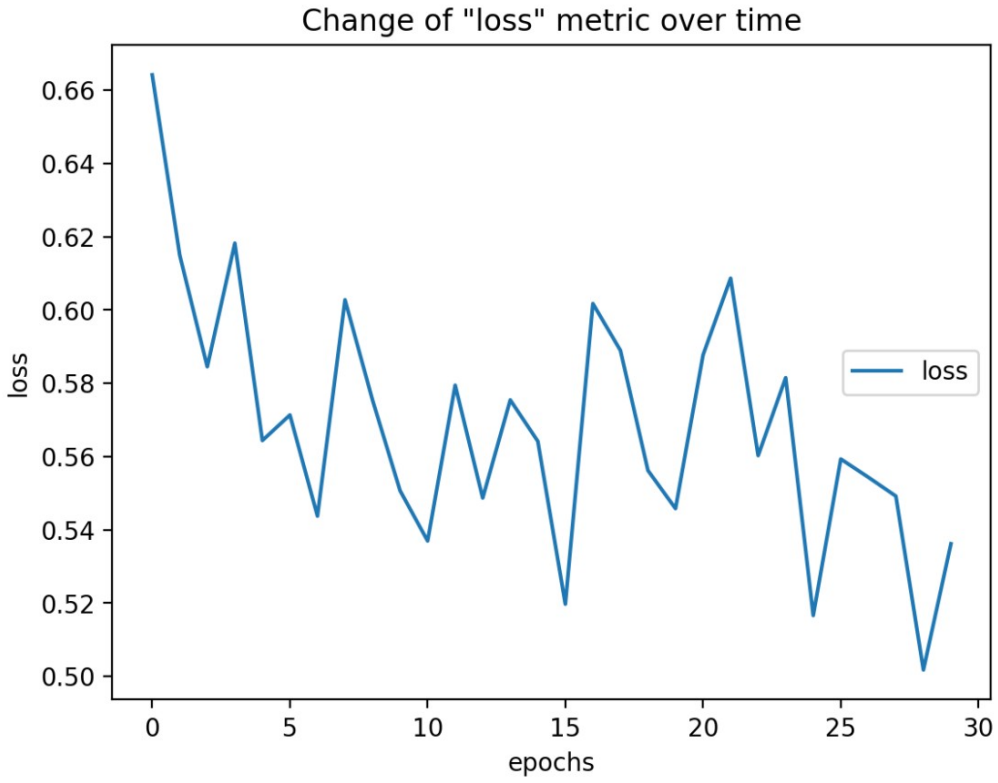
4.3.1. Description

Wrapper methods are characterised by much higher computational complexity than ranking methods. While in ranked methods features are considered independently of each other, in wrappers the relationships between features are crucial. The most favourable set of features is searched for. Different combinations of them are compared. This facilitates the detection of possible interactions between variables.

