PROJEKT PUMA

Dokumentacja projektu zaliczeniowego z przedmiotu PUMA

Politechnika Śląska

Wiktor Hosumbek
Szymon Joszko

Spis treści

1. Dane

1.1 Zbiór danych:

Zbiór danych użytych w projekcie pochodzi ze strony UCI dokładny link:
https://archive.ics.uci.edu/ml/datasets/Wine+Quality
"P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009."

1.2 Opis zbioru danych:

Zbiór ten dotyczy jakości portugalskiego wina "Vinho Verde" i jest podzielony na dwa podzbiory, jeden dla czerwonego wina a drugi dla białego. Dane wejściowe zawierają w sobie tylko dane psychochemiczne takie jak ph, zawartość alkoholu, A nie zawierają informacji na temat marki, ceny, typu winogron itd. jest to spowodowane ochroną prywatności wytwórni tych win. Dane są prawdziwe i nie posiadają brakujących argumentów. Daną wyjściową jest ocena w skali rosnącej (0-10) im wyższa, tym lepsza jakość wina. Dane te nie są zbalansowane co oznacza że istnieje w bazie znaczna większość win średnich niż win bardzo słabych czy wybitnie dobrych.

1.3 Informacje o atrybutach:

Baza danych posiada 4898 rekordów wina białego i 1599 rekordów wina czerwonego co daje łącznie 6497 rekordów

Dane wejściowe:

1 - kwasowość stała
2 - kwasowość lotna
3 - kwas cytrynowy
4 - cukier resztkowy
5 - chlorki
6 - wolny dwutlenek siarki
7 - całkowity dwutlenek siarki
8 - gęstość
9 - pH
10 - siarczany
11 - alkohol

Dane wyjściowe:

12 - Jakość

2. Opis projektu

2.1 Cel projektu:

Jednym z celów projektu było stworzenie programu wykorzystującego różne metody uczenia maszynowego do klasyfikacji jakości “vihno verde” na podstawie danych psycho chemicznych, drugim celem było porównanie jakości klasyfikacji tych metod.

2.2 Metody:

Metoda	opis
svm_method (data):	metoda SVM (C-Support Vector Classification)
pjk_method (data):	metoda (Regresja logistyczna, las losowy)
dt_method (data):	metoda Decision Tree
gaussian_method (data):	metoda Gaussian Naive Bayes (GaussianNB)
split_data_complex (data):	metoda dzieląca dane na zbiór treningowy i testowy z zachowaniem oryginalnych etykiet
split_data(data):	metoda dzieląca dane na zbiór treningowy i testowy z podziałem etykiety label na 3 klasy
qualityclass(x):	metoda służąca do wyznaczenia klasy quality

2.3 Biblioteki:

sklearn - biblioteka zawierająca implementację wszystkiego co potrzebne do pracy z uczeniem maszynowym w Pythonie

Numpy -podstawowy zestaw narzędzi dla języka Python umożliwiający zaawansowane obliczenia matematyczne.

matplotlib - biblioteka do tworzenia wykresów dla języka programowania Python.

* Kod programu znajduje się w pliku projektPUMA.py

2.4 Podział etykiety quality:

W zbiorze danych użytym w projekcie daną wyjściową jest ocena jakości wina w skali 0-10 (0 - bardzo słabe) (10 - bardzo dobre). Aby poprawić jakość klasyfikacji dane zostały poddane obróbce wstępnej w wyniku której została im przydzielona nowa klasa, odpowiednio:

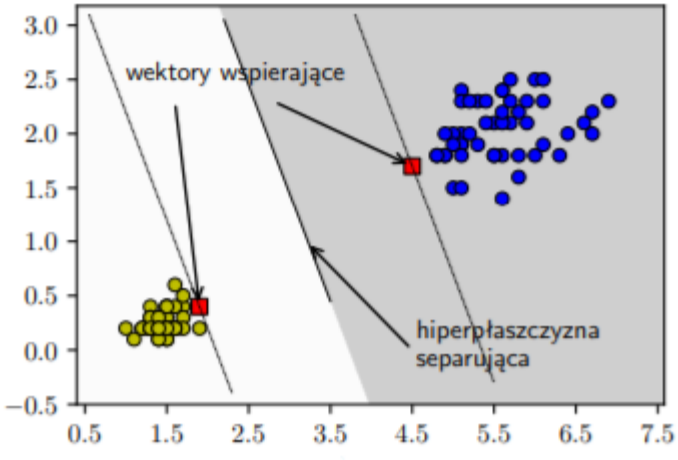
0 - 6 → “low quality”
7 - 8 → “medium quality’
9 - 10 → “high quality”

Poprawiło to znacznie jakość klasyfikacji, jednocześnie nie zmieniając celu projektu, wina wciąż są oceniane jako dobre czy złe.

