Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Uczenie Maszynowe - Projekt

Połączenie Lasu Losowego ID3 z Maszyną Wektorów Nośnych (SVM) w zadaniu klasyfikacji binarnej

Dokumentacja

Bartosz Cywiński, Łukasz Staniszewski

304025, 304098

prowadzący
dr Rafał Biedrzycki

WARSZAWA 24 maja 2022

Spis treści

1. Temat projektu . . . 3
2. Drzewo decyzyjne . . . 3
2.1. Opis algorytmu . . . 3
2.1.1. Przykładowe obliczenia przy podziale zbioru danych: . . . 5
2.2. Predykcje algorytmu . . . 6
3. Algorytm SVM . . . 6
3.1. Opis algorytmu . . . 6
3.2. Optymalizacja . . . 7
3.3. Przykładowe obliczenia . . . 8
4. Las losowy . . . 9
4.1. Opis algorytmu . . . 10
4.2. Predykcje algorytmu . . . 10
5. Eksperymenty . . . 11
5.1. Zbiory danych . . . 11
5.2. Analiza skuteczności hybrydy lasu losowego z SVM . . . 11
5.2.1. Wyniki na zbiorach: Breast Cancer, Ionosphere , Biodegradation . . . 12
5.2.2. Wnioski . . . 13
5.3. Analiza wpływu hiperparametrów na skuteczność hybrydy . . . 14
5.3.1. Wpływ liczby klasyfikatorów . . . 14
5.3.2. Wpływ maksymalnej głębokości drzewa . . . 16
5.3.3. Wpływ parametru λ w SVM . . . 17
5.3.4. Wpływ maksymalnej liczby atrybutów branej pod uwagę przy uczeniu klasyfikatorów . . . 19
6. Aspekty techniczne projektu . . . 20
6.1. Technologie wykorzystane w projekcie . . . 20
6.2. Struktura projektu . . . 20
6.3. Instalacja . . . 21
6.4. Testy . . . 21
7. Podsumowanie . . . 21
Bibliografia . . . 22

1. Temat projektu

Celem projektu była implementacja połączenia lasu losowego z SVM w zadaniu klasyfikacji binarnej. Wykonana została własna implementacja obu algorytmów na podstawie źródeł [1], [2] oraz [3]. Najistotniejszą modyfikacją w stosunku do klasycznego algorytmu lasu losowego było to, że klasyfikatorem w lesie jest zarówno drzewo decyzyjne ID3 jak i SVM, zakładając że jest taka sama liczność obu klasyfikatorów. Proces inferencji w lesie losowym jest niezmienny, a więc ostateczną predykcją dla danego przykładu jest najliczniej przewidywana klasa przez wszystkie klasyfikatory.

2. Drzewo decyzyjne

Pierwszym z klasyfikatorów użytych w zmodyfikowanym lesie losowym jest drzewo decyzyjne. Algorytmem uczenia drzewa jest algorytm ID3. Drzewo decyzyjne zaimplementowane zostało w sposób uniwersalny na tyle, żeby poprawnie działało zarówno dla klasyfikacji binarnej jak i wieloklasowej.

2.1. Opis algorytmu

Algorithm 1 ID3

Input: S: zbiór par uczących, Y: zbiór klas, D: zbiór atrybutów wejściowych, d: obecna głębokość drzewa

if $S = \emptyset$ **then return**
end if

if wszystkie przykłady należą do klasy y **then**
return Liść z klasą y
end if

$(j, t) = \arg\min_{j,t} H(S)$
`Root` = węzeł zbudowany na zbiorze S
if !stop **then**
    `Root.left = ID3(`S_-, Y, D, $d+1$`)`
    `Root.right = ID3(`S_+, Y, D, $d+1$`)`
end if
return `Root`

Drzewo decyzyjne ma strukturę drzewa binarnego. W każdym węźle, podczas konstruowania drzewa, wyszukiwany jest atrybut w zbiorze wszystkich atrybutów zbioru danych oraz jego konkretna wartość, która podzieli zbiór danych na dwa podzbiory w taki sposób, aby suma entropii podzbiorów była jak najmniejsza. Na podstawie powstałych w wyniku operacji podziału podzbiorów rekurencyjnie tworzone są kolejne węzły drzewa decyzyjnego.

Drzewo decyzyjne ma następujące atrybuty:

1. maksymalna głębokość drzewa
2. minimalna różnica między entropią po podziale, a entropią przed podziałem
3. minimalna liczba przykładów w węźle drzewa

Oznaczając zbiór przykładów z etykietami jako S, na początku procesu uczenia drzewa decyzyjnego, drzewo ma tylko jeden węzeł (korzeń), który zawiera wszystkie przykłady ze zbioru S. Kolejno wskutek wywołania algorytmu ID3 rozpoczyna się właściwy proces uczenia drzewa, opisany przedstawionym powyżej pseudokodem.

Linie 1-2: Jeśli w zbiorze S, który był argumentem algorytmu ID3, nie ma już żadnej pary uczącej, proces dalszego uczenia drzewa należy zatrzymać.

Linie 4-6: Jeśli w zbiorze S znajdują się przykłady należące tylko do jednej klasy, to znaczy że zbiór jest już idealnie podzielony i nie należy kontynuować procesu uczenia, więc zwracany jest liść drzewa, którego predykcja jest jedyną klasą w zbiorze S.

Linia 8: Oznaczając zbiór wszystkich atrybutów w zbiorze danych jako D, dla każdego indeksu atrybutu $j = 0, ..., D-1$ oraz dla każdej wartości atrybutu występującej w zbiorze danych t wykonywane są następujące kroki:

1. Zbiór wszystkich par uczących S dzielony jest na dwa podzbiory: $S_- = \{(x, y) | (x, y) \in S, x^{(j)} < t\}, S_+ = \{(x, y) | (x, y) \in S, x^{(j)} \geq t\}$.
2. Oceniana jest jakość podziału. W tym celu liczona jest entropia podziału jako entropia ważona dwóch zbiorów: $H(S_-, S_+) = \frac{|S_-|}{|S|} H(S_-) + \frac{|S_+|}{|S|} H(S_+)$, przy czym wartość entropii H dla zbioru S definiuje się jako: $H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$, gdzie C to zbiór wszystkich klas obecnych w zbiorze S, a $p(c)$ to stosunek liczby przykładów klasy c w zbiorze S do liczby wszystkich przykładów w zbiorze S.

Wykonując powyższe kroki znajdywany jest taki atrybut j oraz taka jego wartość t dla których entropia jest najniższa.

Linia 10: Aby dokonać podziału węzła spełnione muszą być następujące warunki:

1. Nie może być przekroczona maksymalna głębokość drzewa.
2. Różnica między entropią $H(S)$, a entropią ważoną $H(S_-, S_+)$ musi być większa od minimalnej dopuszczalnej różnicy między wartościami entropii.
3. Liczność, zarówno zbioru S_-, jak i zbioru S_+ musi być większa od minimalnej dopuszczalnej liczby przykładów w węźle drzewa.

W przypadku niespełnienia jakiegokolwiek z powyższych warunków dany węzeł drzewa nie zostanie dalej podzielony.