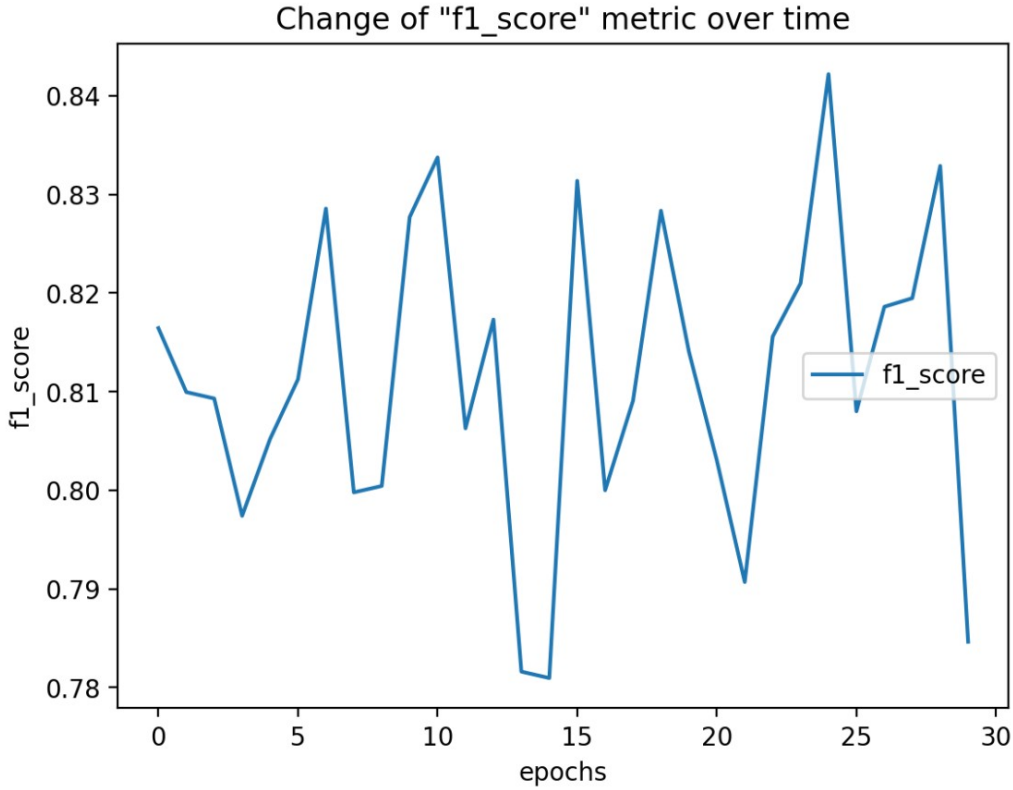
To help find the optimal combination of features in a subset, 2 approaches are used with the best possible computational complexity:

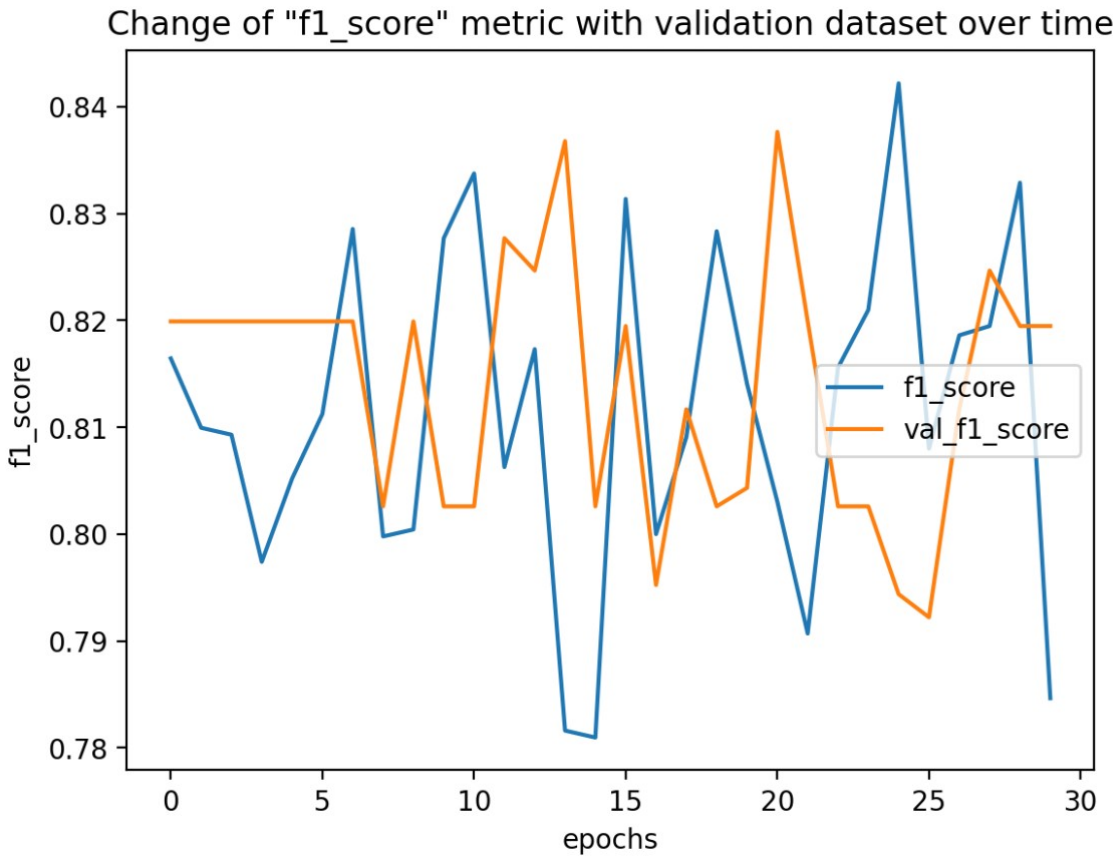
- Forward selection- starts with an empty set and successively adds variables,
- Backward selection- starts with a set of all possible features that are successively discarded.

