WSI LAB 4
Klasyfikacja – drzewo decyzyjne ID3

Będkowski Patryk, 310603

I. WSTĘP

Niniejszy raport przedstawia ćwiczenie polegające na implementacji algorytmu *drzewa ID3* oraz przetestowanie jego modeli.

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

Instalacja potrzebnych modułów odbywa się poprzez wprowadzenie polecenia:

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

Aby uruchomić skrypt należy użyć polecenia:

```
python main.py <dataset_name>
```

III. ZAŁOŻENIA ZBIORU DANYCH

Zbiór danych powinien być formatu o rozszerzeniu *.data* bądź *.csv* oraz powinien mieć uzupełnione nazwy kolumn w pierwszej linijce. Standardowym separatorem kolumn i wartości jest znak `,` (przecinek). Kolumna zawierająca klasy próbek, powinna zajmować ostatnie miejsce.

IV. PRZYGOTOWANE PLIKI

W załączniku raportu przesyłam plik .zip wraz z implementacją algorytmu ID3 oraz wszystkich wymaganych funkcjonalności do testowania programu, tj.:

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

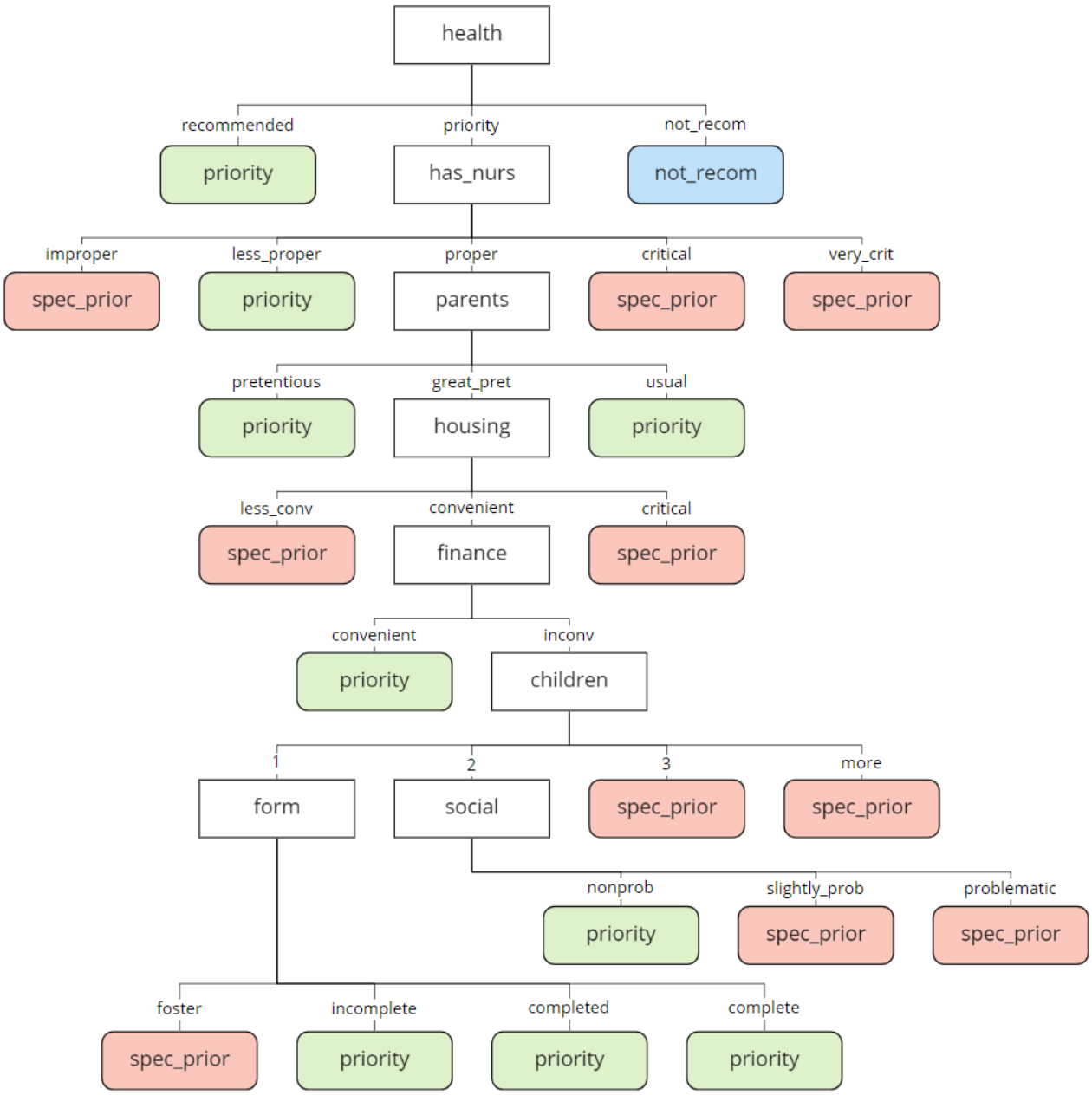
Ze względu na wielkość zaimplementowanego algorytmu i funkcji pomocniczych program został rozdzielony na kilka modułów:

- main.py (główny skrypt uruchomieniowy)
- tree.py (implementacja algorytmu ID3)
- plotter.py (wykresy i funkcje oceny)

V. ROZWIĄZANIE

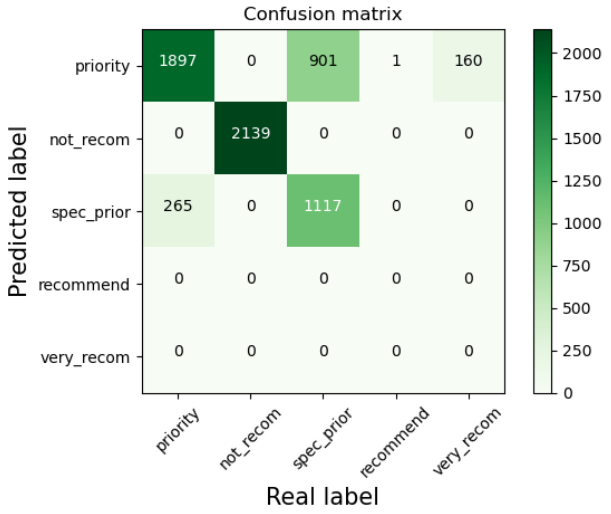
1. Przykładowy wynik programu

Uruchomienie skryptu opisanego w *sekcji II* powoduje uruchomienie algorytmu, podzielenie zbioru danych i zbudowanie drzewa decyzyjnego. Przykład drzewa dla pewnego podzbioru pełnego zbioru danych przedstawia poniższa grafika:



(grafika 1.1) Przykładowe drzewo decyzyjne

Owe drzewo pozwala klasyfikować tylko 3 klasy, zatem można przewidzieć, iż model drzewa został niedouczony i predykcje prowadzone na nowym zbiorze danych mogą okazać się nie do końca trafne. Poniższa grafika przedstawia macierz pomyłek dla pewnego zbioru testowego będącego podzbiorem pełnego zbioru danych i różnym od zbioru treningowego:



(grafika 1.2) Macierz pomyłek

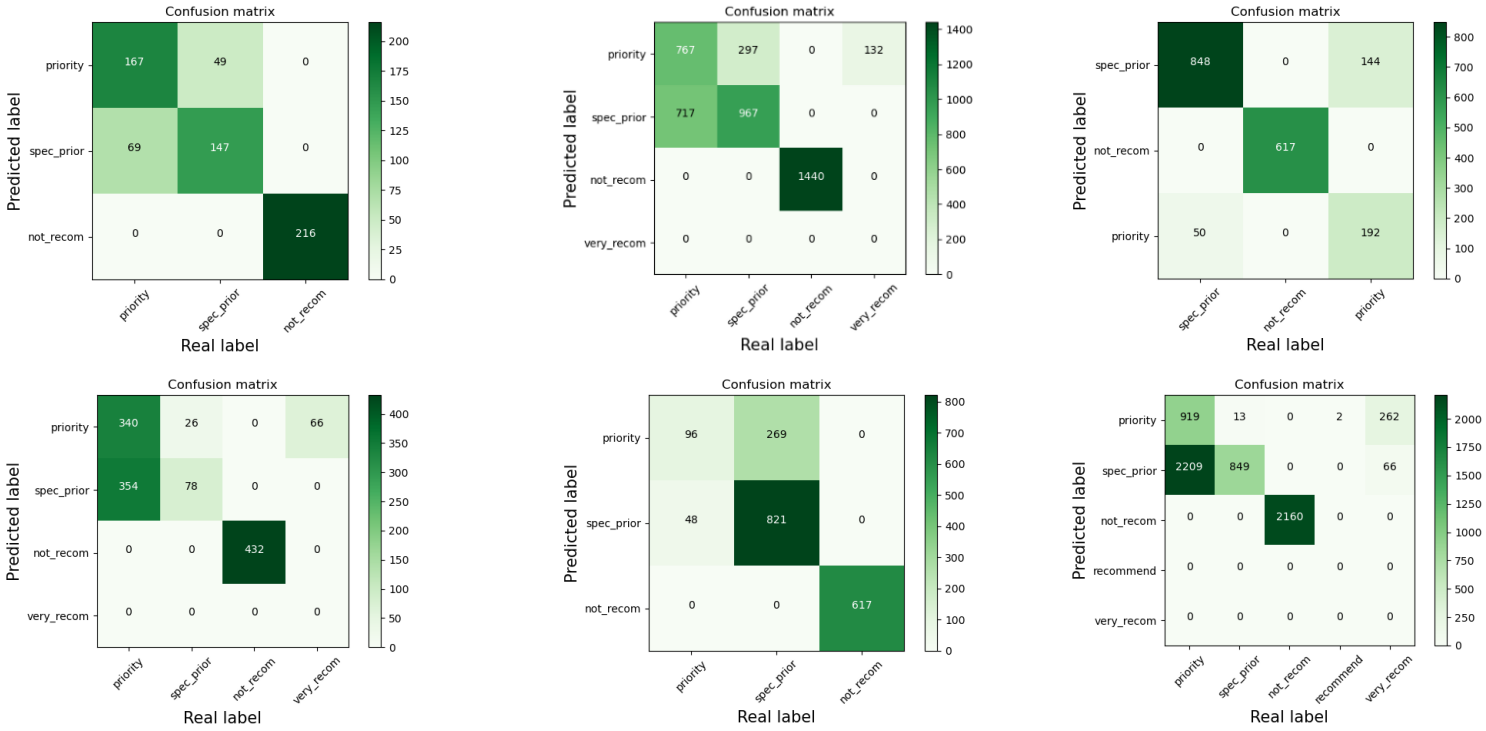
Widoczne jest potwierdzenie przewidzenia tylko 3 klas: *spec_prior*, priority oraz *not_recom*. Możemy wywnioskować, że owe drzewo nie poradziło sobie z predykcją nowych, nieznanych danych. Następnie zbadano miary jakości dla podanej predykcji:

Label	Precision	Recall	Accuracy	F1 score
priority	64.11	87.74	79.52	74.09
not_recom	100.0	100.0	100.0	100.0
spec_prior	80.82	55.35	82.01	65.7
recommend	0.0	0.0	99.98	0.0
very_recom	0.0	0.0	97.53	0.0

(grafika 1.3) Macierz pomyłek

Widzimy, że zbudowany model bezbłędnie poradził sobie w klasyfikowaniu próbek o klasie *not_recom*. Jeżeli chodzi o pozostałe predykcje to widzimy niejednoznaczne wyniki. Stwierdzić można, że owy model nie do końca radzi sobie z wykrywaniem klasy *spec_prior*, z niepowodzeniem nie wykrył tutaj około 45% próbek, co może świadczyć o tym, że model nie jest czuły na dana klasę.