Linie 11-12: Do obecnego korzenia (węzła Root), przypisywane są węzły dzieci. Węzły te tworzone są przez rekursywne wywołanie algorytmu ID3, kolejno dla podzbiorów S_-, jak i S_+, jednocześnie zwiększając obecną głębokość drzewa o 1.

2.1.1. Przykładowe obliczenia przy podziale zbioru danych:

Zakładając, że:

$$S = \{([0,2,5],0),([0,4,6],1),([0,-1,7],0)\}$$

Analizując ten prosty zbiór danych można zauważyć, że atrybut o indeksie $j = 1$ oraz jego wartość $t = 4$ podzieli zbiór S tworząc podzbiory S_- oraz S_+ w sposób następujący:

$$S_- = \{(x,y)|(x,y) \in S, x^{(1)} < 4\} = \{([0,2,5],0),([0,-1,7],0)\},$$
$$S_+ = \{(x,y)|(x,y) \in S, x^{(1)} \geq 4\} = \{([0,4,6],1)\}$$

Zatem licząc entropie poszczególnych zbiorów:

$$H(S_-) = -\frac{2}{2}\log_2\frac{2}{2} + (-\frac{0}{2}\log_2\frac{0}{2}) = 0 + 0 = 0$$
$$H(S_+) = -\frac{0}{1}\log_2\frac{0}{2} + (-\frac{1}{1}\log_2\frac{1}{1}) = 0 + 0 = 0$$
$$H(S_-,S_+) = \frac{2}{3}H(S_-) + \frac{1}{3}H(S_+) = \frac{2}{3}\cdot 0 + \frac{1}{3}\cdot 0 = 0$$

Widać na tym przykładzie, że entropia podziału zbioru S jest równa jej minimalnej możliwej wartości, bo podzbiory S_- i S_+ idealnie podzieliły zbiór S pod względem klas. Dla odróżnienia, gdyby został wybrany ten sam atrybut o indeksie $j = 1$, ale o wartości $t = 2$, podział wyglądałby następująco:

$$S_- = \{(x,y)|(x,y) \in S, x^{(1)} < 2\} = \{([0,-1,7],0)\},$$
$$S_+ = \{(x,y)|(x,y) \in S, x^{(1)} \geq 2\} = \{([0,2,5],0),([0,4,6],1)\}$$

Ponownie licząc entropie poszczególnych zbiorów:

$$H(S_-) = -\frac{0}{1}\log_2\frac{0}{1} + (-\frac{1}{1}\log_2\frac{1}{1}) = 0 + 0 = 0$$
$$H(S_+) = -\frac{1}{2}\log_2\frac{1}{2} + (-\frac{1}{2}\log_2\frac{1}{2}) = \frac{1}{2} + \frac{1}{2} = 1$$
$$H(S_-,S_+) = \frac{1}{3}H(S_-) + \frac{2}{3}H(S_+) = \frac{1}{3}\cdot 0 + \frac{2}{3}\cdot 1 = \frac{2}{3}$$

Co pokazuje, że gdy zbiory byłyby tak podzielone, entropia miałaby większą wartość, dlatego też podział nie zostałby wybrany jako najlepszy możliwy.

2.2. Predykcje algorytmu

Po całkowitym wykonaniu się algorytmu uczącego drzewo ID3, drzewo jest w pełni zbudowane i można na nim wykonywać predykcje. Dla pojedynczej próbki danych algorytm przechodzi od korzenia do liścia, w każdym węźle wybierając dziecko do którego powinien następnie przejść na podstawie wartości atrybutu podziału danego węzła dla próbki danych. Mianowicie, przyjmując za j indeks atrybutu podziału danego węzła, a za t przyjmując wartość tego atrybutu, jeśli dla próbki danych x: $x^{(j)} < t$, to algorytm przechodzi do lewego dziecka. W przeciwnym przypadku, algorytm przejdzie do prawego dziecka. Po dotarciu do liścia drzewa decyzyjnego, jako predykcja zwracana jest klasa większościowa danego liścia.

3. Algorytm SVM

Drugim z klasyfikatorów użytych w implementacji jest Maszyna Wektorów Nośnych (SVM) dopuszczająca pomyłki. SVM zaimplementowany został dla przypadku binarnego ze zbiorem klas $Y = \{-1, 1\}$, a także, dla ułatwienia implementacji, założona została wersja algorytmu bez przekształcenia jądrowego (bazowe jądro liniowe).

3.1. Opis algorytmu

Zadanie polega na znalezieniu funkcji rozgraniczającej $f(x) = x \cdot w - b$, która tworzy hiperpłaszczyznę zapewniającą klasyfikację. Otrzymana funkcja powinna zapewniać jak najmniejszą liczbę pomyłek przy klasyfikowaniu elementów zbioru wejściowego do odpowiedniej klasy.

Klasyfikacja odbywa się poprzez zwrócenie dla danego zestawu cech x klasy $y(x) = -1$ lub $y(x) = 1$, do której przynależność wynika z następującej zależności:

$$y(x) = \begin{cases} -1 & , f(x) \leq 0 \\ 1 & , f(x) > 0 \end{cases} \tag{1}$$

Na końcu procesu inferencji, otrzymane predykcje są mapowane z powrotem do odpowiednich klas ze zbioru X, przykładowo: $-1 \rightarrow 0$.
Ze względu na trenowanie dopuszczające pomyłki, aby otrzymać wyżej wymienioną funkcję f, należy znaleźć parametry (w, b), minimalizujące funkcję straty J:

$$(w, b) = \arg\min_{w,b} J(w, b) \tag{2}$$

$$J(w, b) = \Sigma_i \cdot \xi_i + \lambda \cdot ||w||^2 \tag{3}$$

Przy czym, istotne są odpowiednie ograniczenia (gdzie ξ_i oznacza stratę dla i-tego przykładu trenującego, jeśli klasyfikacja jest błędna; a λ decyduje o istotności szerokości regionu separującego):

$$\lambda > 0 \tag{4}$$

$$\forall_i \, [\xi_i \geq 0 \, \wedge \, y_i \cdot (x_i \cdot w - b) \geq 1 - \xi_i] \tag{5}$$

Wymienione wyżej warunki (4) oraz (5) w połączeniu z postacią funkcji straty (3) implikują ostateczną postać funkcji J:

$$J(w,b) = \Sigma_i \max(1 - f(x_i) \cdot y_i, 0) + \lambda \cdot ||w||^2 \tag{6}$$

Powyższy opis algorytmu można podsumować w postaci algorytmu uczenia (Algorithm 2) oraz algorytmu predykcji (Algorithm 3). W algorytmie predykcji zastosowane zostało wyrażenie logiczne, zwracające 1 gdy jest prawdziwe, a 0 gdy fałszywe, co pozwala mu zwracać numery klas zamiast liczb rzeczywistych. Na końcu, zwracane klasy mapowane są do odpowiednich wartości, przykładowo: $-1 \rightarrow 0$. W algorytmie uczenia, początkowo poprawiane są wejściowe etykiety tak, aby były ze zbioru $\{-1, 1\}$, następnie inicjalizowane są parametry modelu i przekazywane do metody optymalizacyjnej, której opis znajduje się w następnym podrozdziale. Na końcu zwracane są wytrenowane parametry modelu.

