POLITECHNIKA ŚLĄSKA W GLIWICACH

Wydział Automatyki, Elektroniki i Informatyki

Kierunek Informatyka

Magisterskie, stacjonarne, semestr 1

Wizja Komputerowa i Rozpoznawanie Obrazów

Sprawozdanie końcowe z projektu
Klasyfikacja danych dotyczących nowotworów

Samir Abu Safieh
Artur Stalmach
Rafał Gomola
Maksymilian Kisiel

Spis treści:

1. Wstęp

Celem projektu było wykonanie klasyfikatora służącego do przeanalizowania danych medycznych dotyczących nowotworów. Należało do tego celu porównać różne metody selekcji cech, takie jak rankingowe, opakowane i wbudowane, oraz znaleźć najbardziej optymalną.

2. Prosty klasyfikator

2.1. Czym są klasyfikatory?

Klasyfikatory wykorzystywane są w uczeniu maszynowym, jak nazwa wskazuje, do klasyfikacji. Ich działanie polega na tym, że dokonują klasyfikacji danych wejściowych na odpowiednie klasy, na podstawie cech tych danych. Uczą się one rozpoznawać wzorce na podstawie danych treningowych. Klasyfikatory są stosowane m.in. w analizie tekstu, czy też w rozpoznawaniu obrazów.

2.2. Klasyfikator SVM

Maszyna wektorów nośnych (Support Vector Machines, SVM) jest algorytmem używanym do liniowej i nieliniowe klasyfikacji, regresji, czy też do wykrywania wartości odstających. Działa w ten sposób, że dane są mapowane na wielowymiarową przestrzeń, w taki sposób aby możliwe było ich oddzielenie.

2.3. Klasyfikator KNN

K-Nearest Neighbors (KNN) jest prostym klasyfikatorem, który klasyfikuje wybrany punkt na podstawie najbliższych sąsiednich punktów w przestrzeni cech. Aby określić, które punkty danych są najbliższe wybranemu punktowi zapytania, należy obliczyć odległość pomiędzy punktem zapytania a innymi punktami danych.

2.4. Klasyfikator “Random Forest”

Klasyfikator Random Forest jest algorytmem uczenia maszynowego opartym na technice ensemble learning, który tworzy zbiór drzew decyzyjnych z losowo wybranych podzbiorów danych treningowych. Każde drzewo decyzyjne jest trenowane na innym losowym podzbiorze danych oraz cech, a wynik końcowy klasyfikacji jest określany przez większościowy głos wszystkich drzew (dla klasyfikacji) lub średnią wyników (dla regresji).

2.5. Klasyfikator Bayesa

Klasyfikator Bayesa opiera się na twierdzeniu Bayesa (jest to twierdzenie teorii prawdopodobieństwa, wiążące prawdopodobieństwa warunkowe dwóch zdarzeń warunkujących się nawzajem), dzięki czemu zawdzięcza swą nazwę. Twierdzenia używa się go do przewidywania klasy danego punktu danych na podstawie jego cech. Klasyfikator ten oblicza prawdopodobieństwo przynależności punktu do każdej z możliwych klas i przypisuje go do klasy o najwyższym prawdopodobieństwie.

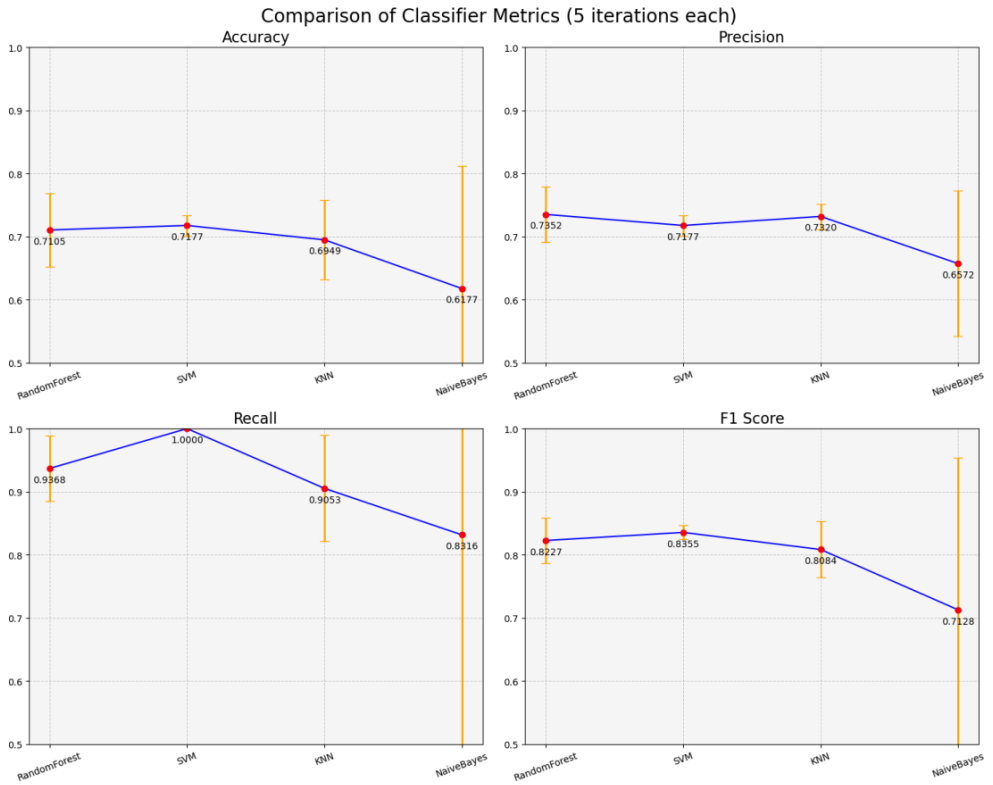
Twierdzenie Bayesa:

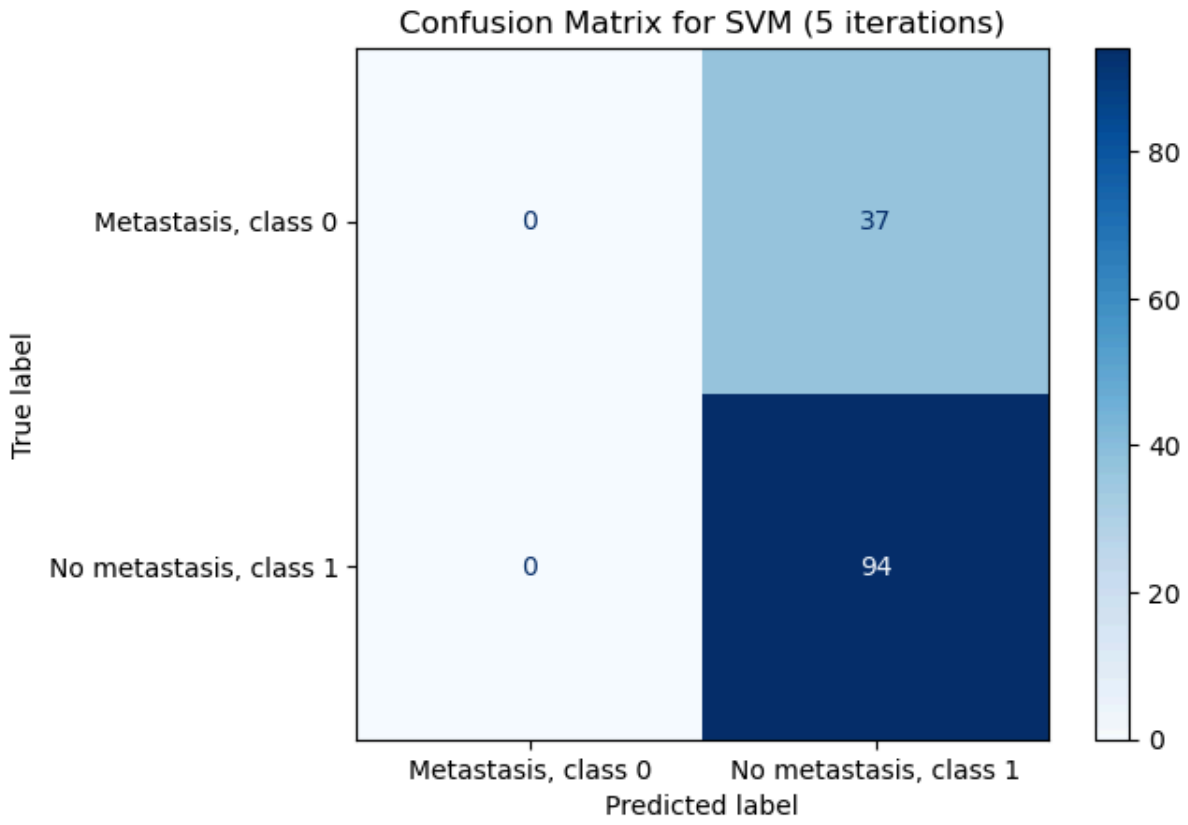
$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(B \mid A)\,\mathrm{P}(A)}{\mathrm{P}(B)},$$

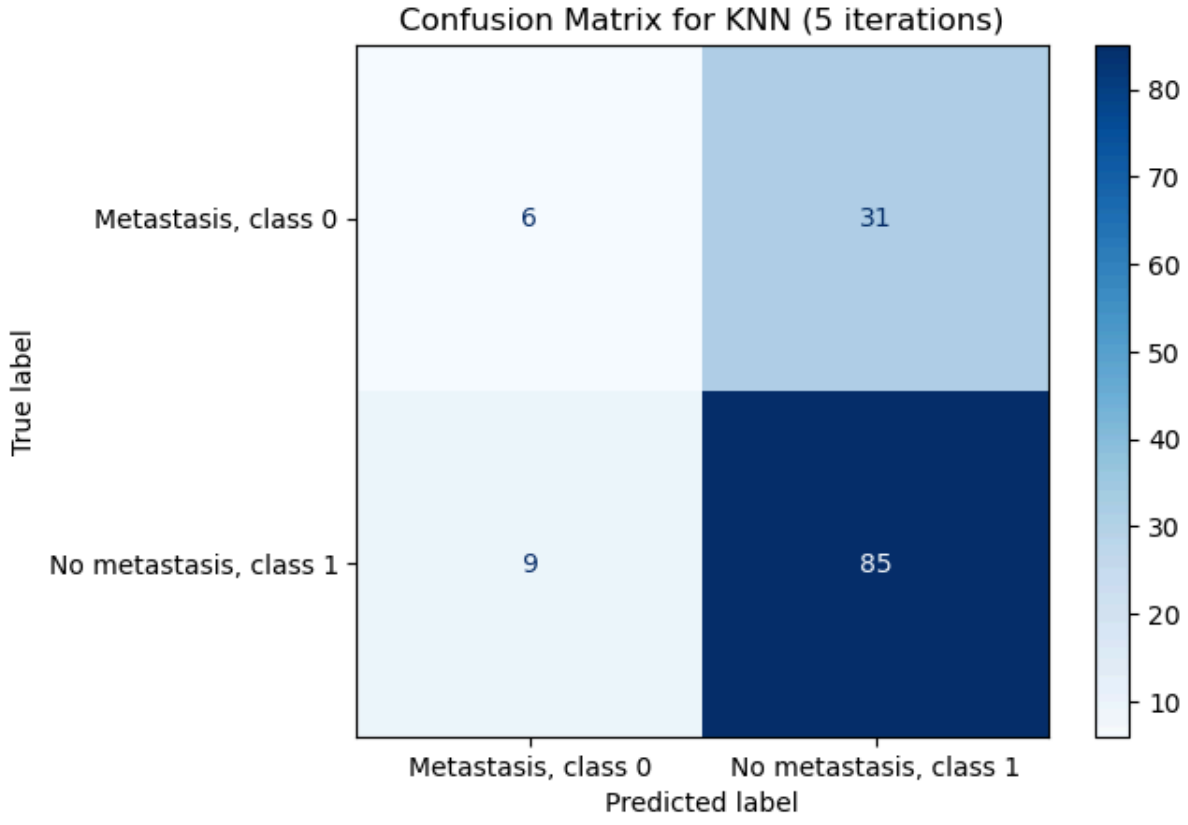
Gdzie A i B są punktami, przy czym:

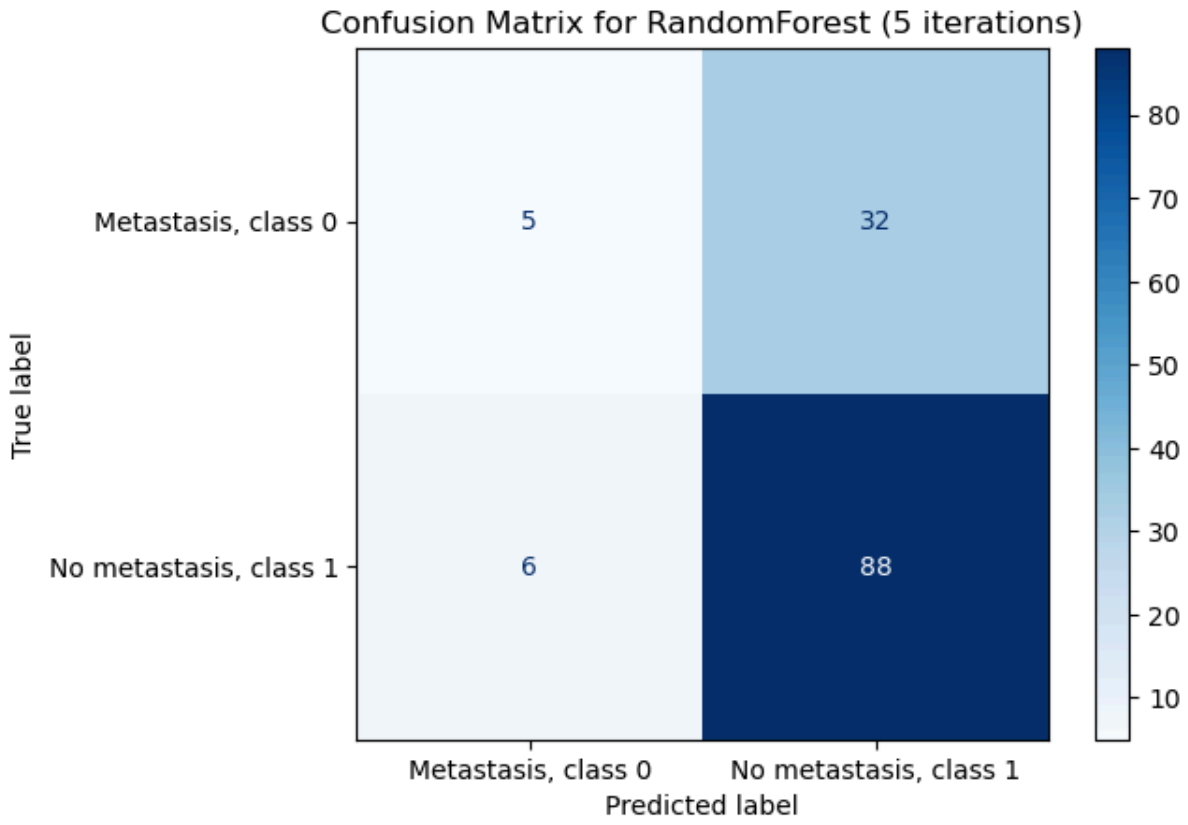
- P(A|B) oznacza prawdopodobieństwo warunkowe, tj. prawdopodobieństwo zajścia zdarzenia A, o ile zajdzie zdarzenie B.
- P(A|B) oznacza prawdopodobieństwo zajścia zdarzenia B, o ile zajdzie zdarzenie A.

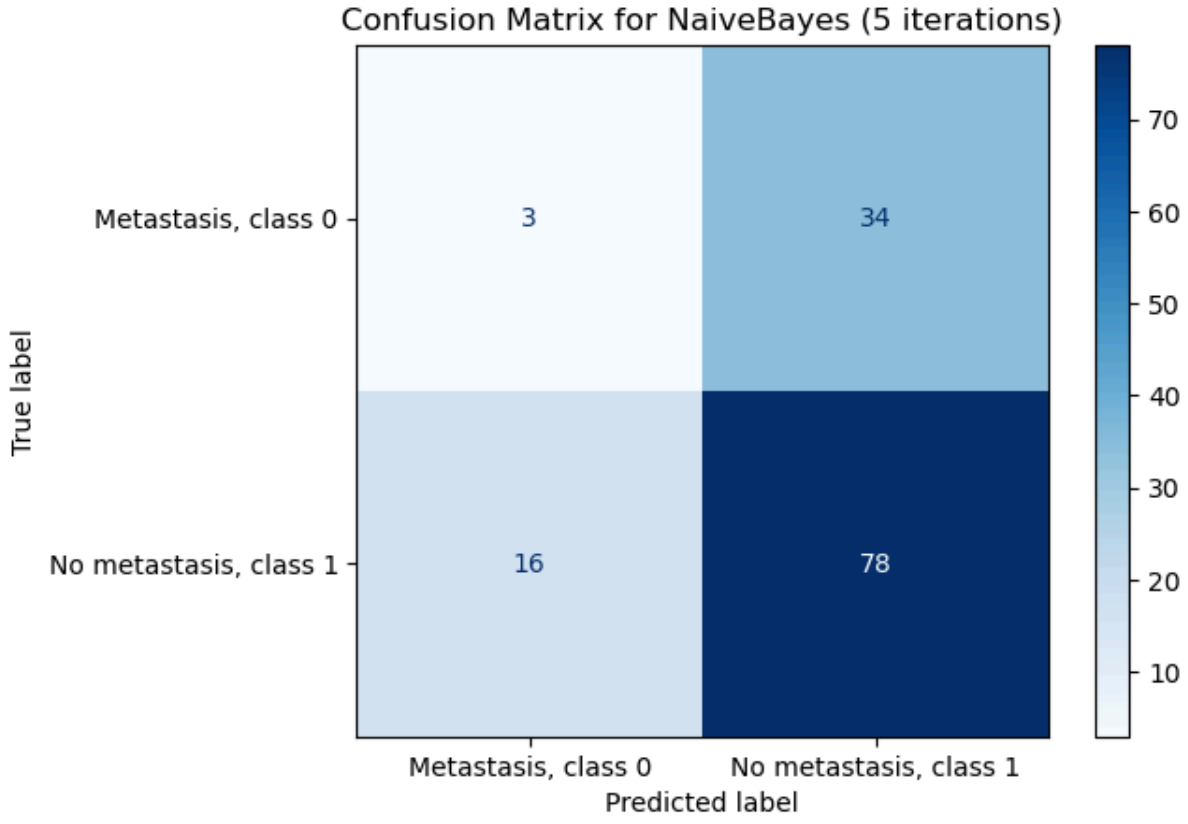
2.6. Wyniki




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

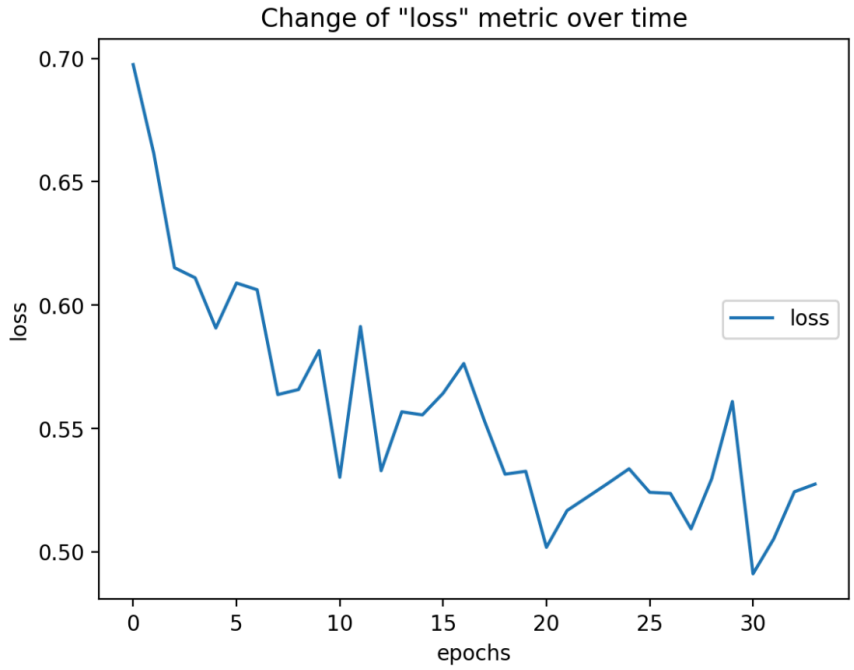

Confusion Matrix for NaiveBayes (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
3
34
16
78
Metastasis, class 0
No metastasis, class 1
Predicted label
70
60
50
40
30
20
10

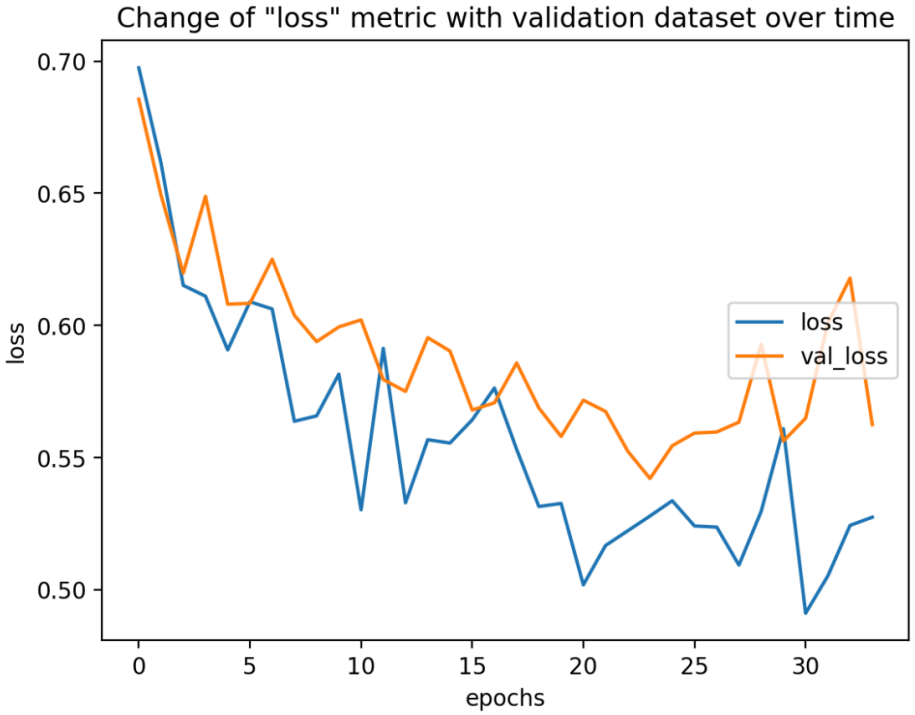
3. Sieć neuronowa konwolucyjna (zaawansowany klasyfikator)

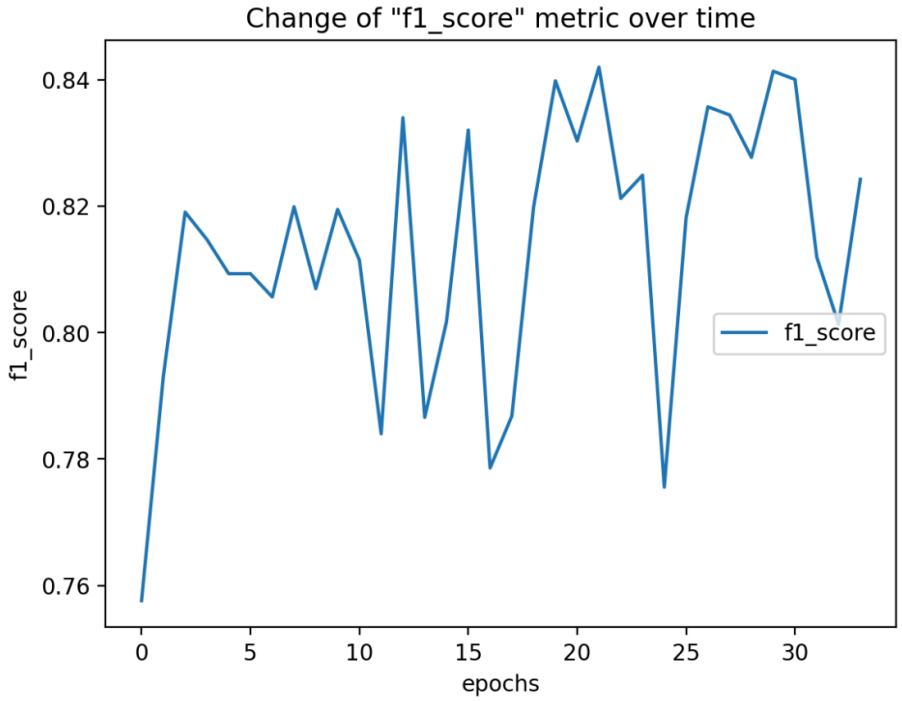
3.1. Czym jest sieć neuronowa konwolucyjna?