Poniższe grafiki przedstawiają przykładowe uzyskane macierze pomyłek dla różnych parametrów k_folds

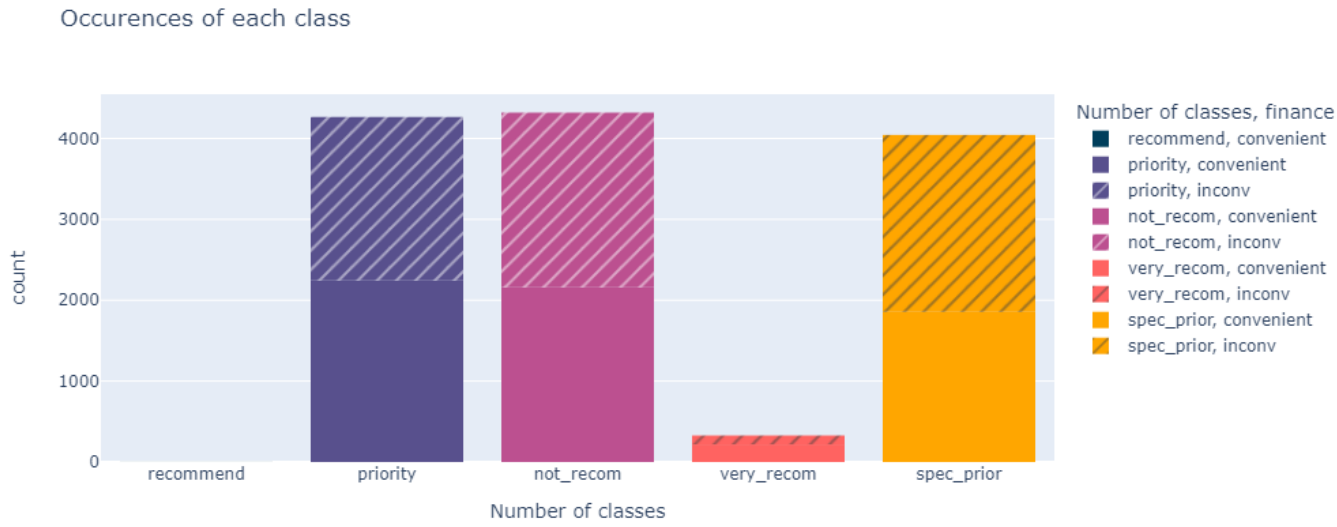


2. Wstępna analiza zbioru danych

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

(grafika 2.1) Ilość wystąpień każdej z klas

(grafika 2.2) Ilość wystąpień każdej z klas dla atrybutu *health*



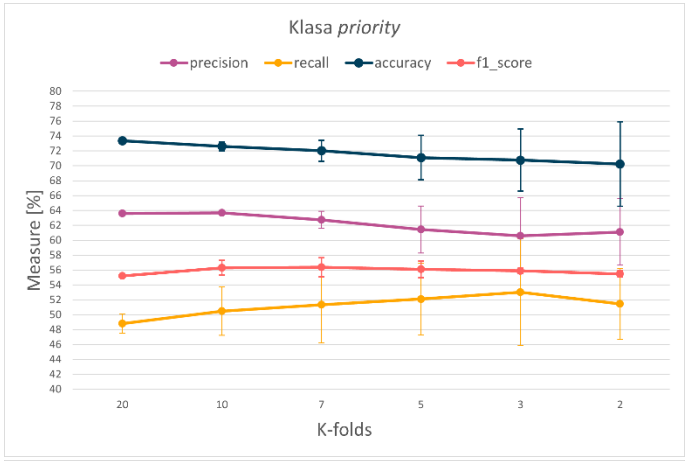
(grafika 2.3) Ilość wystąpień każdej z klas dla atrybutu *finance*

