

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I ZARZĄDZANIA

KIERUNEK: INFORMATYKA
SPECJALNOŚĆ: DANOLOGIA
KURS: INDUKCYJNE METODY ANALIZY DANYCH

Zespoły klasyfikatorów
Dokumentacja ćwiczenia nr 5

AUTOR:
Adam Dłubak

PROWADZĄCY:
Dr inż. Paweł Myszkowski

Spis treści

1. Zbiory Badawcze	4
1.1. Iris Dataset	4
1.2. Wine Dataset	5
1.3. Glass Dataset	7
1.4. Pima Indians Diabetes Dataset	8
2. Wstęp teoretyczny	10
2.1. Zespoły klasyfikatorów	10
2.2. Metoda <i>Bagging</i> - Algorytm <i>BaggingClassifier</i>	10
2.2.1. Badane parametry algorytmu <i>BaggingClassifier</i>	11
2.3. Metoda <i>Boosting</i> - Algorytm <i>AdaBoost</i>	11
2.3.1. Badane parametry algorytmu <i>AdaBoostClassifier</i>	12
2.4. Metoda <i>RandomForest</i>	13
2.4.1. Badane parametry algorytmu <i>RandomForest</i>	14
2.5. Miary jakości klasyfikatora	15
3. Badania i analiza wyników	16
3.1. Normalizacja	16
3.2. Bagging	17
3.2.1. Parametr n-estimators	17
3.2.2. Parametr bootstrap-features	17
3.2.3. Parametr bootstrap	18
3.2.4. Parametr max-samples	18
3.2.5. Parametr max-features	19
3.3. Boosting	19
3.3.1. Parametr n-estimators	19
3.3.2. Parametr learning-rate	20
3.3.3. Parametr algorithm	20
3.4. RandomForest	21
3.4.1. Parametr n-estimators	21
3.4.2. Parametr criterion	21
3.4.3. Parametr min-samples-leaf	22
3.5. Najlepsze uzyskane rezultaty	23

3.6. Porównanie wszystkich przebadanych algorytmów	23
--	----

Rozdział 1

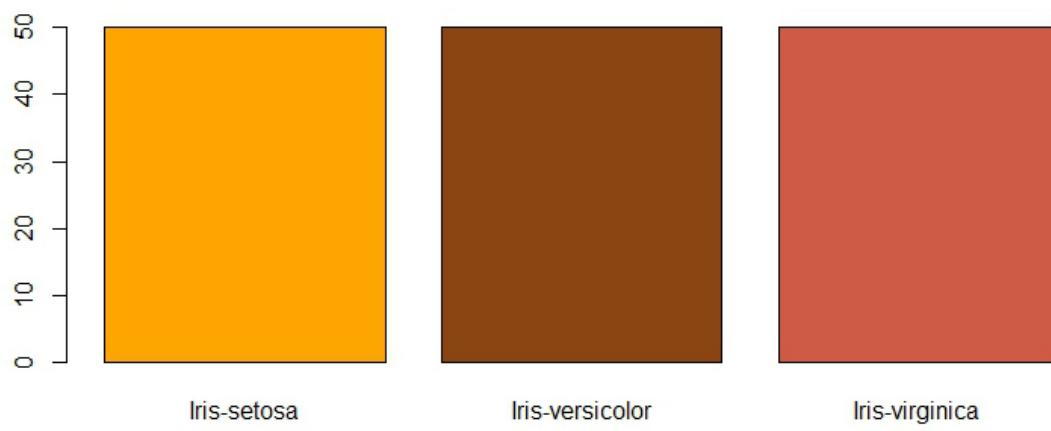
Zbiory Badawcze

W ramach realizacji ćwiczenia, badania zostały oparte o 1 zbiór wykorzystany do wstępnej weryfikacji oraz 3 zbiory danych testowych wykorzystywanych już wcześniej. Wykorzystane zbiory danych to:

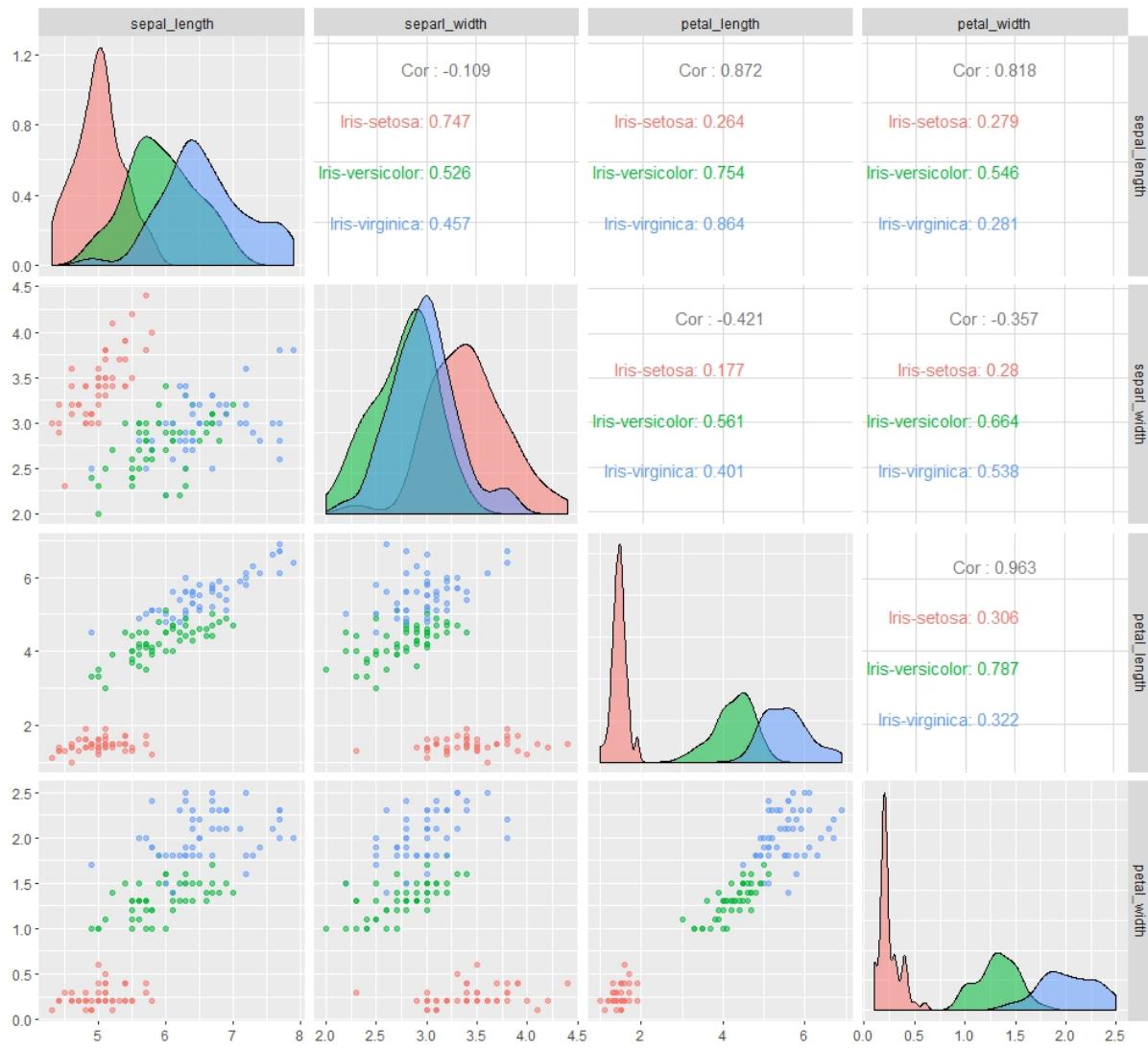
1.1. Iris Dataset

Zestaw pomiarów kwiatów irysa, udostępniony po raz pierwszy przez Ronaldą Fishera w roku 1936. Jeden z najbardziej znanych zbiorów, a zarazem bardzo prosty i użyteczny. Zbiór irysów składa się z 4 wartości pomiarów jego płatków (szerokości i długość) oraz klasy do jakiej należy.