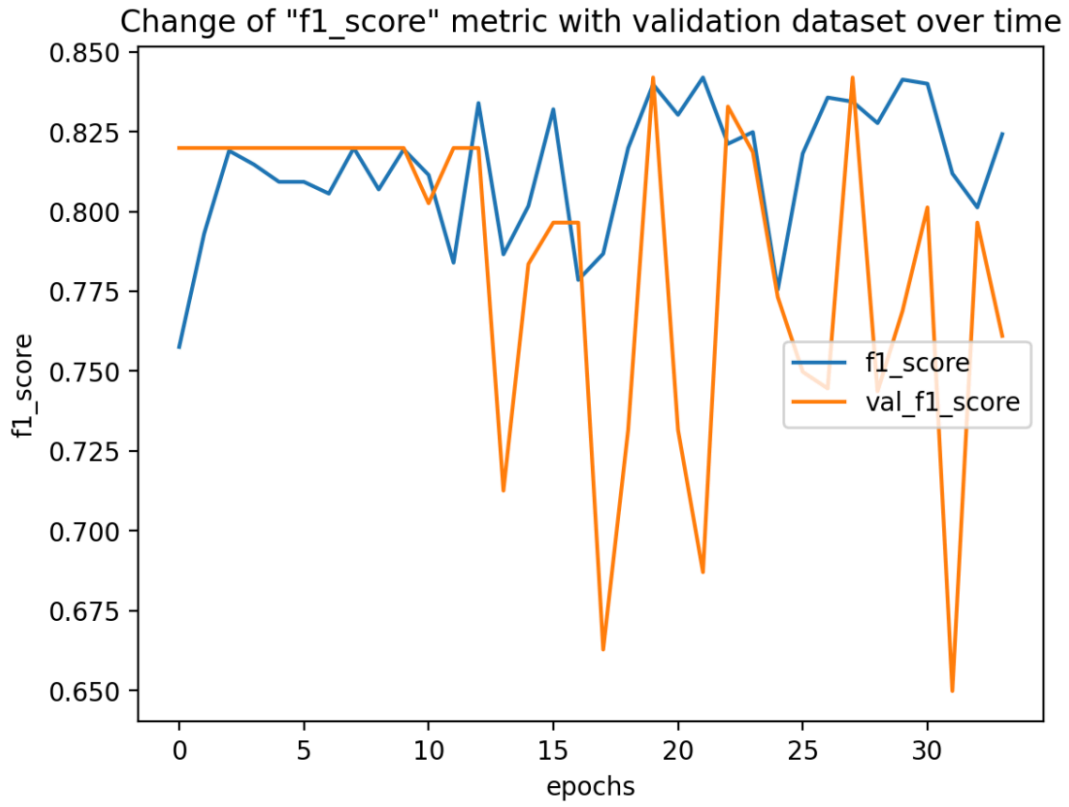
Sieć Neuronowa Konwolucyjna (CNN) to rodzaj sieci neuronowej, zaprojektowany głównie do przetwarzania danych o strukturze siatki, takich jak obrazy. CNN składa się z warstw konwolucyjnych, które używają małych filtrów do wykrywania lokalnych wzorców, takich jak krawędzie i tekstury, oraz warstw ReLU, które wprowadzają nieliniowość. Warstwy poolingowe zmniejszają wymiary danych, zachowując kluczowe informacje. Po przetworzeniu przez kilka takich warstw, dane są spłaszczane i przetwarzane przez w pełni połączone warstwy, które klasyfikują dane. CNN są skuteczne w zadaniach związanych z rozpoznawaniem obrazów, przetwarzaniem wideo i analizą tekstu dzięki swojej zdolności do automatycznego wyodrębniania złożonych cech.

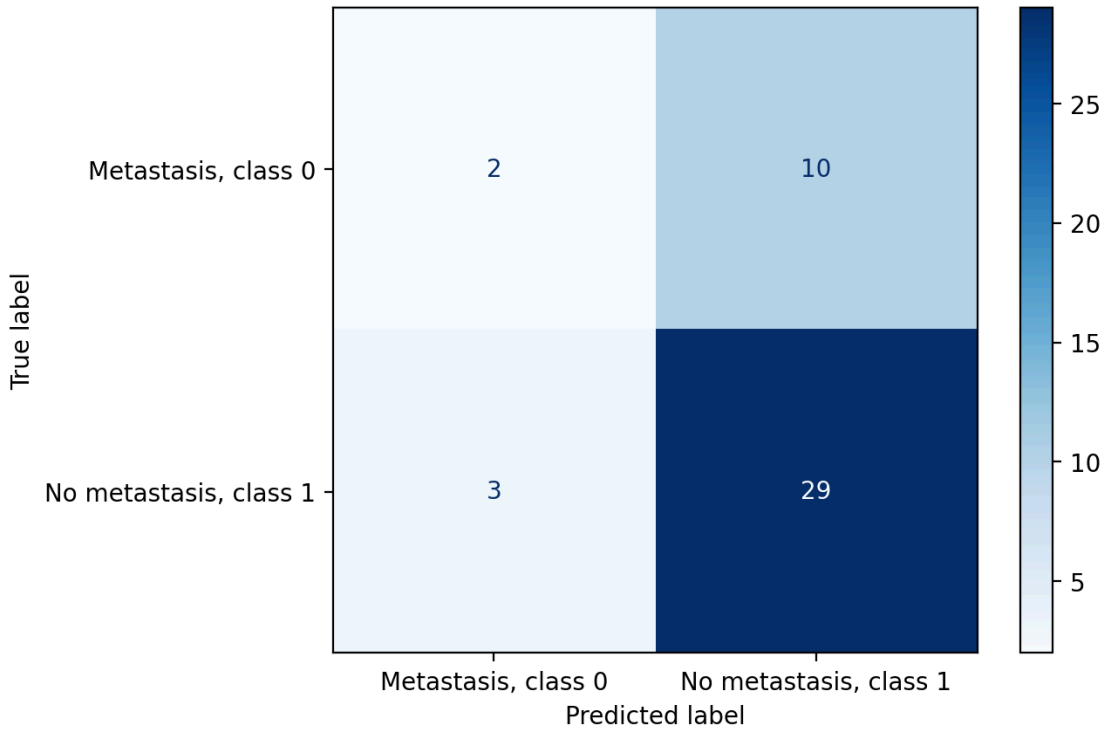
3.2. Wyniki




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


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


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. Metody selekcji cech

4.1. Do czego służą?

Dobór odpowiedniej metody selekcji ma duży wpływ na wydajność algorytmu klasyfikacyjnego. Stosuje się je do redukcji złożoności i wymagań obliczeniowych. Eliminacja nieistotnych cech może poprawić dokładność modelu poprzez skupienie się na bardziej znaczących danych. Redukcja liczby cech przyspiesza proces trenowania modelu i zmniejsza zapotrzebowanie na pamięć.