Algorithm 2 Uczenie SVM

Input: X: zestaw przykładów dla zbioru trenującego, Y: zestaw etykiet dla zbioru trenującego, λ: parametr funkcji straty, V: wektor parametrów dla optymalizatora

1: $Y' =$ `correct_targets`(Y)
2: `initialize`: $w_0 = [0\,0\,...\,0]^T$, $b_0 = 0$
3: $(w,b) =$ `gradient_descent`$(w_0, b_0, X, Y', V, \lambda)$
4: **return** (w,b)

Algorithm 3 Predykcja SVM

Input: X: zestaw przykładów dla zbioru ewaluacyjnego, (w,b): parametry modelu, λ: parametr funkcji straty

1: **return** `repair_targets`$(2 \cdot ((X \cdot w - b) > 0) - 1)$

3.2. Optymalizacja

Istotnym elementem uczenia modelu SVM, jest wyznaczenie jego parametrów poprzez minimalizację funkcji straty J. Zaimplementowany został algorytm Stochastycznego Spadku Gradientowego (Stochastic Gradient Descent / SGD), do którego działania wymagane jest obliczenie gradientu funkcji straty $J(w,b)$ po parametrach modelu w oraz b.

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial w_1} & \cdots & \frac{\partial J}{\partial w_n} & \frac{\partial J}{\partial b} \end{bmatrix}^T \tag{7}$$

Gradient ten jest wektorem pochodnych cząstkowych wyliczanych po kolejnych parametrach modelu (gdzie $x_{k[i]}$ oznacza i-ty atrybut k-tego przykładu).

$$\frac{\partial J}{\partial w_i} = \lambda \cdot 2 \cdot w_i + \Sigma_k (1 \cdot \begin{cases} 0 & , 1 - f(x_k \cdot) y_k \leq 0 \\ -y_k \cdot x_{k[i]} & , 1 - f(x_k) \cdot y_k > 0 \end{cases}) \tag{8}$$

$$\frac{\partial J}{\partial b} = \Sigma_k (1 \cdot \begin{cases} 0 & , 1 - f(x_k) \cdot y_k \leq 0 \\ y_k & , 1 - f(x_k) \cdot y_k > 0 \end{cases}) \tag{9}$$

Zaimplementowany optymalizator SGD można przedstawić w formie algorytmu (Algorithm 4). Metoda zakłada jednokrotne przetworzenie całego zbioru trenującego w ramach jednego kroku i rozpoczyna się od ustalenia parametrów optymalizatora, gdzie max_steps oznacza maksymalną liczbę kroków optymalizacji, min_steps oznacza najmniejszą możliwą normę z parametrów modelu, natomiast β to tzw. learning rate.

Algorithm 4 SGD

Input: (w_0, b_0): zainicjowane parametry modelu, X: zbiór przykładów (wejść), Y: zbiór etykiet (wyjść), V: parametry optymalizatora, λ: współczynnik λ modelu

1: $(max\_steps, \beta, min\_eps) = V$
2: $(w, b) = (w_0, b_0)$
3: $step = 0$
4: **while** $step \leq max\_steps$ **do**
5: $\quad \nabla w = \lambda \cdot 2 \cdot w_i + \Sigma_k (1 \cdot \begin{cases} 0 & , 1 - f(x_k) \cdot y_k \leq 0 \\ -y_k \cdot x_{k[i]} & , 1 - f(x_k) \cdot y_k > 0 \end{cases})$ `for each` $i = 1...M$
6: $\quad \nabla b = \Sigma_k (1 \cdot \begin{cases} 0 & , 1 - f(x_k) \cdot y_k \leq 0 \\ y_k & , 1 - f(x_k) \cdot y_k > 0 \end{cases})$
7: $\quad$ **if** $|w| < min\_eps$ **then return** (w, b)
8: $\quad$ **end if**
9: $\quad w = w - \beta \cdot \nabla w$
10: $\quad b = w - \beta \cdot \nabla b$
11: $\quad step = step + 1$
12: **end while**
13: **return** (w, b)

3.3. Przykładowe obliczenia

Zakładając, że z każdym przykładem związane są 3 atrybuty i istnieje następujący zbiór treningowy T składający się z 3 przykładów oraz zbiór ewaluacyjny E składający się z 2 przykładów:

$$X_T = \begin{vmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 7 & 8 & 7 \end{vmatrix}, Y_T = \begin{vmatrix} 1 \\ 0 \\ 1 \end{vmatrix}, X_E = \begin{vmatrix} -3 & 4 & 1 \\ 4 & 2 & 12 \end{vmatrix}$$

Na początku konieczne jest przerobienie etykiet klas dla przykładów trenujących oraz inicjalizacja modelu:

$$Y'_T = \begin{vmatrix} 1 \\ \mathbf{-1} \\ 1 \end{vmatrix}, w = \begin{vmatrix} 1 \\ -1 \\ 1 \end{vmatrix}, b = 3$$

Przy założeniu, że $max\_steps = 1$, $\lambda = 0.5$ oraz $\beta = 0.2$, wykonywany jest jeden krok SGD:

$$1 - f(X_T) \cdot Y'_T = \begin{vmatrix} 2 \\ 3 \\ -2 \end{vmatrix} \implies \nabla w = \begin{vmatrix} 0.5 \cdot 2 \cdot 1 + \mathtt{sum}(\begin{vmatrix} -1 & 6 & 0 \end{vmatrix}^T) \\ 0.5 \cdot 2 \cdot -1 + \mathtt{sum}(\begin{vmatrix} -2 & 5 & 0 \end{vmatrix}^T) \\ 0.5 \cdot 2 \cdot 1 + \mathtt{sum}(\begin{vmatrix} -3 & 4 & 0 \end{vmatrix}^T) \end{vmatrix} = \begin{vmatrix} 6 \\ 2 \\ 2 \end{vmatrix}$$

$$1 - f(X_T) \cdot Y'_T = \begin{vmatrix} 2 & 3 & -2 \end{vmatrix}^T \implies \nabla b = \mathtt{sum}(\begin{vmatrix} 1 & -1 & 0 \end{vmatrix}^T) = 0$$

Na końcu uczenia ustalane są ostateczne parametry modelu:

$$w = \begin{vmatrix} 1 \\ -1 \\ 1 \end{vmatrix} - 0.2 \cdot \begin{vmatrix} 6 \\ 2 \\ 2 \end{vmatrix} = \begin{vmatrix} -0.2 \\ -1.4 \\ 0.6 \end{vmatrix}, b = 3 - 0.2 \cdot 0 = 3$$

Po ustaleniu parametrów modelu, można przejść do predykcji:

$$Y_E = 2 \cdot ((\begin{vmatrix} -3 & 4 & 1 \\ 4 & 2 & 12 \end{vmatrix} \cdot \begin{vmatrix} -0.2 \\ -1.4 \\ 0.6 \end{vmatrix} - 3) > 0) - 1 = 2 \cdot (\begin{vmatrix} -7.4 \\ 0.6 \end{vmatrix} > 0) - 1 = \begin{vmatrix} -1 \\ 1 \end{vmatrix}$$