- Liczba atrybutów: 4
- Rodzaj atrybutów: wartości typu Float
- Liczba instancji: 150
- Liczba klas: 3



Rys. 1.1: Rozłożenie ilościowe poszczególnych klas zbioru Iris

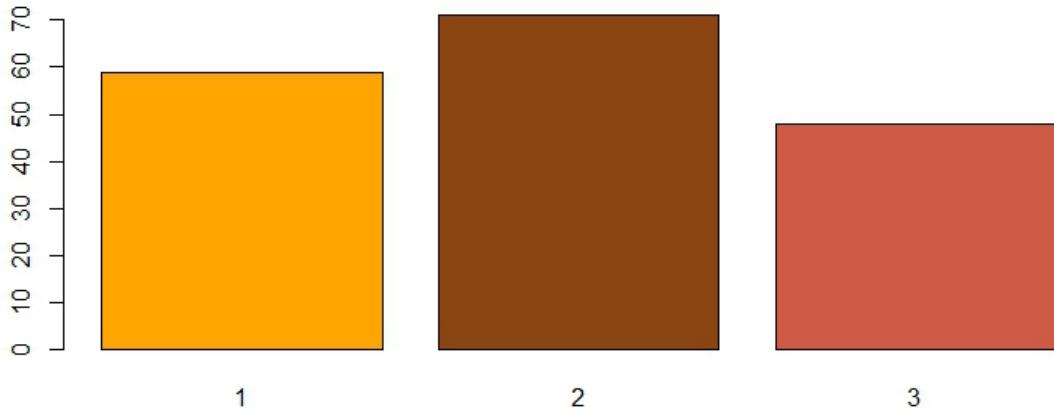


Rys. 1.2: Zestawienie zależności atrybutów i klas zbioru Iris

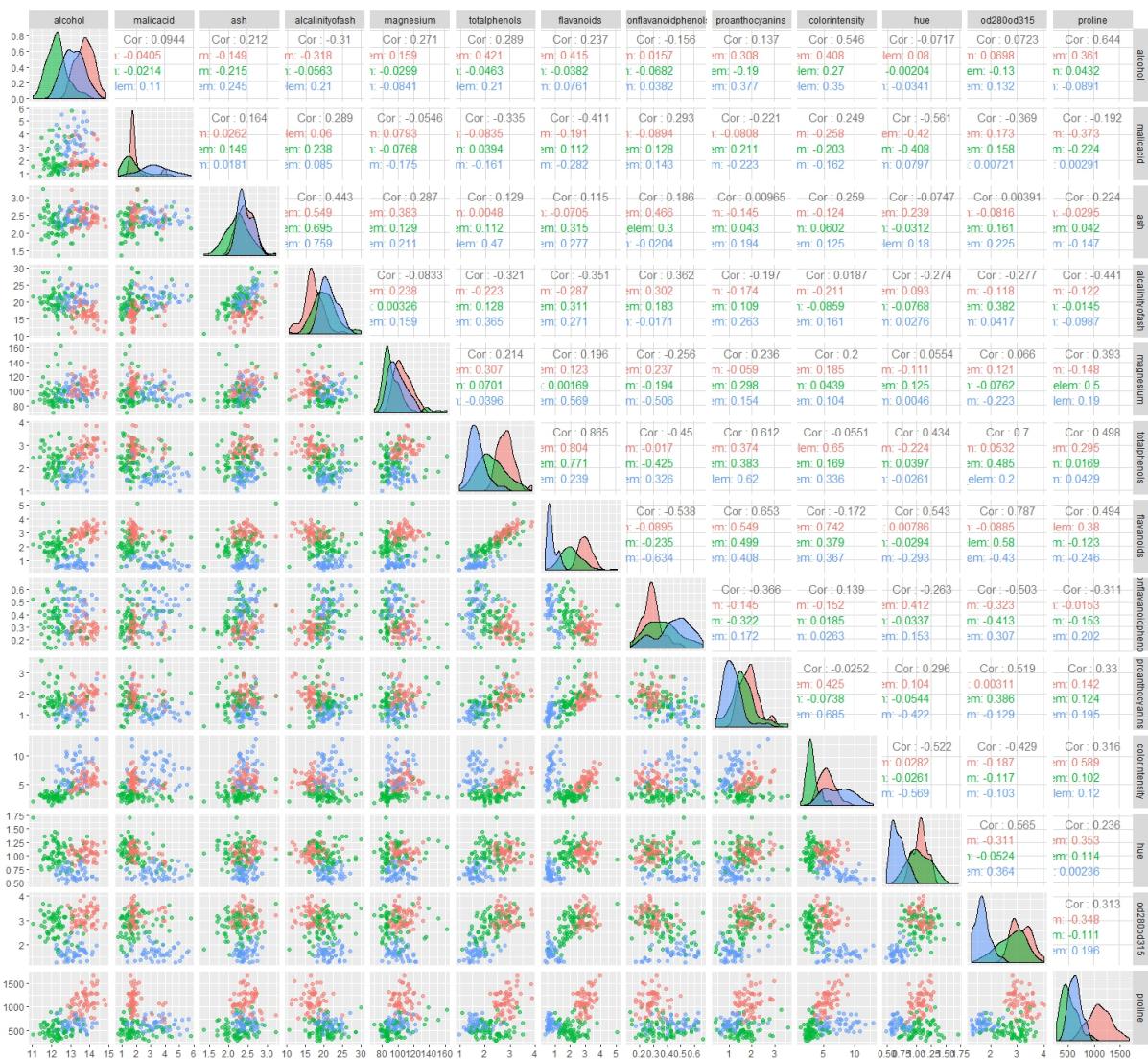
1.2. Wine Dataset

Zbiór danych jest wynikiem analizy chemicznej win uprawianych w tym samym regionie we Włoszech, ale uzyskanych z trzech różnych odmian. W analizie określono 13 składników znalezionych w każdym z trzech rodzajów win.

- Liczba atrybutów: 13.
- Rodzaj atrybutów: wartości typu Float i Intiger.
- Liczba instancji: 178.
- Liczba klas: 3



Rys. 1.3: Rozłożenie ilościowe poszczególnych klas zbioru Wine

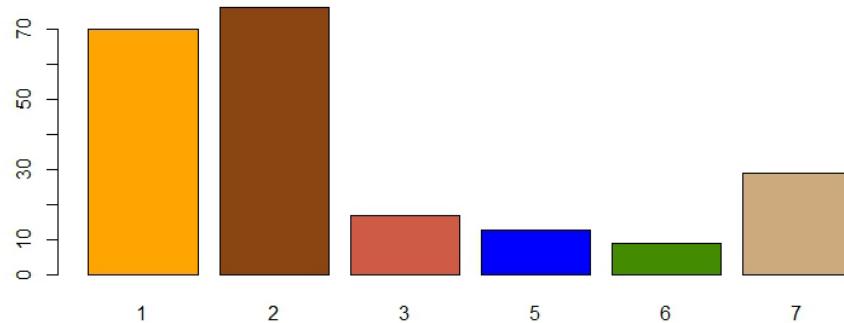


Rys. 1.4: Zestawienie zależności atrybutów i klas zbioru Wine

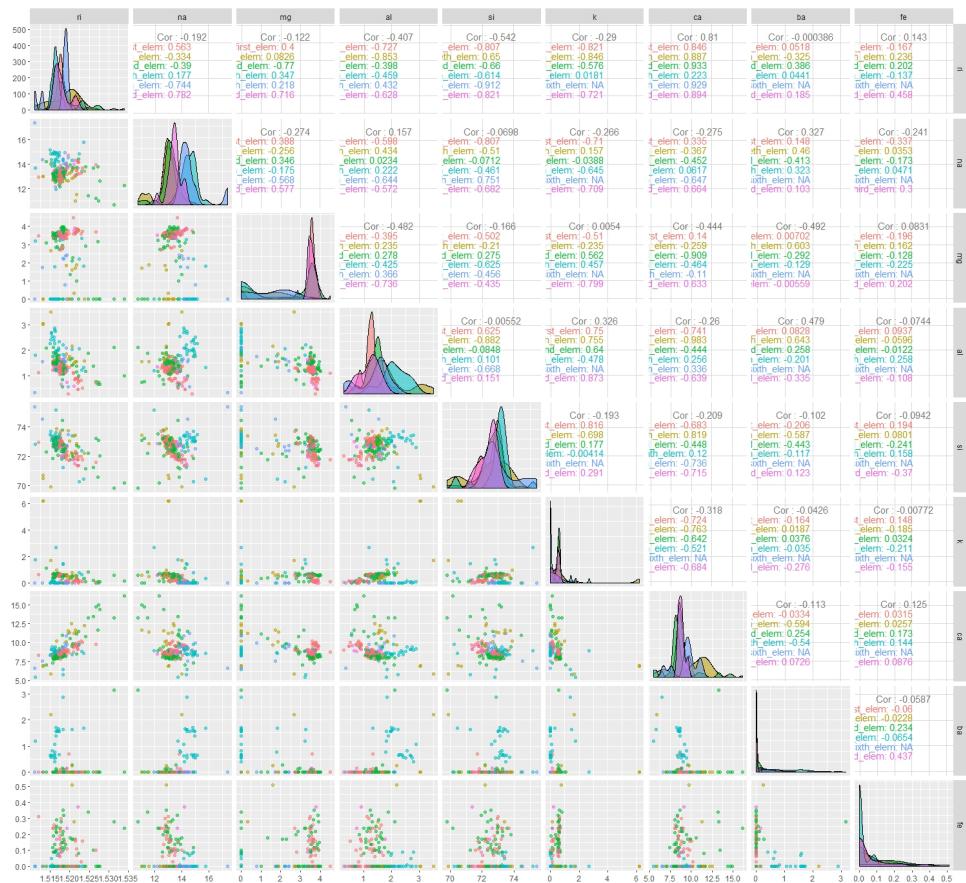
1.3. Glass Dataset

Zbiór danych powstał w wyniku motywacji badania dochodzeń kryminologicznych. Poprawne zidentyfikowanie rodzaju szkła znalezioneego na miejscu przestępstwa, na postawie jego składu pozwala na użycie go jako dowodu w sprawie.

