Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: Informatyka techniczna (ITE)
Specjalność: Systemy informatyki w medycynie (IMT)

PRACA DYPLOMOWA MAGISTERSKA

Zastosowanie systemów wieloklasyfikatorowych do diagnostyki chorób wątroby

Application of multiclassifier systems for the diagnosis of liver diseases

Michał Honc

Opiekun pracy
Prof. dr hab. inż. Marek Kurzyński

Słowa kluczowe: Uczenie maszynowe, systemy wieloklasyfikatorowe

WROCŁAW 2023

Spis treści

1. Cel i zakres pracy 3
2. Opis problemu diagnostycznego 4
2.1. Kontekst medyczny 4
2.2. Opis zbioru danych 5
3. Opis algorytmów 8
3.1. System wieloklasyfikatorowy 8
3.1. Klasyfikatory bazowe 10
3.2. Algorytmy kombinowania wyników zespołu klasyfikatorów 16
3.3. Algorytm selekcji cech 18
4. Projekt systemu 19
4.1. Plan badań 19
4.2. Projekt środowiska eksperymentalnego 22
5. Wyniki badań oraz dyskusja 24
6. Podsumowanie 30
Literatura 31

1. Cel i zakres pracy

Celem pracy magisterskiej jest stworzenie systemów wieloklasyfikatorowych (MCS - *Multiple classifier system),* oraz przeprowadzenie badań eksperymentalnych porównujących ich działanie z klasyfikatorami bazowymi, wykorzystując jako dane medyczny problem diagnostyczny, jakim jest wykrywanie chorób wątroby. System wieloklasyfikatorowy składa się z klasyfikatorów dokładnych, czyli takich dla których częstość błędu klasyfikacji dla nowych obiektów jest mniejsza niż odgadywania losowego. Odpowiedzią takiego systemu jest odpowiedź wyselekcjonowanego klasyfikatora bazowego lub kombinowane odpowiedzi wszystkich / grupy wyselekcjonowanych klasyfikatorów bazowych. Taki system składa się z generowania puli klasyfikatorów bazowych, selekcji klasyfikatora/klasyfikatorów oraz integracji wyników działania wyselekcjonowanych klasyfikatorów. Praca będzie się skupiała głównie na systemach wieloklasyfikatorowych, wykorzystujących dynamiczną selekcję klasyfikatora (DCS - *Dynamic Classifier Selection*) oraz zespołu klasyfikatorów (DES – *Dynamic Ensemble Selection*). Ta technika polega na wyborze i stosowaniu najbardziej odpowiedniego klasyfikatora (lub zestawu klasyfikatorów) dla danego obiektu (próbki) na podstawie jego specyficznych cech.

2. Opis problemu diagnostycznego

2.1. Kontekst medyczny

Choroby wątroby dotykają wiele osób. Kojarzymy je przede wszystkim z ich najczęstszym objawem, jakim jest po prostu ból wątroby. Wątrobowy ból brzucha jest dość charakterystyczny - to ból brzucha pojawiający się z prawej strony pod żebrami. Wątroba sama w sobie nie sprawia dolegliwości bólowych (nie ma unerwienia czuciowego). Ból nie jest również jedynym objawem choroby wątroby. Często zanim wątroba zacznie boleć, pojawiają się przykładowo objawy skórne - świąd czy zażółcenie skóry.[1]

Stwierdzenie, czy dany pacjent ma chorą wątrobę możemy sprowadzić do problemu klasyfikacji. Klasyfikacja polega na przypisaniu rozpoznawanemu obiektowi odpowiedniej klasy na podstawie znajomości wartości wybranych cech. Algorytm podejmowania decyzji o klasie nazywamy algorytmem rozpoznawania lub algorytmem klasyfikacji lub też regułą decyzyjną (*pattern recognition algorithm, classification algorithm, decision rule*). Z kolei realizacja algorytmu (np. program komputerowy) nosi nazwę klasyfikatora lub urządzenia rozpoznającego. W przypadku diagnozowania chorób wątroby badanymi cechami mogą być dane dotyczące organizmu pacjenta takie jak np. wiek, płeć lub zawartość konkretnych substancji we krwi. Na podstawie tych cech możemy przypisać pacjenta do odpowiedniej klasy, która może oznaczać na przykład konkretną chorobę lub sam fakt zachorowania.

2.2. Opis zbioru danych

W celu utworzenia modelu, który będzie w stanie przeprowadzić poprawną klasyfikacje stanu wątroby pacjenta przeanalizowano zbiór danych *Indian Liver Patient Dataset* udostępniony na UCI Machine Learning Repository. Zawiera on dane, na podstawie których można stwierdzić czy dany pacjent ma chorą wątrobę. Próbki należące do tego zbioru zostały pobrane zarówno od osób zdrowych jak i chorych.[2]

Dla badanego problemu klasyfikacji zostały zdefiniowane dwie klasy. Poniżej przedstawiono ich rozkład w rozpatrywanym zbiorze danych:

- Pacjent ze zdrową wątrobą - 167 próbek (27,46%)
- Pacjent z chorą wątrobą - 441 próbek (72,54%)

Badane cechy - Każda próbka z badanego zbioru danych została opisana dziesięcioma atrybutami istotnymi z punktu widzenia diagnozowania chorób wątroby. Poniżej przedstawiono listę tych atrybutów wraz z ich opisem.

a) Cechy liczbowe dyskretne

Age – wiek pacjenta – wiek pacjenta jest zawsze istotnym czynnikiem mającym wpływ na stan jego zdrowia. Wraz z wiekiem zdolność organizmu do regeneracji oraz radzenia sobie ze szkodliwymi czynnikami obniża się, co może mieć wpływ na podatność na różne choroby, w tym choroby wątroby.

b) Cechy kategorialne

Gender – płeć pacjenta – również może mieć znaczenie przy diagnozowaniu chorób wątroby. Znaczące są tutaj nie tylko różnice w budowie ciała, ale także różnice w nawykach żywieniowych, na przykład tendencje do spożywania większych ilości alkoholu wśród mężczyzn.[3]

c) Cechy liczbowe ciągłe

tot_bilirubin – bilirubina całkowita – bilirubina to barwnik żółciowy, który pochodzi z rozpadu krwinek czerwonych. Jej stężenie we krwi pozwala ocenić funkcjonowanie wątroby. Wzrost tego stężenia może być przyczyną chorób wątroby. Bilirubinę całkowitą tworzą bilirubiny pośrednie i bezpośrednie.[4]

direct_bilirubin – bilirubina bezpośrednia – postać bilirubiny sprzężona z kwasem glukuronowym. Do wzrostu stężenia bilirubiny bezpośredniej we krwi dochodzi na skutek cholestazy wewnątrzwątrobowej lub zewnątrzwątrobowej oraz w wyniku uszkodzenia komórek wątroby.[4]

alkphos – fosfataza alkaliczna – fosfataza alkaliczna jest enzymem występującym w organizmie człowieka w czterech odmianach produkowanych przez różne narządy - wątrobę, kości, nerki i łożysko podczas ciąży. Podwyższenie poziomu fosfatazy alkalicznej może wskazywać na cholestazę (zatrzymanie żółci w przewodach żółciowych).[5]

sgpt – aminotransferaza alaninowa – aminotransferaza alaninowa jest enzymem wewnątrzkomórkowym. Najwyższe stężenie tego enzymu występuje w wątrobie i w nerkach. Poziom aminotransferazy alaninowej pozwala stwierdzić chorobę lub uszkodzenie wątroby.[6]

sgot – aminotransferaza asparaginianowa – aminotransferaza asparaginianowa jest enzymem wewnątrzkomórkowym, spotykanym w wątrobie, mięśniu sercowym, w mięśniach szkieletowych, nerkach ikrwinkach czerwonych. Określenie aktywności tego enzymu we krwi pozwala na wczesne wykrycie chorób wątroby.[6]

tot_proteins – białko całkowite – na białko całkowite składają się wszystkie białka krążące w organizmie człowieka. Wyróżniamy jego dwie główne postaci: globuliny i albuminy. Zbyt niski poziom białka całkowitego we krwi może być spowodowany marskością wątroby.[7]

albumin – albuminy – albuminy stanowią podstawową część białek krwi. Odpowiadają za utrzymanie tzw. ciśnienia onkotycznego w naczyniach krwionośnych. Produkowane są przez wątrobę, więc ich ilość we krwi odzwierciedla czynność tego narządu.[8]

ag_ratio – stosunek albuminy do globuliny – u zdrowej osoby stosunek albuminy do globuliny powinien wynosić nieco powyżej 1. Niższy współczynnik może wskazywać na choroby wątroby lub nerek.[9]