4. Las losowy

Las losowy wykorzystuje oba algorytmy opisane powyżej. Za klasycznym algorytmem lasu losowego stoi idea baggingu. W algorytmie konstruuje się wiele nieskorelowanych drzew losowych, a następnie się je uśrednia. W przypadku modyfikacji algorytmu, wykonywanego w ramach projektu, zamiast samych drzew decyzyjnych, występuje zbiór klasyfikatorów - drzew decyzyjnych oraz SVM. Każdy klasyfikator wytrenowany jest na zbiorze próbek danych losowanych z oryginalnego zbioru ze zwracaniem. Ponadto w procesie uczenia drzewa losowego przed każdym podziałem węzła losowany jest bez zwracania podzbiór indeksów atrybutów uwzględnianych w procesie podziału. Natomiast w procesie uczenia SVM, podzbiór indeksów atrybutów losowany jest raz, przed rozpoczęciem procedury treningowej. Zasadniczą ideą baggingu jest uśrednienie wielu modeli z dość małym obciążeniem, w skutek czego zmniejszana jest ich wariancja.

4.1. Opis algorytmu

Algorithm 5 Las losowy

Input: S: zbiór par uczących, Y: zbiór klas, D: zbiór atrybutów wejściowych, B: liczba klasyfikatorów

1: **for** b = 1 to B: **do**
2: Wylosuj ze zwracaniem $|S|$ próbek danych ze zbioru S.
3: **if** b mod 2 == 0 **then**
4: Wytrenuj drzewo decyzyjne algorytmem ID3 na wylosowanym zbiorze, przed każdym podziałem węzła losując bez zwracania podzbiór $m \leq |D|$ atrybutów uwzględnianych przy jego podziale.
5: **else**
6: Wylosuj bez zwracania podzbiór podzbiór $m \leq |D|$ atrybutów. Wytrenuj SVM na wylosowanym zbiorze, uwzględniając tylko wylosowane atrybuty.
7: **end if**
8: Dodaj wytrenowany klasyfikator do zbioru klasyfikatorów lasu losowego.

Linia 1: Liczba wszystkich klasyfikatorów w lesie losowym jest hiperparametrem algorytmu lasu losowego, z zastrzeżeniem że musi być ona podzielna przez 2, ponieważ zakładamy że las składa się z drzew decyzyjnych oraz SVM na przemian.

Linia 2: Trenując każdy model w lesie losowym na różnych podzbiorach oryginalnego zbioru danych (z założeniem, że próbki danych w podzbiorze mogą się powtarzać) redukowana jest wariancja modelu, w skutek czego zmniejszane jest przeuczenie lasu.

Linie 4-6: Przed każdym podziałem węzła w drzewie decyzyjnym losowany jest bez zwracania podzbiór atrybutów rozważanych podczas podziału. Ta modyfikacja klasycznego algorytmu uczenia drzewa zmniejsza korelacje między poszczególnymi drzewami w lesie. Gdyby ta modyfikacja nie była zastosowana, atrybutami podziału węzłów w większości drzew w lesie byłyby takie, które najskuteczniej dzielą zbiór danych. Wskutek tego las składałby się ze skorelowanych drzew, co nie zwiększyłoby skuteczności modelu, ponieważ słabe klasyfikatory w lesie byłyby zgodne co do złych predykcji, a to by skutkowało błędnymi ostatecznymi predykcjami lasu. Podobnie przed uczeniem SVM losowany jest bez zwracania podzbiór atrybutów uwzględnianych w procesie uczenia - model uczy się tylko na tych atrybutach.

4.2. Predykcje algorytmu

Po wytrenowaniu wszystkich klasyfikatorów, predykcją lasu losowego jest klasa większościowa w zbiorze predykcji każdego z klasyfikatorów w drzewie.

5. Eksperymenty

5.1. Zbiory danych

Eksperymenty zostały wykonane na 3 zbiorach danych opisanych w poniższej tabeli. Dla każdego ze zbiorów została wykonana walidacja krzyżowa, gdzie ostateczny wynik na zbiorze testowym był estymowany na zasadzie makro-uśredniania, tzn. jako średnia wartość wskaźników jakości uzyskanych na wszystkich podziałach.

Nazwa zbioru	Opis	Liczba przykładów pozytywnych	Liczba przykładów negatywnych
Breast Cancer [4]	Zbiór danych składający się z 569 przykładów posiadających 30 atrybutów ciągłych.	212	357
Ionosphere [5]	Zbiór danych składający się z 351 przykładów posiadających 34 atrybuty ciągłe.	225	126
QSAR biodegradation [6]	Zbiór danych składający się z 1055 przykładów posiadających 41 atrybuty ciągłe.	356	699

Tabela 5.1. Opis zbiorów danych, które zostaną wykorzystane do eksperymentów numerycznych.

5.2. Analiza skuteczności hybrydy lasu losowego z SVM

Wykonane zostały symulacje mające na celu porównanie zaimplementowanej hybrydy Lasu Losowego i SVM z klasycznym Lasem Losowym (z samymi drzewami decyzyjnymi), Lasem Losowym z samymi modelami SVM, modelem SVM i modelem Drzewa Decyzyjnego w zadaniu klasyfikacji binarnej pod względem metryk dokładności (accuracy), odzysku (recall) oraz precyzji (precision), a także metryki F1. Wnioski i analizy przeprowadzone zostały na zagregowanych wynikach z 25 uruchomień, na których wyliczona została średnia arytmetyczna, odchylenie standardowe czy wartości minimalne i maksymalne. Badania wykonane zostały na 3 zbiorach danych opisanych w poprzednim podpunkcie. Hiperparametry dla modeli zostały dobra te, które dawały najlepsze rezultaty w kolejnym eksperymencie:

- Liczba klasyfikatorów w lesie losowym: 14.
- Maksymalna liczba atrybutów dla lasu losowego: 16.
- Maksymalna głębokość drzewa: 6.
- Minimalna różnica entropii: 0.01.
- Minimalna liczba przykładów w węźle: 40.
- Parametr λ dla SVM: 0.005.
- Dla SGD: parametr kroku $\beta = 0.01$, $\epsilon_{min} = 10^{-17}$ oraz $n\_steps_{max} = 10000$.

5.2.1. Wyniki na zbiorach: Breast Cancer, Ionosphere , Biodegradation

Model	Accuracy	Recall	Precision	F1
Model SVM	0.88 ±0.05 min:0.75 max:0.92	0.91 ±0.07 min:0.66 **max:0.98**	0.92 ±0.02 min:0.87 max:0.95	0.90 ±0.04 min:0.76 max:0.94
Drzewo decyzyjne	0.90 ±0.01 min:0.88 max:0.91	0.94 ±0.02 min:0.91 max:0.97	0.91 ±0.01 min:0.88 max:0.93	0.92 ±0.01 min:0.91 max:0.93
Las losowy - hybryda	**0.93** ±0.01 min:**0.91** max:**0.94**	**0.95** ±0.01 min:**0.93** max:**0.98**	**0.93** ±0.01 min:0.90 max:**0.95**	**0.94** ±0.01 min:**0.93** max:**0.96**
Las losowy - same SVM	0.90 ±0.02 min:0.85 max:0.93	0.92 ±0.03 min:0.83 max:**0.98**	0.92 ±0.01 min:**0.91** max:0.94	0.92 ±0.01 min:0.87 max:0.95
Las losowy - klasyczny	0.92 ±0.01 min:0.90 max:**0.94**	**0.95** ±0.01 min:**0.93** max:0.97	**0.93** ±0.01 min:**0.91** max:0.94	**0.94** ±0.01 min:0.92 max:0.95

Tabela 5.2. Porównanie działania różnych typów modeli pod względem metryk accuracy, recall, precision i f1 na zbiorze Breast Cancer [4].