- Liczba atrybutów: 9.
- Rodzaj atrybutów: realistyczne, ciągłe.
- Liczba instancji: 214.
- Liczba klas: 7.



Rys. 1.5: Rozłożenie ilościowe poszczególnych klas zbioru Glass

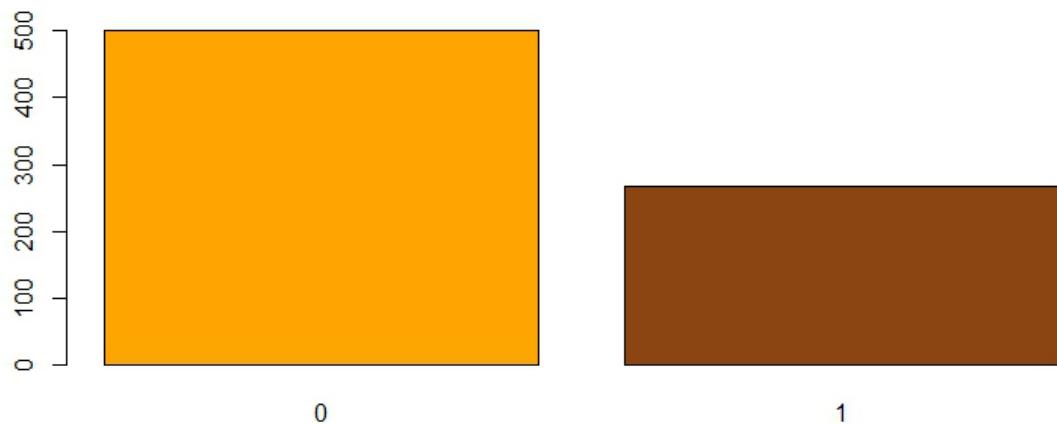


Rys. 1.6: Zestawienie zależności atrybutów i klas zbioru Glass

1.4. Pima Indians Diabetes Dataset

Zbiór danych Pima Indian Diabetes przewidywanie wystąpienie cukrzycy w oparciu o badania diagnostyczne. Pochodzi on z *National Institute of Diabetes and Digestive and Kidney Diseases*. Zawiera on dane dotyczące zachorowań na cukrzycę wśród kobiet z indiańskiego plemienia Pima. Każdy z 768 obiektów zbioru opisany jest przy pomocy 8 cech zawierających następujące informacje: ile razy pacjentka była w ciąży, test tolerancji glukozy, ciśnienie rozkurczowe, grubość zagięcia skóry, poziom insuliny, masę ciała, czy ktoś w rodzinie był chory na cukrzycę oraz wiek pacjentki. Każdy z obiektów przynależy do jednej z dwóch klas. Pierwsza klasa oznacza, że pacjentka nie choruje na cukrzycę, a druga klasa oznacza, że dana kobieta jest diabetykiem.

- Liczba atrybutów: 8.
- Rodzaj atrybutów: realistyczne, ciągłe i typu Intiger (wiek i ilość dotychczasowych ciąży).
- Liczba instancji: 768.
- Liczba klas: 2 - wartość 1 (pozytywna) lub 0 (negatywna).



Rys. 1.7: Rozłożenie ilościowe poszczególnych klas zbioru Diabetes



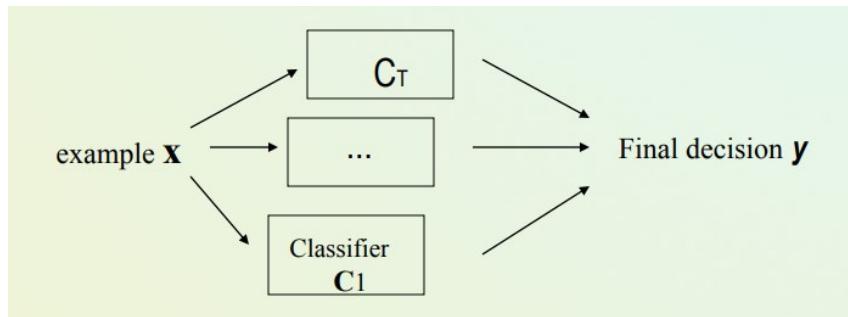
Rys. 1.8: Zestawienie zależności atrybutów i klas zbioru Diabetes

Rozdział 2

Wstęp teoretyczny

2.1. Zespoły klasyfikatorów

Zespoły klasyfikatorów (ang. *multiple classifiers* lub *ensemble methods*) to zbiór indywidualnych klasyfikatorów, których predykcje agregowane są do jednej decyzji w celu polepszenia zdolności predykcyjnej. Głównym problemem w tej kategorii są pytania typu: jak budować klasyfikatory składowe oraz jak agregować odpowiedzi. Istnieje wiele rozwiązań tych problemów, najważniejsze z nich to: Bagging, Boosting oraz Random Forest, które zostaną omówione poniżej.

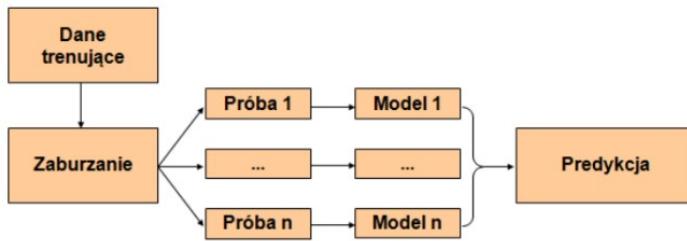


Rys. 2.1: Schemat działania zespołu klasyfikatorów

2.2. Metoda *Bagging* - Algorytm *BaggingClassifier*

W przebiegu pierwszej z omawianych technik grupowych: bagging, której nazwa pochodzi od angielskiego sformułowania bootstrap aggregating, generowana jest rodzina zaburzonych zbiorów trenujących T_1, \dots, T_N , z których każdy jest wynikiem losowania przykładów ze zwracaniem z pierwotnego zbioru trenującego T . W ten sposób, w każdym z zaburzonych zbiorów trenujących pewne przykłady mogą występować wielokrotnie, a inne nie pojawić się wcale. Liczność każdego z zaburzonych zbiorów może być taka sama, jak zbioru pierwotnego lub inna, co zależy od rodzaju bazowego algorytmu klasyfikacji oraz rozmiaru dostępnych danych treningowych. Następnie, w oparciu o i-ty zaburzony zbiór budowany jest osobny klasyfikator bazowy hi. Aby wyznaczyć predykcję otrzymanego w ten sposób klasyfikatora złożonego, należy najpierw dla danego przykładu x wyznaczyć klasy przewidywane przez poszczególne klasyfikatory

bazowe, a następnie wybrać najczęściej pojawiającą się klasę. Tak otrzymana etykieta jest więc wynikiem głosowania klasyfikatorów bazowych. Schemat techniki bagging przedstawiony został na rys. 2.2



Rys. 2.2: Schemat techniki Bagging

Wyniki eksperymentalne przedstawione w licznej literaturze wykazują, że otrzymane w ten sposób predykcje są dla większości zbiorów danych dokładniejsze niż predykcje pojedynczych klasyfikatorów. Dodatkowo, w przypadku danych o dużych rozmiarach wybór grupy niewielkich prób losowych może znacząco ograniczyć czas budowy klasyfikatora względem wykorzystania całości dostępnych danych w charakterze zbioru trenującego. Do ograniczeń techniki bagging zaliczyć można fakt, że każdy z klasyfikatorów bazowych jest równouprawniony - nawet pomimo ewentualnych znaczących różnic w ich indywidualnej dokładności. Dodatkowo każdy z przykładów trenujących także traktowany jest jednakowo, pomimo że niektóre z nich sprawiać mogą większą trudność w prawidłowym zaklasyfikowaniu. Zaletą tej metody jest bardzo łatwe zrównoleglanie, która to cecha wynika z faktu, że modele bazowe mogą zostać budowane w sposób całkowicie niezależny.

2.2.1. Badane parametry algorytmu *BaggingClassifier*

- **n estimators** - (domyślana wartość: 10) - Liczba podstawowych estymatorów w zespole.
- **max samples** - (domyślana wartość: 1) - Liczba próbek do pobrania z X w celu wyszkolenia każdego estymatora bazowego.
- **max features** - (domyślana wartość: 1) - Liczba cech do wyciągnięcia z X, aby wyszkolić każdy estymator bazowy.
- **bootstrap** - (domyślana wartość: True) - Czy próbki są pobierane ze zwracaniem.
- **bootstrap features** - (domyślana wartość: False) - Czy cechy są pobierane ze zwracaniem.

