PSZT-U - zima 2020/2021

Adam Steciuk (300263)
Łukasz Pokorzyński (300251)

Treść zadania:

Przewidywanie czy grzyb jest jadalny przy użyciu własnej implementacji algorytmu ID3.
Testy tożsamościowe.

Doprecyzowanie:

Testy tożsamościowe - przy podziale na poddrzewa tworzone jest tyle poddrzew ile różnych wartości atrybutu wybranego do podziału

Podział zadań:

Wspólnie:

- Analiza danych i wyników
- Dokumentacja
- Implementacja walidacji krzyżowej

Adam Steciuk:

- Założenie repozytorium
- Implementacja algorytmu ID3

Łukasz Pokorzyński:

- Stworzenie struktury projektu
- Przeprowadzenie testów

Analiza danych wejściowych:

Otrzymane dane składają się z dwóch zestawów. Pierwszy z nich składa się z 8416 grzybów i zawiera pełne nazwy atrybutów. Zostanie on wykorzystany jedynie do demonstracji zbudowanego drzewa. W drugim natomiast atrybuty oznaczone są jednoliterowymi skrótami i to on użyty będzie do wszystkich testów i badań.

Wszystkich grzybów jest 8124, jadalnych jest 4208 (51.8%), a trujących 3916 (48.2%). Klasy są w miarę równo rozdystrybuowane, zatem wystarczającą miarą jakości modelu będzie jego dokładność *accuracy = (TP + TN) / (TP + TN + FP + FN).* Każdy model mający dokładność większą niż 0.518 będzie modelem lepszym niż model traktujący wszystkie grzyby jako jadalne.

2480 grzybów ma brakującą wartość w atrybucie *“stalk-root”.* Traktowanie braku wartości jako osobnej wartości atrybutu nie wpływa zauważalnie na jakość modelu.

Przeprowadziliśmy porównanie modeli budowanych na danych pozbawionych wybrakowanych wierszy z modelami budowanymi również na danych zawierających braki uznawane tak jak opisano powyżej i nie zauważyliśmy żadnych wyraźnych różnic. Niestety na szczegółowe wyniki tych testów zabrakło miejsca w dokumentacji (max 4 strony).

Decyzje projektowe:

W przypadku przewidywania klasy dla instancji grzyba, który (dla atrybutu dzielącego na poddrzewa) ma wartość niewystępującą w drzewie - zostaje wybrane poddrzewo o występującej najczęściej (przy tworzeniu drzewa) wartości atrybutu. Z tego powodu w każdym korzeniu zapisywana jest ilość instancji ze zbioru uczącego “pasujących” do danego korzenia.

Drzewo budowane jest jako rekursywnie budowany słownik o kluczach zawierających etykiety informacji przechowywanych w wartościach temu kluczowi odpowiadającemu.
Każdy węzeł (*node)* to słownik zawierający parę (*“count”: int)* zliczający instancje grzybów pasujących do ów klucza. Ponadto, jeśli węzeł jest liściem zawiera parę (*“label”: class*), gdzie *class* to wartość, którą przewidujemy, a jeśli nie to zawiera pary (*“attribute”: attribute*), gdzie *attribute* to atrybut wybrany do podziału na poddrzewa oraz (*“children”: {value: node}*), gdzie wartością dla klucza *“children”* jest słownik o kluczach będących różnymi wartościami atrybutu *attribute*, którym przyporządkowane są odpowiednie, kolejne węzły drzewa.