4.2. Metoda rankingowa

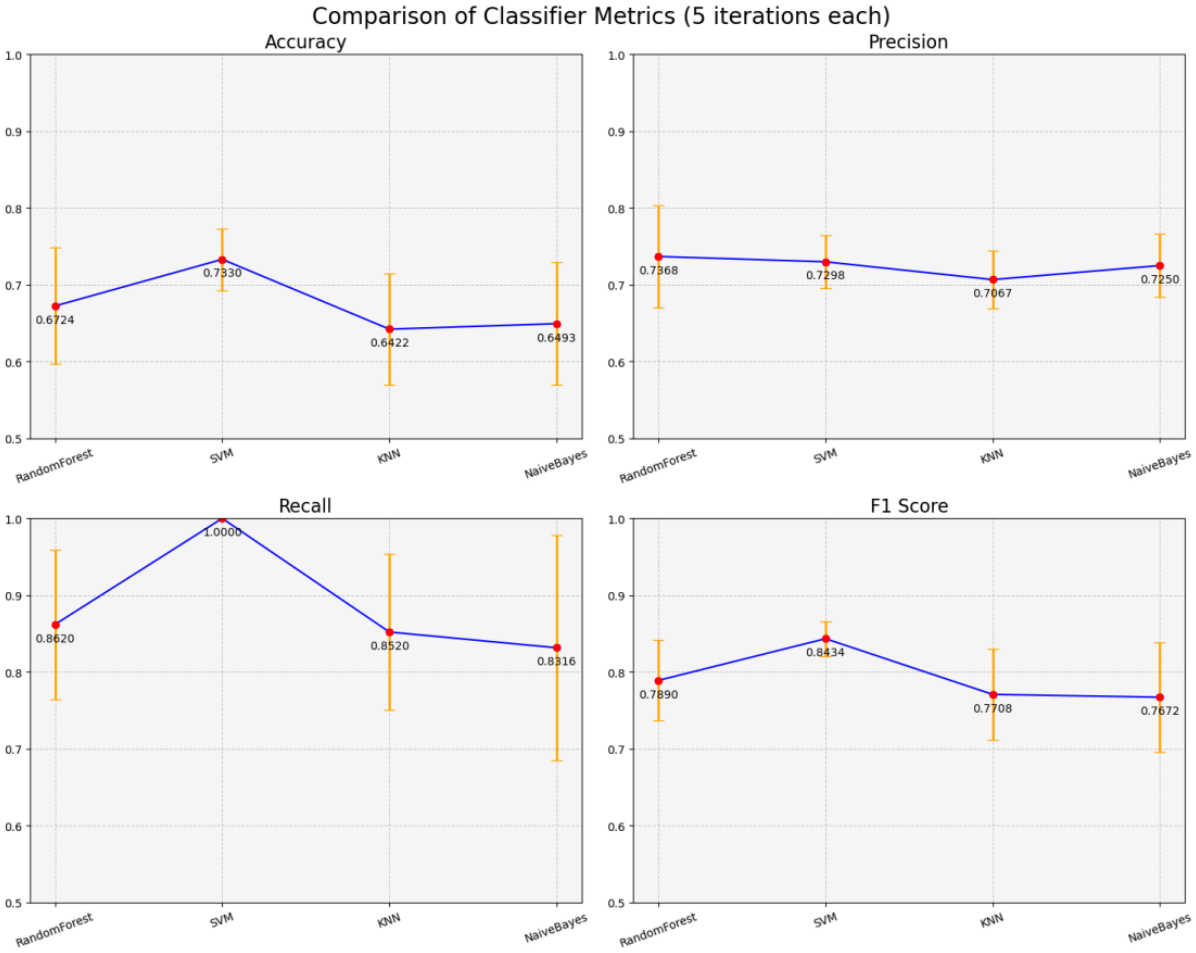
4.2.1. Opis

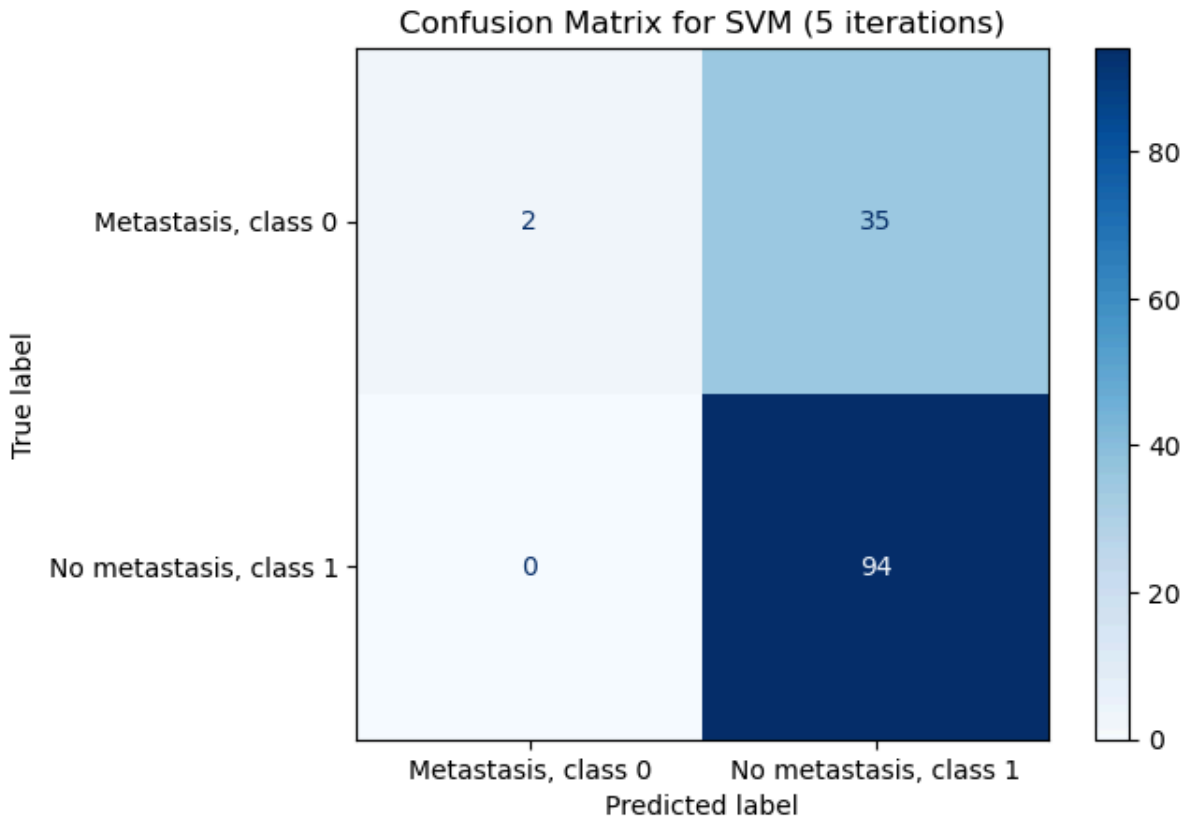
Metody rankingowe, nazywane również metodami filtracyjnymi (ang. filter methods) są jednymi z najprostszych metod selekcji danych. Ich najważniejszą cechą jest niska złożoność obliczeniowa.

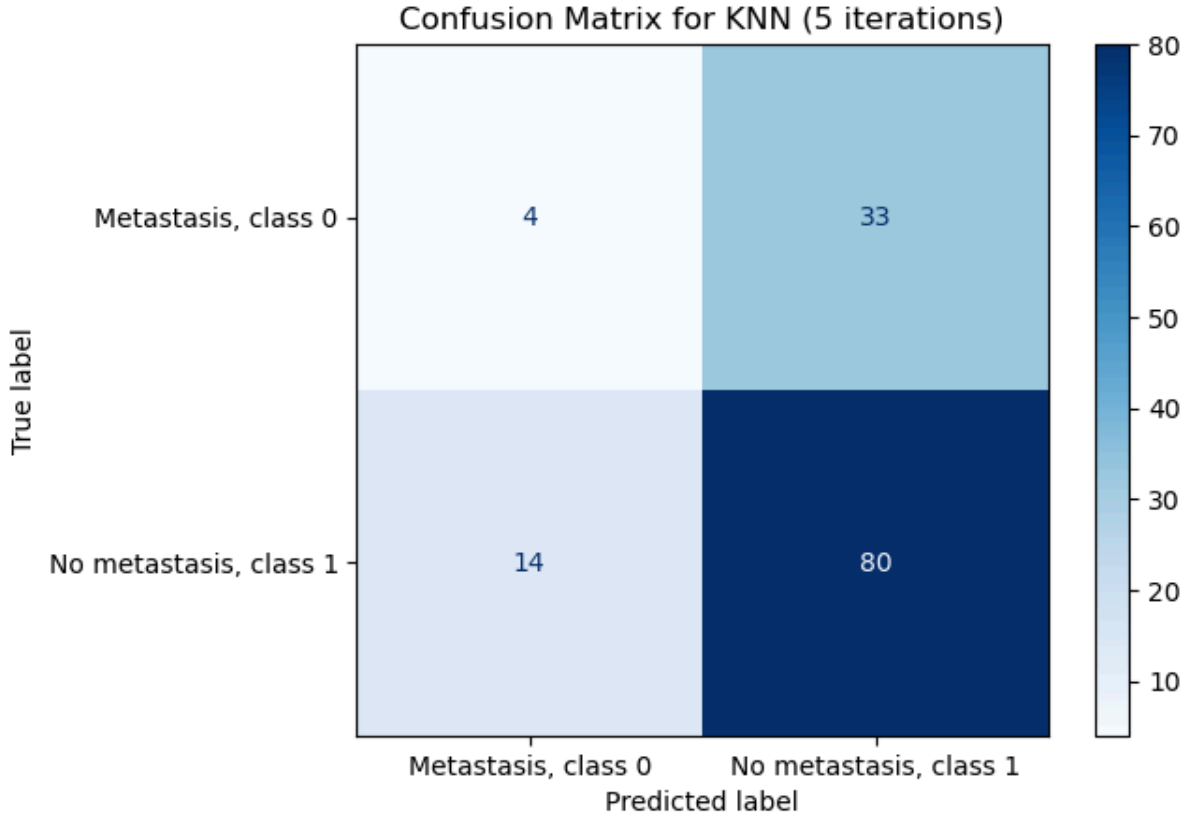
Aby stworzyć taką metodę należy do każdej cechy przyporządkować indeks określający jej jakość na podstawie przyjętych kryteriów. Wartość indeksu wykorzystywana jest do posortowania cech. Dzięki tej metodzie wybierane są najlepsze cechy (znajdujące się najwyżej w rankingu), a najgorsze odrzucane. Próg wyznaczający cechy do odrzucenia może być ustalony według potrzeb i opierać się na określonej wartości indeksu, bądź też opierać się na konkretnej liczbie najlepszych cech do wyselekcjonowania.

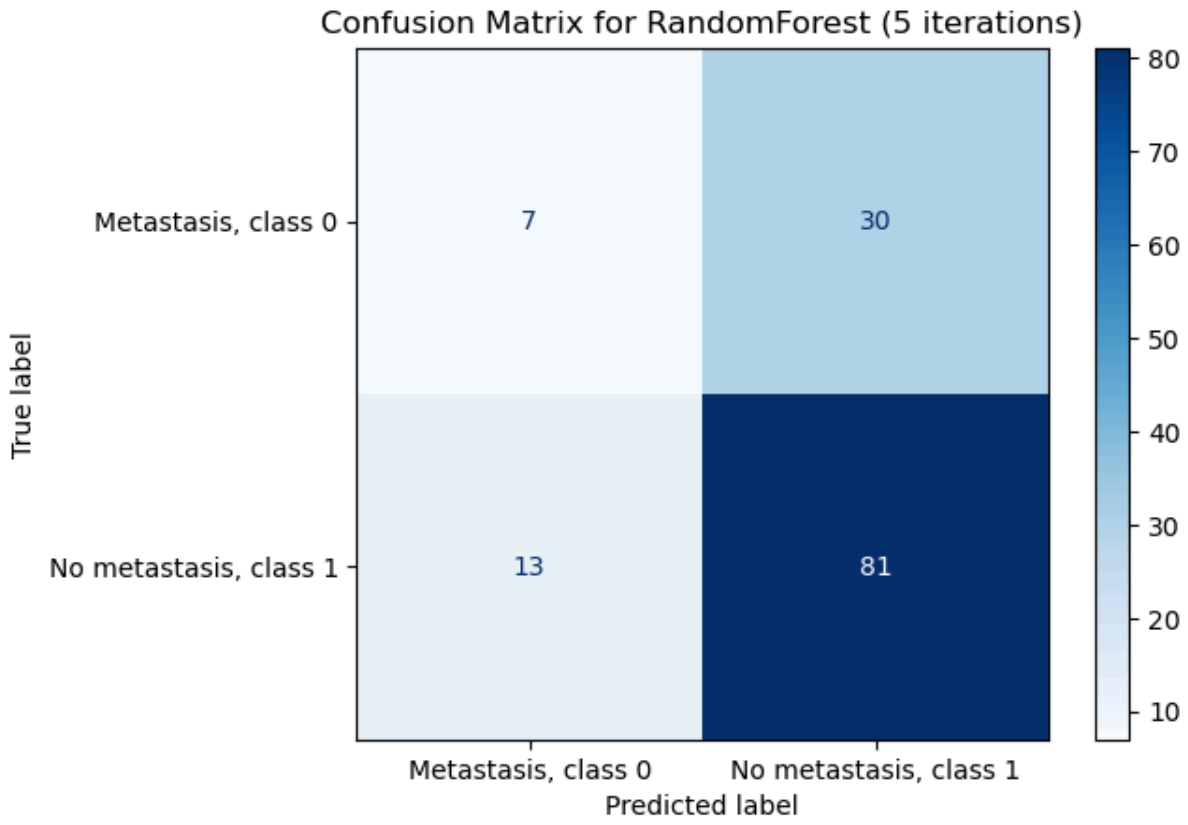
4.2.2. Wyniki dla klasyfikatorów prostych