2.3. Metoda *Boosting* - Algorytm *AdaBoost*

Boosting jest ogólną metodą służącą zwiększeniu skuteczności dowolnego algorytmu uczenia. Idea Budowanie “mocnego i złożonego klasyfikatora” ze “słabych i prostych klasyfikatorów”. Metoda AdaBoost została opracowana przez Yoava Freunda i Roberta Schapire'a. Nazwa wzięła się od *Adaptive Boosting* (z ang. wzmacnianie adaptacyjne). AdaBoost w kolejnych t iteracjach trenuje t „słabych” klasyfikatorów na zbiorze przykładów D ze zmienianymi wagami. Co

należy rozumieć poprzez określenie słaby "klasyfikator? „Słabym” klasyfikatorem nazywamy klasyfikator stosunkowo prosty, o niezbyt dużej sile wyrażania, potrafiący klasyfikować dane testowe ze skutecznością większą niż 50%.

- Algorytm na wejściu otrzymuje zbiór treningowy $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$, gdzie każdy x_i należy do pewnej dziedziny problemu X , natomiast każda etykieta (decyzja) y_i należy do pewnego zbioru Y . Dla ułatwienia będziemy na razie zakładać, że $Y = \{-1, +1\}$.
- AdaBoost (Adaptive Boosting) wywołuje wybrany “słaby” algorytm uczący w serii T iteracji. Zakładamy, że błąd uzyskiwanych klasyfikatorów na zbiorze treningowym jest mniejszy niż $\frac{1}{2}$.
- Jedną z głównych idei algorytmu jest strojenie rozkładu (lub wag elementów) dla zbioru treningowego. Wagę i -tego elementu ze zbioru treningowego w iteracji t będziemy oznaczali przez $D_t(i)$.
- Początkowo wszystkie wagi są ustawione na równe wartości.
- Po każdej iteracji, wagi elementów źle klasyfikowanych są zwiększane. Dzięki temu mamy możliwość skierowania uwagi “słabego” klasyfikatora na pewne elementy (trudne do wyuczenia) ze zbioru treningowego.
- Zadaniem “słabego” algorytmu uczącego jest zbudowanie klasyfikatora (ang. hypothesis) $h_t : X \rightarrow Y$ odpowiedniego dla aktualnego rozkładu D_t .
- Skuteczność takiego klasyfikatora jest mierzona przez jego błąd (z uwzględnieniem rozkładu D_t).
- W praktyce “słaby” algorytm uczący może być algorytmem, który uwzględnia rozkład D_t . Nie jest to jednak konieczne. Kiedy algorytm nie pozwala na bezpośrednie uwzględnienie rozkładu D_t , losuje się (względem D_t) podzbiór zbioru treningowego, na którym następnie wywołuje się algorytm uczący.
- Kiedy AdaBoost dostaje klasyfikator h_t dobierany jest parametr α_t . Intuicyjnie α_t odpowiada za wagę jaką przykładamy do klasyfikatora h_t . Zauważmy, że $\alpha_t \geq 0$ gdy $\varepsilon_t \leq \frac{1}{2}$. Ponadto α_t rośnie kiedy ε_t maleje.
- Rozkład D_t jest następnie zmieniany tak, aby zwiększyć (zmniejszyć) wagi elementów zbioru treningowego, które są źle (dobrze) klasyfikowane przez h_t . Stąd wagi mają tendencję do skupiania się na “trudnych” przykładach.
- Wynikowy klasyfikator powstaje za pomocą ważonego głosowania klasyfikatorów h_t , gdzie α_t jest wagą przypisaną klasyfikatorowi h_t .

2.3.1. Badane parametry algorytmu *AdaBoostClassifier*

- **n estimators** - (domyślna wartość: 50) - Maksymalna liczba estymatorów, przy których boosting jest zakańczany. W przypadku idealnego dopasowania procedura uczenia zostaje zatrzymana wcześniej.
- **learning rate** - (domyślna wartość: 1.0) - zmniejsza wkład każdego klasyfikatora o podaną wartość.

- **algorithm** - (domyślna wartość: 'SAMME.R') - algorytm 'SAMME.R' użyty zostanie algorytmu boostingu. Klasyfikator podany jako base estimator musi obsługiwać obliczanie prawdopodobieństwa klas. Algorytm 'SAMME' używa metody SAMME Discrete Boosting. Algorytm SAMME.R zazwyczaj zbiega się szybciej niż SAMME, uzyskując niższy błąd testowy z mniejszą liczbą iteracji

2.4. Metoda *RandomForest*

To metoda klasyfikacji (i regresji) polegająca na tworzeniu wielu drzew decyzyjnych na podstawie losowego zestawu danych. Idea tego algorytmu polega na zbudowaniu konsylium ekspertów z losowych drzew decyzyjnych, gdzie w odróżnieniu od klasycznych drzew decyzji, losowe drzewa budowane są na zasadzie, iż podzbiór analizowanych cech w węźle dobierany jest losowo. Ponadto, poszczególne drzewa z losowych lasów drzew budowane są zgodnie z koncepcją Bagging.

Cechy Algorytmu Random Forest:

- wielu przypadkach jest najlepszy jeśli chodzi o dokładność wśród pozostałych algorytmów
- działa skutecznie na dużych bazach danych
- utrzymuje dokładność w przypadku braku danych
- daje oszacowanie, które zmienne są istotne w klasyfikacji
- nie ma potrzeby przycinania drzew
- lasy mogą być zapisane i wykorzystane w przyszłości dla innego zbioru danych
- nie wymaga wiedzy eksperckiej
- nie jest podatny na overfitting

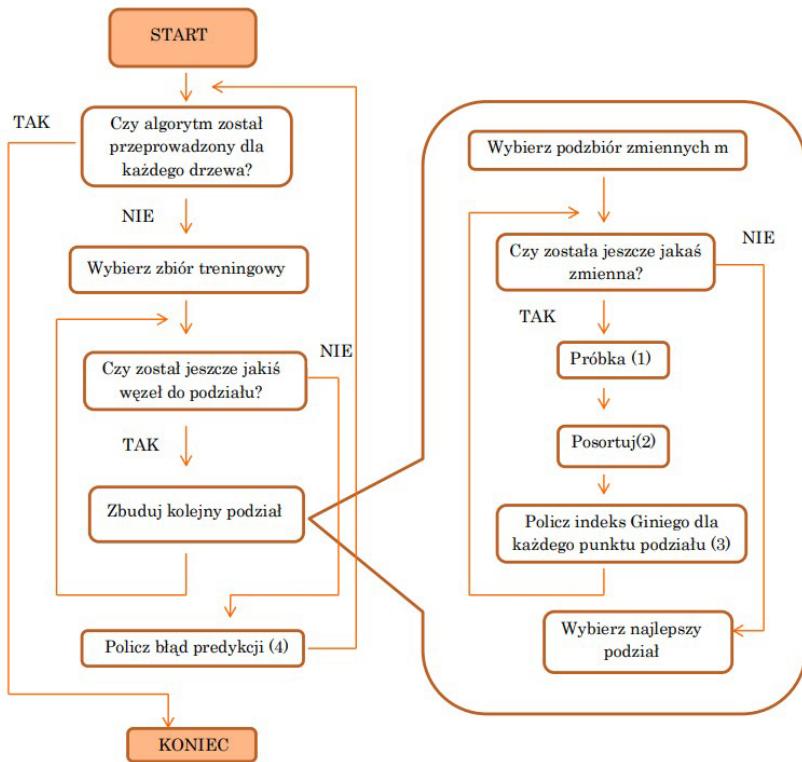
Sposób działania:

- Losujemy ze zwracaniem z n -elementowej próby uczącej n wektorów obserwacji. Na podstawie takiej pseudopróby stworzone zostanie drzewo.
- W każdym węźle podział odbywa się poprzez wylosowanie bez zwracania m spośród p atrybutów, następnie w kolejnym węźle k spośród m atrybutów itd ($p \geq m \geq k$) (parametr m jest jedynym elementem algorytmu, który trzeba ustalić, wartość dająca dobre wyniki dla modeli decyzyjnych to około $m = \sqrt{p}$, dla modeli regresyjnych $\frac{m}{3}$).
- Proces budowania drzewa bez przycinania trwa, jeżeli to możliwe do momentu uzyskania w liściach elementów z tylko jednej klasy.

