WSI LAB 7
Klasyfikacja – Modele bayesowskie

I. WSTĘP

Niniejszy raport przedstawia ćwiczenie polegające na implementacji Naiwnego Klasyfikatora Bayesa i jego przetestowanie.

II. URUCHOMIENIE

Do uruchomienia skryptu potrzebne są następujące narzędzia:

- Python w wersji 3.8.8
- Moduły:
    - tabulate 0.8.9
    - pandas 1.2.4
    - numpy 1.20.1
    - sklearn 1.0.1
    - matplotlib 3.3.4
    - sympy 1.8

Aby uruchomić skrypt należy użyć polecenia:

```
python main.py
```

III. PRZYGOTOWANE PLIKI

W załączniku raportu przesyłam plik .zip wraz z implementacją opisanego algorytmu oraz wszystkich wymaganych funkcjonalności do testowania programu, tj.:

- Budowanie macierzy pomyłem dla wielu klas
- Funkcje do oceny modelu

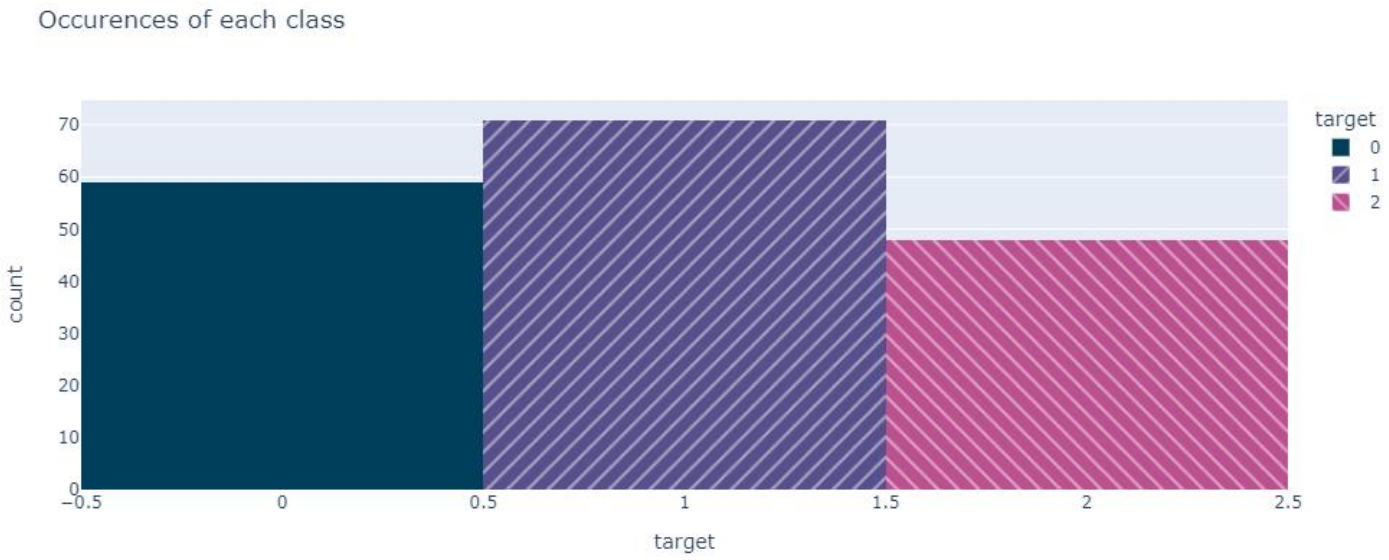
Ze względu na wielkość zaimplementowanego algorytmu i funkcji pomocniczych program został rozdzielony na kilka modułów:

- main_.py (główny skrypt uruchomieniowy)
- NaiveBayesianClassifier.py (implementacja algorytmu „Naiwny klasyfikator Bayesa”)
- tools.py (wczytywanie danych z biblioteki sklearn)
- plotter.py (wykresy i funkcje oceny)