3. Opis algorytmów

3.1. System wieloklasyfikatorowy

System wieloklasyfikatorowy (MCS - *Multiple classifier system)* – składa się z klasyfikatorów dokładnych, czyli takich dla których częstość błędu klasyfikacji dla nowych obiektów jest mniejsza niż odgadywania losowego. Odpowiedzią takiego systemu jest odpowiedź wyselekcjonowanego klasyfikatora bazowego lub kombinowane odpowiedzi wszystkich / grupy wyselekcjonowanych klasyfikatorów bazowych. Taki system składa się z generowania puli klasyfikatorów bazowych, selekcji klasyfikatora/klasyfikatorów oraz integracji wyników działania wyselekcjonowanych klasyfikatorów.

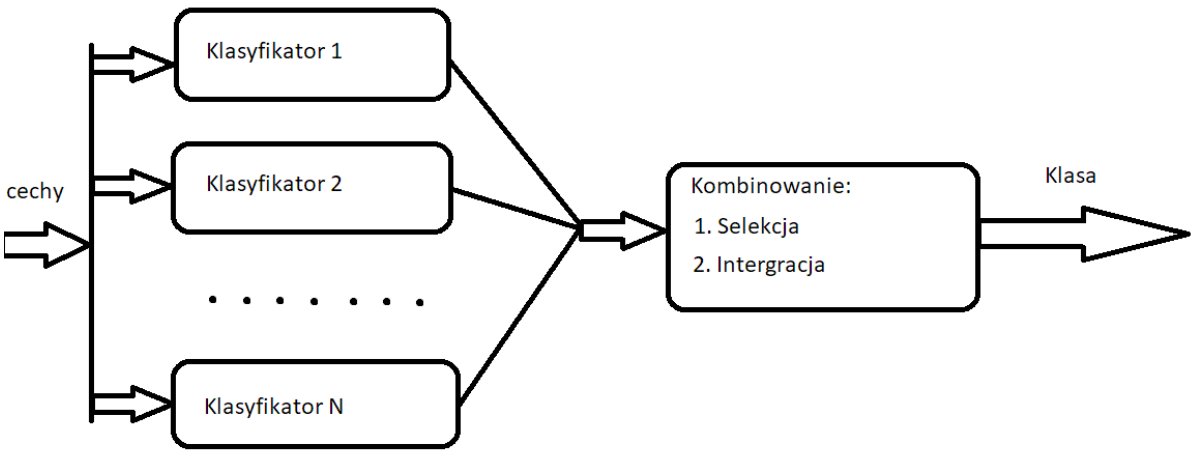
Systemy MCS wykorzystują podział danych na trzy zbiory: zbiór uczący, walidacyjny oraz testowy. Zbiór uczący służy do nauki klasyfikatorów bazowych, na podstawie tych danych uczą się one zwracać odpowiednie predykcje. Zbiór walidacyjny służy do oceny wydajności klasyfikatorów w różnych obszarach przestrzeni cech, co pomaga w wyborze najbardziej odpowiedniego klasyfikatora dla danego wektora wejściowego. Po trenowaniu i dostrojeniu modeli na zbiorach uczącym i walidacyjnym, zbiór testowy jest używany do oceny ostatecznej wydajności modelu. Zbiór testowy daje szacunek "realnego" działania modelu, ponieważ dane w tym zestawie nie były używane w żadnej części procesu uczenia czy dostrojenia modelu.

Generowanie puli klasyfikatorów bazowych można podzielić na podejście homogeniczne oraz heterogeniczne. Pula homogeniczna składa się z tych samych klasyfikatorów bazowych (np. drzewo decyzyjne), lecz dla zapewnienia różnorodności wykorzystuje się różne przestrzenie cech, różne zbiory uczące lub różne parametry klasyfikatorów. Przykładową techniką

zapewniającą różnorodność puli homogenicznej jest technika bootstrapingu, która polega na losowaniu wielu próbek ze zwracaniem z pierwotnego zbioru uczącego, a następnie używanie tych próbek do szkolenia klasyfikatorów z puli. Pula heterogeniczna natomiast składa się z klasyfikatorów różnego rodzaju (np. drzewo decyzyjne oraz kNN).

Selekcja (klasyfikatora/klasyfikatorów) dzieli się na selekcję statyczną – w której wybieramy ten sam klasyfikator / te same klasyfikatory dla wszystkich obiektów testujących (wynik selekcji nie jest zależny od cech obiektu), oraz dzieli się na dynamiczna selekcję klasyfikatora (DCS) / zespołu klasyfikatorów (DES) – wybieramy klasyfikator/klasyfikatory dla każdego obiektu testowego (wynik selekcji zależy od cech klasyfikowanego obiektu).

Integracja wyników jest potrzebna, gdy wybieramy zespół klasyfikatorów. Przykładową metodą integracji jest głosowanie większościowe, gdzie każdy model wskazuje na daną etykietę, a większość głosów decyduje o podjętej decyzji.[10][11]



Rys. 1. Schemat systemu wieloklasyfikatorowego.

3.2. Klasyfikatory bazowe

Fundamentem systemów wieloklasyfikatorowych są klasyfikatory bazowe. Poniżej znajduje się opis poszczególnych metod.

a.) *Gaussian Naive Bayes* – Naiwny klasyfikator bayesowski

Wykorzystuje teorię prawdopodobieństwa Bayesa. Ponieważ zakłada niezależność cech, co w praktyce rzadko jest prawdziwe, jest nazywany „naiwnym". Ogólnie rzecz biorąc, zasada działania jest dość prosta. Rozważmy problem klasyfikacji binarnej, w którym możemy mieć dwa możliwe wyniki: 1 lub 0. Chociaż mamy zestaw cech $\boldsymbol{X} = (\boldsymbol{x1}, \boldsymbol{x2}, \ldots, \boldsymbol{xn})$, chcemy obliczyć prawdopodobieństwo obu wyników dla danego zestawu cech. W tym celu korzystamy z twierdzenia Bayesa dotyczącego prawdopodobieństwa warunkowego. Dla zestawu cech X i klasy C (1 lub 0) oblicza się je następująco:

$$\boldsymbol{P(B|X)} = \frac{\boldsymbol{P(X|B)P(B)}}{\boldsymbol{P(X)}}$$

Klasyfikator naiwny Bayesa zakłada, że każda cecha xi w zestawie cech X jest niezależna od pozostałych. To pozwala nam uprościć obliczenie P(X|B) do sumy prawdopodobieństw indywidualnych cech:

$$\boldsymbol{P(X|B)} = \boldsymbol{P(x1|B)} \times \boldsymbol{P(x2|B)} \times \ldots \times \boldsymbol{P(xn|B)}$$

Gdzie:

$\boldsymbol{P(B|X)}$ – prawdopodobieństwo, że przypadek należy do klasy B, pod warunkiem że ma zestaw cech X,

$\boldsymbol{P(X|B)}$ – prawdopodobieństwo wystąpienia zestawu cech X, pod warunkiem że przypadek należy do klasy B,

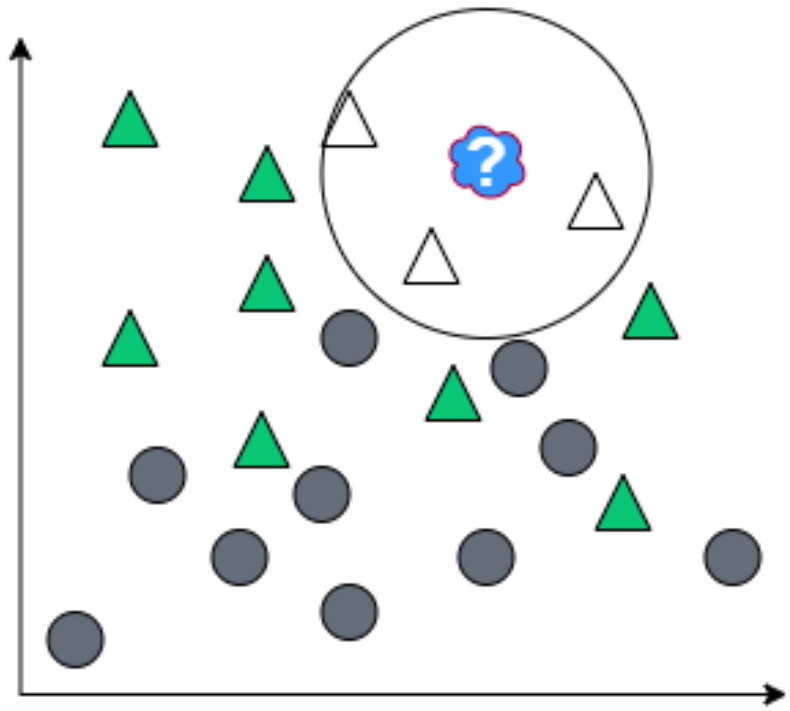
$\boldsymbol{P(B)}$ – prawdopodobieństwo wystąpienia klasy B (tzw. prawdopodobieństwo a priori),

$\boldsymbol{P(X)}$ – prawdopodobieństwo wystąpienia zestawu cech X.

W praktyce klasyfikator naiwnego Bayesa przypisuje przypadek do klasy, w której $\boldsymbol{P(X|B)\times P(B)}$ jest największe.[12]

b.) *K-Nearest Neighbors* (kNN) – Metoda K Najbliższych Sąsiadów

Algorytm działa w taki sposób, że bierze najbliższą okolicę danego punktu który chcemy sklasyfikować, sprawdza jakiej klasy obiektów jest w tej okolicy najwięcej, i przyporządkowuje obiekt do klasy większościowej. Najprościej działanie metody można pokazać na rysunku, gdzie dla liczby sąsiadów k=3 klasą większościową jest „trójkąt". Taką klasę uzyska więc klasyfikowany punkt w danym przykładzie:

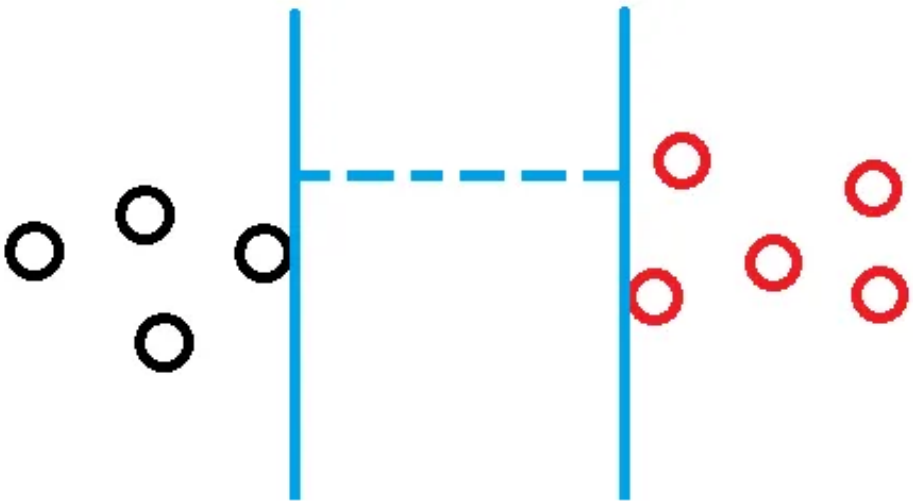


Rys. 2. Działanie metody kNN.

Niemniej jednak należy podkreślić, że do zastosowania algorytmu najbliższych sąsiadów w danym zbiorze konieczne jest posiadanie zdolności do obliczenia odległości między elementami tego zbioru. Może być to na przykład metryka euklidesowa.[13]

c.) SVC – C-Support Vector Classification

Algorytm, który klasyfikuje punkty danych w dwóch lub więcej kategoriach. Po sklasyfikowaniu wszystkich punktów algorytm zaczyna śledzić linie na krawędzi separacji między klasami, mając na celu maksymalizację odległości między nimi. Miejsce, w którym znajduje się największa odległość, to miejsce, w którym znajduje się najlepsza linia separacji. Poniżej znajduje się przykładowy rysunek pokazujący separację liniową:



Rys. 3. Separacja liniowa.

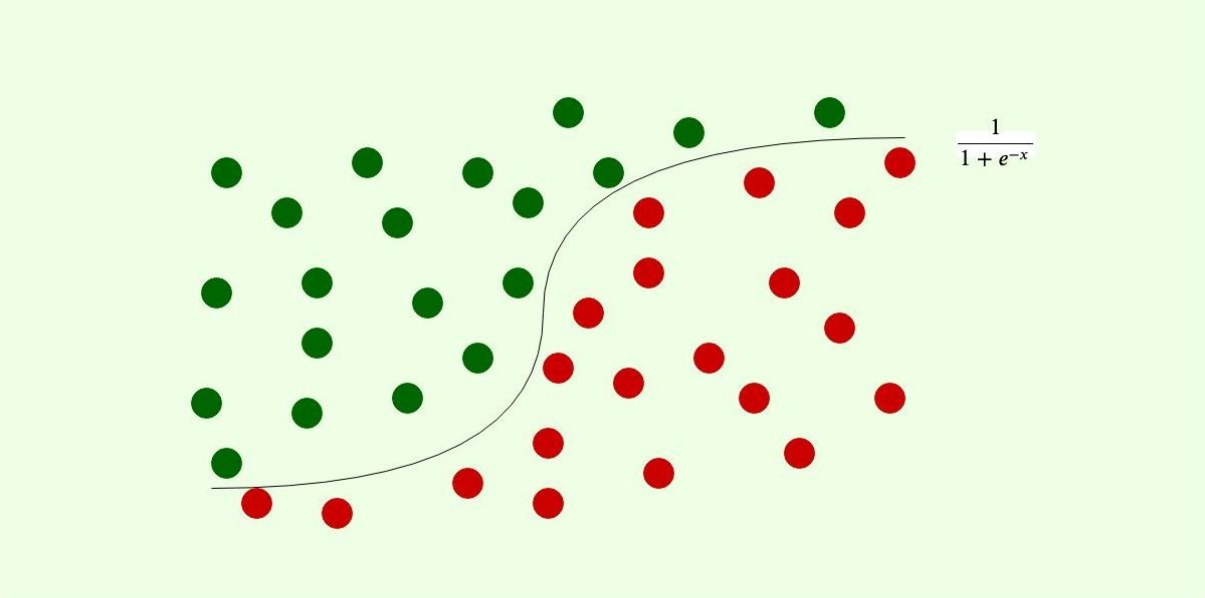
Jednak separacja może być inna niż liniowa, np. Rbf - *Gaussian Radial Basis Function,* która zamiast prostych linii tworzy krzywe Gaussa.[12]