3. Opis metod

3.1 Metoda svm - Maszyna Wektorów wspierających (Support Vector Machine) ma na celu znalezienie takiej prostej(hiperpłaszczyzny separującej), która oddziela przykłady ze zbioru treningowego z maksymalnym marginesem.
Granice marginesu to hiperpłaszczyzny, które odpowiadają skrajnym przypadkiem ze zbioru treningowego. Oznacza to, że wszystkie przypadki z klasy *+1* powinny spełniać warunek *g(x) > 0* (bo są separowalne liniowo) oraz *g(x) >= 1* (bo należą do klasy 1). Podobnie dla klasy *−1*. Przypadki, dla których zachodzi *g(x) = 1* oraz *g(x) = −1* to **wektory wspierające lub wektory nośne.**
Twardy margines - Wewnątrz twardego marginesu nie mogą się znaleźć przypadki z żadnej z tych klas!
Miękki margines - dopuszczamy pojawienie się błędów klasyfikacji.
Danie nie zawsze są liniowo separowalne, czasem trzeba je zmapować do innej przestrzeni stosując tzw **“kernel trick”**

Przykład zastosowania:



Parametry:

C - parametr regularyzacji służący do sterowania algorytmu im wyższa wartość tym mniejsza ilość błędów lecz mniejsza generalizacja
kernel - wybór jądra algorytmu np. “poly” , “rbf”.
gamma - współczynnik jądra dla rbf.
degree - Stopień wielomianowej funkcji jądra („poly”).

Metody znajdowania najlepszych parametrów:

GridSearchCV() - przeszukuje podane parametry jeden po drugim

RandomizedSearchCV() - przeszukuje podane parametry losowo

obie metody służą do znajdowania najlepszych parametrów pracy algorytmu.

Wzory:

$$\min_{w} \frac{\|w\|^2}{2} + C \sum_{i=1}^{m} \zeta_i,$$

gdzie:

$$y_i(w \cdot x_i + b) \geqslant 1 - \zeta_i$$

$$\zeta_i \geqslant 0.$$

ζ - zmienne osłabiające

Uwagi:

Do uczenia klasyfikowania do wszystkich klas użyliśmy parametrów:

```
parameters = {'kernel': ('linear', 'rbf'),
              'C': [2 ** -2, 2 ** 2],
              'gamma': [2 ** -2, 2 ** 2],
              'degree': [1, 2, 3, 4]}
```

Do uczenia klasyfikowania do trzech klas użyliśmy parametrów:

```
parameters = {'kernel': ('linear', 'rbf'),
              'C': [2 ** -6, 2 ** 6],
              'gamma': [2 ** -6, 2 ** 6],
              'degree': [1, 2, 3, 4, 5, 6, 7, 8]}
```

Dodatkowo dla zbioru wina czerwonego użyliśmy poszerzonego zakresu C i gamma do [2**-4,2**4] ale nie przyniosło to lepszych wyników

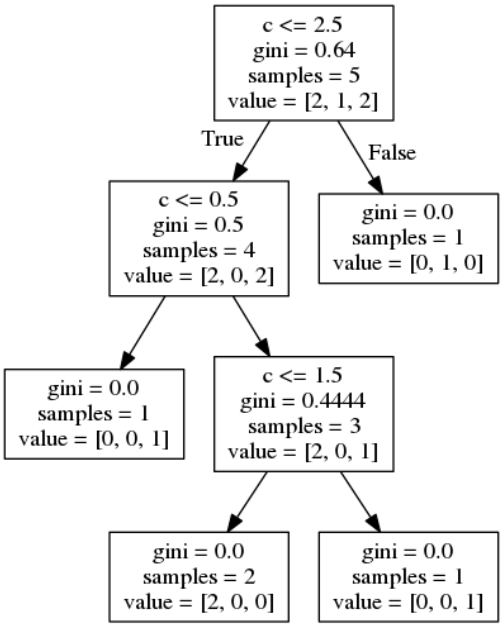
W początkowej fazie projektu, doboru metod i parametrów, szukaliśmy najlepszych parametrów używając również metody poly. Zrezygnowaliśmy z tej metody ponieważ czas i możliwości obliczeniowe, którymi dysponowaliśmy były zbyt małe do wyszukiwania najlepszych parametrów uwzględniając również tą metodę, a wyniki jakie dawała były podobne do metody liniowej.