Model	Accuracy	Recall	Precision	F1
Model SVM	0.83 ±0.01 min:0.8 max:0.86	0.93 ±0.02 min:0.89 max:0.90	0.83 ±0.01 min:0.81 max:0.85	0.88 ±0.01 min:0.85 max:0.9
Drzewo decyzyjne	**0.87** ±0.02 min:0.81 max:**0.9**	0.95 ±0.01 min:0.93 max:0.98	**0.87** ±0.03 min:0.8 max:**0.92**	**0.9** ±0.01 min:**0.88** max:**0.93**
Las losowy - hybryda	0.86 ±0.01 min:**0.84** max:0.89	**0.98** ±0.01 min:**0.96** max:0.99	0.84 ±0.01 min:0.81 max:0.86	**0.9** ±0.01 min:**0.88** max:0.92
Las losowy - same SVM	0.85 ±0.01 min:0.83 max:0.87	0.95 ±0.01 min:0.93 max:0.96	0.84 ±0.01 min:**0.82** max:0.85	0.89 ±0.01 min:**0.88** max:0.9
Las losowy - klasyczny	0.85 ±0.02 min:0.82 max:0.88	**0.98** ±0.01 min:**0.96** max:**1.0**	0.82 ±0.02 min:0.79 max:0.85	0.89 ±0.01 min:0.87 max:0.91

Tabela 5.3. Porównanie działania różnych typów modeli pod względem metryk accuracy, recall, precision i f1 na zbiorze Ionosphere [5].

Model	Accuracy	Recall	Precision	F1
Model SVM	0.76 ±0.02 min:0.73 max:0.79	**0.69** ±0.15 min:0.41 max:**0.95**	0.73 ±0.07 min:0.62 max:**0.86**	0.63 ±0.08 min:0.47 max:0.74
Drzewo decyzyjne	0.78 ±0.01 min:0.76 max:0.81	0.67 ±0.03 min:**0.61** max:0.71	0.69 ±0.02 min:0.67 max:0.74	0.67 ±0.02 min:0.64 max:0.72
Las losowy - hybryda	**0.82** ±0.01 min:**0.81** max:**0.84**	0.65 ±0.05 min:0.55 max:0.73	**0.79** ±0.02 min:**0.74** max:0.85	**0.71** ±0.03 min:0.65 max:**0.75**
Las losowy - same SVM	0.81 ±0.01 min:0.78 max:**0.84**	0.64 ±0.07 min:0.44 max:0.75	0.78 ±0.02 min:0.72 max:0.81	0.67 ±0.04 min:0.57 max:**0.75**
Las losowy - klasyczny	0.81 ±0.01 min:0.8 max:0.82	0.65 ±0.03 min:0.58 max:0.73	0.75 ±0.02 min:0.73 max:0.79	0.69 ±0.02 min:**0.66** max:0.74

Tabela 5.4. Porównanie wyników różnych modeli na zbiorze Biodegradation [6].

5.2.2. Wnioski

Na podstawie wyników przeprowadzonych eksperymentów można zauważyć, że zaimplementowana hybryda lasu losowego z Drzewami Decyzyjnymi i SVM w większości przypadków osiąga najlepsze wyniki względem pozostałych modeli, które brały udział w badaniu. W przypadku zbiorów Breast Cancer i Biodegradation, hybrydowy Las Losowy osiąga najlepsze wyniki pod względem metryk Accuracy oraz F1, które są najbardziej pożądanymi metrykami. W przypadku zbioru Ionosphere możemy natomiast zauważyć brak dominacji hybrydy względem pozostałych modeli na tle tych dwóch metryk - spowodowane to może być mniejszym skomplikowaniem domeny tego zbioru - w takim przypadku bardziej pożądane mogą okazać się modele prostsze, takie jak model SVM czy Drzewo Decyzyjne, których trening trwa zdecydowanie krócej, a interferencja osiąga bardzo podobne wyniki. Istotnym jest również zwrócenie uwagi na stabilność w wynikach modelu - zauważyć można, że, spośród wszystkich badanych modeli, wyniki hybrydy osiągają największą stabilność (posiadają najmniejsze odchylenie standardowe na wszystkich zbiorach w niemal wszystkich metrykach) - co jest dużą zaletą tego rozwiązania. Ciekawe wyniki można zauważyć w przypadku zbioru Biodegradation - tylko dla niego ma miejsce sytuacja, gdzie model hybrydowy wypada najgorzej pod względem metryki Recall i najlepiej pod względem metryki Precision na tle pozostałych modeli, co może być związane z faktem, że ten zbiór jest najmniej zbilansowanym w porównaniu do pozostałych - stąd też tutaj bardziej miarodajna wydaje się metryka F1, dla której hybryda osiąga najlepsze wyniki. Na podstawie tych wniosków, można stwierdzić, że wykonana implementacja osiągnęła bardzo korzystne wyniki.

5.3. Analiza wpływu hiperparametrów na skuteczność hybrydy

W ramach kolejnego eksperymentu zbadano wpływ hiperparametrów na skuteczność zaimplementowanej hybrydy lasu losowego z SVM. Wszystkie eksperymenty przeprowadzone zostały dla 3 zbiorów danych wymienionych wcześniej. Dla każdego hiperparametru dobrana została lista badanych jego wartości i dla każdej z tych wartości wyliczono wartość metryki accuracy, precision, recall i f1-score jako średnia arytmetyczna, odchylenie standardowe, wartości minimalne i maksymalne na podstawie wyników zagregowanych z 25 uruchomień. W każdym eksperymencie została zastosowana 5-krotna walidacja krzyżowa, a ostateczny wynik metryki dla każdego uruchomienia to średnia wartość metryki dla 5 podzbiorów testowych. Gdy dane hiperparametry nie są badane w eksperymencie, przyjmują one wartości odpowiednio:

- Liczba klasyfikatorów w lesie losowym: 14.
- Maksymalna liczba atrybutów dla lasu losowego: 16.
- Maksymalna głębokość drzewa: 6.
- Minimalna różnica entropii: 0.01.
- Minimalna liczba przykładów w węźle: 40.
- Parametr λ dla SVM: 0.005.
- Dla SGD: parametr kroku $\beta = 0.01$, $\epsilon_{min} = 10^{-17}$ oraz $n\_steps_{max} = 10000$.