Przykład drzewa zbudowanego na całym zbiorze dostępnych danych:

```
odor (8416)
    ALMOND: EDIBLE (400)
    ANISE: EDIBLE (400)
    NONE: spore-print-color (3808)
        BLACK: EDIBLE (1424)
        BROWN: EDIBLE (1472)
        CHOCOLATE: EDIBLE (48)
        GREEN: POISONOUS (72)
        WHITE: habitat (648)
            LEAVES: cap-color (88)
                BROWN: EDIBLE (48)
                CINNAMON: EDIBLE (24)
                WHITE: POISONOUS (8)
                YELLOW: POISONOUS (8)
            WASTE: EDIBLE (192)
            WOODS: gill-size (40)
                NARROW: POISONOUS (32)
                BROAD: EDIBLE (8)
            GRASSES: EDIBLE (288)
            PATHS: EDIBLE (40)
        YELLOW: EDIBLE (48)
        ORANGE: EDIBLE (48)
        BUFF: EDIBLE (48)
    PUNGENT: POISONOUS (256)
    CREOSOTE: POISONOUS (192)
    FOUL: POISONOUS (2160)
    FISHY: POISONOUS (576)
    SPICY: POISONOUS (576)
    MUSTY: POISONOUS (48)
```

Wyniki testów przedstawiane są jako *DataFrame* z biblioteki pandas zawierające wszystkie logi testu pozwalające na jego odtworzenie. Aby móc przedstawić tablicę pomyłek, wymagane jest sprecyzowanie “klasy pozytywnej”. Jako taką uznaliśmy grzyby jadalne, więc poprawne sklasyfikowanie grzyba jadalnego uznane jest jako “True Positive” (TP) a trującego - “True Negative” (TN). Aby to zmienić, należy w pliku config.py zmienić positive_class z “e” na “p”.

Narzędzia i biblioteki:

- Python 3.7
- math
- pandas
- numpy
- Microsoft Excel

Odtworzenie wyników:

Z uwagi na wykorzystane biblioteki należy upewnić się, że zainstalowane są pakiety pandas i numpy:

- pip install pandas
- pip install numpy

Aby odtworzyć wyniki każdego z badań należy uruchomić odpowiadający mu skrypt: ex1.py, ex2.py, ex3.py.

Wyniki zostaną wyświetlone w konsoli oraz wyeksportowane do plików .csv.

Badania:

ex1: Badanie wpływu parametru k w k-walidacji krzyżowej na jakość modelu:

Dla pełnolicznego zbioru ucząco-walidującego, hiperparametr k przebadany zostanie dla wartości z przedziału <2, 7>.
Teza: Im większe k tym większy zbiór uczący, a co za tym idzie jakość modelu większa.
Wyniki:

data_size	k	TP	TN	FP	FN	accuracy	global_seed	num_of_reruns
8124	2	4207.68	3915.28	0.72	0.32	0.9998719842442148	1	25
8124	3	4208.0	3916.0	0.0	0.0	1.0	1	25
8124	4	4208.0	3916.0	0.0	0.0	1.0	1	25
8124	5	4208.0	3916.0	0.0	0.0	1.0	1	25
8124	6	4208.0	3916.0	0.0	0.0	1.0	1	25
8124	7	4208.0	3916.0	0.0	0.0	1.0	1	25

Interpretacja: Model nawet dla k=2 jest już bardzo dokładny, nie ma jednak stuprocentowej dokładności, tak jak dla większych wartości k. Różnica ta jednak jest na tyle mała, że nie można potwierdzić postawionej tezy.

ex2: Badanie wpływu wielkości zbioru na jakość modelu przy użyciu k-walidacji krzyżowej:

Dla ustalonego k=3 zbiór ucząco-walidujący będzie zmniejszany w każdej iteracji o połowę dopóki wielkość zbioru będzie większa niż 1/1000 początkowego zbioru.
Teza: Im mniejszy zbiór danych tym mniejszy zbiór uczący, a co za tym idzie jakość modelu mniejsza.
Wyniki:

data_size	k	TP	TN	FP	FN	accuracy	global_seed	num_of_reruns
8124	3	4207.84	3915.84	0.16	0.16	0.999961	1	25
4062	3	2089.28	1971.36	0.64	0.72	0.999665	1	25
2031	3	1017.52	1009.76	1.24	2.48	0.998168	1	25
1016	3	504.08	508.68	1.32	1.92	0.996811	1	25
508	3	250.16	253.48	2.52	1.84	0.991417	1	25
254	3	127.08	120.68	4.32	1.92	0.975433	1	25
127	3	65.6	55.72	4.28	1.4	0.955276	1	25
63	3	28.76	33.76	0.24	0.24	0.992381	1	25
32	3	14.2	16.4	0.6	0.8	0.956250	1	25
16	3	3.04	10.76	0.24	1.96	0.862500	1	25