* Wszystkie wyniki dostępne w załącznikach opisanych na końcu dokumentacji

3.2 Drzewo decyzyjne:

Drzewo decyzyjne to nieparametryczna metoda uczenia maszynowego nadzorowanego stosowana do klasyfikacji i regresji. Jej celem stworzenie jest modelu przewidującego wartość docelową poprzez utworzenie reguł decyzyjnych na podstawie cech danej próbki.

Przykładowe drzewo:



opis:

1. Wiedza jest reprezentowana w postaci drzewa.
2. Węzły drzewa określają sposób podziału przestrzeni cech na obszary/klasy.
3. Liście drzewa określają klasę, do której należy klasyfikowany obiekt.
4. Proces klasyfikacji polega na przejściu od korzenia drzewa do liści.

3.3 Naiwny Klasyfikator Bayesowski:

Naiwny klasyfikator bayesowski - jest prostym probabilistycznym klasyfikatorem zakłada on wzajemną niezależność zmiennych niezależnych (naiwność).Nazywany też jako „model cech niezależnych”. Model prawdopodobieństwa można wyprowadzić korzystając z twierdzenia Bayesa.

Twierdzenie bayesa:

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

gdzie A i B są zdarzeniami oraz $P(B) > 0$, przy czym

$P(A \mid B)$ oznacza prawdopodobieństwo warunkowe, tj. prawdopodobieństwo zajścia zdarzenia A, o ile zajdzie zdarzenie B.

$P(B \mid A)$ oznacza prawdopodobieństwo zajścia zdarzenia B,o ile zajdzie zdarzenie A

Opis:

1. Twierdzenie Bayesa określa prawdopodobieństwa warunkowe dwóch zdarzeń warunkujących się wzajemnie.
2. Wyliczane prawdopodobieństwo to prawdopodobieństwo a posteriori.
3. Wnioskowanie bayesowskie polega na sekwencyjnym wykorzystaniu reguły Bayesa.
4. Wnioskowanie bayesowskie pozwala na aktualizację prawdopodobieństw, które mogą służyć do aktualizacji prawdopodobieństw zajścia zdarzeń z nimi współzależnych

3.4 Regresja logistyczna - metoda do szacowania prawdopodobieństwa przynależności przykładu do określonej klasy. Jest to klasyfikator binarny czyli jeśli prawdopodobieństwo przekracza 50% to próbka należy do klasy pozytywnej i w odwrotnym przypadku do negatywnej

Funkcja logistyczna:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

3.5 Lasy losowe - metoda zespołowego uczenia maszynowego, która polega na konstruowaniu wielu drzew decyzyjnych w czasie uczenia i generowaniu klasy, która jest dominantą Losowe lasy decyzyjne poprawiają tendencję drzew decyzyjnych do nadmiernego dopasowywania się do zestawu treningowego.

4. Wyniki

- wyniki dokładności klasyfikacji dla 10 klas wina czerwonego

	Decision Tree	gaussianNB	SVM	Random Forest	Logistic Regression
zb. treningowy	1.0	0.5460232350312779	0.5915996425379804	0.683646	0.589812
zb. testowy	0.5770833333333333	0.5479166666666667	0.6041666666666666	0.652083	0.620833

- wyniki dokładności klasyfikacji dla 10 klas wina białego

	Decision Tree	gaussianNB	SVM	Random Forest	Logistic Regression
zb. treningowy	1.0	0.4582847141190198 3	0.9967911318553092	0.580222	0.542299
zb. testowy	0.5727891156462585	0.4489795918367347	0.5850340136054422	0.527211	0.521769