5.3.1. Wpływ liczby klasyfikatorów

Na podstawie uzyskanych wyników w tabeli 5.5 widać wyraźnie, że algorytm osiąga słabe wyniki, kiedy liczba klasyfikatorów jest bardzo mała. Nie jest to zaskakujące, pojedyncze klasyfikatory w lesie są słabymi modelami, a gdy jest ich za mało to nawet uśrednienie predykcji lasu losowego przez przedstawienie ostatecznej predykcji jako klasy większościowej nic nie daje. Dla małej liczby klasyfikatorów zauważalne są też duże wartości odchylenia standardowego. Wzrost ten jest jednak ograniczony. Można zauważyć, że nie ma już bardzo znaczącej różnicy między skutecznością algorytmu złożonego z 14, a złożonego z 20 klasyfikatorów. Również przy większej liczbie klasyfikatorów odchylenia standardowe bardzo maleją, co świadczy o stabilności i dobrej generalizacji modelu. Ciekawą obserwacją jest również to, że głównie na poprawę skuteczności algorytmu wraz ze wzrostem liczby klasyfikatorów wpływa poprawa metryki recall. Metryka precision za to zmienia się w bardzo małym stopniu. Znaczne zwiększenie liczby klasyfikatorów w drzewie do liczby 200 zwiększa skuteczność modelu, chociaż nie zmienia się ona diametralnie. Da się jednak zauważyć na podstawie wyników w tabeli, że większość metryk zwiększa swoje wartości. Wniosek z tego taki, że model, gdy liczba klasyfikatorów w drzewie jest większa potrafi lepiej generalizować, a więc osiąga lepsze wyniki na zbiorze testowym.

Liczba klasyfikatorów	Nazwa zbioru	Accuracy	Recall	Precision	F1
2	Breast Cancer [4]	0.79 ±0.08 min:0.59 max:0.85	0.71 ±0.14 min:0.52 max:0.91	0.92 ±0.08 min:0.73 max:0.98	0.76 ±0.12 min:0.40 max:0.93
8	Breast Cancer [4]	0.92 ±0.17 min:0.85 max:0.93	0.93 ±0.03 min:0.82 max:0.97	**0.93** ±0.02 min:0.91 max:0.96	0.93 ±0.02 min:0.85 max:0.95
14	Breast Cancer [4]	0.92 ±0 min:0.90 max:0.93	0.95 ±0.02 min:0.92 max:0.97	**0.93** ±0.01 min:0.90 max:0.96	0.94 ±0 min:0.93 max:0.95
20	Breast Cancer [4]	0.92 ±0 min:0.91 max:0.93	0.95 ±0.01 min:0.94 max:0.97	**0.93** ±0.01 min:0.90 max:0.95	0.94 ±0 min:0.93 max:0.95
200	Breast Cancer [4]	**0.93** ±0.01 min:0.92 max:0.94	**0.97** ±0 min:0.96 max:0.97	0.92 ±0.01 min:0.91 max:0.93	**0.95** ±0.01 min:0.93 max:0.95
2	Ionosphere [5]	0.84 ±0.02 min:0.81 max:0.87	0.90 ±0.03 min:0.85 max:0.95	**0.87** ±0.02 min:0.83 max:0.90	0.88 ±0.01 min:0.86 max:0.91
8	Ionosphere [5]	**0.87** ±0.01 min:0.84 max:0.89	0.98 ±0 min:0.95 max:0.99	0.84 ±0.01 min:0.82 max:0.85	**0.90** ±0 min:0.88 max:0.92
14	Ionosphere [5]	**0.87** ±0 min:0.85 max:0.88	0.98 ±0 min:0.97 max:0.98	0.84 ±0 min:0.83 max:0.84	**0.90** ±0 min:0.89 max:0.91
20	Ionosphere [5]	0.86 ±0.01 min:0.85 max:0.89	0.98 ±0 min:0.98 max:0.99	0.84 ±0.01 min:0.82 max:0.84	**0.90** ±0 min:0.91 max:0.92
200	Ionosphere [5]	0.86 ±0.01 min:0.85 max:0.87	**0.99** ±0 min:0.98 max:1.0	0.83 ±0.01 min:0.81 max:0.84	**0.90** ±0 min:0.89 max:0.91
2	QSAR biodegradation [6]	0.74 ±0.02 min:0.71 max:0.76	0.35 ±0.11 min:0.14 max:0.53	0.69 ±0.13 min:0.41 max:0.83	0.41 ±0.11 min:0.25 max:0.62
8	QSAR biodegradation [6]	0.80 ±0.01 min:0.78 max:0.81	0.59 ±0.06 min:0.45 max:0.68	**0.79** ±0.03 min:0.76 max:0.82	0.66 ±0.04 min:0.60 max:0.72
14	QSAR biodegradation [6]	0.81 ±0.01 min:0.80 max:0.83	0.62 ±0.07 min:0.50 max:0.72	**0.79** ±0.04 min:0.75 max:0.85	0.68 ±0.04 min:0.67 max:0.72
20	QSAR biodegradation [6]	0.82 ±0 min:0.80 max:0.83	0.65 ±0.04 min:0.57 max:0.67	0.78 ±0.02 min:0.77 max:0.83	0.70 ±0.03 min:0.67 max:0.73
200	QSAR biodegradation [6]	**0.83** ±0.01 min:0.82 max:0.84	**0.69** ±0.01 min:0.68 max:0.7	**0.79** ±0.01 min:0.77 max:0.81	**0.74** ±0.01 min:0.73 max:0.75

Tabela 5.5. Analiza wpływu liczby klasyfikatorów na działanie algorytmu.

5.3.2. Wpływ maksymalnej głębokości drzewa

Maksymalna głębokość drzewa	Nazwa zbioru	Accuracy	Recall	Precision	F1
2	Breast Cancer [4]	**0.92** ±0 min:0.91 max:0.93	**0.95** ±0.01 min:0.94 max:0.98	**0.93** ±0.02 min:0.90 max:0.94	**0.94** ±0 min:0.93 max:0.95
6	Breast Cancer [4]	**0.92** ±0 min:0.91 max:0.93	0.94 ±0.01 min:0.93 max:0.96	**0.93** ±0.02 min:0.90 max:0.95	**0.94** ±0 min:0.93 max:0.95
10	Breast Cancer [4]	**0.92** ±0.01 min:0.89 max:0.94	**0.95** ±0.02 min:0.93 max:0.97	**0.93** ±0.01 min:0.92 max:0.95	**0.94** ±0.01 min:0.93 max:0.96
2	Ionosphere [5]	**0.86** ±0.01 min:0.84 max:0.87	**0.98** ±0 min:0.96 max:0.99	**0.84** ±0.01 min:0.82 max:0.85	**0.90** ±0 min:0.89 max:0.91
6	Ionosphere [5]	**0.86** ±0 min:0.85 max:0.87	**0.98** ±0 min:0.95 max:0.99	0.83 ±0.01 min:0.82 max:0.84	**0.90** ±0 min:0.88 max:0.92
10	Ionosphere [5]	**0.86** ±0.01 min:0.83 max:0.88	**0.98** ±0.01 min:0.97 max:0.99	0.83 ±0.02 min:0.80 max:0.86	**0.90** ±0.01 min:0.88 max:0.91
2	QSAR biodegradation [6]	0.79 ±0.01 min:0.78 max:0.81	0.59 ±0.05 min:0.49 max:0.61	0.77 ±0.03 min:0.70 max:0.81	0.65 ±0.03 min:0.58 max:0.69
6	QSAR biodegradation [6]	**0.81** ±0.01 min:0.80 max:0.83	**0.65** ±0.04 min:0.59 max:0.66	**0.78** ±0.03 min:0.75 max:0.83	**0.70** ±0.02 min:0.67 max:0.73
10	QSAR biodegradation [6]	**0.81** ±0.01 min:0.77 max:0.83	0.63 ±0.06 min:0.48 max:0.72	**0.78** ±0.02 min:0.78 max:0.82	0.69 ±0.04 min:0.57 max:0.72

Tabela 5.6. Analiza wpływu maksymalnej głębokości drzewa decyzyjnego na działanie algorytmu.