Proces klasyfikacji

- Dany wektor obserwacji jest klasyfikowany przez wszystkie drzewa, ostatecznie zaklasyfikowany do klasy, w której wystąpił najczęściej.
- W przypadku elementów niewylosowanych z oryginalnej podpróby, każdy taki i-ty element zostaje poddany klasyfikacji przez drzewa, w których budowie nie brał udziału. Taki element

zostaje następnie przyporządkowany klasie, która osiągana była najczęściej (w ten sposób zaklasyfikowane zostały wszystkie elementy z oryginalnej próby).



Rys. 2.3: Schemat działania algorytmu *RandomForest*

2.4.1. Badane parametry algorytmu *RandomForest*

- **n estimators** - (domyślna wartość: 10) - Liczba podstawowych estymatorów w zespole.
- **criterion** - (domyślana wartość: 'gini') - Funkcja pomiaru jakości podziału. Obsługiwane kryteria to "gini" dla wskaźnika Gini impurity i "entropy" dla zysku informacyjnego. Uwaga: parametr ten jest specyficzny dla drzewa.
- **min samples split** - (domyślana wartość: 2) - Minimalna liczba próbek potrzebnych do podziału węzła wewnętrznego. Jeśli podana jest wartość int, należy rozważyć min samples split jako minimalną liczbę próbek.

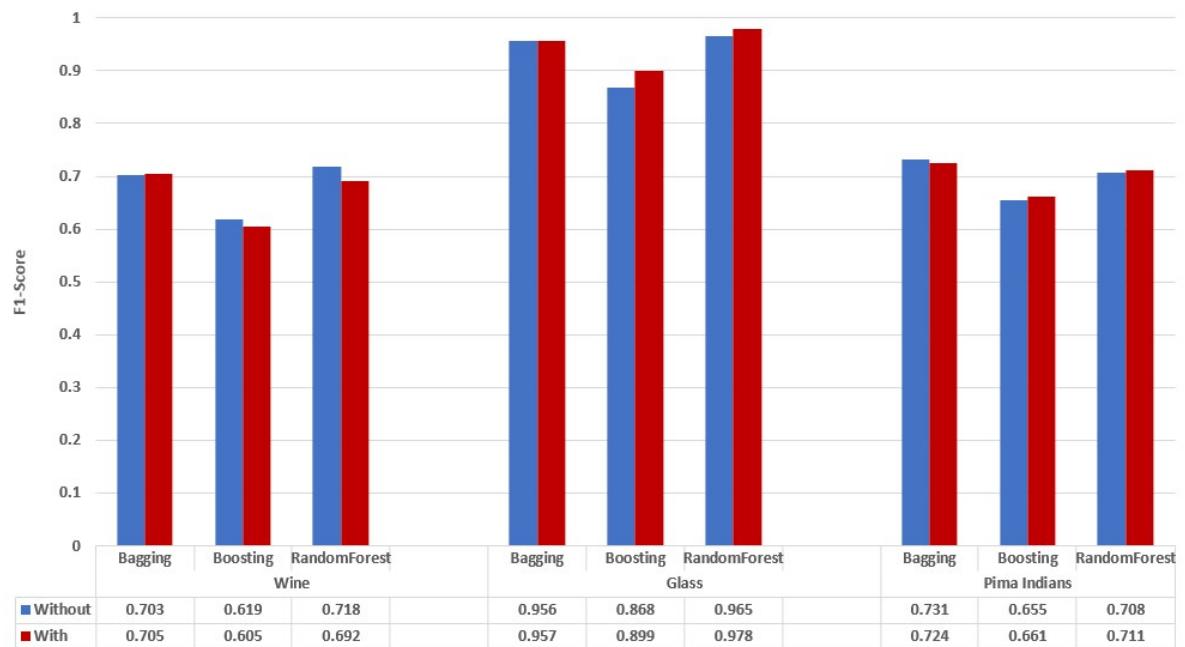
2.5. Miary jakości klasyfikatora

- **Trafność** (*ang. Accuracy*) - określa, jaka część prognozowanych etykiet jest zgodna z rzeczywistymi wynikami. Oznacza to procent poprawnie sklasyfikowanych etykiet.
- **Precyzja** (*ang. Precision*) - określa liczbę adekwatnych elementów w zbiorze wyników. W kontekście klasyfikacji jest to liczba poprawnych etykiet z wszystkich zbiorów klasyfikowanych etykiet. Wyniki są uśredniane dla wszystkich etykiet.
- **Czułość** (*ang. Recall*) - określa liczbę poprawnych wyników względem liczby wszystkich poprawnych etykiet. W kontekście klasyfikacji jest to liczba poprawnie sklasyfikowanych etykiet w zbiorze podzielona przez łączną liczbę etykiet ze zbioru. Wyniki są uśredniane.
- **Wskaźnik F1** (*ang. F1 Score*) - jest to średnia harmoniczna precyzji i czułości. Najczęściej stosowana jest dla niezrównoważonych zbiorów danych w celu ustalenia, czy klasyfikator działa dobrze dla wszystkich klas.

Rozdział 3

Badania i analiza wyników

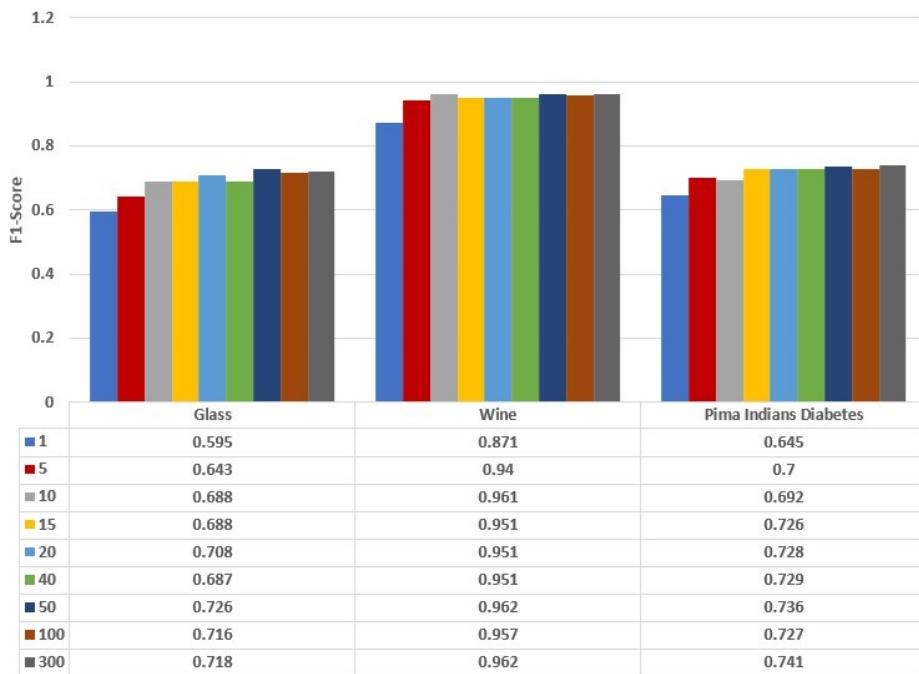
3.1. Normalizacja



Rys. 3.1: Porównanie wyników osiągniętych przy oraz bez zastosowania normalizacji