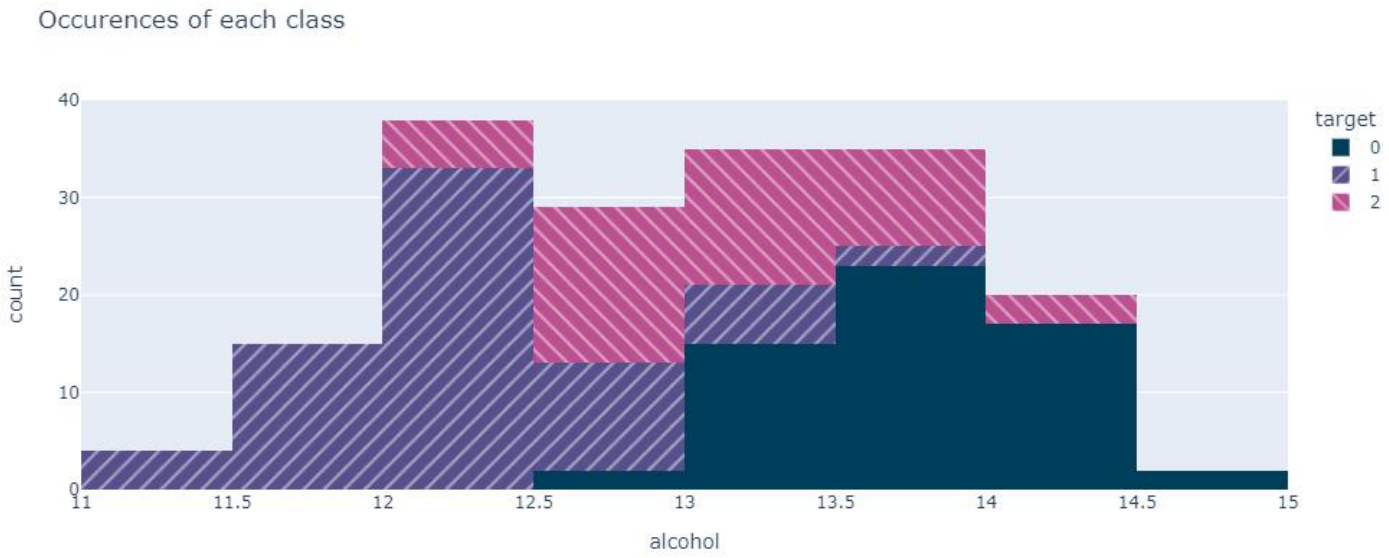
IV. ROZWIĄZANIE

1. Wstępna analiza zbioru danych

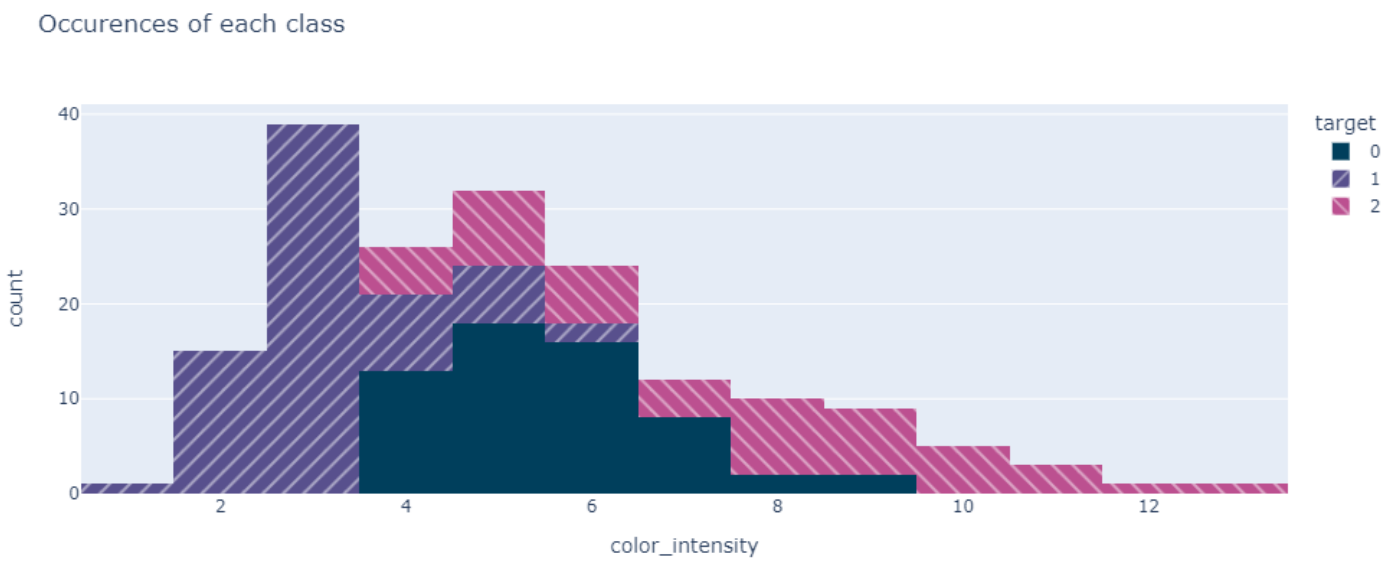
Dane zostały zbadane pod względem częstości wystąpienia klas dla niektórych atrybutów. Uzyskano następujące rezultaty:



(grafika 1.1) Ilość wystąpień każdej z klas



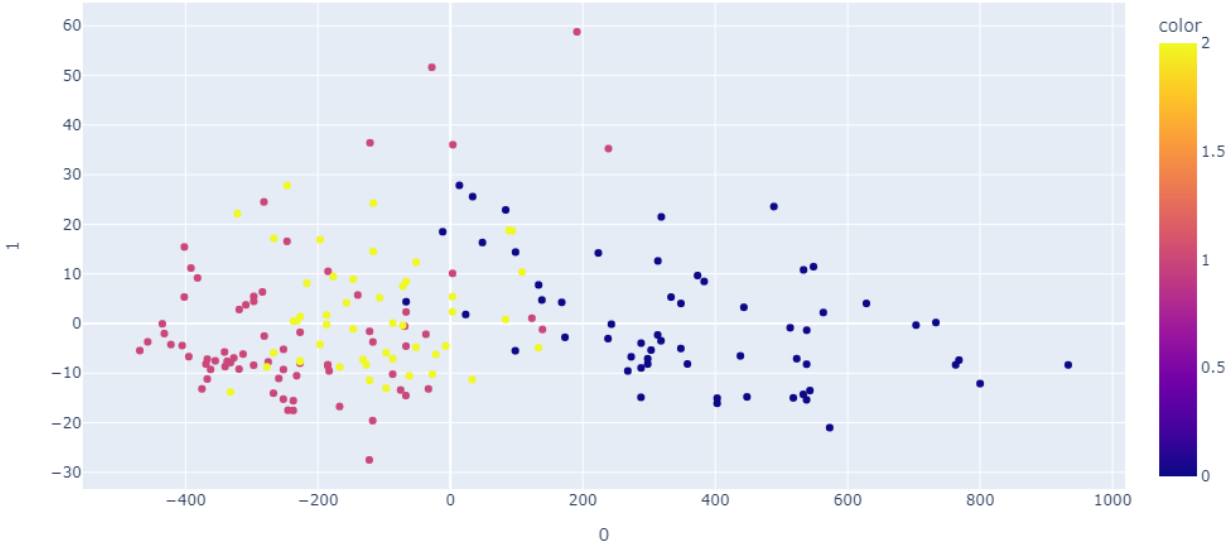
(grafika 1.2) Ilość wystąpień każdej z klas dla atrybutu *alcohol*



(grafika 1.2) Ilość wystąpień każdej z klas dla atrybutu *color_intensity*

Zbadano ilość klas występujących w całym zbiorze danych oraz w dwóch atrybutach: *alcohol* oraz *color_intensity*. Widzimy, że najmniej liczną klasę stanowi klasa *2*.
Kolejno zbadano bardziej szczegółowo wystąpienia klas w atrybucie *alcohol,* widzimy, że każda z klas preferuje szczególne wartości tego atrybutu. Większe wartości tego atrybutu przeważają dla klasy *0*. Klasa *1* preferuje niższe wartości, natomiast klasa *2* opisywana jest przez wartości pomiędzy najmniejszymi i największymi wartościami tego atrybutu.
Analizując atrybut *color_intensity* możemy również stwierdzić podział klas. Klasę *1* charakteryzują mniejsze wartości tego atrybutu, klasa 2 wyższe, które w mniejszym stopniu są zgrupowane dla konkretnej wartości. Natomiast klasa *0* przyjmuje wartości pomiędzy najmniejszymi i największymi wartościami atrybutu *color_intensity*.

Poniżej przedstawiono zbiór danych z wykorzystaniem metody redukcji wymiarów PCA.