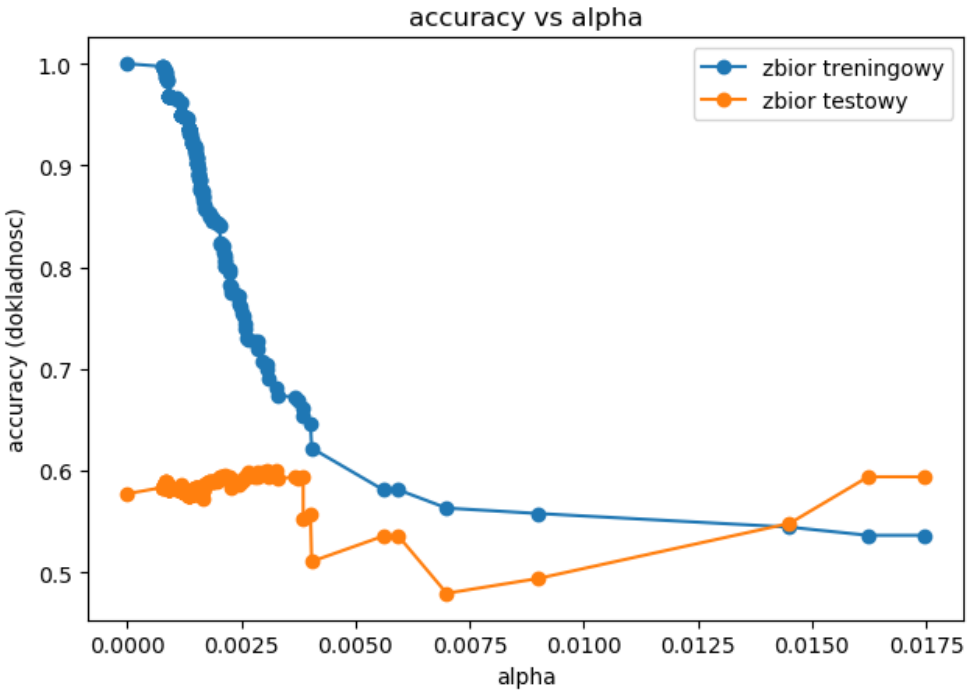
- wyniki dokładności klasyfikacji dla 3 klas wina czerwonego

	Decision Tree	gaussianNB	SVM	Random Forest	Logistic Regression
zb. treningowy	1.0	1.0	1.0	0.999106	0.993744
zb. testowy	0.9958333333333333	1.0	1.0	0.989583	0.991667

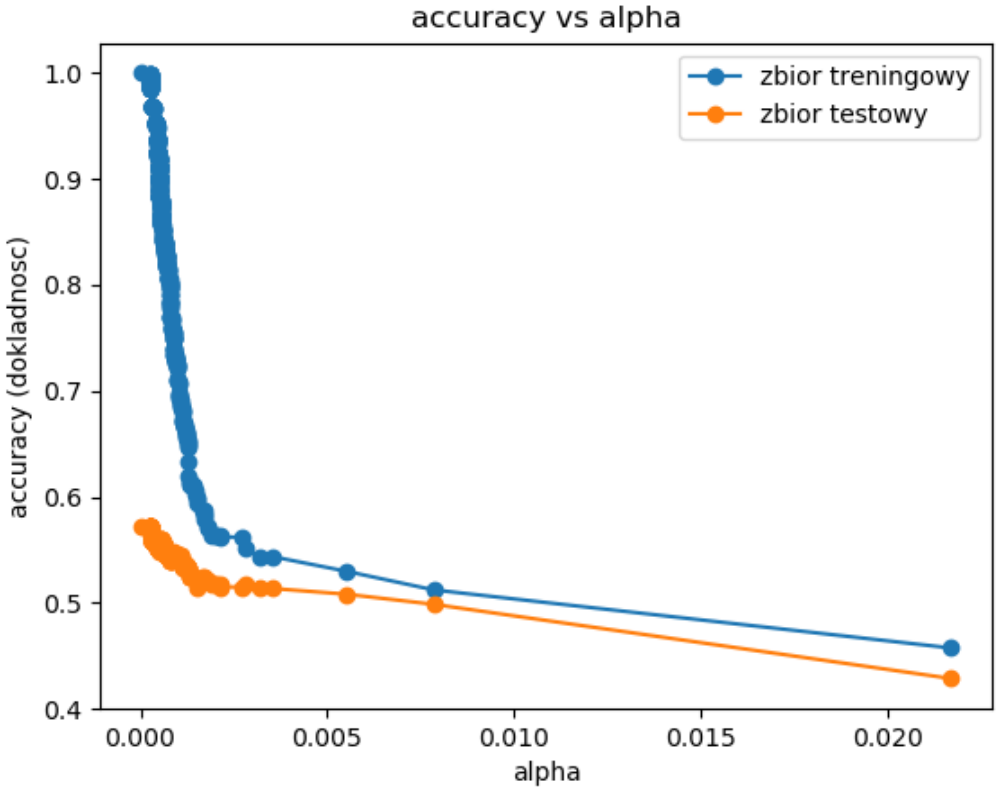
- wyniki dokładności klasyfikacji dla 3 klas wina białego