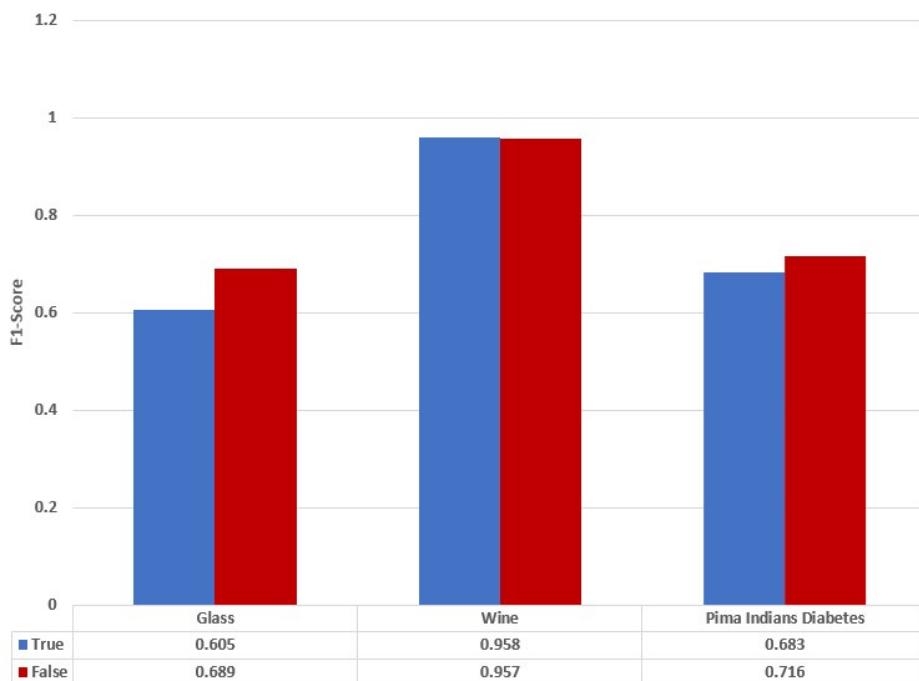
3.2. Bagging

3.2.1. Parametr n-estimators



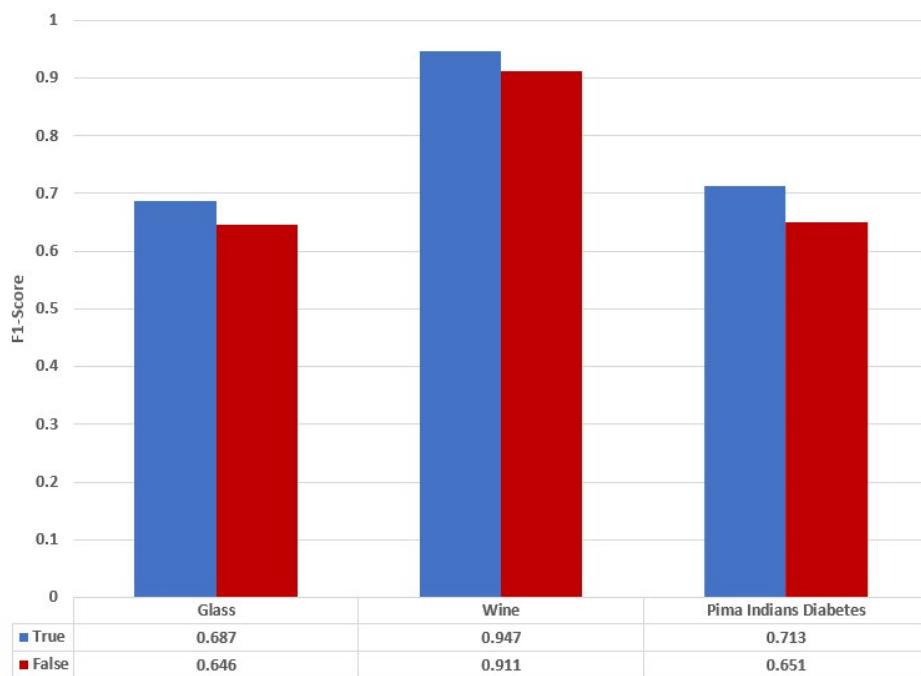
Rys. 3.2: Bagging - Wyniki pomiarów parametru N-Estimators

3.2.2. Parametr bootstrap-features



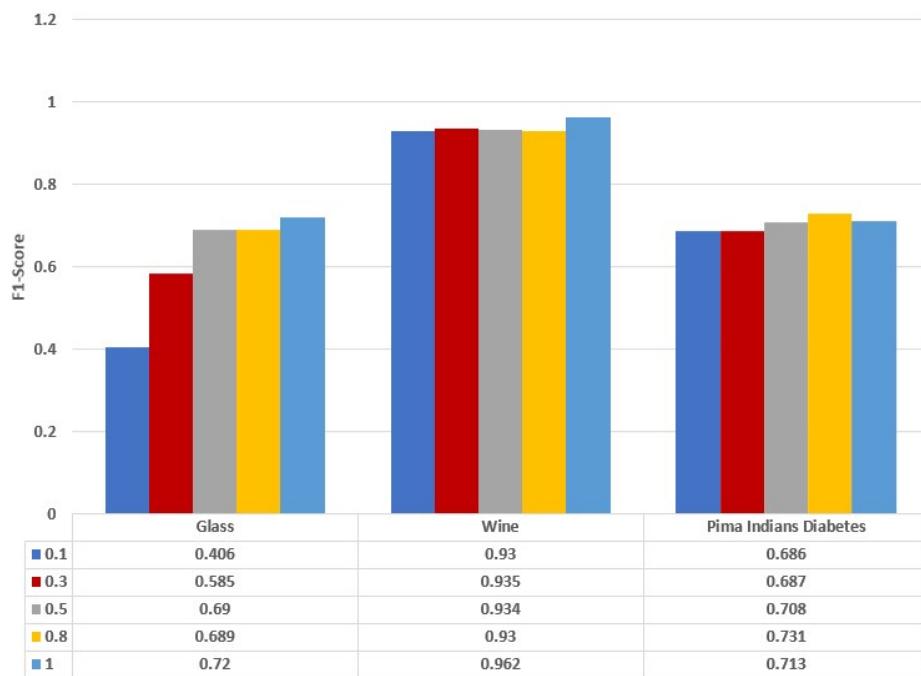
Rys. 3.3: Bagging - Wyniki pomiarów parametru Bootstrap-Features

3.2.3. Parametr bootstrap



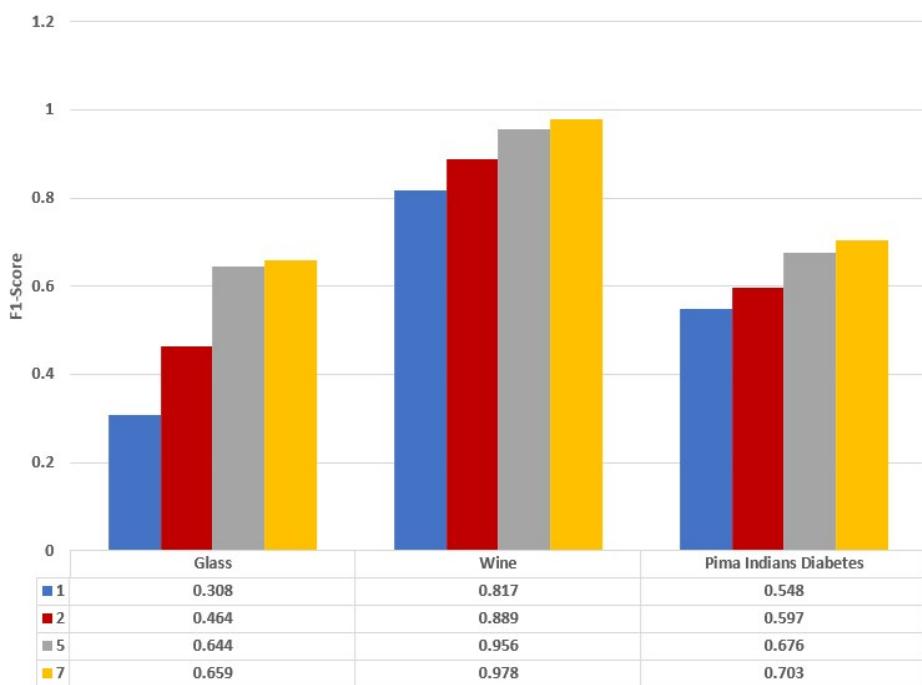
Rys. 3.4: Bagging - Wyniki pomiarów parametru Bootstrap