Interpretacja: Im mniej danych w zbiorze, tym model ma mniejszą dokładność. Postawioną tezę można potwierdzić. Dodatkowo można zauważyć, że uznanie w poprzednim teście różnicy w dokładności modelu dla k=2 od dokładności modelu dla innych wartości k za mieszczące się w granicy błędu było słuszne (tym razem model dla k=3 i pełnego zbioru ucząco-walidującego, ale inaczej posortowanego, nie ma dokładności 100%). Badanie pokazuje również, że bardzo wysoka dokładność modelu wynika z charakterystyki zbioru danych a nie jest wynikiem błędu.

ex3: Badanie wpływu wielkości zbioru uczącego na jakość modelu testowanego na całym dostępnym zbiorze danych.

Zbiór uczący będzie stopniowo zmniejszany, tak długo, aż pozostanie w nim 1 element. Dla każdej jego wielkości zostanie utworzony model drzewa ID3, który zostanie przetestowany na całym dostępnym zbiorze danych.
Teza: Dokładność klasyfikacji będzie spadała szybciej niż w przypadku poprzedniego testu, a dla zbioru uczącego wielkości 1 elementu będzie wynosiła około 0.5.
Wyniki:

train_size	test_size	TP	TN	FP	FN	accuracy	global_seed	num_of_reruns
8124	8124	4208.0	3916.0	0.0	0.0	1	1	25
4062	8124	4207.84	3915.76	0.24	0.16	0.999951	1	25
2031	8124	4205.6	3910.8	5.2	2.4	0.999065	1	25
1016	8124	4202.64	3905.36	10.64	5.36	0.998031	1	25
508	8124	4196.8	3892.0	24.0	11.2	0.995667	1	25
254	8124	4194.8	3867.36	48.64	13.2	0.992388	1	25
127	8124	4177.12	3814.88	101.12	30.88	0.983752	1	25
63	8124	4181.36	3704.88	211.12	26.64	0.970734	1	25
32	8124	4107.04	3528.4	387.6	100.96	0.939862	1	25
16	8124	4020.16	3170.0	746.0	187.84	0.885052	1	25
8	8124	3611.52	2489.52	1426.48	596.48	0.750990	1	25
4	8124	3008.32	1581.28	2334.72	1199.68	0.564943	1	25
2	8124	2660.48	1559.04	2356.96	1547.52	0.519389	1	25
1	8124	3029.76	1096.48	2819.52	1178.24	0.507907	1	25

Interpretacja: Część tezy o “szybszym” spadaniu dokładności modelu okazała się błędna. Model dla większych zbiorów testowych jest w stanie poprawnie sklasyfikować
podobny procent grzybów. Okazuje się również, że jest bardzo dokładny nawet dla małych zbiorów uczących. Ucząc się na tylko 63 grzybach jest w stanie poprawnie sklasyfikować 97% z ponad 8 tysięcy. Druga część tezy okazuje się poprawna - Dla modelu zbudowanego na jednym grzybie - wszystkie będą klasyfikowane jako jadalne lub wszystkie jako trujące - stąd dokładność rzędu 0.5. Warto również zauważyć, że model budowany na wszystkich wierszach poprawnie klasyfikuje wszystkie wiersze. Nie jest to zaskakujące, ale pokazuje, że drzewo budowane i odtwarzane jest bezbłędnie.

Czego nauczyliśmy się podczas realizacji projektu?

- Pracy na zbiorach danych w Pythonie przy pomocy biblioteki Pandas
- Budowy i implementacji drzew decyzyjnych ID3
- Odstępowania od obiektowej struktury projektu tam gdzie nie jest to potrzebne
- Analizy zagregowanych wyników testów