	Decision Tree	gaussianNB	SVM	Random Forest	Logistic Regression
zb. treningowy	1.0	0.9997082847141119	1.0	0.989790	1.000000
zb. testowy	1.0	0.9993197278911564	1.0	0.987075	1.000000

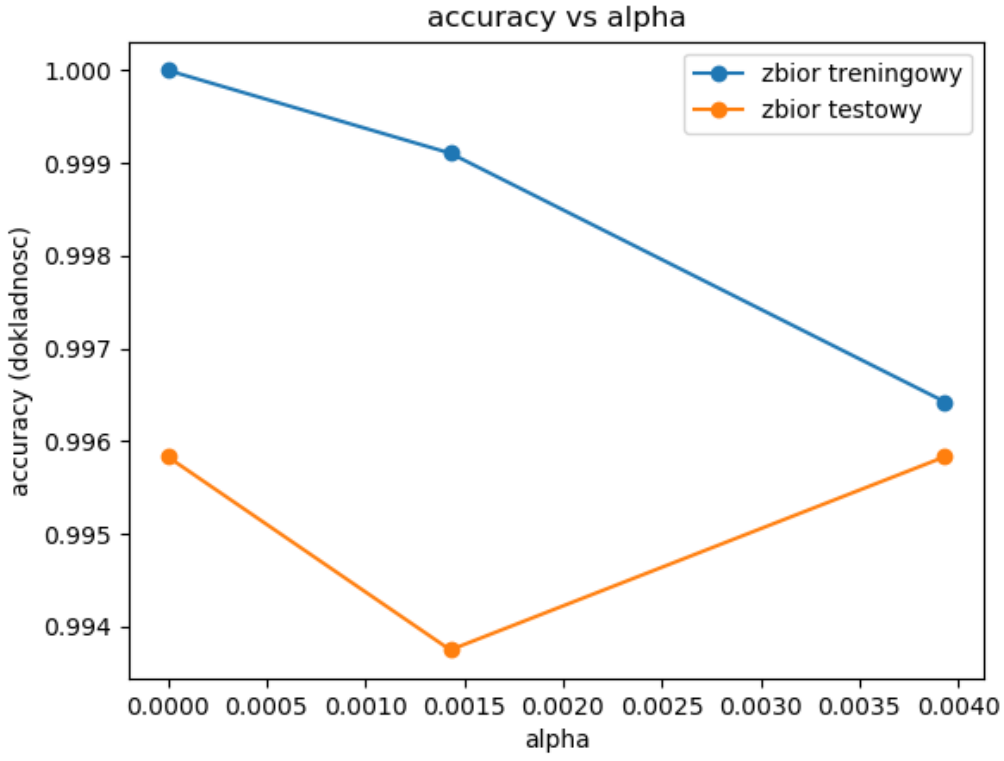
- Wyniki accuracy vs alpha dla wina czerwonego i 10 klas