3.2.4. Parametr max-samples



Rys. 3.5: Bagging - Wyniki pomiarów parametru Max-Samples

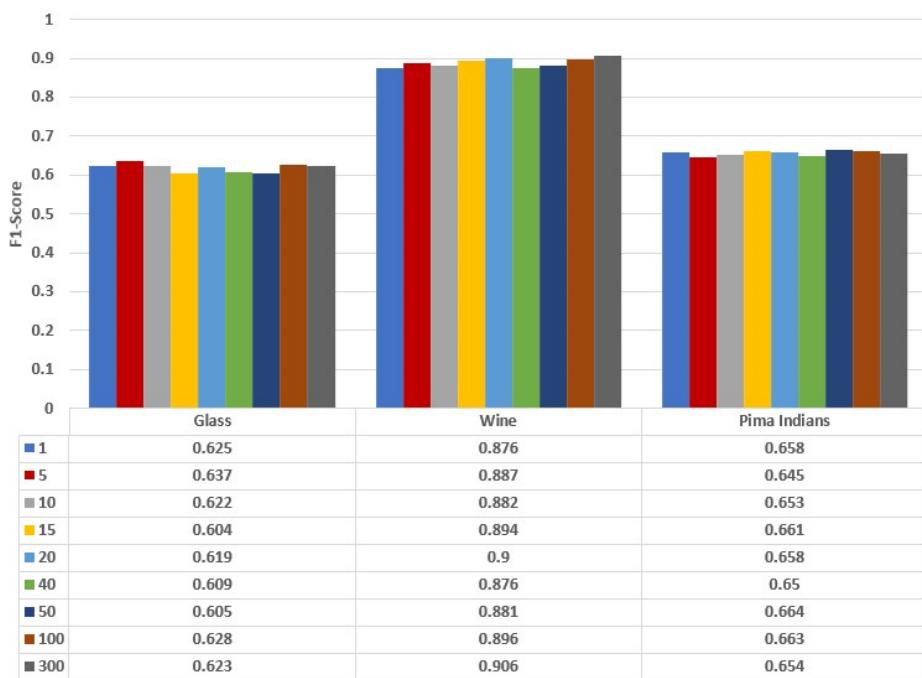
3.2.5. Parametr max-features



Rys. 3.6: Bagging - Wyniki pomiarów parametru Max-Features

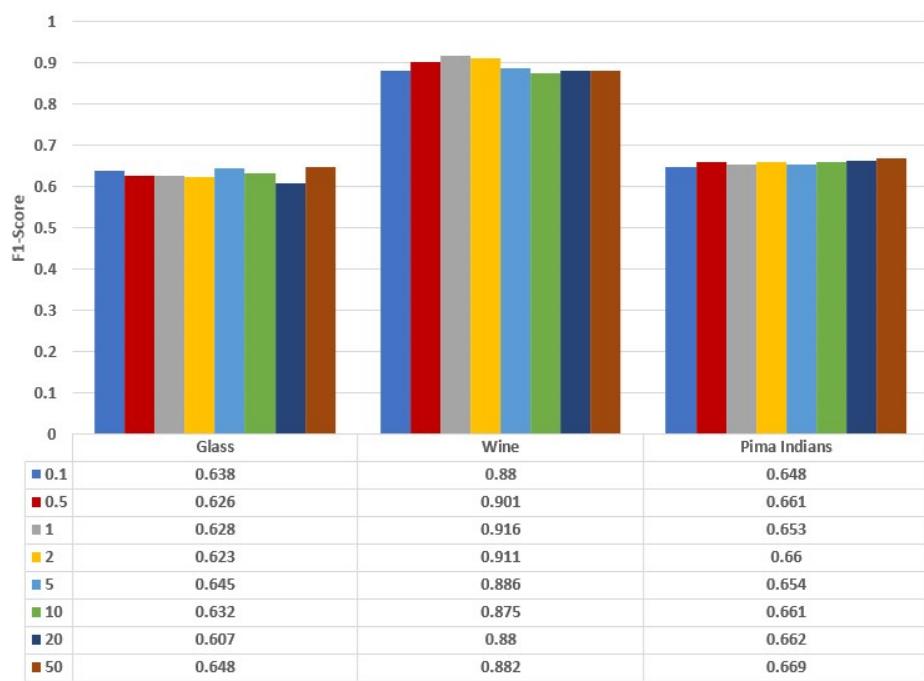
3.3. Boosting

3.3.1. Parametr n-estimators



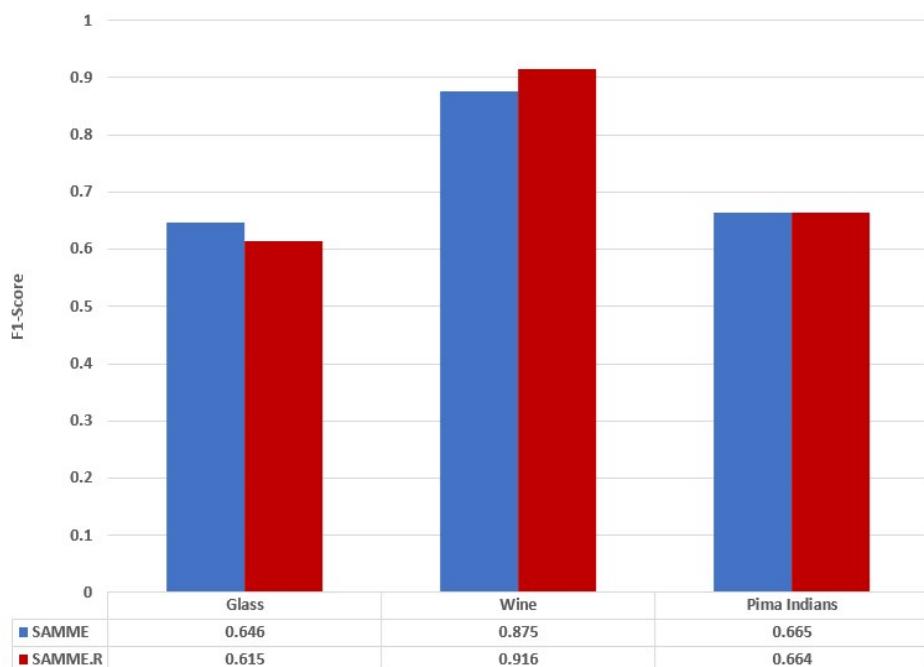
Rys. 3.7: Boosting - Wyniki pomiarów parametru N-Estimators