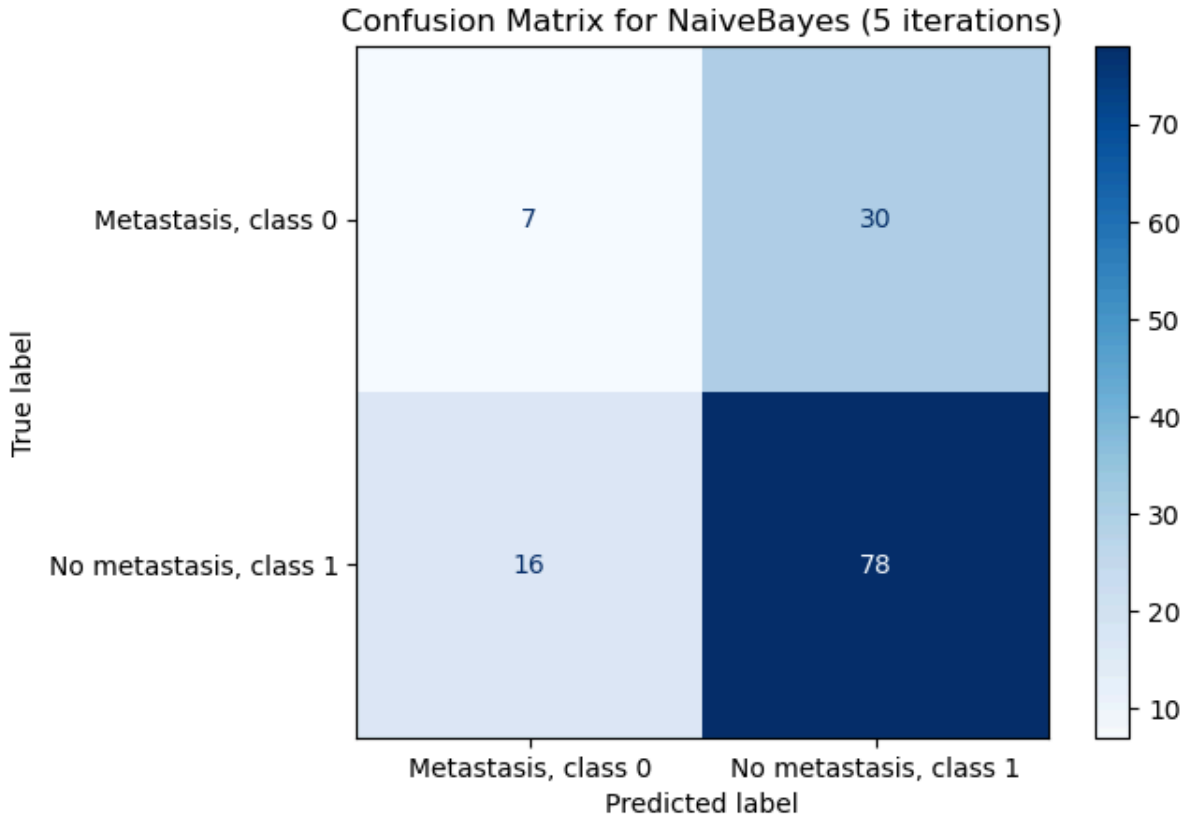
Metoda rankingowa została również zastosowana dla prostych klasyfikatorów.



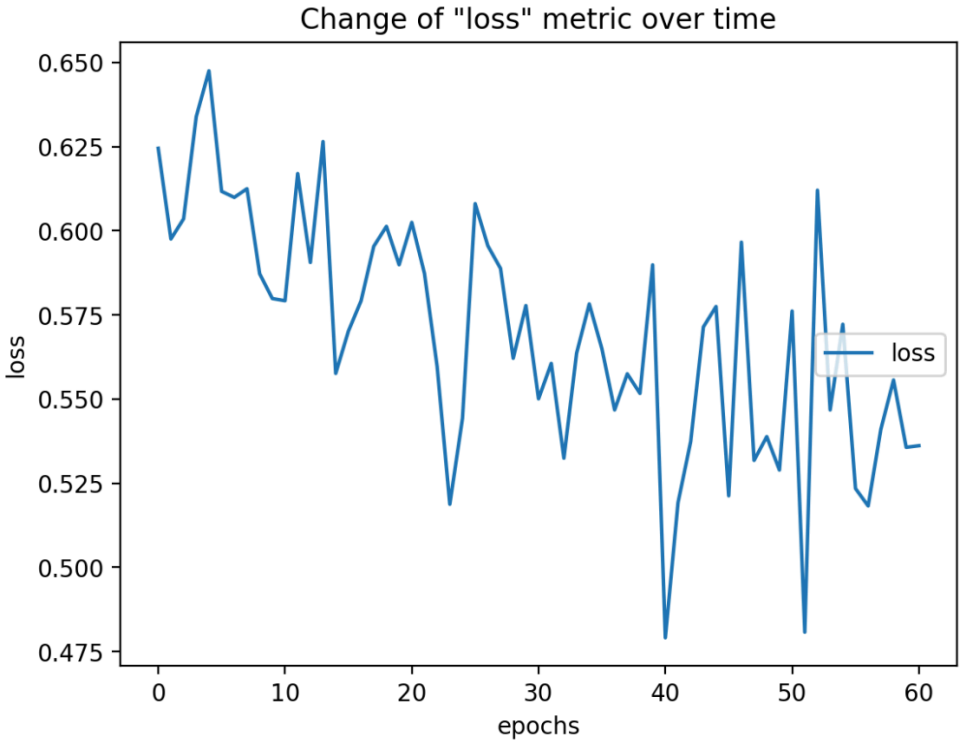

Confusion Matrix for SVM (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
2
35
0
94
Metastasis, class 0
No metastasis, class 1
Predicted label
80
60
40
20
0


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

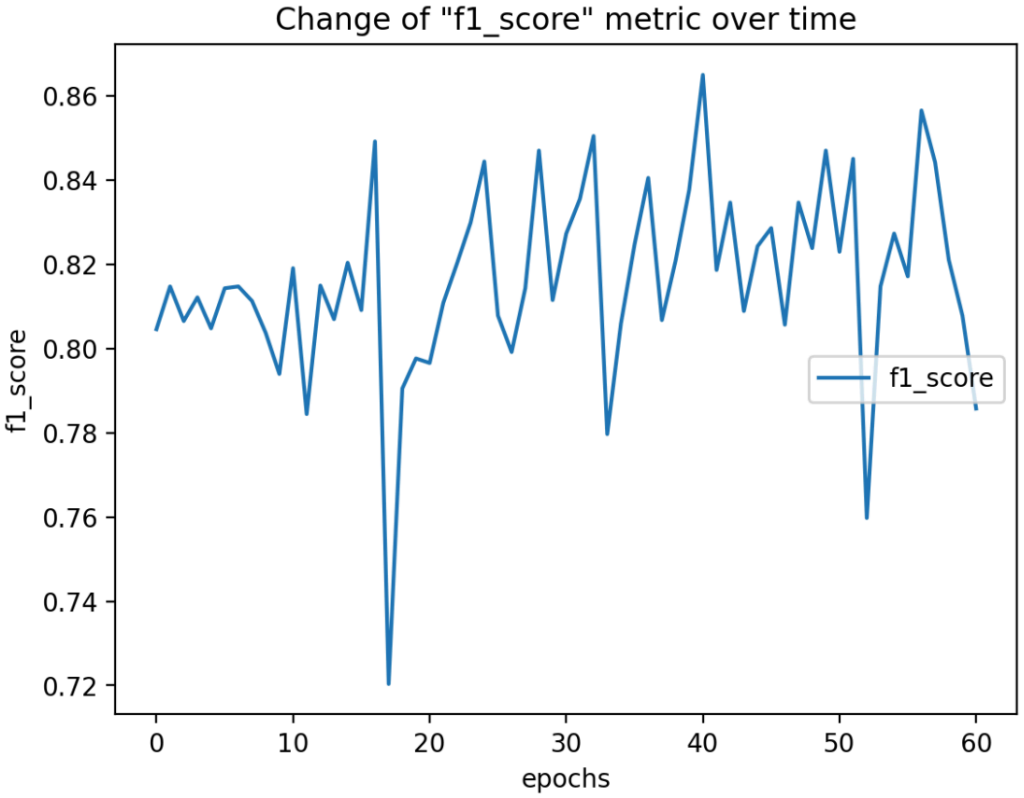

Confusion Matrix for RandomForest (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
7
30
13
81
Metastasis, class 0
No metastasis, class 1
Predicted label
80
70
60
50
40
30
20
10

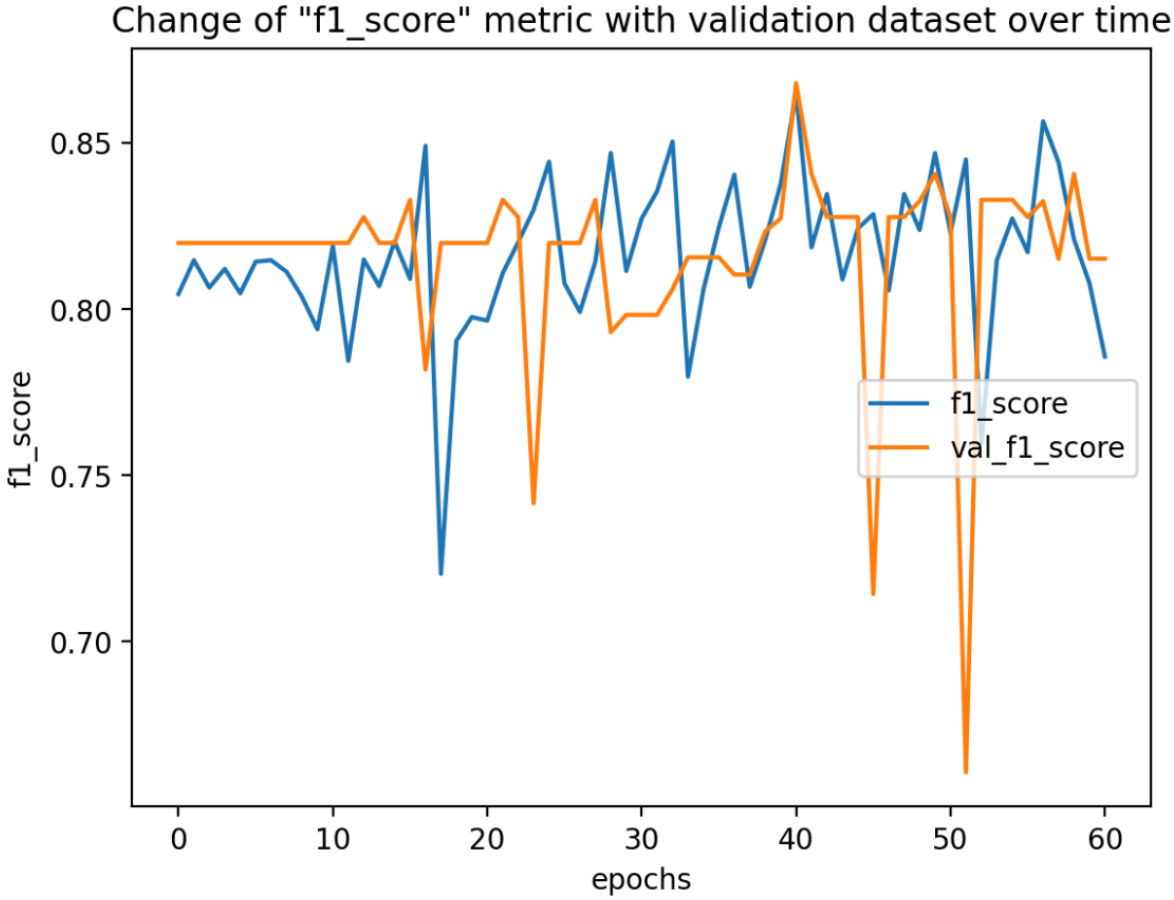

Confusion Matrix for NaiveBayes (5 iterations)
Metastasis, class 0
No metastasis, class 1
True label
7
30
16
78
Metastasis, class 0
No metastasis, class 1
Predicted label
70
60
50
40
30
20
10