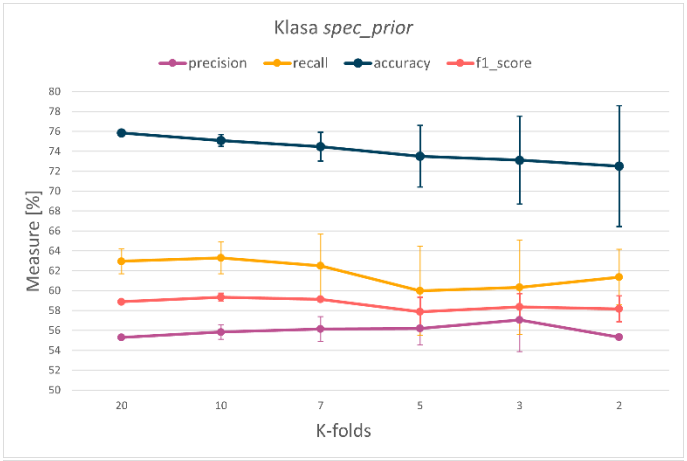
Zbadano ilość klas występujących w całym zbiorze danych oraz w dwóch atrybutach : *health* oraz *finance*. Widzimy, że najmniej liczne klasy to *recommend* oraz *very_recom*. Przyrównując te spostrzeżenia do grafiki przykładowego drzewa decyzyjnego (grafika 1.1), możemy potwierdzić rzadkość tych klas, iż nie zostały one zawarte w przewidywaniach drzewa decyzyjnego dla danego zbioru trenującego.

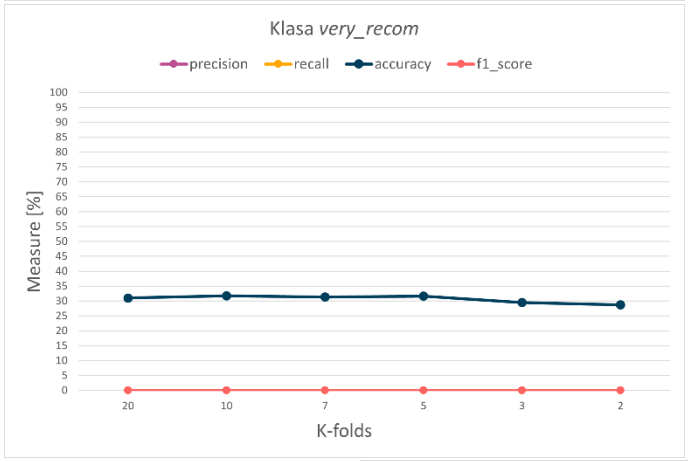
Kolejno zbadano bardziej szczegółowo wystąpienia klas w atrybucie *health,* widzimy, że atrybut ten zawiera prawie wszystkie wystąpienia klasy *not_recom*, co również zgadza się ze zbudowanym powyżej drzewem decyzyjnym. Kolejnym krokiem było zbadanie atrybutu – *finance*, który wydawał się mieć mniejsze znaczenie, ponieważ dla przykładowego drzewa decyzyjnego, węzeł go opisujący leży dalej od korzenia drzewa.