3.3.2. Parametr learning-rate



Rys. 3.8: Boosting - Wyniki pomiarów parametru Learning-Rate

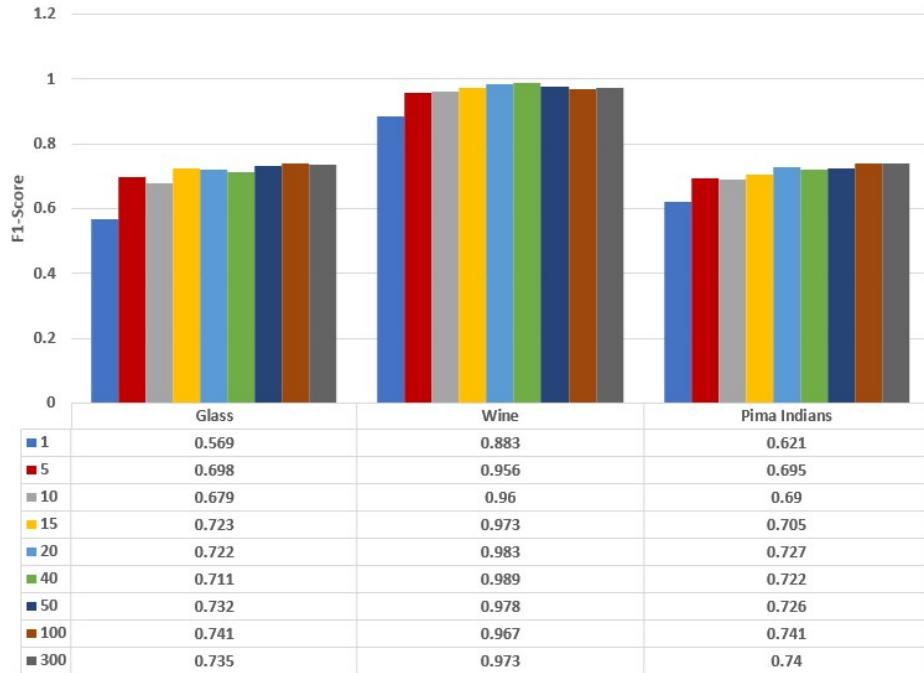
3.3.3. Parametr algorithm



Rys. 3.9: Boosting - Wyniki pomiarów parametru Algorithm

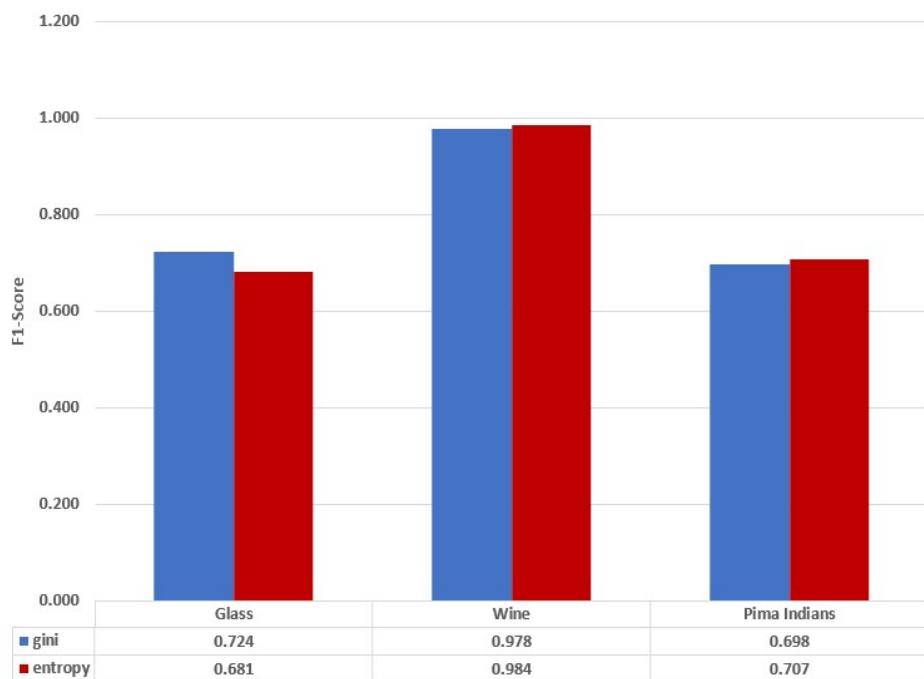
3.4. RandomForest

3.4.1. Parametr n-estimators



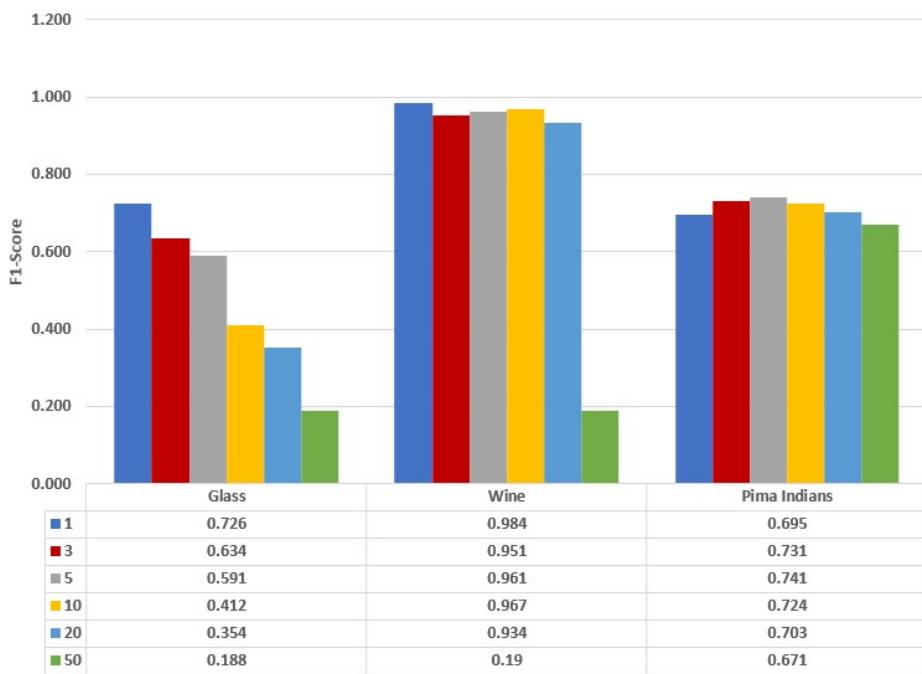
Rys. 3.10: RandomForest - Wyniki pomiarów parametru N-Estimators

3.4.2. Parametr criterion



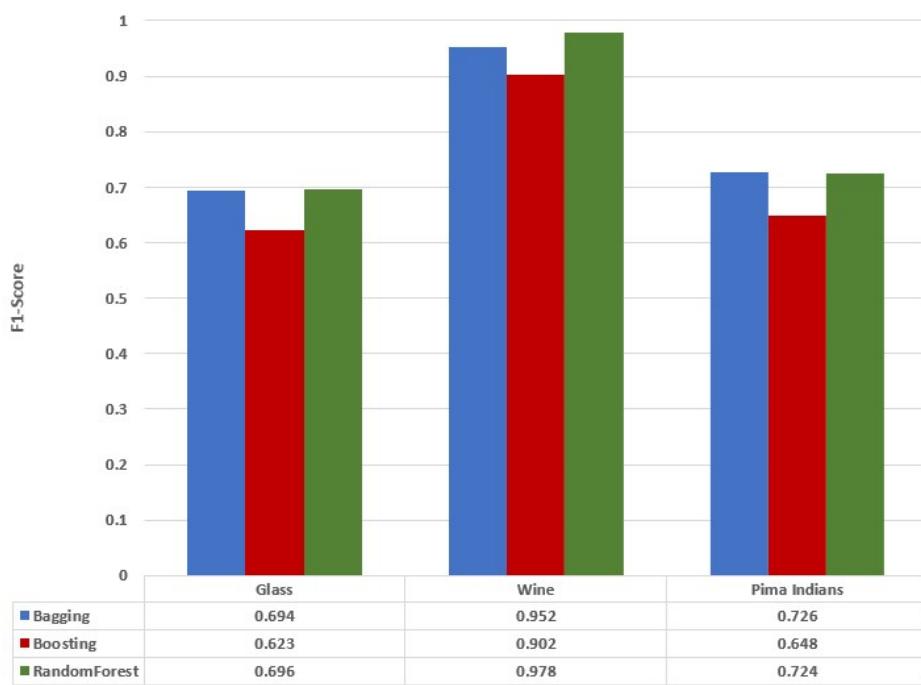
Rys. 3.11: RandomForest - Wyniki pomiarów parametru Criterion

3.4.3. Parametr min-samples-leaf



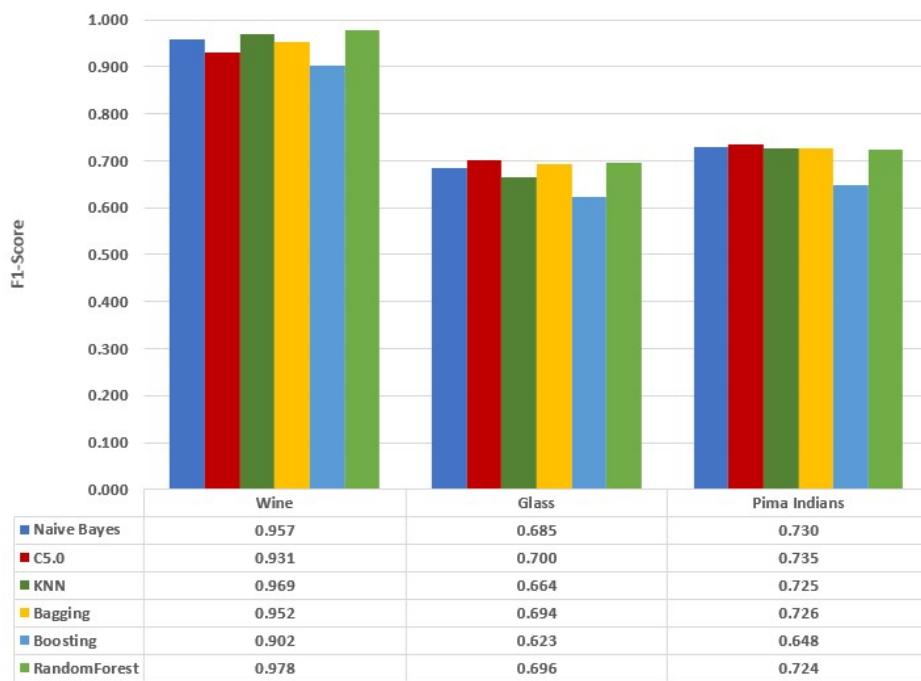
Rys. 3.12: RandomForest - Wyniki pomiarów parametru Min-Samples-Leaf

3.5. Najlepsze uzyskane rezultaty



Rys. 3.13: Porównanie wyników osiągniętych przez algorytmy

3.6. Porównanie wszystkich przebadanych algorytmów



Rys. 3.14: Porównanie wszystkich do tej pory przebadanych algorytmów

Spis rysunków

1.1.	Rozłożenie ilościowe poszczególnych klas zbioru Iris	4
1.2.	Zestawienie zależności atrybutów i klas zbioru Iris	5
1.3.	Rozłożenie ilościowe poszczególnych klas zbioru Wine	6
1.4.	Zestawienie zależności atrybutów i klas zbioru Wine	6
1.5.	Rozłożenie ilościowe poszczególnych klas zbioru Glass	7
1.6.	Zestawienie zależności atrybutów i klas zbioru Glass	7
1.7.	Rozłożenie ilościowe poszczególnych klas zbioru Diabetes	8
1.8.	Zestawienie zależności atrybutów i klas zbioru Diabetes	9
2.1.	Schemat działania zespołu klasyfikatorów	10
2.2.	Schemat techniki Bagging	11
2.3.	Schemat działania algorytmu RandomForest	14
3.1.	Porównanie wyników osiągniętych przy oraz bez zastosowania normalizacji	16
3.2.	Bagging - Wyniki pomiarów parametru N-Estimators	17
3.3.	Bagging - Wyniki pomiarów parametru Bootstrap-Features	17
3.4.	Bagging - Wyniki pomiarów parametru Bootstrap	18
3.5.	Bagging - Wyniki pomiarów parametru Max-Samples	18
3.6.	Bagging - Wyniki pomiarów parametru Max-Features	19
3.7.	Boosting - Wyniki pomiarów parametru N-Estimators	19
3.8.	Boosting - Wyniki pomiarów parametru Learning-Rate	20
3.9.	Boosting - Wyniki pomiarów parametru Algorithm	20
3.10.	RandomForest - Wyniki pomiarów parametru N-Estimators	21
3.11.	RandomForest - Wyniki pomiarów parametru Criterion	21
3.12.	RandomForest - Wyniki pomiarów parametru Min-Samples-Leaf	22
3.13.	Porównanie wyników osiągniętych przez algorytmy	23
3.14.	Porównanie wszystkich do tej pory przebadanych algorytmów	23