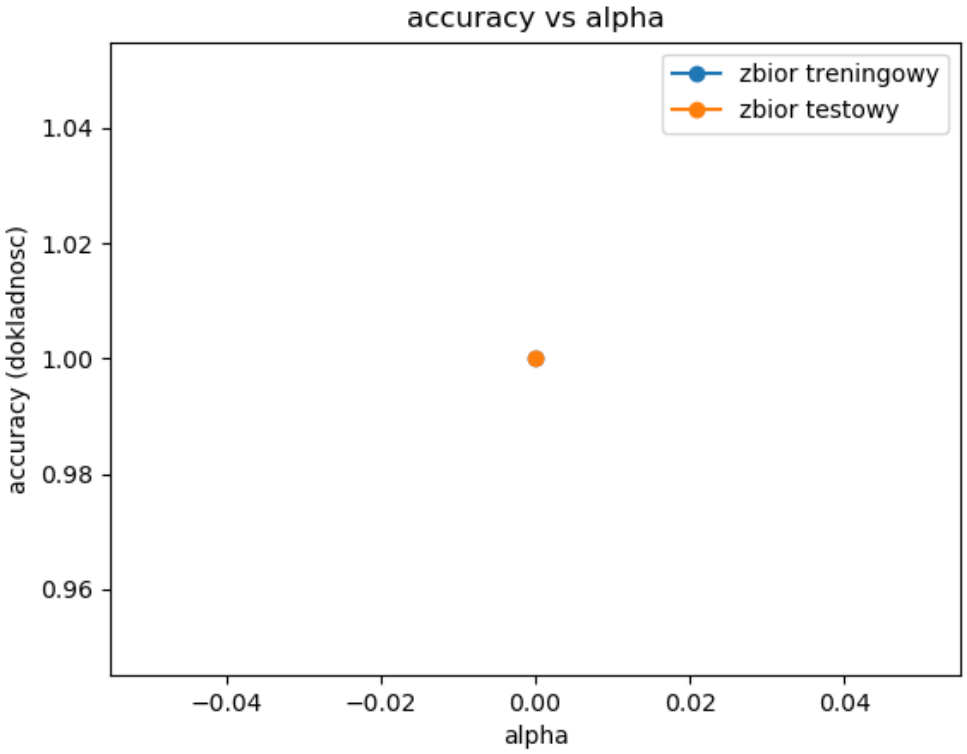
- Wyniki accuracy vs alpha dla wina białego 10 klas



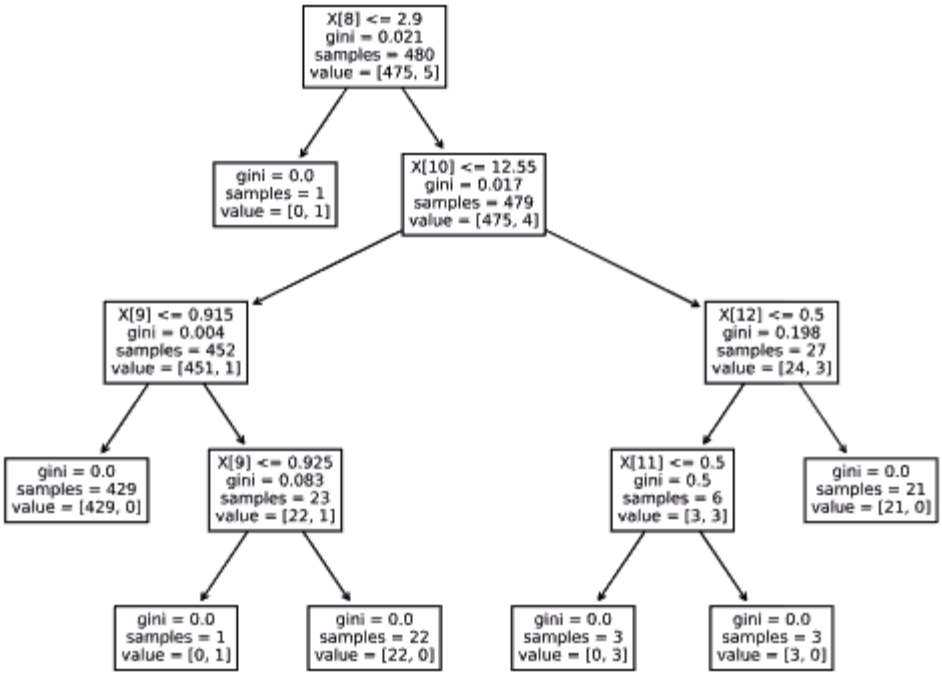
- Wyniki accuracy vs alpha dla wina czerwonego i 3 klas