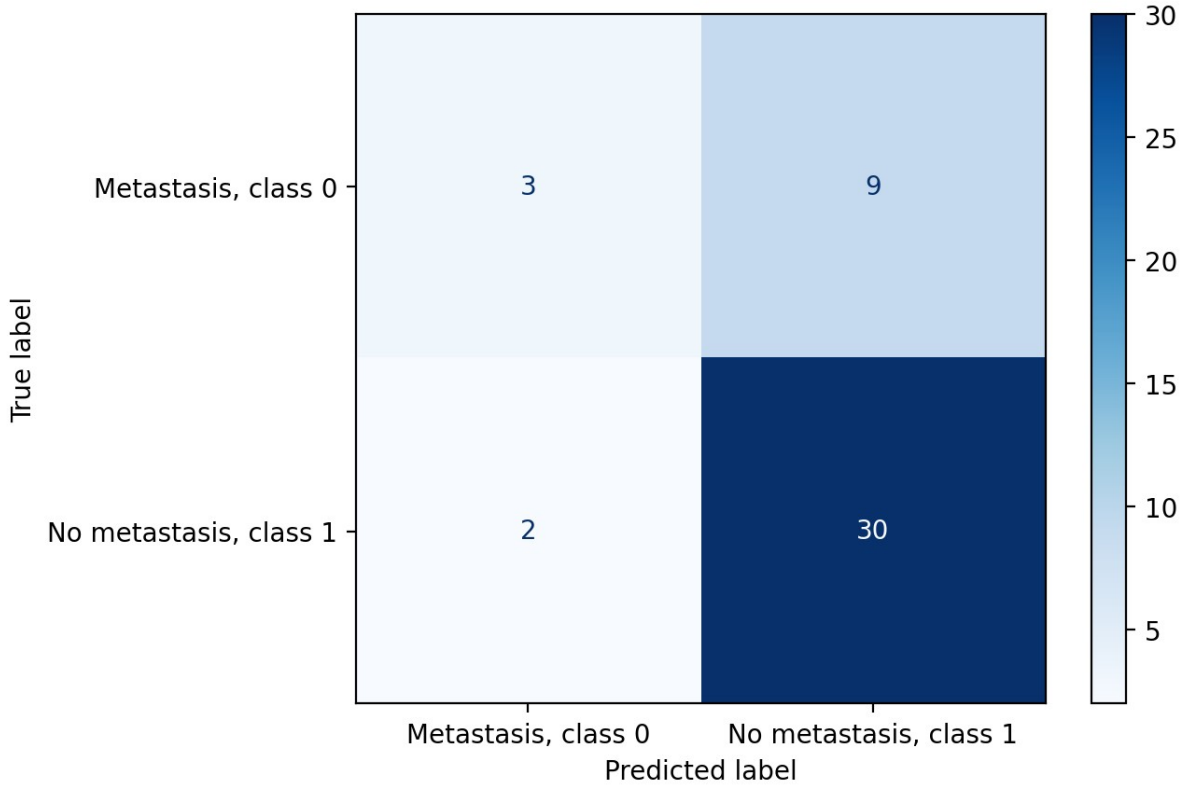
4.3.2. Results




Change of "loss" metric with validation dataset over time
loss
epochs
loss
val_loss
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0.50
0
5
10
15
20
25
30