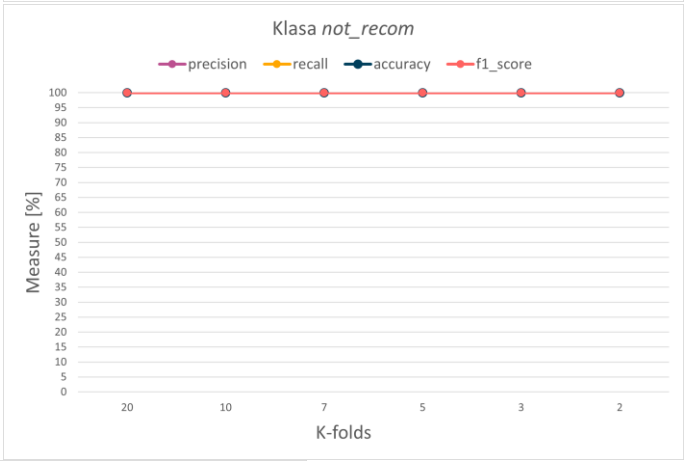
3. Wpływ parametru *k-folds* walidacji krzyżowej na wyniki

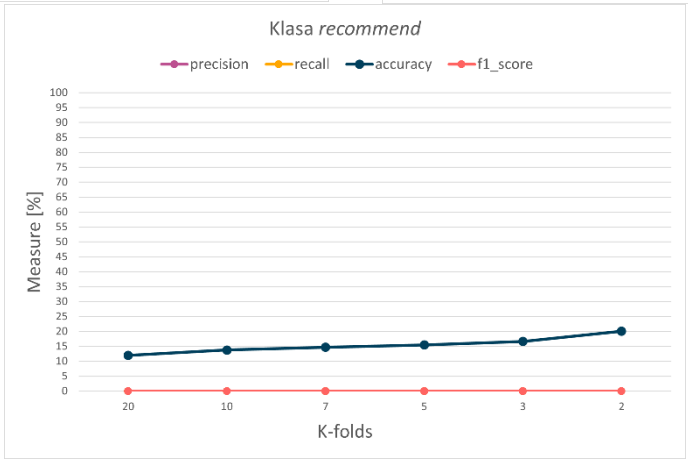
Zbadano wpływ parametru k-folds (ilość podzbiorów) walidacji krzyżowej na wyniki algorytmu klasyfikacji. Wyniki zostały przeprowadzone 20 razy i zostały uśrednione.











(grafika 3.1) Wpływ parametru k-folds na metryki jakości przewidywań

Z powyższych wykresów możemy stwierdzić, iż parametr *k-folds* wpływa na miary jakości dla podanego przewidywania. Wraz ze spadkiem wartości tego parametru spadają takie metryki jak precyzja (*precision*) i dokładność (*accuracy*). Wzrasta także odchylenie standardowe tych metryk, co może oznaczać że budowane modele bardziej różnią się od siebie klasyfikacją próbek. Z drugiej jednak strony klasa *priority*, wraz ze spadkiem parametru *k-folds* uzyskuje lepszą miarę czułości (recall), co oznacza, że wykrywa więcej przypadków danej klasy.

Klasa *spec_prior* również potwierdza, że większe wartość parametru *k-folds* generalnie buduje skutkuje lepszymi modelami. Wszystkie metryki uzyskują większe wartości dla dużych wartości *k-folds*.

Klasa *not_recom* uzyskała takie same wyniki dla przeprowadzonych testów. Wyniki są stale na poziomie 100% dla każdej metryki, co może oznaczać, że budowane modele skutecznie wykrywają daną klasę.

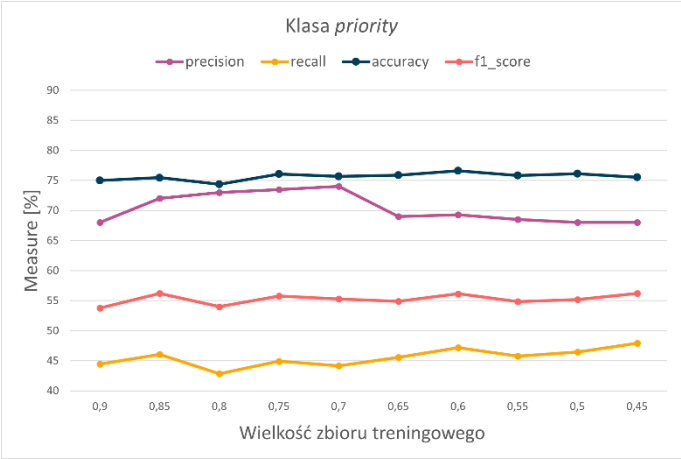
Pozostałe klasy nie uzyskały zadowalających wyników, co oznacza że zbudowane modele nie faworyzowały tej klasy w budowie drzewa, co może być powodem działania algorytmu bądź zbiorem danych.