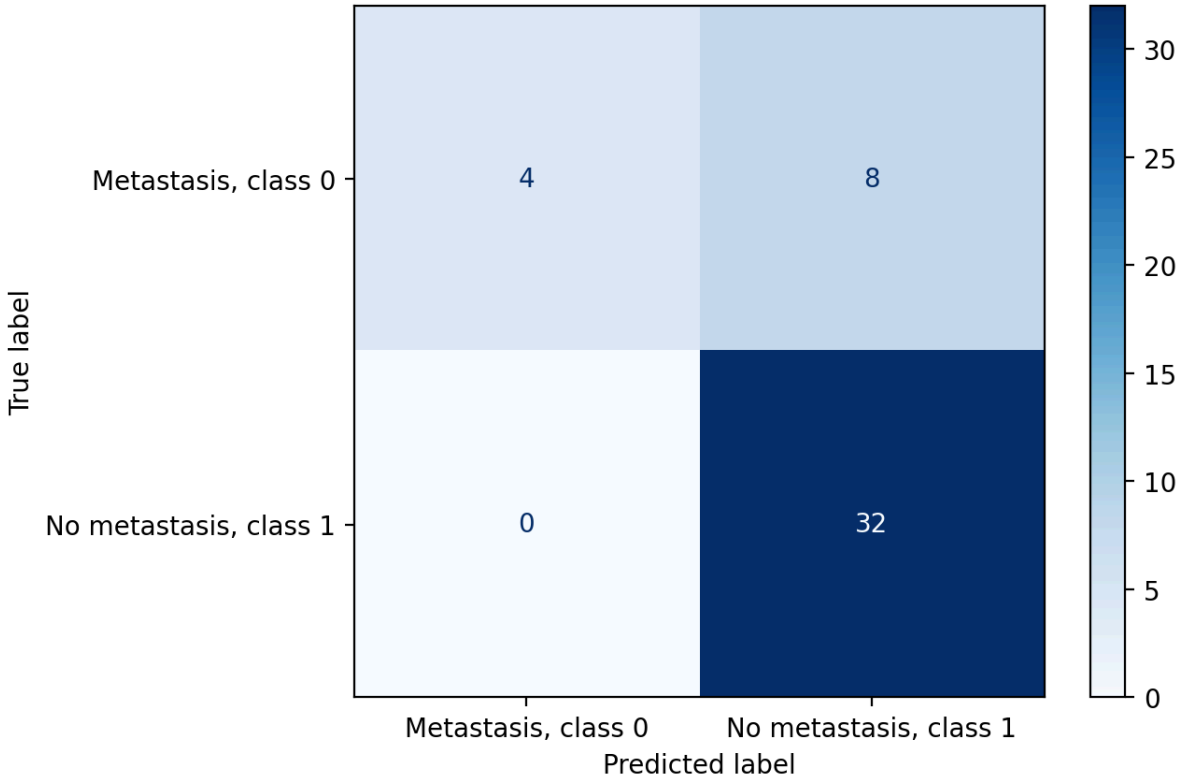
4.2.3. Wyniki dla sieci konwolucyjnej




Change of "loss" metric with validation dataset over time
loss
epochs
loss
val_loss
0.70
0.65
0.60
0.55
0.50
0.45
0
10
20
30
40
50
60


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
0.85
0.80
0.75
0.70
0
10
20
30
40
50
60
epochs
f1_score
val_f1_score



4.3. Metody opakowane (wrapped)

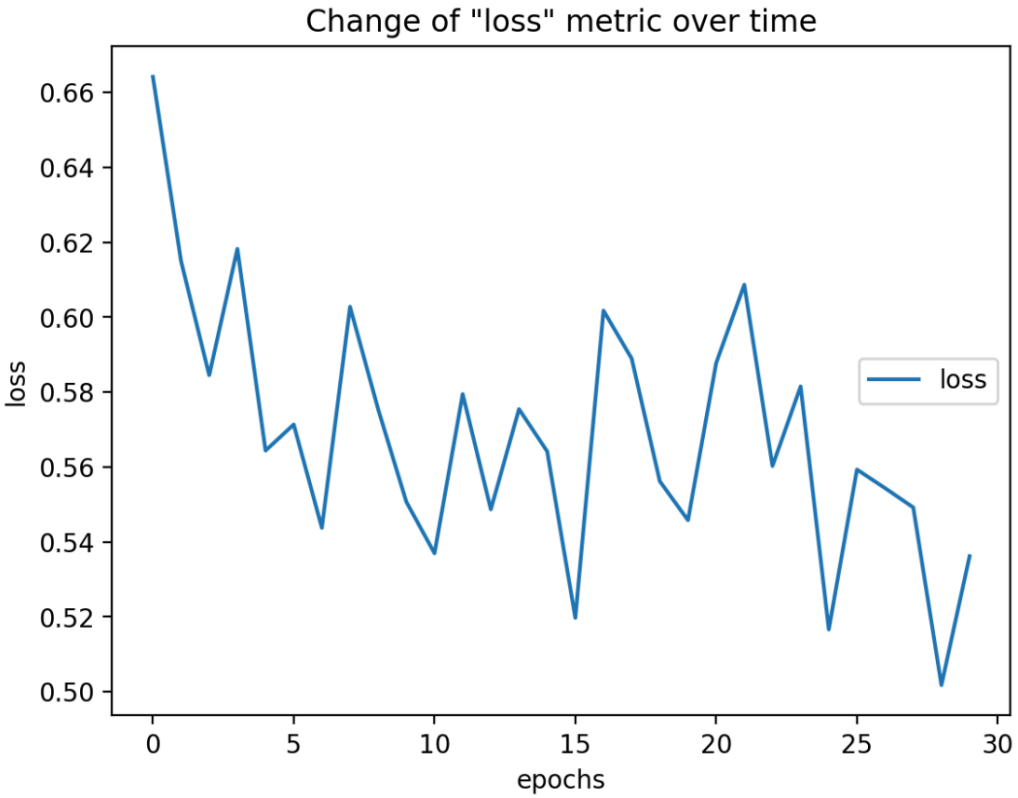
4.3.1. Opis

Metody opakowane cechują się dużo większą złożonością obliczeniową, niż metody rankingowe. Podczas gdy w przypadku metod rankingowych cechy rozpatrywane są niezależnie od siebie, we wrapperach kluczowe są zależności między cechami. Wyszukiwany jest najkorzystniejszy zestaw cech. Porównuje się różne ich kombinacje. Ułatwia to wykrycie możliwych interakcji pomiędzy zmiennymi.

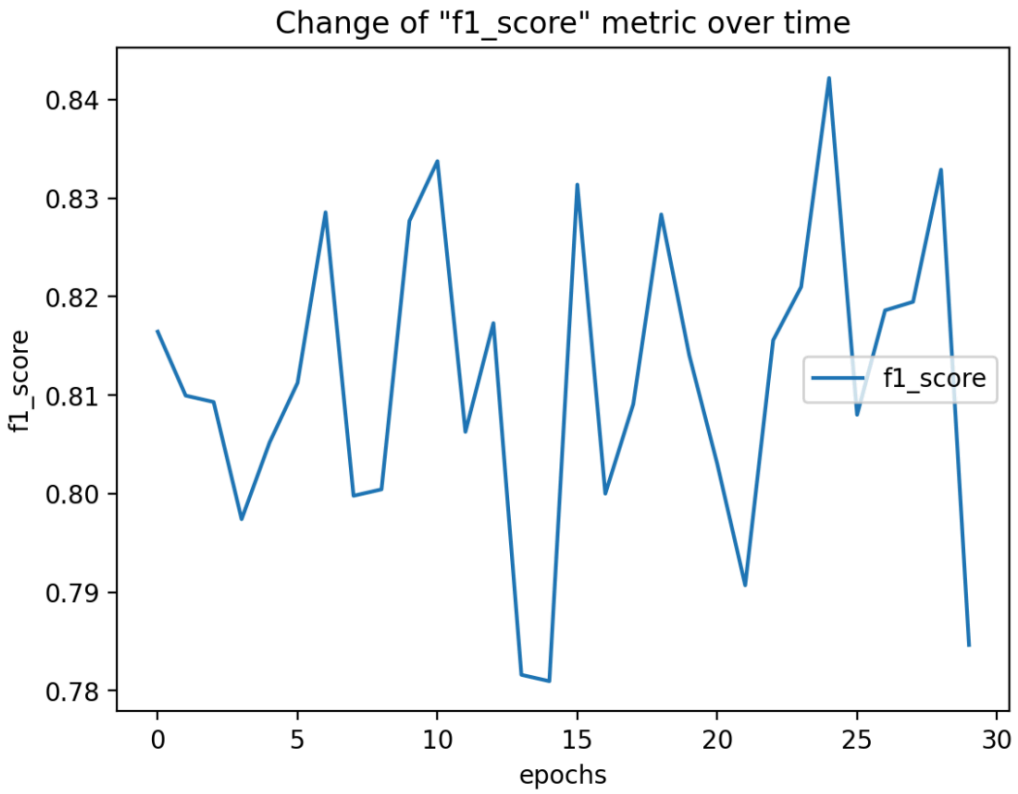
Aby ułatwić znalezienie optymalnej kombinacji cech w podzbiorze, przy możliwie najlepszej złożoności obliczeniowej stosuje się 2 podejścia:

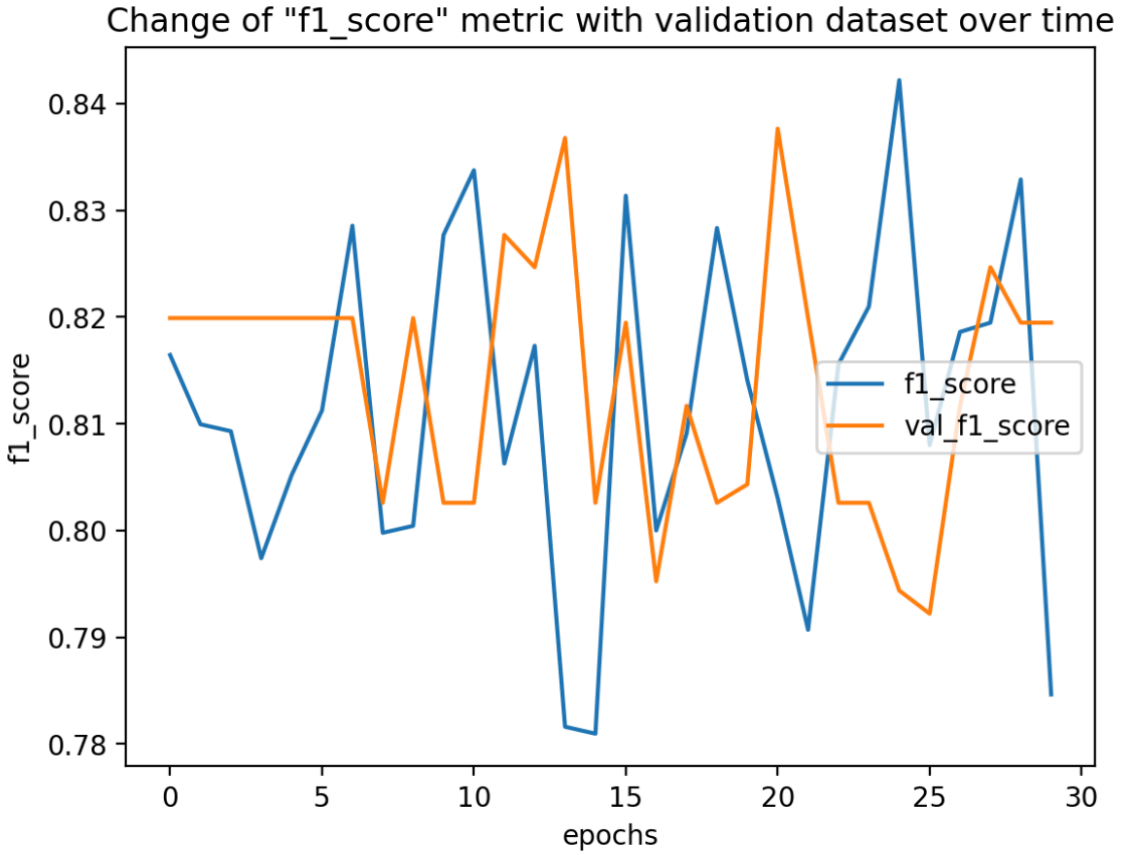
- Forward selection- zaczyna się od pustego zbioru i kolejno dodaje zmienne,
- Backward selection- zaczyna się od zbioru wszystkich możliwych cech, które kolejno się odrzuca.