Change of "f1_score" metric over time
f1_score
epochs
f1_score
0.84
0.83
0.82
0.81
0.80
0.79
0.78
0
5
10
15
20
25
30


Change of "f1_score" metric with validation dataset over time
f1_score
epochs
f1_score
val_f1_score
0.84
0.83
0.82
0.81
0.80
0.79
0.78
0
5
10
15
20
25
30

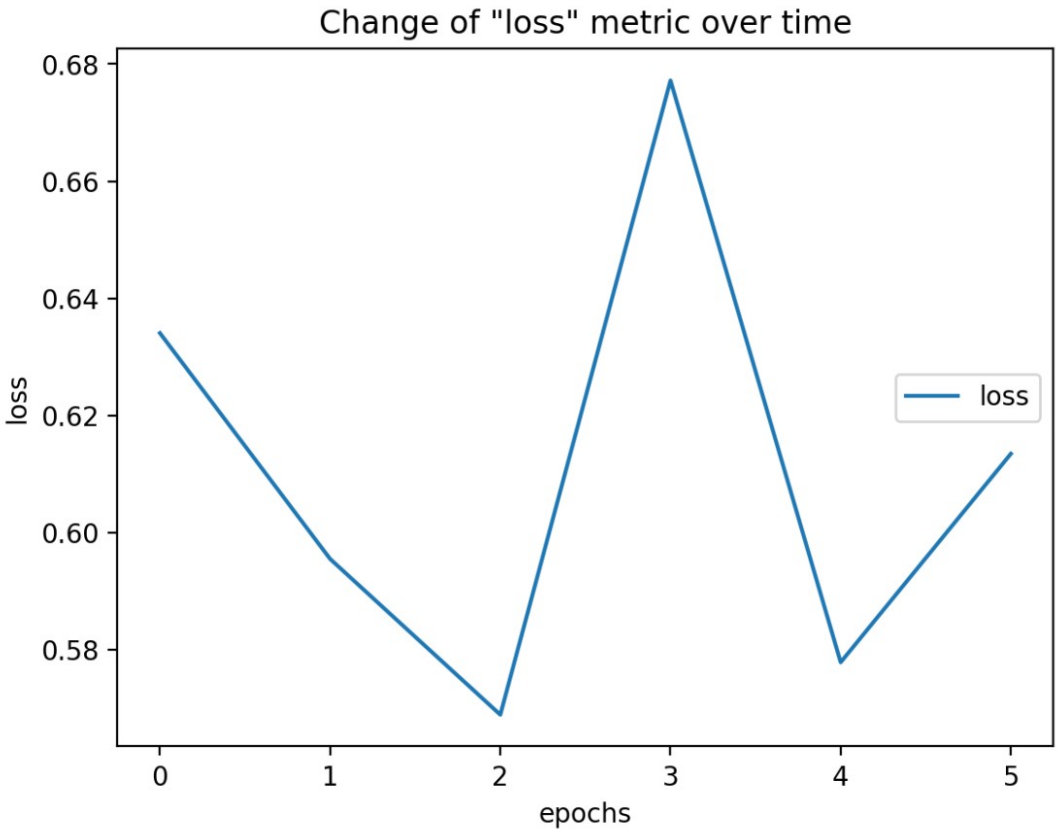

Metastasis, class 0
No metastasis, class 1
True label
3
9
2
30
Metastasis, class 0
No metastasis, class 1
Predicted label
30
25
20
15
10
5