d.) *Logistic regression* – Regresja logistyczna

Zakłada, że można sklasyfikować dane za pomocą linii lub płaszczyzny n-wymiarowej. Innymi słowy, klasyfikacja następuje poprzez obliczenie wartości wielomianu pierwszego stopnia:

$$\boldsymbol{y = \omega 1 \times x1 + \omega 2 \times x2 + \ldots + \omega n \times xn}$$

gdzie x jest daną wejściową, ω jest wagą przypisaną do tego parametru, a n to liczba wszystkich wejściowych parametrów. Kolejnym krokiem jest przepuszczenie tego wyniku y przez funkcję logistyczną, taką jak na przykład sigmoid, aby uzyskać wartość w przedziale od 0 do 1. Na podstawie prostej reguły możemy zaklasyfikować daną wejściową do grupy A lub grupy B po uzyskaniu tej wartości. W przypadku, gdy y jest większe niż 0,5, jest to klasa A, a w przeciwnym razie klasa B. Proces uczenia maszynowego polega na doborze wag ω. Polega to na wykorzystaniu algorytmu gradientu prostego do wyliczenia funkcji błędu i minimalizacji błędu.[15]

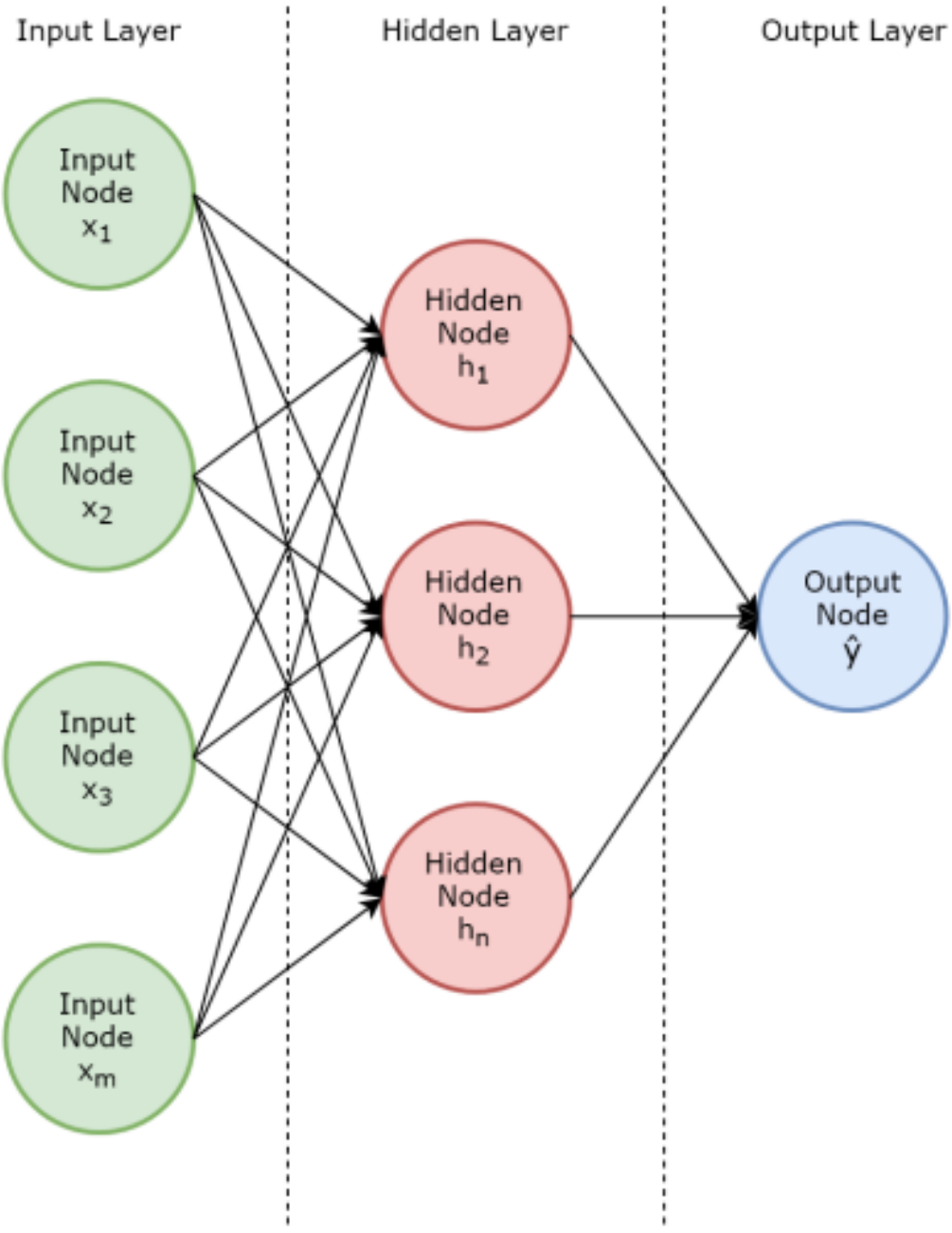


Rys. 4. Funkcja logistyczna – sigmoid

e.) *Multi-layer Perceptron* – Perceptron wielowarstwowy

Najbardziej popularny rodzaj sztucznych sieci neuronowych. Zawiera co najmniej trzy warstwy: warstwę wejściową, warstwę ukrytą i warstwę wyjściową. Wiele warstw wejściowych jest możliwych. Neurony McCullocha-Pittsa są najczęściej obecne w warstwach ukrytych. Do połączeń między neuronami przypisane są wagi – liczby rzeczywiste.

Schemat wielowarstwowego perceptronu przedstawiono na rysunku poniżej. Uczenie tego algorytmu opiera się na zmienianiu wag połączeń między neuronami, aż sieć osiągnie wystarczającą dokładność. W tym procesie najczęściej stosuje się metodę wstecznej propagacji błędów, która przenosi błąd z wyjścia sieci do warstwy wejściowej.[16]

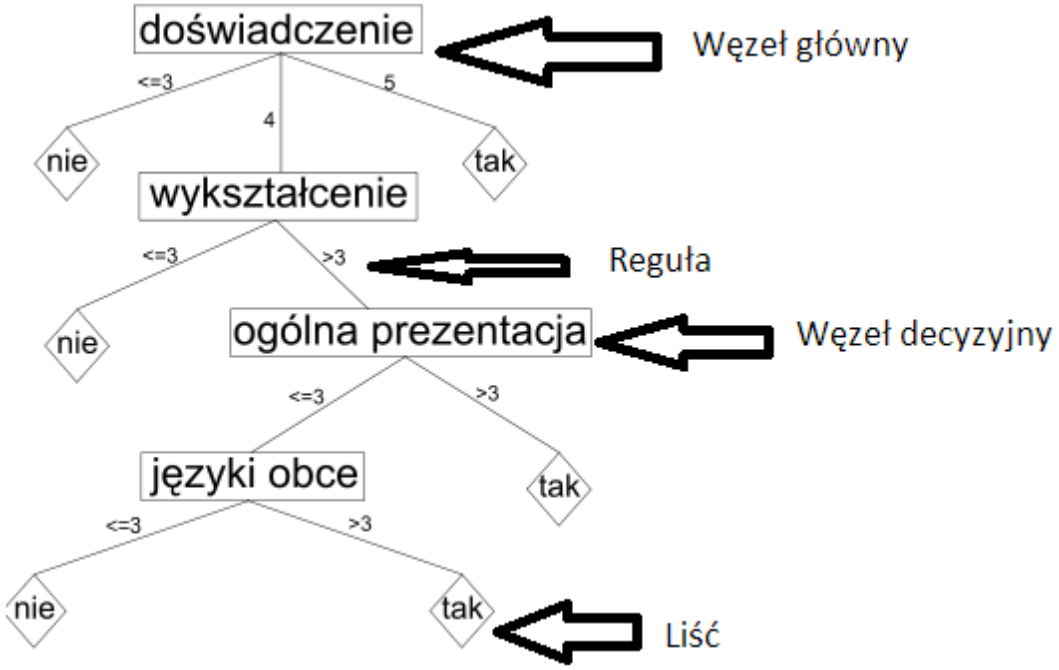


Rys. 5. Wielowarstwowy perceptron z jedną warstwą ukrytą.