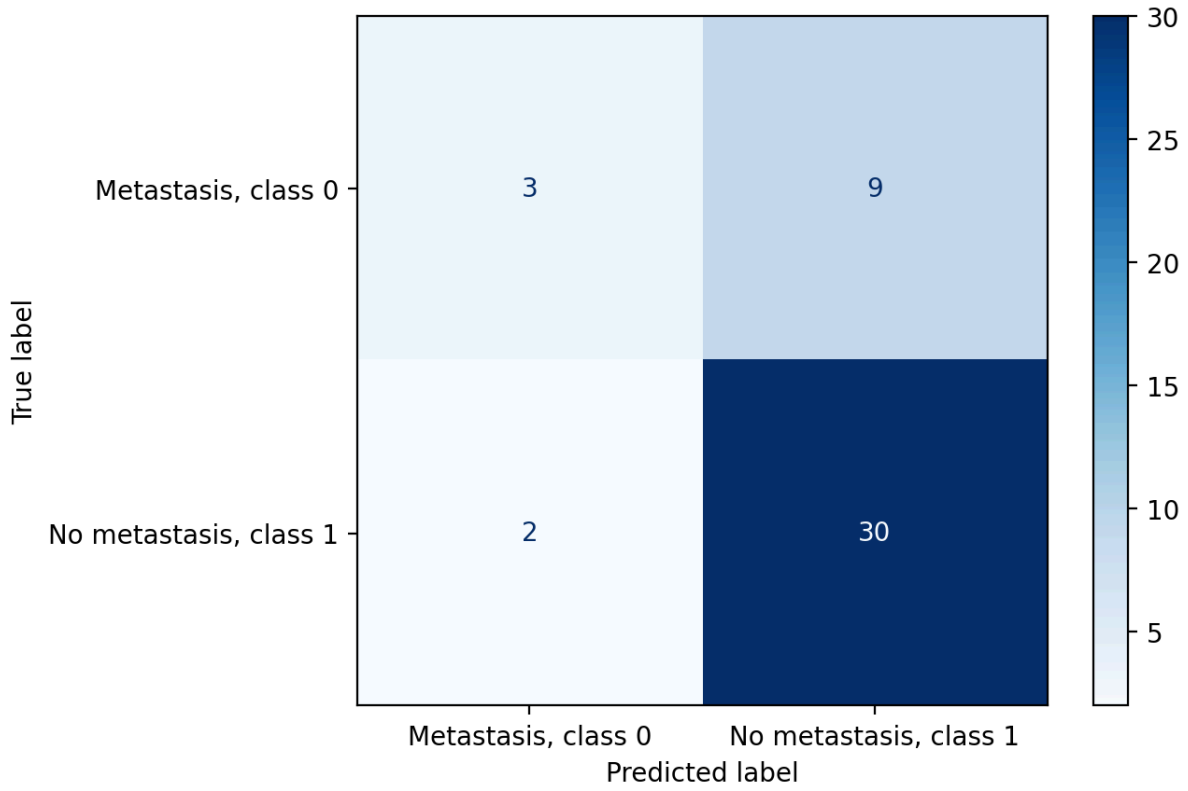
4.3.2. Wyniki


Change of "loss" metric over time
loss
epochs
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
loss


Change of "loss" metric with validation dataset over time
loss
epochs
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
loss
val_loss


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

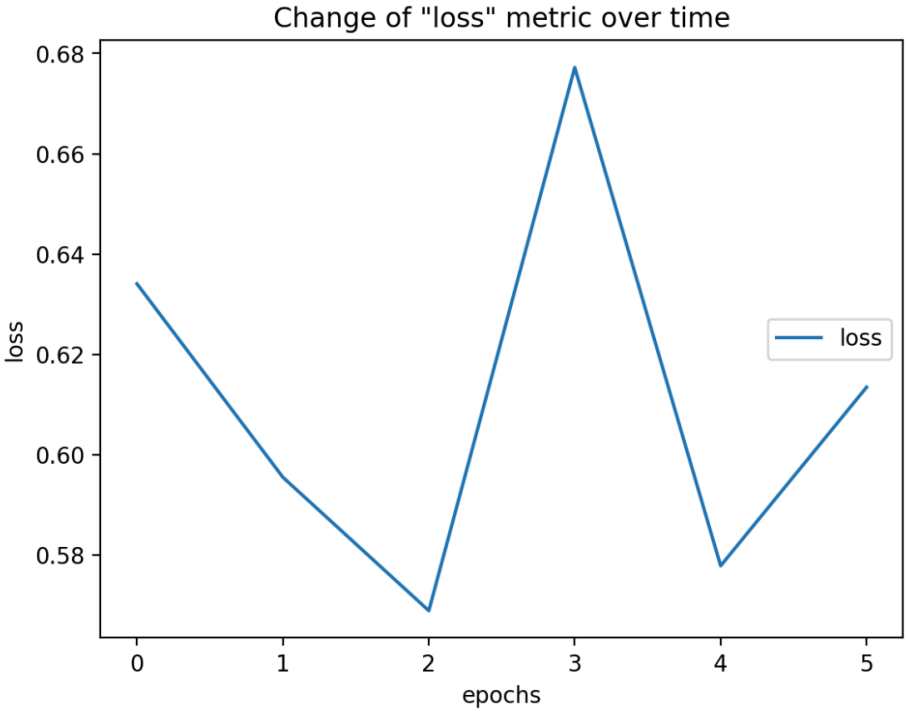


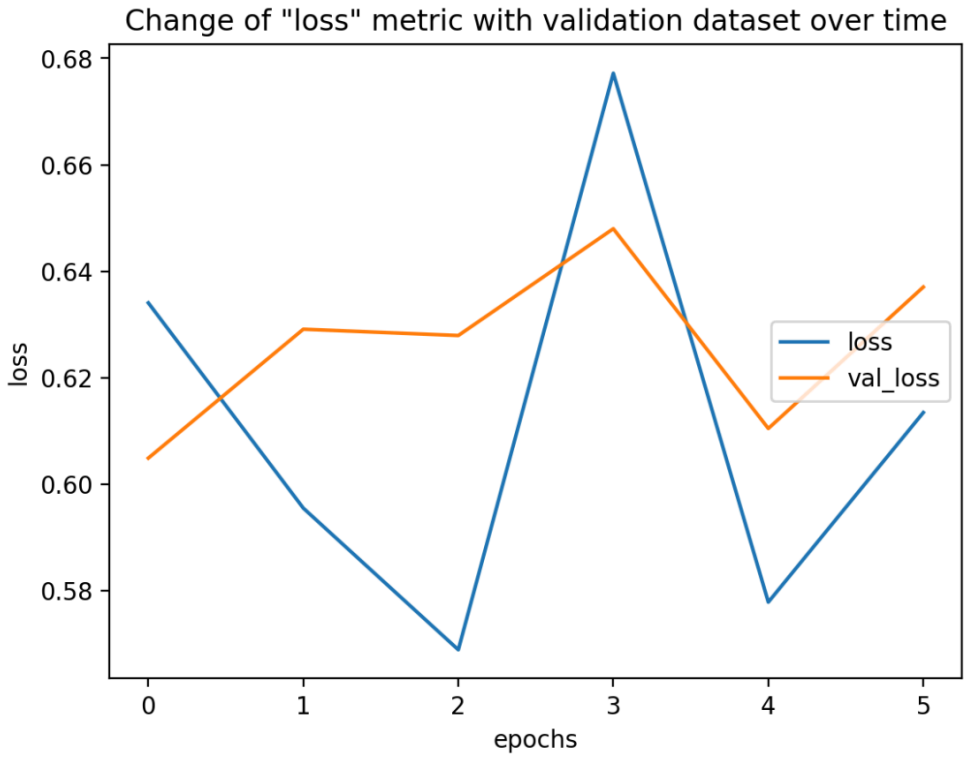
4.4. Metody wbudowane (embedded)