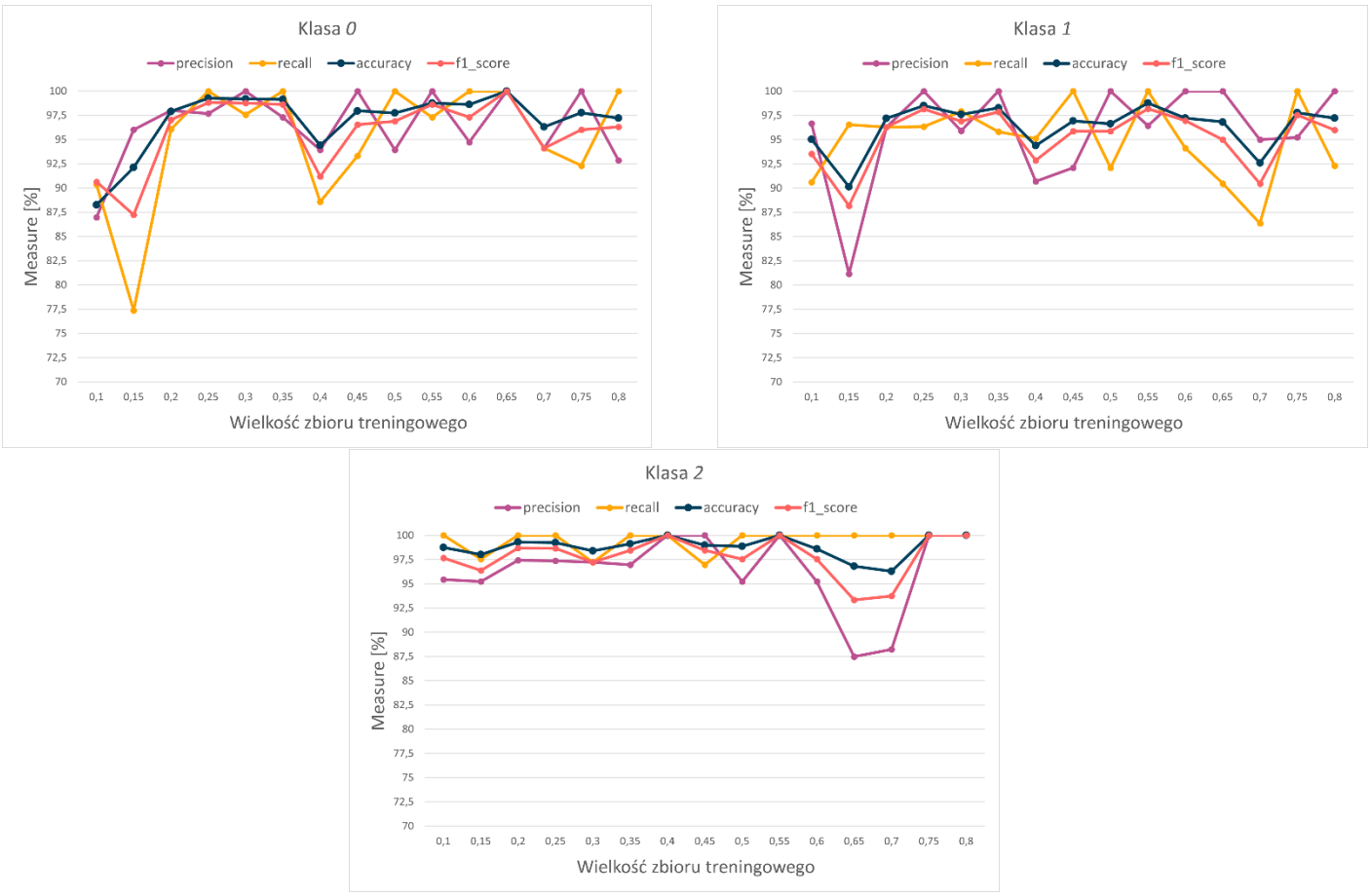
(grafika 1.3) Redukcja wymiarów PCA

Z powyższego wykresu możemy zauważyć wyraźnie, że większość obserwacji klasy 0 znacząco odstaje od próbek pozostałych klas. Odróżnienie przypadków klasy 1 i 2 nie jest już takie wyraźne, mimo tego że obserwacje są bardziej skupione w jednej grupie niż klasy *0*.