f.) *Decision tree* – drzewo decyzyjne

Struktura drzewiasta podobna do wykresu przepływu, w której wewnętrzny węzeł reprezentuje cechę (lub atrybut), gałąź reprezentuje regułę decyzyjną, a każdy węzeł liścia reprezentuje wynik. Najwyższy węzeł w drzewie decyzyjnym jest znany jako węzeł główny. Dzieli drzewo w sposób rekurencyjny, zwany partycjonowaniem rekurencyjnym., na podstawie atrybutów. Działanie tej metody można przedstawić w następujących poniżej krokach.[17]

1. Wybierz najlepszy atrybut, aby podzielić rekordy.
2. Uczyń ten atrybut węzłem decyzyjnym i podziel zbiór danych na mniejsze podzbiory.
3. Rozpocznij budowanie drzewa, powtarzając ten proces rekurencyjnie dla każdego węzła, aż jeden z warunków zostanie spełniony:
    - Węzeł zostaje liściem(wynikiem).
    - Nie ma więcej pozostałych atrybutów.
    - Nie ma więcej przypadków.



Rys. 6. Przykład uproszczonego drzewa decyzyjnego.

3.3. Algorytmy kombinowania wyników zespołu klasyfikatorów

a.) *Overall Local Accuracy (OLA)*

Jest to metoda DCS. Miarą kompetencji klasyfikatora bazowego w punkcie $x \in X$ jest frakcja poprawnie sklasyfikowanych obiektów walidacyjnych spośród k najbliższych obiektowi x. Metodę można opisać w poniższych krokach.[18]

1. Dla każdego punktu x w zbiorze testowym X znajdujemy k najbliższych sąsiadów tego punktu(w zbiorze walidacyjnym). Następnie każdy klasyfikator z puli dokonuje predykcji na obszarze lokalnym.
2. Dla każdego punktu x zostaje wybrany klasyfikator, który uzyskał najlepszy lokalny wynik. Jeśli więcej niż jeden klasyfikator uzyskał najlepszy wynik, do puli zostaje wybrany pierwszy.
3. Mając przyporządkowany klasyfikator w każdym punkcie x, dokonujemy predykcji na zbiorze testowym.

Poniżej znajduje się kod metody *OLA* zaimplementowany w języku Python. Funkcja przyjmuje jako parametry wejściowe pulę wytrenowanych klasyfikatorów na zbiorze treningowym, zbiór walidacyjny, zbiór testowy oraz liczbę najbliższych sąsiadów. Wynikiem jej działania jest klasyfikacja punktów zbioru testowego. Dalsze zaimplementowane metody zostaną jedynie opisane teoretycznie, ze względu na podobną konstrukcję (parametry wejściowe, wyjściowe, użycie metody najbliższych sąsiadów do wyznaczenia sąsiedztwa punktów).

Listing 1. Kod metody OLA.

```python
def overall_local_accuracy(classifiers, X_val, y_val, X_test, k=7):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_val, y_val)
    y_pred = []

    for x in X_test:
        # Find the k nearest neighbors of the instance in the training set
        neighbors = knn.kneighbors([x], return_distance=False)
        local_accuracies = []

        # Calculate the local accuracy for each classifier
        # on the k nearest neighbors
        for clf in classifiers:
            local_predictions = clf.predict(X_val[neighbors.ravel()])
            local_accuracies.append(accuracy_score(y_val[neighbors.ravel()], local_predictions))

        # Select the classifier with the highest local accuracy
        best_classifier_idx = np.argmax(local_accuracies)
        y_pred.append(classifiers[best_classifier_idx].predict([x])[0])

    return y_pred
```

b.) *k-Nearest Oracle-Eliminate (KNORA-E)*

Jedna z metod DES. Polega na wyszukiwaniu lokalnych wyroczni, które poprawnie klasyfikują wszystkie próbki należące do regionu kompetencji próbki testowej. Wybierane są wszystkie klasyfikatory o doskonałej wydajności w regionie kompetencji (czyli te które sklasyfikują poprawnie k najbliższych sąsiadów punktu x w zbiorze walidacyjnym). W przypadku, gdy żaden klasyfikator nie osiąga doskonałej dokładności, rozmiar regionu kompetencji jest zmniejszany (poprzez usunięcie najdalszego sąsiada), a wydajność klasyfikatorów jest ponownie oceniana. Wyniki wybranego zespołu klasyfikatorów są łączone przy użyciu schematu głosowania większościowego. Jeśli nie zostanie wybrany żaden klasyfikator bazowy, do klasyfikacji używana jest cała pula.[19]

c.) *Dynamic Ensemble Selection performance (DES-P)*

Metoda DES, która wybiera wszystkie klasyfikatory bazowe, które osiągają wydajność klasyfikacji w obszarze kompetencji (dla k najbliższych sąsiadów punktu) wyższą niż klasyfikator losowy (RC). Wydajność klasyfikatora losowego jest określona przez RC = 1/L, gdzie L jest liczbą klas w problemie (dla naszego problemu jest to więc ½). Jeśli nie zostanie wybrany żaden klasyfikator bazowy, do klasyfikacji używana jest cała pula, gdzie wynikiem zostaje etykieta która otrzymała największą ilość głosów (głosowanie większościowe).[20][21]

3.4. Algorytm selekcji cech

Do przeprowadzenia selekcji cech wybrano jednoczynnikową analizę wariancji (*One-Way ANOVA*) należącą do grupy *Univariate Filter Methods*. Jest ona najczęściej stosowana, gdy analizowane zmienne niezależne mają charakter liczbowy, a zmienna zależna kategorialny. Opiera się ona na analizie różnicy między wartościami średnimi cech dla poszczególnych klas oraz wariancji wewnątrz klas. Przyjmujemy, że jeżeli średnie wartości danej cechy dla zdefiniowanych klas są do siebie podobne, to jest ona mało znaczącą cechą, gdyż jej wartości nie dają informacji o klasie, do której należy dana próbka. Analogicznie, jeżeli między wartościami średnimi dla danych klas będą znaczące różnice, to cecha uznawana jest za przydatną z punktu widzenia klasyfikacji. Dla każdej cechy wyliczamy wartość F, która jest wyznacznikiem przydatności cechy. Wyższa wartość F oznacza lepsze właściwości dyskryminacyjne danego atrybutu. Poniżej przedstawiono sposób wyznaczania tej wartości.

$$\boldsymbol{F = \frac{(\sum_{i=1}^{K} n_i\,(\overline{X_i} - \overline{X})^2)/(K-1)}{(\sum_{i=1}^{K}(n_i - 1)\,\sigma^2)/(N-K)}}$$

gdzie K to liczba klas, N to liczba wszystkich próbek ni to liczebność i-tej klasy, σ 2 i to wariancja wewnątrz i-tej klasy.[22]

4. Projekt systemu

4.1. Plan badań

Poniżej znajduje się plan przeprowadzenia badań w punktach:

1. Wczytanie oraz przygotowanie danych.

W miejsce kategorii *male* i *female* wstawiono odpowiednio liczby 1 i 0. Przed przystąpieniem do obliczenia przydatności danych cech przy klasyfikacji sprawdzono zbiór danych pod kątem brakujących wartości. W czterech próbkach brakowało wartości cechy *alkphos*. W celu uzupełnienia brakujących danych obliczono średnią wartość tej cechy w badanym zbiorze danych, a następnie wstawiono ją w miejsce brakujących wartości.