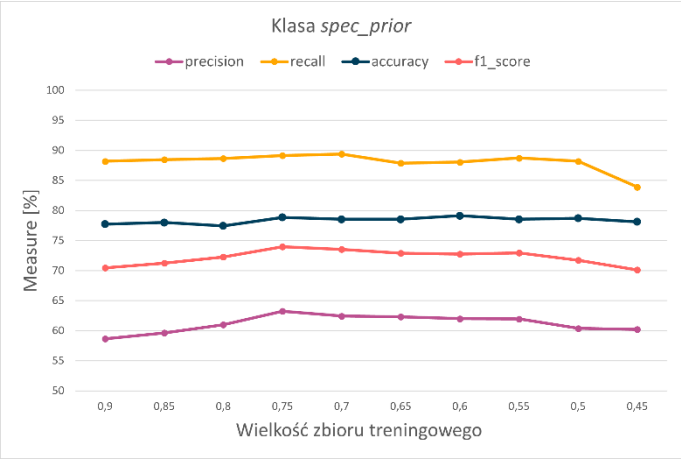
4. Wpływ podziału danych zbioru treningowego i testowego

Zbadano różne wielkości zbioru treningowego oraz testowego, oraz ich wpływ na testowane miary jakości. Test przeprowadzono dla danych wstępnie posortowanych (nie zmieniano kolejności próbek w wejściowym zbiorze danych) oraz na danych przemieszanych:

a. Dane losowo przemieszane

Na wykresach przedstawiono miary jakości dla różnych podzbiorów treningowych całego zbioru danych (wielkość części zbioru testowego wynosi 1 – <wielkość części zbioru testowego>)