Wyniki dla wszystkich zmierzonych wartości maksymalnej głębokości drzewa są bardzo zbliżone, co widać w tabeli 5.6. Ciekawym spostrzeżeniem jest większa wartość odchylenia standardowego dla większej maksymalnej głębokości w większości przypadków. Może to być oznaką lekkiego przeuczania się modelu - drzewa stają się zbyt rozbudowane przez co za bardzo dopasowują się do zbioru danych. Niemniej jednak, jako że otrzymano podobne wyniki w każdym eksperymencie, można wysnuć wniosek, że zmiana wartości hiperparametrów tylko jednego z dwóch użytych klasyfikatorów w naszym algorytmie nie będzie drastycznie wpływała na skuteczność całego modelu hybrydowego.

5.3.3. Wpływ parametru λ w SVM

λ	Nazwa zbioru	Accuracy	Recall	Precision	F1
0.005	Breast Cancer [4]	**0.93** ±0 min:0.92 max:0.94	**0.95** ±0.01 min:0.92 max:0.98	**0.93** ±0.01 min:0.92 max:0.95	**0.94** ±0 min:0.93 max:0.94
0.05	Breast Cancer [4]	0.92 ±0 min:0.91 max:0.93	**0.95** ±0.01 min:0.93 max:0.96	**0.93** ±0.01 min:0.92 max:0.96	**0.94** ±0 min:0.93 max:0.95
0.5	Breast Cancer [4]	0.91 ±0.02 min:0.89 max:0.94	0.93 ±0.03 min:0.90 max:0.97	**0.93** ±0.02 min:0.91 max:0.94	0.93 ±0.02 min:0.92 max:0.94
5	Breast Cancer [4]	0.92 ±0 min:0.89 max:0.93	0.94 ±0.02 min:0.88 max:0.97	**0.93** ±0.02 min:0.91 max:0.95	0.93 ±0 min:0.91 max:0.94
0.005	Ionosphere [5]	**0.87** ±0.01 min:0.85 max:0.88	0.98 ±0 min:0.97 max:0.98	**0.84** ±0.01 min:0.82 max:0.86	**0.91** ±0 min:0.90 max:0.93
0.05	Ionosphere [5]	0.86 ±0.01 min:0.86 max:0.87	0.98 ±0 min:0.96 max:0.99	0.83 ±0.01 min:0.82 max:0.85	0.90 ±0 min:0.89 max:0.91
0.5	Ionosphere [5]	0.86 ±0.01 min:0.85 max:0.88	0.98 ±0 min:0.97 max:0.98	0.83 ±0.01 min:0.83 max:0.84	0.90 ±0 min:0.89 max:0.91
5	Ionosphere [5]	0.84 ±0.01 min:0.83 max:0.86	**0.99** ±0 min:0.98 max:0.99	0.81 ±0.01 min:0.79 max:0.82	0.89 ±0 min:0.88 max:0.90
0.005	QSAR biodegradation [6]	**0.82** ±0 min:0.80 max:0.83	**0.66** ±0.11 min:0.57 max:0.71	0.79 ±0.02 min:0.78 max:0.81	**0.71** ±0.03 min:0.67 max:0.73
0.05	QSAR biodegradation [6]	0.81 ±0 min:0.80 max:0.82	0.63 ±0.05 min:0.53 max:0.73	0.78 ±0.02 min:0.76 max:0.83	0.69 ±0.02 min:0.67 max:0.72
0.5	QSAR biodegradation [6]	0.79 ±0.01 min:0.77 max:0.82	0.53 ±0.05 min:0.44 max:0.57	**0.81** ±0.03 min:0.78 max:0.84	0.62 ±0.04 min:0.58 max:0.67
5	QSAR biodegradation [6]	0.76 ±0.02 min:0.74 max:0.79	0.36 ±0.08 min:0.18 max:0.46	0.80 ±0.10 min:0.61 max:0.92	0.47 ±0.08 min:0.42 max:0.61

Tabela 5.7. Analiza wpływu parametru λ w SVM na działanie algorytmu.

Na podstawie analizy wyników z tego eksperymentu, przedstawionych w tabeli 5.7, dojść można do ciekawych obserwacji, które były już zauważalne podczas analizy wpływu maksymalnej głębokości drzewa decyzyjnego na skuteczność hybrydy, niemniej jednak w dużo mniejszym stopniu. Zmiana hiperparametru pojedynczego typu klasyfikatora w hybrydzie dużo bardziej oddziałuje na skuteczność algorytmu na trudniejszym zbiorze danych, niż na względnie łatwym. Podobnie jak to było we wspomnianym wcześniej

eksperymencie, wyniki dla każdej możliwej wartości λ dla dwóch pierwszych badanych zbiorów danych są bardzo podobne, po czym można by wysnuć wniosek, że parametr ten nie wpływa znacząco na skuteczność hybrydy. Jednakże, po analizie wyników dla ostatniego zbioru danych widać, że parametr ten wpływa i to w bardzo znaczącym stopniu. Najbardziej widoczna jest różnica w metryce F1, gdzie w najlepszym przypadku jest ona równa 0.71, a w najgorszym 0.47. Znowu jak to było we wcześniejszych eksperymentach, tak i tu zmienia się głównie wartość metryki recall. Z eksperymentów można wyciągnąć wniosek, że model działa dużo lepiej dla małego współczynnika regularyzacji w klasyfikatorze SVM. Może być tak z tego powodu, że współczynnik ten spełnia bardzo istotną rolę zapobiegając przeuczeniu, gdy SVM uczony jest klasycznie tj. na całym zbiorze danych i wszystkich atrybutach. W tym przypadku pamiętać trzeba, że pojedynczy klasyfikator uczony jest tylko na podzbiorze atrybutów i danych, co już zapobiega nadmiernemu dopasowywaniu się modeli. Dlatego też nie jest konieczny wysoki współczynnik λ.