2. Wpływ wielkości zbioru treningowego oraz posortowania danych na wyniki

Zbadano wpływ wielkości zbioru treningowego na wyniki algorytmu klasyfikacji. Wyniki predykcji każdej z klas zostały przedstawione poniżej.

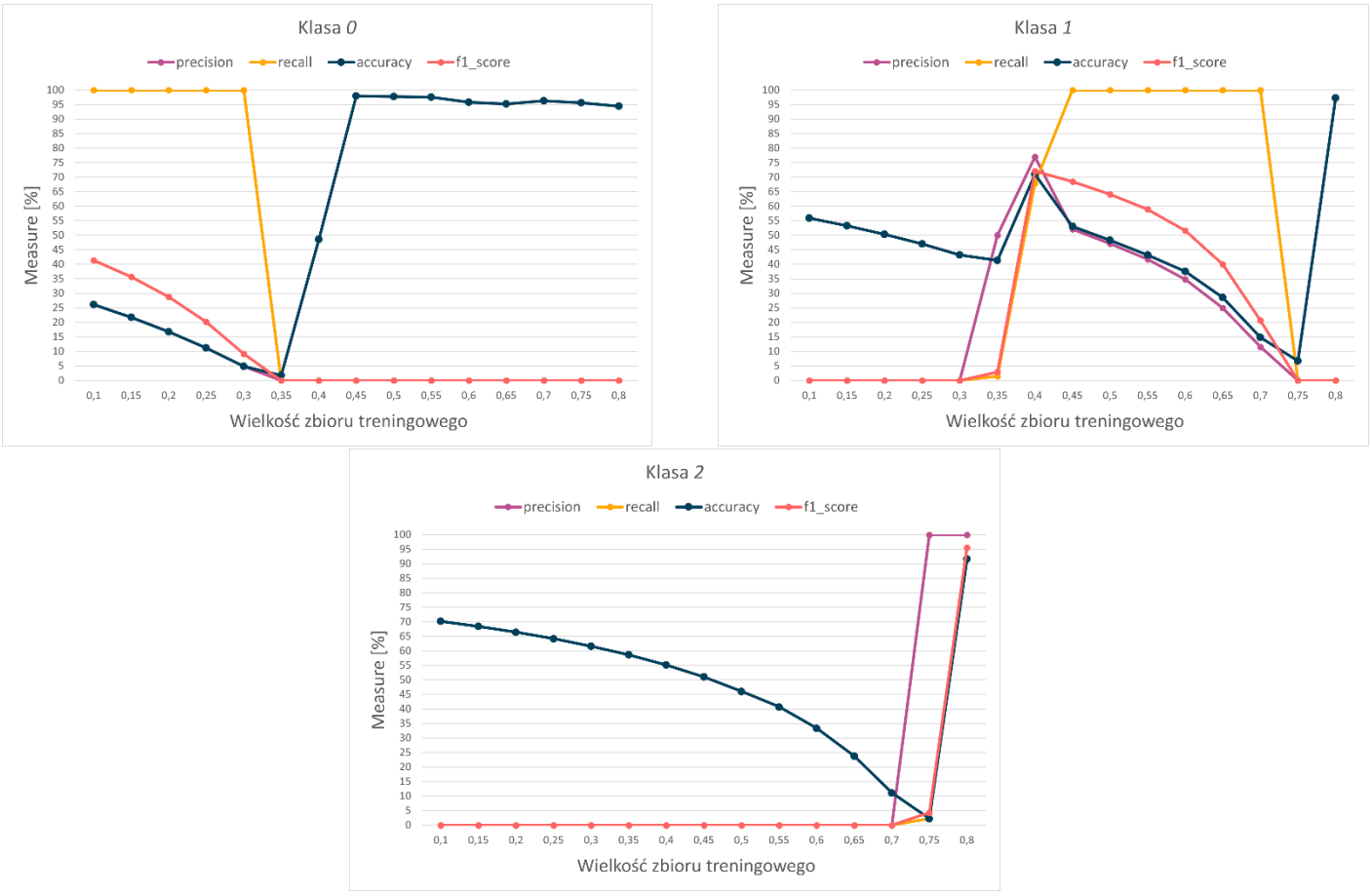
a. Zbiór danych losowo pomieszanych



(grafika 2.1) Wpływ wielkości zbioru treningowego na metryki jakości przewidywań

Z powyższych wykresów możemy stwierdzić, iż wielkość zbioru treningowego wpływa na miary jakości dla przewidywania klas. Wyniki różnią się dla każdej z klas. Klasa numer *2* uzyskuje wysokie wyniki nawet dla bardzo małej wielkości zbioru treningowe, w przeciwieństwie do pozostałych klas. Może to wskazywać na fakt, że zbiór treningowy składał się głównie z klas numer *2*, jednocześnie posiadając zbyt mało przypadków pozostałych klas. Wraz ze wzrostem wielkości zbioru treningowego, metryki pozostałych klas wzrastają, lecz z powodu losowości mieszania zbioru danych, nie uzyskują one stale utrzymujących się wyników metryk jakości. Warto zwrócić uwagę, że dla prawie wszystkich wartości tego parametru, nie istnieje sytuacja gdzie wszystkie klasy jednocześnie uzyskują wyniki na poziomie powyżej *95%,* co wskazuje na to, że w tworzonych zbiorach treningowych brakuje próbek przedstawiających jedną z klas.

b. Zbiór danych posortowanych klasami



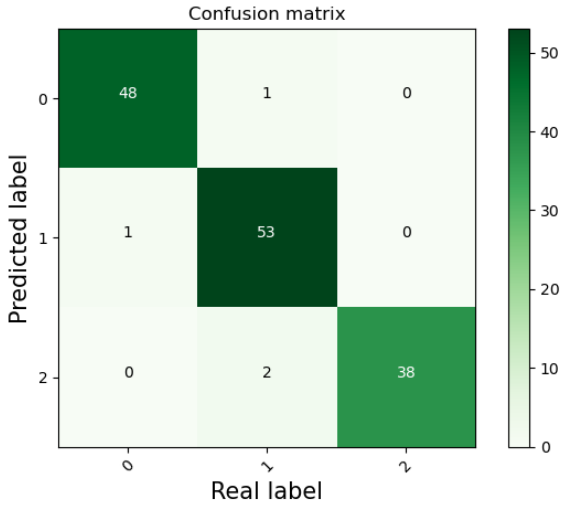