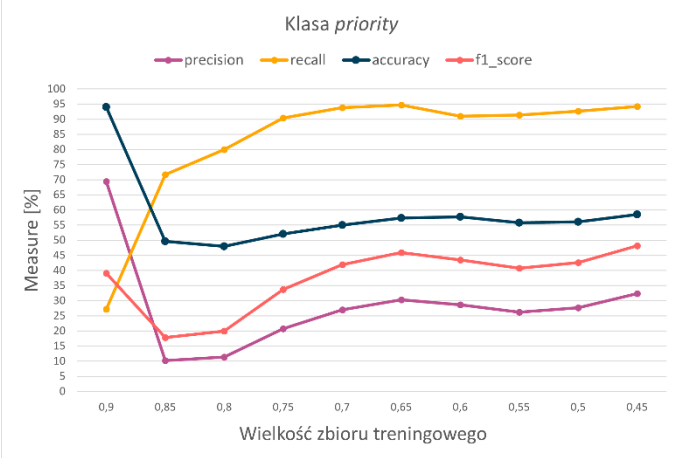


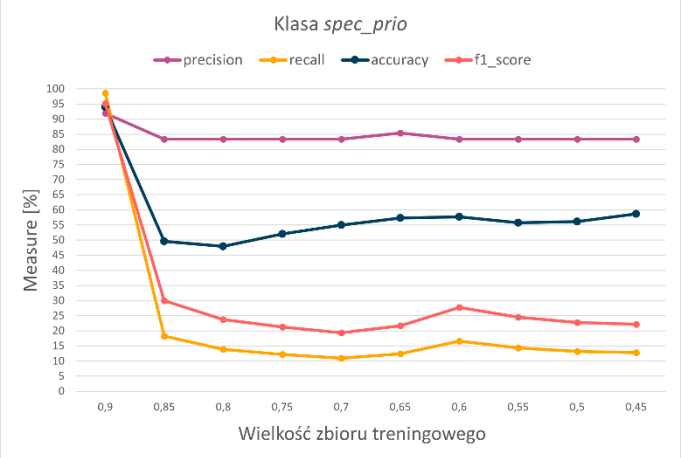
(grafika 4.1) Miary jakości dla różnych wielkości zbioru treningowego

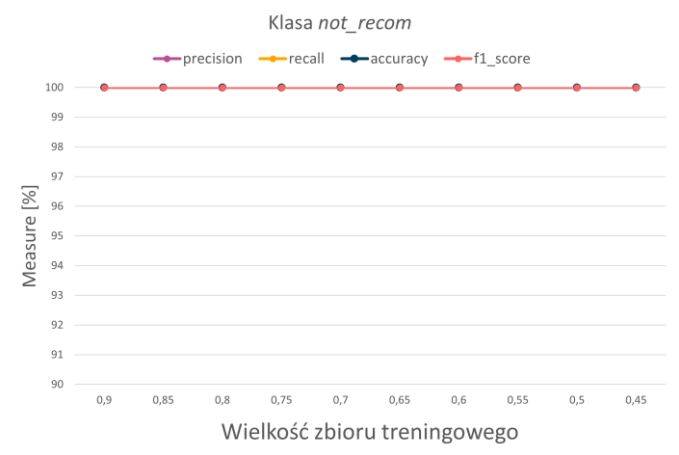
Powyższe wykresy wykazują, iż stosunek wielkości zbioru treningowego ma duży wpływ na osiągnięte wyniki miary jakości przy klasyfikacji nowego zbioru danych. Widzimy, że najlepsze wyniki zostały osiągnięte gdy zbiór treningowy nie wynosił mniej niż 70% całego zbioru danych. Dla pozostałych parametrów widzimy, że model został wytrenowany ze zbyt dużym przypasowaniem do wprowadzanych danych, przez co nie poradził sobie na klasyfikacji danych nie widzianych wcześniej.

b. Dane wstępnie posortowane

Na wykresach przedstawiono miary jakości dla różnych podzbiorów treningowych całego zbioru danych (wielkość części zbioru testowego wynosi 1 – <wielkość części zbioru testowego>)







(grafika 4.2) Miary jakości dla różnych wielkości zbioru treningowego

Powyższe wykresy wykazują, iż uzyskaliśmy inne wyniki niż w przypadku gdy zbiory treningowe i testowe były losowo wymieszane. Klasa *not_recom* została bezbłędnie sklasyfikowana, co może świadczyć, że w każdym przypadku uzyskano zbiór treningowy, który posiadał możliwie dużo zróżnicowanych próbek.
Wykresy dwóch pozostałych klas mogą sugerować, że im większy zbiór treningowy tym większe prawdopodobieństwo, że model zostanie wytrenowany na większej ilości różniących się próbek, przez lepiej poradzi sobie w klasyfikacji zbioru testowego. Tym samym zwiększenie zbioru testowego (zmniejszenie treningowego) prowadzi do sytuacji gdy model, wytrenowany na zbyt małej ilości danych, z niepowodzeniem wykona predykcję klas.

5. Podsumowanie

Zaimplementowana przeze mnie algorytm ID3 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.
Dodatkowo przebadana przeze mnie walidacja krzyżowa pozwoliła znaleźć optymalne parametry, zwiększające prawdopodobieństwo uzyskania lepszego modelu dla danego zbioru danych. Ten raport również przekonująco utwierdził, iż stosunek wielkości zbioru treningowego do zbioru testowego ma wpływ na jakoś finalnego modelu.