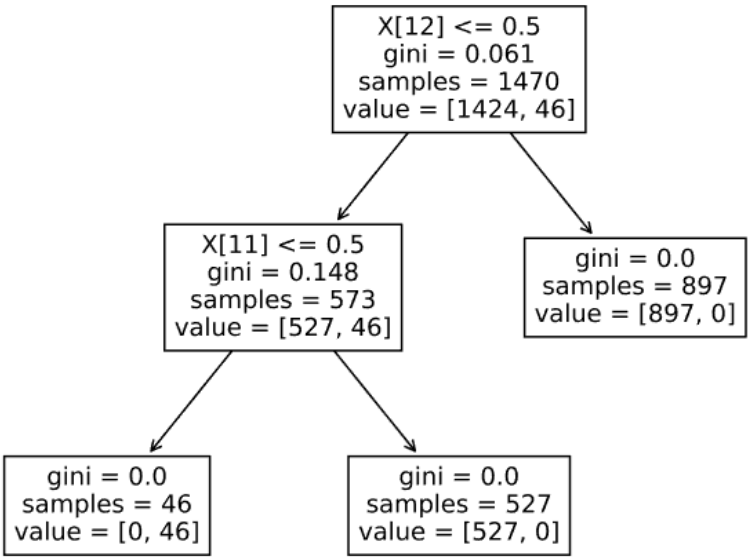
- Wyniki accuracy vs alpha dla wina białego i 3 klas



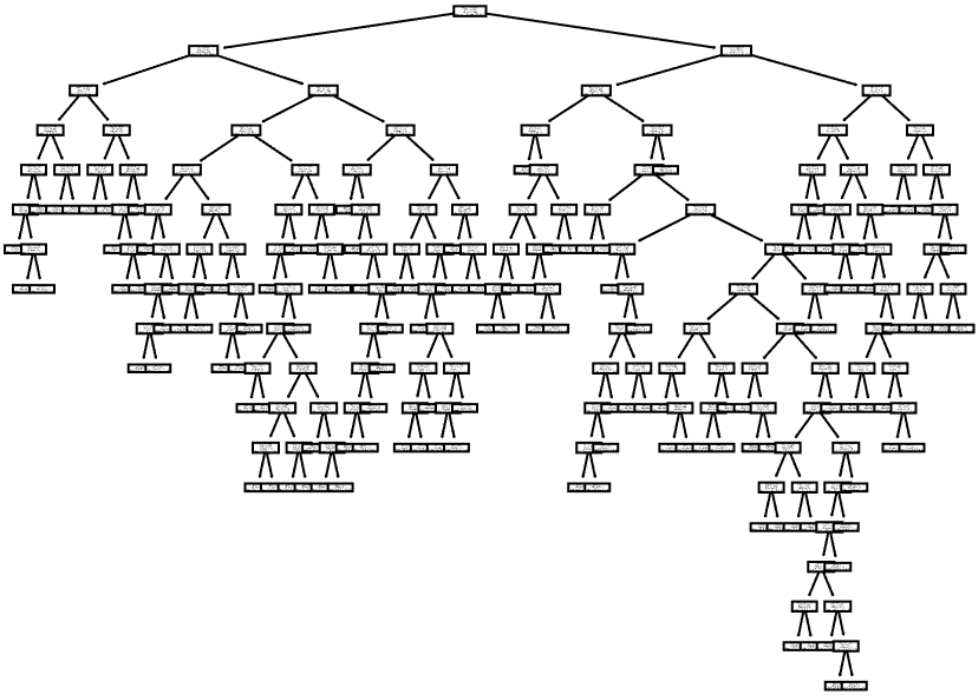
wygląd drzewa dla wina czerwonego i 3 klas