(grafika 2.2) Wpływ wielkości zbioru treningowego na metryki jakości przewidywań

Z powyższych wykresów możemy wyciągnąć dwa wnioski. Pierwszy z nich dotyczy wpływu posortowania zbioru danych na wyniki predykcji klas. Wykresy z poprzedniego podpunktu przedstawiały wyniki klasyfikacji dla losowo pomieszanego zbioru danych. Powyższe wykresy przedstawiają wyniki dla zbioru danych posortowanych klasami co zdecydowanie wpływa na wyniki predykcji, ponieważ zbiór treningowy nie będzie zawierał jednocześnie wszystkich klas, przez co predykcja próbek nie da zadowalających wyników.

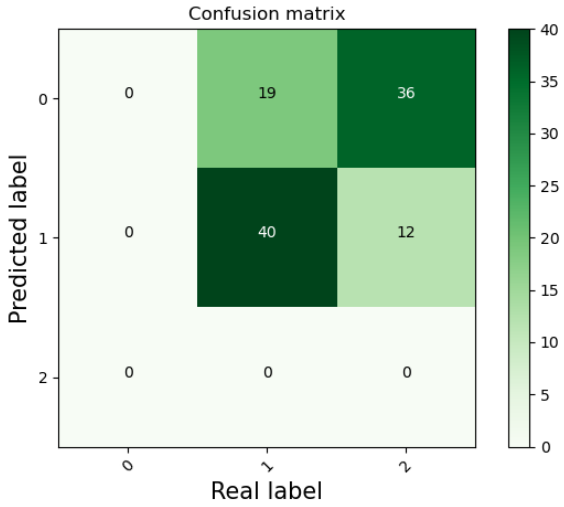
Kolejnym wnioskiem będzie wpływ wielkości zbioru treningowego na wyniki predykcji. Gdy wartość tego parametru jest niska, predykcje wszystkich klas nie są zadowalające. Wyjątkiem będzie tutaj klasa *0,* która posiada wartości miar jakości różne od 0, ponieważ w tym przypadku zbiór treningowy składał się tylko z próbek danej klasy. Dodatkowo możemy stwierdzić, że niewielka ilość próbek tej klasy nie wystarczyła na uzyskanie wysokich wartości miar jakości. Wraz ze wzrostem wielkości zbioru treningowego, możemy zauważyć, że od wartości *0,35* zbiór treningowy zawierał również próbki z *klasy 1,* co wpłynęło na uzyskanie niezerowych wartości miar jakości dla tej klasy. Jednocześnie miary dla pozostałych klas spadły do niezadowalających wartości.