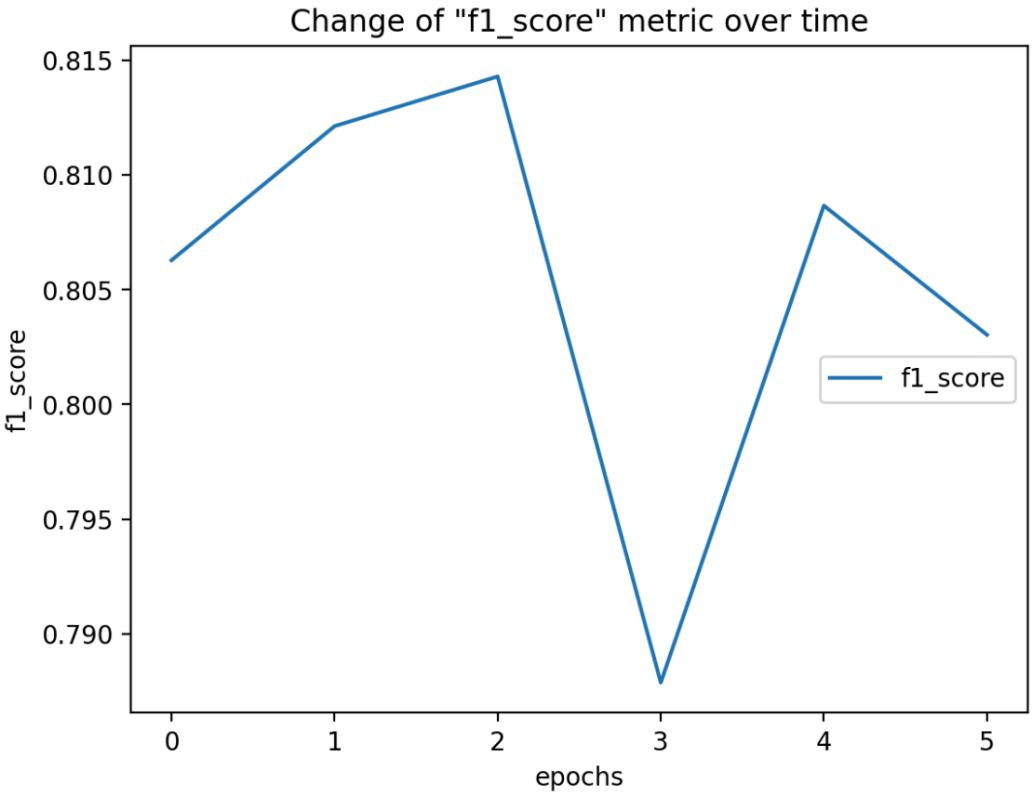
4.4.1. Opis

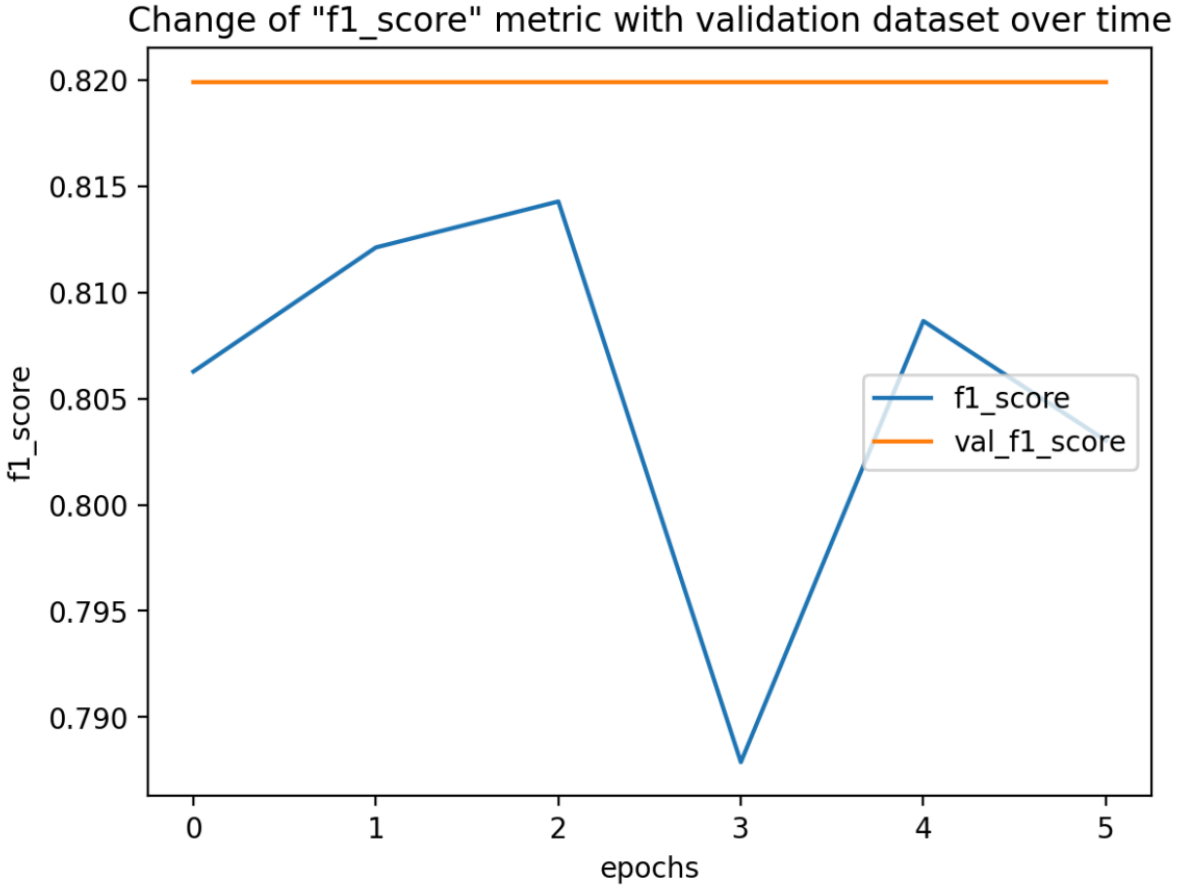
Metody wbudowane (embedded methods) korzystają z wewnętrznych reprezentacji wybranych klasyfikatorów, które podczas procesu uczenia dokonują pośrednio oceny przydatności cech, ich ważenia bądź wręcz selekcji. Metody wbudowane są bardziej złożone, ale mogą dawać lepsze wyniki, uwzględniając interakcje między cechami.

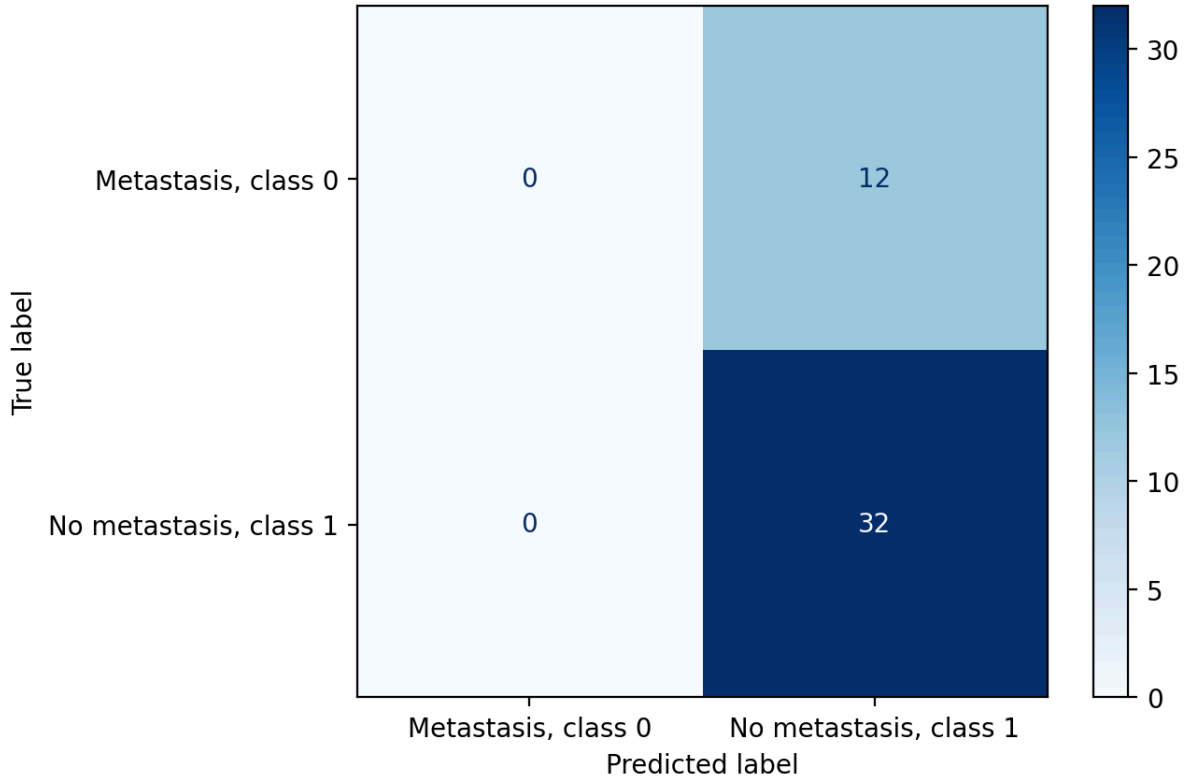
4.4.2. Wyniki


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


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



5. Wnioski

Wnioski z projektu dotyczącego klasyfikacji danych z niezbalansowanymi klasami:

- **Niezbalansowane klasy**:

Zbiór danych składał się z 37 przypadków metastazy (klasa "0") oraz 94 przypadków bez metastazy (klasa "1"), co powoduje znaczącą nierównowagę klas. Taka dysproporcja może prowadzić do sytuacji, w której model nauczy się rozpoznawać jedynie dominującą klasę, ignorując klasę mniejszościową. Przy podziale na zbiór treningowy i testowy zastosowano parametr `stratify`, aby zachować proporcję klas (30% do 70%) w każdym zbiorze. Dzięki temu uniknięto ryzyka, że zbiór treningowy będzie zawierał tylko przypadki jednej klasy, co zapewnia bardziej reprezentatywne i zrównoważone dane do treningu modelu. Ten krok jest kluczowy, aby model mógł nauczyć się rozpoznawać obie klasy, a nie tylko tę dominującą.

- **Selekcja cech**:

Początkowo wybrano 5 cech do selekcji, jednak okazało się to niewystarczające. Model nie osiąga zadowalających wyników, ponieważ nie uwzględniał wystarczającej ilości informacji potrzebnych do dokładnej klasyfikacji. Skuteczność modelu poprawiła się dopiero przy selekcji około 30 cech. Większa liczba cech dostarczyła modelowi więcej istotnych informacji, co przełożyło się na lepsze wyniki. Ten wynik podkreśla znaczenie odpowiedniego doboru liczby cech w procesie selekcji.

- **Proste klasyfikatory**:

Proste klasyfikatory, takie jak regresja logistyczna i drzewa decyzyjne, działały zadowalająco. Były one w stanie skutecznie klasyfikować dane, mimo ich prostoty.Porównano wyniki 5 iteracji dla każdego z klasyfikatorów. Średnie wartości i odchylenia standardowe z tych iteracji przedstawiono na wykresie słupkowym błędów. Taki sposób prezentacji pozwala na łatwe porównanie stabilności i efektywności poszczególnych klasyfikatorów, co jest pomocne w wyborze najbardziej odpowiedniego modelu.

- **Optymalizacja hiper parametrów**:

Optymalizacja hiper parametrów przyczyniła się do zwiększenia skuteczności konwolucyjnej sieci neuronowej (CNN). Przeprowadzono wiele eksperymentów, aby znaleźć najlepsze ustawienia hiper parametrów, co pozwoliło na uzyskanie wyższej dokładności modelu. Ta optymalizacja jest kluczowa, aby w pełni wykorzystać potencjał sieci neuronowej.

- **Różne metody selekcji cech**:

Metoda rankingowa okazała się najlepszą metodą selekcji cech, zapewniając najlepsze wyniki w klasyfikacji. Metoda ta polega na ocenie cech na podstawie ich znaczenia i wyborze tych o najwyższym rankingu. Były najmniej skuteczne. W tej metodzie cechy są wybierane jako część procesu treningu modelu, ale w tym przypadku nie dostarczyły one wystarczająco dobrych wyników, co sugeruje, że nie były w stanie skutecznie wyodrębnić najważniejszych cech z danych. Dodatkowo stosując metodę rankingową (najlepszą) na klasyfikatorach prostych zauważono znaczną poprawę otrzymanych wyników.

- **Klasyfikator Bayesowski**:

Klasyfikator Bayesowski wykazywał znaczne odchylenia pomiędzy poszczególnymi iteracjami, co sugeruje niestabilność modelu. Ta niestabilność może wynikać z wrażliwości modelu na zmiany w danych treningowych, co sprawia, że wyniki są mniej przewidywalne.

- **Klasyfikacja SVM z selekcją cech (embedded)**:

SVM w połączeniu z metodą embedded sklasyfikowało większość przypadków do klasy negatywnej z powodu biasu zbioru. Skutkowało to wysoką skutecznością ponad 70%, ale kosztem błędnego klasyfikowania przypadków klasy pozytywnej. Ze względu na przewagę przypadków klasy negatywnej, model miał trudności z nauką klasy pozytywnej, co wpływało na jego ogólną skuteczność. To pokazuje, jak ważne jest radzenie sobie z biasem w danych, aby model mógł skutecznie rozpoznawać obie klasy.

- **Metryki oceny modelu**:

F1-score okazała się lepszą metryką do oceny skuteczności CNN w porównaniu do innych metryk, biorąc pod uwagę niezbalansowane klasy. F1-score jest szczególnie użyteczny, ponieważ łączy zarówno precyzję, jak i recall, co daje lepszy obraz ogólnej wydajności modelu w przypadku nierównomiernie rozłożonych klas. Dzięki temu model jest oceniany bardziej sprawiedliwie, biorąc pod uwagę jego zdolność do rozpoznawania zarówno klasy dominującej, jak i mniejszościowej.

Podsumowując na podstawie projektu można stwierdzić, iż:

- Zastosowanie parametru `stratify` jest kluczowe dla prawidłowego podziału danych przy nierównowadze klas, co zapewnia bardziej reprezentatywne zbiory treningowe i testowe.
- Skuteczność modeli klasyfikacyjnych może być znacznie zwiększona poprzez odpowiednią selekcję cech oraz optymalizację hiper parametrów. Odpowiednia liczba cech i optymalne ustawienia hiper parametrów są kluczowe dla uzyskania wysokiej dokładności modeli.
- Proste klasyfikatory mogą działać dobrze, ale ich skuteczność można poprawić poprzez iteracyjne podejście i analizę wyników, co pozwala na lepsze zrozumienie stabilności i efektywności modeli.
- F1-score jest preferowaną metryką do oceny modeli w przypadku nierównowagi klas, ponieważ lepiej uwzględnia zarówno precyzję, jak i recall, co jest szczególnie ważne przy analizie danych z niezbalansowanymi klasami.

6. Źródła

- 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