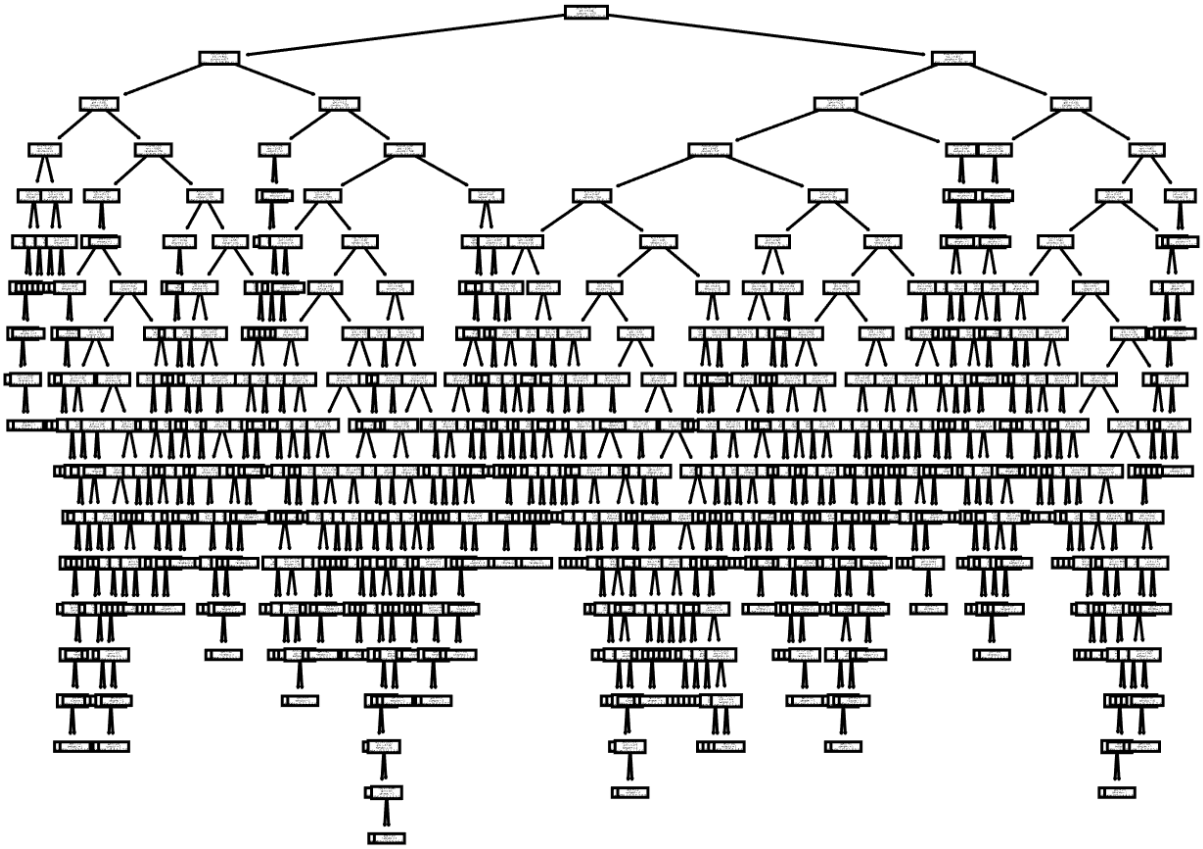
wygląd drzewa dla wina białego i 3 klas



wygląd drzewa dla wina czerwonego i 10 klas



wygląd drzewa dla wina białego i 10 klas



* Diagramy drzew dostępne w formacie SVG w załącznikach opisanych na końcu dokumentacji

5. Podsumowanie

Do sklasyfikowania jakości wina na podstawie jego składu użyliśmy pięciu metod z różnymi parametrami. Ze względu na nie równomierne rozproszenie danych (większość danych klasyfikowała się do środkowego zakresu) żadna z metod nie osiągnęła pożądanego przez nas efektu. Do klasyfikacji na 10 różnych jakości najlepsza okazała się metoda lasu losowego - dla wina czerwonego i metoda svm dla wina białego. Nie przekroczyły one ale dokładności 70% dla zbioru testowego. Wobec tego postanowiliśmy klasyfikować wina do 3 grup. Niskiej jakości, średniej i wysokiej jakości. Dla tak złagodzonych kryteriów, każda z pięciu testowanych metod osiągnęła dokładność powyżej 98% dla zbioru testowego. Wszystkie metody dawały porównywalne rezultaty na bardzo zadowalającym poziomie.

6. Załączniki

Dla klasyfikacji 3-jakościowej:
acc_vs_apl_pjk_red_3.png
dt_red_3.txt
gauss_red_3.txt
pjk_red_3.txt
svm_red_bigger_range_3.txt
tree.svg
acc_vs_apl_pjk_white_3.png
dt_white_3.txt
gauss_white_3.txt
pjk_white_3.txt
svm_white_bigger_range_3.txt
tree_white.svg
Dla klasyfikacji 10-jakościowej:
acc_vs_alp_pjk_red.png
dt_white.txt
grid_search_svm_progress.txt
svm_red.txt
tree.svg
acc_vs_alp_pjk_white.png
gauss_red.txt
pjk_red.txt
svm_red_bigger_range.txt
tree_white.svg
dt_red.txt
gauss_white.txt
pjk_white.txt
svm_white.txt
* Załączniki zawierają wyniki klasyfikacji oraz progres uczenia (dla svm), wykresy, oraz diagramy.
** Załączniki znajdują się odpowiednio w folderach Wyniki i Wyniki_Complex dla klasyfikacji 3-jakosściowej i 10-jakościowej