2. Zbalansowanie zbioru danych.

Przed rozpoczęciem eksperymentów dane zostały zbalansowane za pomocą metody *SMOTE* (*Synthetic Minority Oversampling Technique*), która polega na wygenerowaniu dodatkowych instancji klasy mniejszościowej na podstawie instancji znajdujących się w zestawie danych.

3. Stworzenie rankingu cech.

Badania zostały przeprowadzone wykorzystując algorytm *ANOVA* do stworzenia rankingu cech. Eksperymenty zostały przeprowadzone zaczynając od wybrania pięciu najlepszych cech, kończąc na wszystkich cechach, czyli dziesięciu (przedział od 5 do 10). W tabeli 1 przedstawiono obliczone wartości F dla poszczególnych atrybutów. Posortowano je w kolejności od najwyższego wyniku do najniższego. Z poniższej tabeli wynika, że najbardziej przydatną podczas klasyfikacji cechą jest *direct_bilirubin*. Natomiast zdecydowanie najmniej przydatną cechą jest *sgpt*.

Tabela 1. Obliczone wartości F dla każdego z atrybutów.

Nazwa cechy	F
direct_bilirubin	92.895
tot_bilirubin	73.605
tot_proteins	58.760
albumin	41.314
alkphos	37.964
ag_ratio	35.466
sgot	33.309
age	13.505
sgpt	1.360
gender	0.177

4. Stworzenie puli klasyfikatorów.

Utworzenie puli klasyfikatorów homogenicznych oraz heterogenicznych. Pula homogeniczna składa się z pięćdziesięciu klasyfikatorów bazowych tego samego typu oraz o tych samych parametrach (np. 50 klasyfikatorów kNN, dla których parametr k = 5), jednak dla różnorodności wykorzystano technikę bootstrapingu, która polega na losowaniu wielu próbek ze zwracaniem z pierwotnego zbioru uczącego, a następnie używanie tych próbek do szkolenia klasyfikatorów z puli. Pula heterogeniczna natomiast składa się w konkretnie opisywanym eksperymencie z sześciu klasyfikatorów bazowych opisanych we wcześniejszych rozdziałach.

5. Podział danych.

Dane zostają podzielone na zbiór uczący (wykorzystywany do wyuczenia klasyfikatorów znajdujących się w puli), zbiór walidacyjny (z którego korzystają metody MCS), oraz zbiór testowy (do wyznaczania końcowej predykcji systemu). Zbiór został podzielony w proporcjach 60/20/20, gdzie zbiór uczący liczy 60%, natomiast zbiór walidacyjny i testowy liczą po 20%. Do podziału wykorzystano 5-krotną walidację krzyżową powtórzoną pięć razy (walidacja 5x5). Główną ideą walidacji krzyżowej jest podział zbioru na K równych części. Następnie kolejno każda z tych części używana jest jako zbiór testowy, a pozostałe jako zbiór uczący oraz walidacyjny. Proces ten wykonywany jest K razy i za każdym razem budujemy nowy model od nowa. W naszym przypadku proces powtarzamy również K razy, a uzyskane w ten sposób rezultaty (dokładność klasyfikacji) są agregowane w jeden wynik.

6. Otrzymanie wyników zespołu klasyfikatorów oraz klasyfikatorów bazowych dla zadanego zakresu cech.

Mając dane podzielone w taki sposób, otrzymujemy wyniki działania klasyfikatorów bazowych (które są uczone na zbiorze uczącym oraz walidacyjnym – pojedynczy klasyfikator nie potrzebuje zbioru walidacyjnego), oraz otrzymujemy wyniki systemów wieloklasyfikatorowych (zagregowane wyniki klasyfikatorów z puli). Wynikiem badań jest jakość klasyfikacji oraz jej odchylenie standardowe. Jakość klasyfikacji mówi nam o tym, jaki procent instancji w zbiorze testowym został przez nas zaklasyfikowany poprawnie.

Odchylenie standardowe to statystyczna miara rozproszenia danych wokół średniej. Im większe odchylenie standardowe, tym większa jest różnorodność wartości danych. Odchylenie standardowe można wyrazić za pomocą wzoru poniżej:

$$\boldsymbol{\sigma} = \sqrt{\frac{\sum(\boldsymbol{x} - \boldsymbol{\mu})^{\mathbf{2}}}{\boldsymbol{N}}}$$

Gdzie x to wartość poszczególnej próbki, μ to wartość średnia całego zbioru danych, a N to liczba punktów w zbiorze danych.[23]

4.2. Projekt środowiska eksperymentalnego

W celu przeprowadzenia eksperymentów przygotowane zostało odpowiednie środowisko programistyczne w języku Python. Stworzono je przy pomocy bibliotek *scikit-learn, imblearn, pandas* oraz *numpy*. Na samym początku wczytano dane oraz przygotowano je do dalszego wykorzystania, wstawiając w miejsce kategorii *male* i *female* odpowiednio liczby 1 i 0, a także wstawiając w miejsca brakujących cech średnią wartość danej cechy w badanym zbiorze danych. Następnie zbalansowano zbiór danych przy pomocy klasy **SMOTENC**, która umożliwia zastosowanie metody *SMOTE*. Mając tak przygotowane dane, skorzystano z metody **RepeatedKFold** (do stworzenia walidacji krzyżowej 5x5), oraz **train_test_split**, aby podzielić dane na zbiór treningowy, walidacyjny oraz testowy. Przy implementacji algorytmu selekcji cech wykorzystano metodę **SelectKBest**(score_func=f_classif, k=feature_number), która implementuje algorytm *ANOVA*. Do utworzenia homogenicznych puli klasyfikatorów skorzystano z metody **resample**, która umożliwia użycie techniki bootstrapingu. Do implementacji klasyfikatorów bazowych(dokładnych) użyto następujących metod: **GaussianNB, KNeighborsClassifier, SVC, LogisticRegression, MLPClassifier** oraz **DecisionTreeClassifier**. Główne algorytmy tematu pracy, czyli systemy DCS oraz DES zostały zaimplementowane przez autora pracy(kod metody *OLA* znajduje się w rozdziale 4.2). Ostateczne porównanie dokładności systemów MCS oraz klasyfikatorów bazowych, zostało wykonane przy pomocy funkcji

accuracy_score, która zwraca ułamek poprawnie sklasyfikowanych próbek. Do wyliczenia odchyleń standardowych skorzystano z funkcji **std** z biblioteki numpy. [24][25][26][27]

5. Wyniki badań oraz dyskusja

W tym rozdziale przedstawione zostały wyniki systemów wieloklasyfikatorowych *OLA*, *KNORAE* oraz *DESP*, składających się z puli homogenicznych (pięćdziesięciu klasyfikatorów tego samego rodzaju o tych samych parametrach wyuczonych korzystając z bootstrapingu), oraz puli heterogenicznych (składa się ona ze wszystkich sześciu klasyfikatorów opisanych wcześniej). Wyniki systemów MCS zostały porównane z działaniem pojedynczych klasyfikatorów bazowych. Eksperyment przeprowadzono dla liczby najbardziej znaczących cech w przedziale <5, 10>. Wszystkie klasyfikatory posiadają parametry domyślne, kluczowe z nich zostaną podane, natomiast dla metod *OLA*, *KNORAE* oraz *DESP*, które przyjmują parametr ilości najbliższych sąsiadów, ustawiono ten parametr na k = 7.

a.) **GaussianNB**

Tabela 2. Jakość klasyfikacji dla klasyfikatora bazowego GaussianNB dla cech <5, 10>.