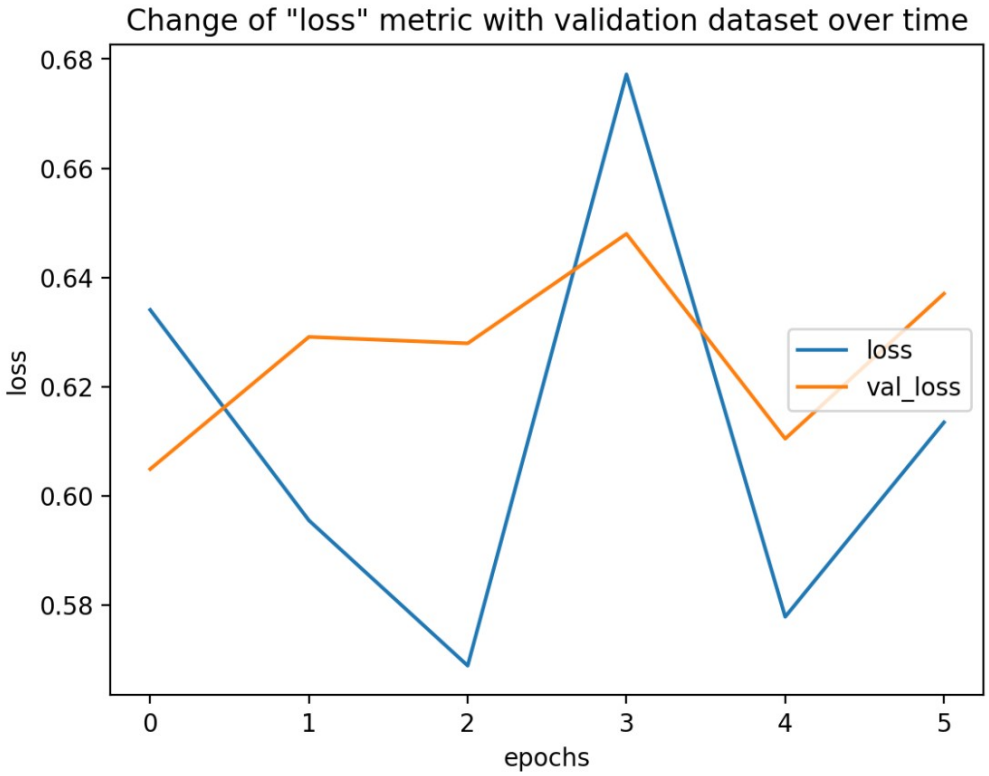
4.4. Embedded methods

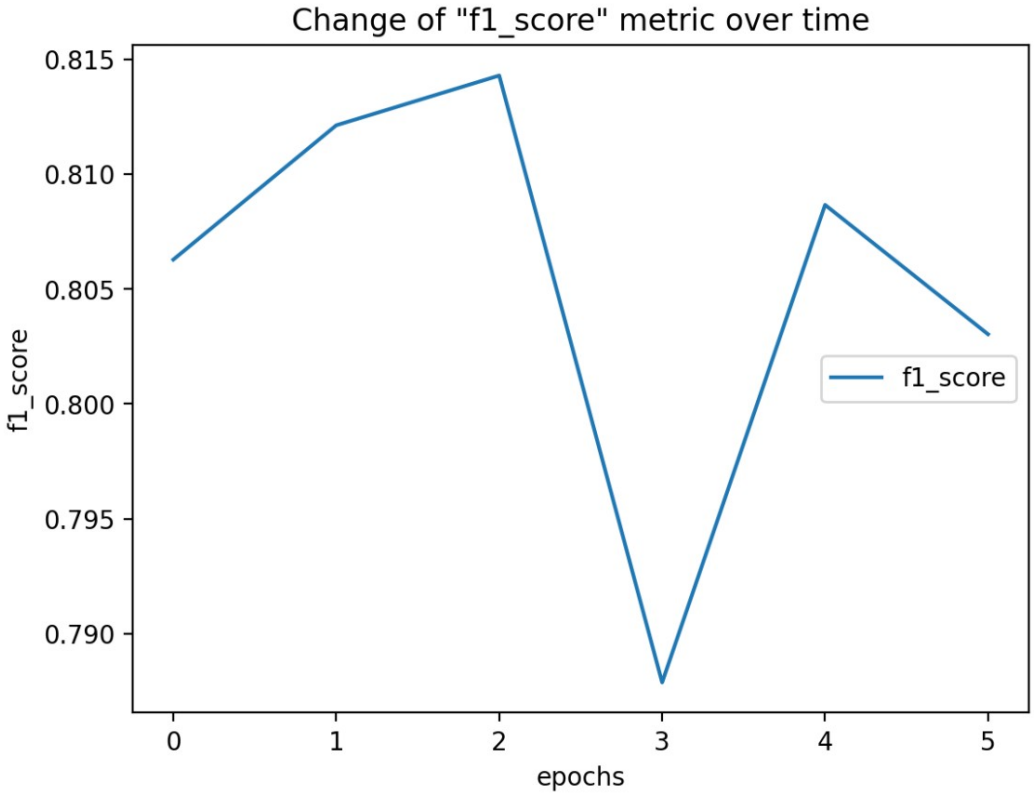
4.4.1. Description

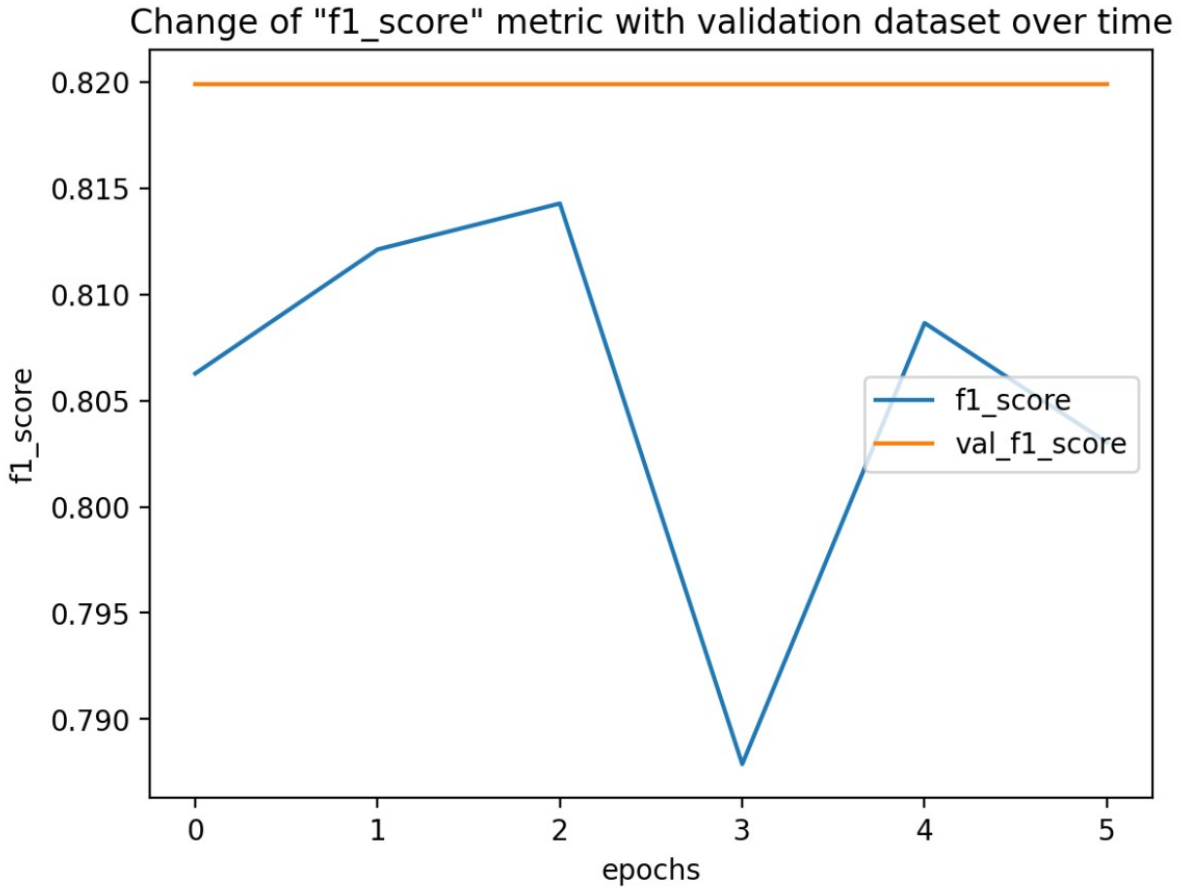
Embedded methods use internal representations of selected classifiers that implicitly perform feature relevance assessment, weighting or even selection during the learning process. Embedded methods are more complex, but can produce better results by taking into account interactions between features.