5.3.4. Wpływ maksymalnej liczby atrybutów branej pod uwagę przy uczeniu klasyfikatorów

Maksymalna liczba atrybutów	Nazwa zbioru	Accuracy	Recall	Precision	F1
8	Breast Cancer [4]	0.92 ±0 min:0.90 max:0.93	0.94 ±0.02 min:0.94 max:0.98	**0.93** ±0.01 min:0.90 max:0.95	**0.94** ±0 min:0.92 max:0.95
16	Breast Cancer [4]	0.92 ±0 min:0.90 max:0.93	0.95 ±0.02 min:0.92 max:0.96	**0.93** ±0.01 min:0.91 max:0.94	0.93 ±0 min:0.92 max:0.95
24	Breast Cancer [4]	**0.93** ±0 min:0.92 max:0.94	**0.96** ±0.01 min:0.92 max:0.97	**0.93** ±0.01 min:0.91 max:0.93	**0.94** ±0 min:0.92 max:0.94
8	Ionosphere [5]	0.83 ±0.01 min:0.83 max:0.96	**0.99** ±0 min:0.97 max:0.99	0.80 ±0.01 min:0.79 max:0.83	0.88 ±0 min:0.87 max:0.90
16	Ionosphere [5]	0.86 ±0.01 min:0.85 max:0.89	0.98 ±0 min:0.97 max:0.99	0.83 ±0.01 min:0.82 max:0.86	0.90 ±0 min:0.90 max:0.92
24	Ionosphere [5]	**0.87** ±0 min:0.85 max:0.89	0.98 ±0 min:0.97 max:0.99	**0.85** ±0 min:0.83 max:0.87	**0.91** ±0 min:0.90 max:0.91
8	QSAR biodegradation [6]	0.80 ±0.01 min:0.79 max:0.82	0.58 ±0.06 min:0.42 max:0.63	0.79 ±0.02 min:0.76 max:0.84	0.65 ±0.04 min:0.52 max:0.69
16	QSAR biodegradation [6]	**0.81** ±0.01 min:0.78 max:0.83	0.59 ±0.06 min:0.48 max:0.70	**0.80** ±0.03 min:0.76 max:0.84	0.67 ±0.03 min:0.60 max:0.69
24	QSAR biodegradation [6]	**0.81** ±0.01 min:0.79 max:0.84	**0.64** ±0.05 min:0.60 max:0.71	0.78 ±0.02 min:0.75 max:0.82	**0.69** ±0.03 min:0.67 max:0.72

Tabela 5.8. Analiza wpływu maksymalnej liczby atrybutów branej pod uwagę przy uczeniu klasyfikatorów.

Analizując wyniki z tabeli 5.8, widać że na zbadanych zbiorach danych nie obserwuje się znacznego wpływu maksymalnej liczby atrybutów na skuteczność algorytmu. Jest to optymistyczna obserwacja, ponieważ klasyfikatory należące do hybrydy były w stanie dobrze nauczyć się już na tylko kilku atrybutach. Może to być też spowodowane tym, że w lesie użyto 14 klasyfikatorów. Można domyślać się, że gdyby klasyfikatorów było mniej, różnice w wynikach byłyby bardziej zauważalne.

6. Aspekty techniczne projektu

6.1. Technologie wykorzystane w projekcie

Projekt rozwijany był z użyciem repozytorium kodu na GitHub. Implementacje wykonano z użyciem języka programowania Python 3.8.10 na systemie operacyjnym Ubuntu 20.04 LTS. W celu implementacji operacji macierzowych i matematycznych wykorzystano bibliotekę NumPy, do testów jednostkowych narzędzie PyTest, a do formatowania kodu narzędzie Black.

6.2. Struktura projektu

Po rozpakowaniu folderu z projektem, ukazuje się następująca struktura katalogów i plików:

```
.
├── experiments/            ← folder z eksperymentami
│   ├── *.py                ← skrypty przeprowadzające eksperymenty
│   └── logs/               ← folder z logami
│       ├── logs.log        ← logi
│       └── *.csv           ← wyniki eksperymentów
├── data/                   ← folder ze zbiorami danych
├── src/                    ← folder z kodem implementacji
│   ├── SVM/                ← folder z implementacją SVM
│   │   ├── *.py            ← kod SVM
│   │   └── tests/          ← folder z testami SVM
│   └── random_forest/      ← folder z implementacją lasu losowego
│       ├── *.py            ← kod lasu losowego
│       └── tests/          ← folder z testami lasu losowego
├── main.py                 ← skrypt uruchamiający model
├── main_experiments.py     ← skrypt uruchamiający eksperymenty
└── requirements.txt        ← moduły Python do zainstalowania
```

6.3. Instalacja

W celu zainstalowania projektu i uruchomienia eksperymentów, należy wykonać następujące kroki:

1. Uruchomić system operacyjny Ubuntu z zainstalowanym Pythonem i pobranym repozytorium.
2. Przejść do folderu z projektem.
3. Zainstalować niezbędne moduły języka Python:

```
$ pip install -r requirements.txt
```

4. Uruchomić model hybrydowy z wybranymi hiperparametrami, przykładowo:

```
$ python3 main.py --dataset breast_cancer --n_folds 5
    --num_classifiers 4 --tree_max_depth 4
    --tree_min_entropy_diff 0.001
    --tree_min_node_size 34 --svm_lambda 0.05
```

5. Uruchomić eksperymenty porównujące modele:

```
$ python3 main_experiments.py -WHAT models
```

6. Uruchomić eksperymenty porównujące wartości hiperparametrów:

```
$ python3 main_experiments.py -WHAT parameters
```

7. Uruchomić testy jednostkowe:

```
$ python3 -m pytest
```

6.4. Testy

W celu sprawdzenia czy implementacja nie zawiera błędów, działanie algorytmu sprawdzone zostało ręcznie - w tym celu zostały wykonane testy jednostkowe, które znajdują się w folderach tests/ w folderze src/. Poza sprawdzeniem działania podstawowych, logicznych elementów algorytmów, dodatkowo sprawdzono działanie przy małym, spreparowanym zbiorze danych.

7. Podsumowanie

W ramach wykonanego projektu nauczyliśmy się tworzyć modele, które cechuje poprawność i uniwersalność względem używanych danych, jak i pogłębiliśmy wiedzę odnośnie modeli SVM oraz Drzew Decyzyjnych. Udało nam się zauważyć, jak różnie można oceniać modele w zależności od stosowanej metryki, zestawu hiperparametrów jak i charakterystki zbioru danych. Zauważyliśmy, jak istotny jest dobór odpowiednich hiperparametrów do danego modelu oraz jak zróżnicowane są względem siebie poszczególne modele w uczeniu maszynowym.

Bibliografia

[1] Paweł Zawistowski, "Wykłady z przedmiotu Wprowadzenie do Sztucznej Inteligencji (WSI)", *Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska*, 2020.

[2] Paweł Cichosz, "Wykłady z przedmiotu Uczenie Maszynowe (UMA)", *Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska*, 2022.

[3] R. T. Trevor Hastie i J. Friedman, *The Elements of Statistical Learning*. Stanford, California: Springer, 2008.

[4] University of Wisconsin, "Breast Cancer Wisconsin (Diagnostic) Data Set", 1995.

[5] Space Physics Group, Johns Hopkins University, "Ionosphere Data Set", 1989.

[6] Milano Chemometrics and QSAR Research Group, Università degli Studi di Milano Bicocca), "QSAR biodegradation Data Set", 2013.