3. Macierze błędów

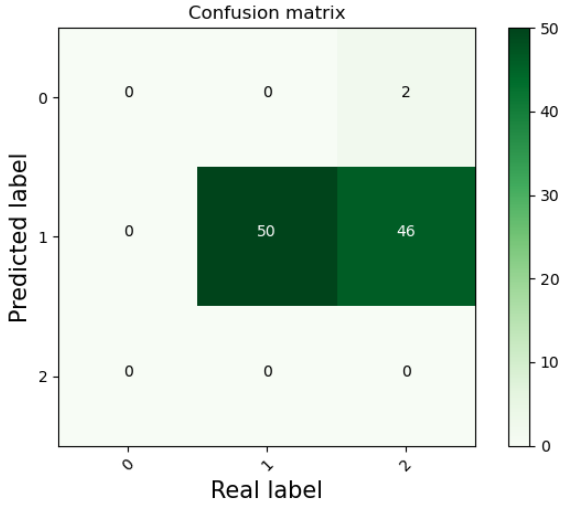
Poniższe wykresy przedstawiają macierze błędów dla pewnych przypadków posortowania zbioru danych oraz wielkości zbioru treningowego:



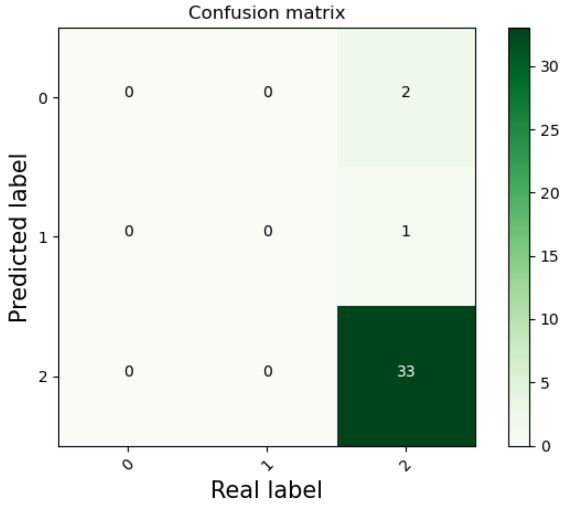
(grafika 3.1) Macierz błędów



(grafika 3.2) Macierz błędów



(grafika 3.3) Macierz błędów



(grafika 3.4) Macierz błędów

Grafika 3.1. przedstawia przypadek gdzie początkowy zbiór danych został losowo pomieszany. Skutkuje to dobrymi wartościami predykcji. Kolejna grafika numer 3.2. przedstawia przypadek gdzie początkowy zbiór danych został posortowany klasami, lecz zbiór treningowy jest zbyt mały, żeby przewidzieć klasy dla niewidzianych próbek. Grafika 3.3 przedstawia przypadek gdzie zbiór danych został posortowany klasami, oraz zbiór testowy jest zbyt mały i nie posiada przypadków jednej z klas, co widoczne jest na grafice. Ostatnia grafika numer 3.4 wskazuje na przypadek, gdzie początkowy zbiór danych został posortowany klasami oraz zbiór testowy jest bardzo mały i zawiera przypadki tylko jednej z klas.

4. Podsumowanie

Zaimplementowana przeze mnie algorytm naiwnego klasyfikatora Bayesa został przebadany pod różnymi aspektami wpływającymi na zbudowane modele. Widzimy, że na wyniki zbudowanych modeli (pozornie trywialnego algorytmu) ma wpływ wiele czynników. Okazuje się jednak, że nawet sortowanie danych wejściowych ma wpływ na finalną jakość modeli.
Ten raport również przekonująco utwierdził, iż stosunek wielkości zbioru treningowego do zbioru testowego ma wpływ na jakoś finalnego modelu.