	5	6	7	8	9	10
GaussianNB	0,6899	0,6893	0,6923	0,6900	0,6899	0,6942
OLA	0,6995	0,7007	0,6965	0,7008	0,7025	0,7014
KNORAE	0,6983	0,6959	0,6941	0,6978	0,6977	0,6996
DESP	0,6971	0,6935	0,6929	0,6960	0,6965	0,6954

Tabela 3. Odchylenia standardowe dla klasyfikatora bazowego GaussianNB dla cech <5, 10>.

	5	6	7	8	9	10
GaussianNB	0,0193	0,0285	0,0209	0,0352	0,0491	0,0356
OLA	0,0297	0,0278	0,0183	0,0353	0,0482	0,0348
KNORAE	0,0238	0,0303	0,0201	0,0335	0,0454	0,0399
DESP	0,0229	0,0249	0,0211	0,0343	0,0493	0,0418

b.) **KNeighborsClassifier** – dla liczby najbliższych sąsiadów k = 5

Tabela 4. Jakość klasyfikacji dla klasyfikatora bazowego kNN dla cech <5, 10>.

	5	6	7	8	9	10
kNN	0,7092	0,6815	0,7066	0,6911	0,7020	0,7104
OLA	0,7140	0,7249	0,7199	0,6935	0,7188	0,7176
KNORAE	0,7199	0,7031	0,7247	0,7237	0,7128	0,7128
DESP	0,7271	0,6996	0,7078	0,7212	0,7044	0,7296

Tabela 5. Odchylenia standardowe dla klasyfikatora bazowego kNN dla cech <5, 10>.

	5	6	7	8	9	10
kNN	0,0107	0,0329	0,0301	0,0218	0,0199	0,0132
OLA	0,0343	0,0187	0,0599	0,0153	0,0331	0,0176
KNORAE	0,0341	0,0426	0,0378	0,0186	0,0291	0,0147
DESP	0,0244	0,0191	0,0229	0,0145	0,0264	0,0278

c.) SVC

Tabela 6. Jakość klasyfikacji dla klasyfikatora bazowego SVC dla cech <5, 10>.

	5	6	7	8	9	10
SVC	0,6689	0,6682	0,6658	0,6682	0,6647	0,6676
OLA	0,6725	0,6730	0,6736	0,6766	0,6791	0,6779
KNORAE	0,6755	0,6809	0,6923	0,6839	0,6954	0,6851
DESP	0,6701	0,6766	0,6736	0,6700	0,6785	0,6760

Tabela 7. Odchylenia standardowe dla klasyfikatora bazowego SVC dla cech <5, 10>.

	5	6	7	8	9	10
SVC	0,0383	0,0332	0,0438	0,0439	0,0498	0,0313
OLA	0,0437	0,0328	0,0456	0,0498	0,0520	0,0419
KNORAE	0,0474	0,0236	0,0490	0,0451	0,0529	0,0372
DESP	0,0409	0,0320	0,0504	0,0479	0,0532	0,0324

d.) **LogisticRegression**

Tabela 8. Jakość klasyfikacji dla klasyfikatora bazowego LogisticRegression dla cech <5, 10>.

	5	6	7	8	9	10
LogisticRegression	0,7128	0,7121	0,7134	0,7092	0,7133	0,7091
OLA	0,7170	0,7236	0,7248	0,7206	0,7200	0,7115
KNORAE	0,7296	0,7283	0,7217	0,7308	0,7266	0,7265
DESP	0,7182	0,7212	0,7139	0,7134	0,7200	0,7223

Tabela 9. Odchylenia standardowe dla klasyfikatora LogisticRegression dla cech <5, 10>.

	5	6	7	8	9	10
LogisticRegression	0,0367	0,0315	0,0343	0,0263	0,0262	0,0276
OLA	0,0396	0,0321	0,0268	0,0245	0,0262	0,0332
KNORAE	0,0367	0,0274	0,0300	0,0188	0,0246	0,0277
DESP	0,0443	0,0297	0,0298	0,0241	0,0215	0,0227

e.) MLPClassifier – ilość neuronów warstwy ukrytej = 100

Tabela 10. Jakość klasyfikacji dla klasyfikatora bazowego MLPClassifier dla cech <5, 10>.

	5	6	7	8	9	10
MLP	0,6851	0,6851	0,6791	0,6869	0,6845	0,6840
OLA	0,6995	0,6863	0,6700	0,6917	0,6977	0,6930
KNORAE	0,7127	0,6995	0,6971	0,7092	0,7170	0,6960
DESP	0,7037	0,7001	0,7013	0,7013	0,7067	0,6954

Tabela 11. Odchylenia standardowe dla klasyfikatora MLPClassifier dla cech <5, 10>.

	5	6	7	8	9	10
MLP	0,0389	0,0205	0,0336	0,0254	0,0224	0,0520
OLA	0,0338	0,0260	0,0315	0,0347	0,0349	0,0364
KNORAE	0,0369	0,0202	0,0235	0,0465	0,0377	0,0315
DESP	0,0362	0,0144	0,0288	0,0299	0,0278	0,0408

f.) **DecisionTreeClassifier**

Tabela 12. Jakość klasyfikacji dla klasyfikatora bazowego DecisionTree dla cech <5, 10>.

	5	6	7	8	9	10
DecisionTree	0,7097	0,7188	0,7134	0,7217	0,7284	0,7158
OLA	0,7152	0,7212	0,7284	0,7254	0,7386	0,7194
KNORAE	0,7380	0,7416	0,7302	0,7476	0,7265	0,7410
DESP	0,7560	0,7674	0,7741	0,7819	0,7656	0,7747

Tabela 13. Odchylenia standardowe dla klasyfikatora bazowego DecisionTree dla cech <5, 10>.

	5	6	7	8	9	10
DecisionTree	0,0372	0,0239	0,0361	0,0363	0,0386	0,0450
OLA	0,0302	0,0387	0,0268	0,0320	0,0275	0,0282
KNORAE	0,0335	0,0321	0,0222	0,0212	0,0333	0,0370
DESP	0,0302	0,0248	0,0247	0,0323	0,0162	0,0425

g.) Pula heterogeniczna

Tabela 14. Jakość klasyfikacji dla puli heterogenicznej dla cech <5, 10>.

	5	6	7	8	9	10
GaussianNB	0,6941	0,6940	0,6857	0,6869	0,6923	0,6845
kNN	0,6947	0,6827	0,7265	0,7182	0,6930	0,7122
SVC	0,6526	0,6454	0,6562	0,6677	0,6737	0,6725
Logistic Regression	0,6983	0,7145	0,7049	0,7079	0,7091	0,7200
MLP	0,6917	0,7097	0,6989	0,6881	0,6863	0,6905
DecisionTree	0,7193	0,7152	0,7157	0,7236	0,7019	0,7170
OLA	0,7115	0,7212	0,7308	0,7283	0,7236	0,7289
Knorae	0,7265	0,7332	0,7380	0,7440	0,7284	0,7374
DESP	0,7223	0,7199	0,7307	0,7530	0,7302	0,7428

Tabela 15. Odchylenia standardowe dla puli heterogenicznej dla cech <5, 10>.