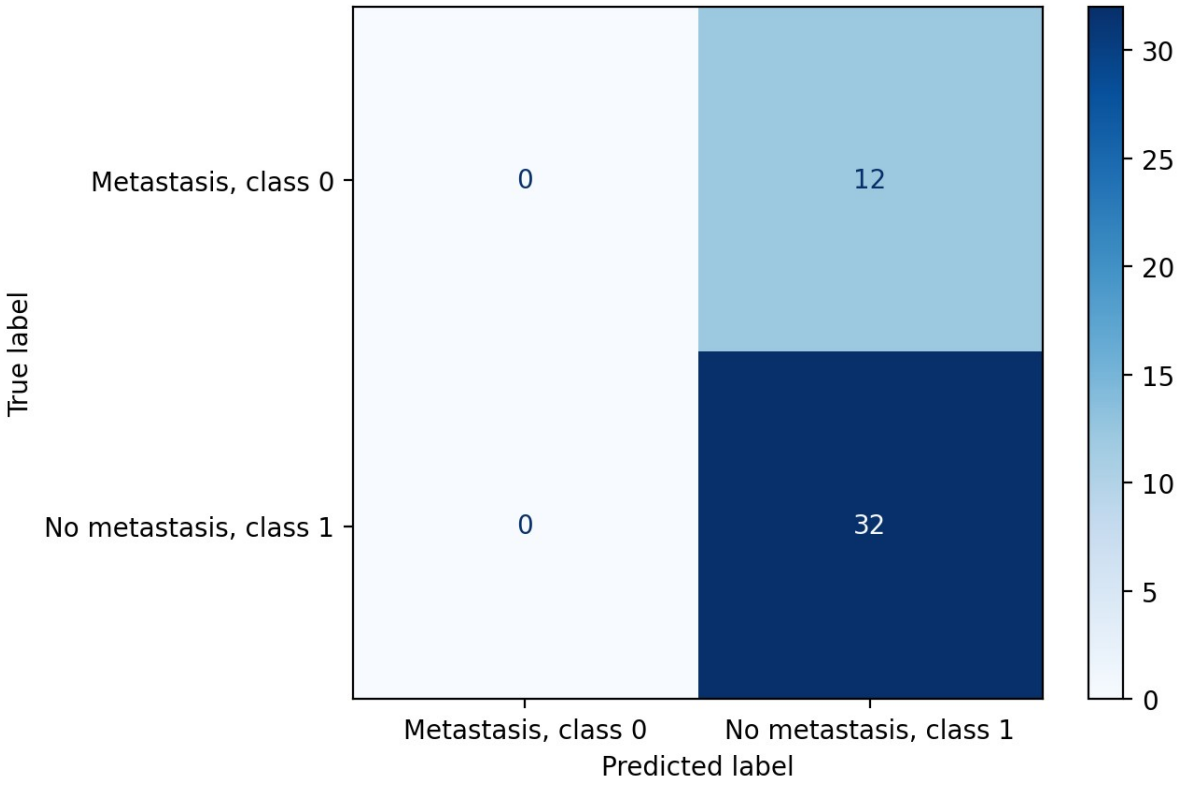
4.4.2. Results




Change of "loss" metric with validation dataset over time
0.68
0.66
0.64
0.62
0.60
0.58
loss
0
1
2
3
4
5
epochs
loss
val_loss


Change of "f1_score" metric over time
0.815
0.810
0.805
0.800
0.795
0.790
f1_score
0
1
2
3
4
5
epochs
f1_score


Change of "f1_score" metric with validation dataset over time
0.820
0.815
0.810
0.805
0.800
0.795
0.790
f1_score
0
1
2
3
4
5
epochs
f1_score
val_f1_score


Metastasis, class 0
No metastasis, class 1
True label
0
12
0
32
Metastasis, class 0
No metastasis, class 1
Predicted label
30
25
20
15
10
5
0

5. Conclusions

Conclusions from a data classification project with unbalanced classes

- **Unbalanced classes**

  The dataset consisted of 37 cases with metastasis (class ‘0’) and 94 cases without metastasis (class ‘1’), resulting in a significant class imbalance. This imbalance can lead to a situation where the model learns to recognise only the dominant class, ignoring the minority class. When dividing the training and test sets, a stratify parameter was used to maintain the proportion of classes (30% to 70%) in each set. This avoids the risk of the training set containing only instances of one class, providing more representative and balanced data to train the model. This step is crucial so that the model can learn to recognise both classes and not just the dominant one.