	5	6	7	8	9	10
GaussianNB	0,0330	0,0275	0,0322	0,0371	0,0300	0,0304
kNN	0,0288	0,0309	0,0286	0,0235	0,0305	0,0259
SVC	0,0303	0,0362	0,0295	0,0328	0,0351	0,0236
LogisticRegression	0,0277	0,0357	0,0342	0,0307	0,0348	0,0280
MLP	0,0303	0,0329	0,0441	0,0460	0,0382	0,0389
DecisionTree	0,0275	0,0275	0,0301	0,0368	0,0319	0,0321
OLA	0,0299	0,0415	0,0336	0,0359	0,0293	0,0280
Knorae	0,0317	0,0427	0,0381	0,0385	0,0318	0,0299
DESP	0,0350	0,0387	0,0290	0,0330	0,0312	0,0279

Spoglądając na tabele wyników można jasno stwierdzić, że metody *OLA*, *KNORAE* oraz *DESP*, korzystające z puli klasyfikatorów homogenicznych dają lepsze wyniki od pojedynczych klasyfikatorów bazowych (np. pula agregująca wyniki 50 klasyfikatorów **kNN** daje lepszy wynik od pojedynczego klasyfikatora **kNN**). Biorąc pod uwagę pulę heterogeniczą, systemy MCS również uzyskały lepszy wynik od każdego klasyfikatora bazowego z którego składała się ta pula (tabela 14.).

Analizując odchylenia standardowe, ciężko tutaj wyciągnąć jednoznaczne wnioski. Metody MCS nie zawsze generują większy bądź mniejszy rozrzut wyników wokół średniej.

Przechodząc do oceny wyników pod kątem liczby cech, można zauważyć, że średnio najlepsze wyniki pojawiają się dla ośmiu oraz dziewięciu cech. Można wywnioskować, że najmniej istotna cecha, jaką jest płeć (tabela 1.) jest szumem i nie niesie ze sobą przydatnych informacji. Z kolei biorąc pod uwagę siedem najlepszych cech lub mniej, tracimy znaczące informacje.

Oceniając poszczególne klasyfikatory widzimy, że najlepszy wynik zwrócił klasyfikator **DecisionTree**, uzyskując najlepszy wynik 72,84% poprawnie sklasyfikowanych próbek dla dziewięciu cech (tabela 12.).

Najgorzej poradził sobie natomiast klasyfikator **SVC**, który samodzielnie uzyskał najlepszy wynik 67,37% dla dziewięciu cech (tabela 14.).

Najlepszy wynik ogółem zwróciła metoda *DESP* korzystając z puli homogenicznej składającej się z 50 klasyfikatorów DecisionTree – 78,19% dla ośmiu cech (tabela 12.). Można z tego wywnioskować, że lepiej radzą sobie systemy MCS korzystające z puli homogenicznej skonstruowanej z najlepiej radzących sobie klasyfikatorów bazowych, niż systemy MCS korzystające z puli heterogenicznych. Przechodząc do samych metod MCS, możemy zauważyć, że metody *KNORAE* oraz *DESP* poradziły sobie lepiej niż metoda *OLA*, wniosek – metody DES dają lepsze rezultaty niż metody DCS.

6. Podsumowanie

Celem pracy magisterskiej było stworzenie kilku systemów wieloklasyfikatorowych (MCS - *Multiple classifier system),* wykorzystujących dynamiczną selekcję klasyfikatora / klasyfikatorów (DCS/DES) oraz przeprowadzenie badań eksperymentalnych porównujących ich działanie z klasyfikatorami bazowymi, wykorzystując jako dane medyczny problem diagnostyczny, jakim jest wykrywanie chorób wątroby. Cel ten udało się osiągnąć, uzyskując spodziewany rezultat, jakim jest poprawa jakości klasyfikacji systemów MCS względem klasyfikatorów bazowych.

Literatura

[1] https://www.poradnikzdrowie.pl/zdrowie/wzw/choroby-watroby-objawy-chorej-watroby-przyczyny-i-leczenie-aa-tsXu-TZD2-gaQn.html, [01.06.2023].

[2] https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset), [01.05.2023].

[3] D. J. French, K. A. Sargent-Cox, S. Kim, K. J. Anstey. Gender differences in alcohol consumption among middle-aged and older adults in australia, the united states and korea. Australian and New Zealand Journal of Public Health, 38(4):332–339, 2014.

[4] J. Fevery. Bilirubin in clinical practice: a review. Liver International, 28(5):592–605, 2008.

[5] U. Sharma, D. Pal, R. Prasad. Alkaline phosphatase: an overview. Indian Journal of Clinical Biochemistry, 29(3):269–278, 2014.

[6] J. A. Cohen, M. M. Kaplan. The sgot/sgpt ratio—an indicator of alcoholic liver disease. Digestive diseases and sciences, 24(11):835–838, 1979.

[7] C. Huguet, B. Nordlinger, P. Bloch, J. Conard. Tolerance of the human liver to prolonged normothermic ischemia: a biological study of 20 patients submitted to extensive hepatectomy. Archives of Surgery, 113(12):1448–1451, 1978.

[8] H. Moshage, J. Janssen, J. Franssen, J. Hafkenscheid, S. Yap, i in. Study of the molecular mechanism of decreased liver synthesis of albumin in inflammation. The Journal of clinical investigation, 79(6):1635– 1641, 1987.

[9] W. B. RAWLS, S. Weiss, V. L. Collins. Liver function in rheumatoid (chronic infectious) arthritis. Annals of Internal Medicine, 12(9):1455–1462, 1939.

[10] L. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, Wiley Interscience 2014.

[11] T. Dietterich, Ensemble methods in machine learning, LNCS 1857, 1-15 (2000).

[12] https://python.astrotech.io/machine-learning/bayes/naive.html, [04.06.2023].

[13] https://aigeekprogrammer.com/pl/k-najblizszych-sasiadow-w-klasyfikacji-pisma/, [04.06.2023].

[14] https://towardsdatascience.com/diving-into-c-support-vector-classification-221ced32e4b4, [04.06.2023].

[15] https://aigeekprogrammer.com/pl/klasyfikacja-binarna-regresja-logistyczna-keras/, [06.06.2023].

[16] https://home.agh.edu.pl/~horzyk/lectures/biocyb/BIOCYB-SieciNeuronowe.pdf, [06.06.2023].

[17] https://www.datacamp.com/tutorial/decision-tree-classification-python, [06.06.2023].

[18] Woods, Kevin, W. Philip Kegelmeyer, and Kevin Bowyer. “Combination of multiple classifiers using local accuracy estimates.” IEEE transactions on pattern analysis and machine intelligence 19.4 (1997): 405-410.

[19] Britto, Alceu S., Robert Sabourin, and Luiz ES Oliveira. “Dynamic selection of classifiers—a comprehensive review.” Pattern Recognition 47.11 (2014): 3665-3680.

[20] Woloszynski, Tomasz, et al. “A measure of competence based on random classification for dynamic ensemble selection.” Information Fusion 13.3 (2012): 207-213.

[21] Woloszynski, Tomasz, and Marek Kurzynski. “A probabilistic model of classifier competence for dynamic ensemble selection.” Pattern Recognition 44.10 (2011): 2656-2668.

[22] R. M. Heiberger, E. Neuwirth. One-Way ANOVA, strony 165–191. Springer New York, New York, NY, 2009.

[23] https://numpy.org/doc/stable/reference/generated/numpy.std.html, [12.06.2023].

[24] https://scikit-learn.org/stable/index.html, [12.06.2023].

[25] https://imbalanced-learn.org/stable/, [12.06.2023].

[26] https://pandas.pydata.org/, [12.06.2023].

[27] https://numpy.org/, [12.06.2023].