- **Feature selection**

  Initially, 5 features were selected for selection, but this proved to be insufficient. The model underperformed because it did not include enough information needed for accurate classification. The effectiveness of the model only improved with a selection of around 30 features. A higher number of features provided the model with more relevant information, which translated into better results. This result highlights the importance of selecting the number of features appropriately in the selection process.

- **Simple classifiers**

  Simple classifiers, such as logistic regression and decision trees, performed satisfactorily. They were able to classify the data effectively, despite their simplicity.The results of 5 iterations for each classifier were compared. The mean values and standard deviations from these iterations are presented in an error bar chart. This kind of presentation makes it easy to compare the stability and effectiveness of the different classifiers, which is helpful in selecting the most suitable model.

- **Optimisation of hyperparameters**

  The optimisation of the hyperparameters helped to improve the performance of the convolutional neural network (CNN). Many experiments were conducted to find the best hyper parameter settings, resulting in higher model accuracy. This optimisation is crucial to realise the full potential of the neural network.

- **Different feature selection methods**

  The ranking method proved to be the best method for feature selection, providing the best classification results. This method evaluates features based on their importance and selects those with the highest ranking. They were the least effective. In this method, features are selected as part of the model training process, but in this case they did not provide good enough results, suggesting that they were not able to effectively extract the most important features from the data. In addition, using the ranked (best) method on simple classifiers, a significant improvement in the results obtained was noted.

- **Bayes classifier**

  The Bayes classifier showed significant variation between iterations, suggesting instability in the model. This instability may be due to the sensitivity of the model to changes in the training data, making the results less predictable.

- **SVM classification with feature selection (embedded)**

  The SVM combined with the embedded method classified most cases into the negative class due to harvest bias. This resulted in a high efficiency of over 70%, but at the cost of misclassifying positive class cases. Due to the predominance of negative class cases, the model had difficulty learning the positive class, which affected its overall effectiveness. This shows the importance of dealing with bias in the data so that the model can effectively recognise both classes.

- **Model evaluation metrics**

  The F1-score proved to be a better metric for assessing CNN performance compared to other metrics, given unbalanced classes. The F1-score is particularly useful as it combines both precision and recall, giving a better picture of the overall performance of the model in the case of unevenly distributed classes. This results in a more equitable evaluation of the model, taking into account its ability to recognise both dominant and minority classes.

To summarise, it can be stated that

- The use of the stratify parameter is key to correctly partitioning the data under class imbalance, ensuring more representative training and test sets.
- The effectiveness of classification models can be significantly improved by appropriate feature selection and hyper parameter optimisation. The right number of features and optimal hyper parameter settings are key to achieving high model accuracy.
- Simple classifiers can work well, but their effectiveness can be improved through an iterative approach and analysis of the results, allowing a better understanding of the stability and effectiveness of the models.
- F1-score is the preferred metric for evaluating models in the case of class imbalance, as it better captures both precision and recall, which is particularly important when analysing data with unbalanced classes.

6. Sources

- https://www.cs.put.poznan.pl/ibladek/students/ed/lab2/ml.pdf
- http://fizyka.umk.pl/ftp/pub/publications/kmk/Prace-Mgr/07-Jakub-mag6.pdf
- https://www.ibm.com/topics/random-forest
- https://www.ibm.com/docs/pl/spss-modeler/saas?topic=models-how-svm-works
- https://gdudek.el.pcz.pl/images/Dydaktyka/Wyklad10_UM_KB.pdf
- https://en.wikipedia.org/wiki/Bayes%27_theorem
- https://www.ibm.com/topics/convolutional-neural-networks