Probabilistyczne uczenie maszynowe
Projekt 1

Michał Maternik, Maksym Telepchuk

7 maja 2020

Streszczenie

Zadanie polegało na przeprowadzeniu eksploracyjnej analizy wybranego zbioru danych oraz implementacji dwóch modeli probabilistycznych, w tym graficznego modelu probabilistycznego. Zostało ono zrealizowane dla zbioru dotyczącego grzybów, na podstawie którego przewidujemy czy grzyb jest jadalny czy nie.

1 Eksploracyjna analiza danych

Wybrany model danych dotyczy grzybów, które opisane są 23 cechami wyszczególnionymi poniżej. Ten zestaw danych zawiera opisy hipotetycznych próbek odpowiadające 23 gatunkom grzybów. Każdy gatunek jest identyfikowany jako zdecydowanie jadalne, zdecydowanie trujące lub o nieznanej jadalności. Ta ostatnia klasa została połączona z trującymi w jedną. Eksperty wyraźnie stwierdzają, że nie ma prostej zasady określania jadalności grzyba, jednak są cechy grzybów, które są bardziej lub mniej skorelowane między sobą.

Poniżej znajdują się statystyki cech.

	Column	Non-Null Count	Dtype
0	class	8124 non-null	category
1	cap-shape	8124 non-null	category
2	cap-surface	8124 non-null	category
3	cap-color	8124 non-null	category
4	bruises?	8124 non-null	category
5	odor	8124 non-null	category
6	gill-attachment	8124 non-null	category
7	gill-spacing	8124 non-null	category
8	gill-size	8124 non-null	category
9	gill-color	8124 non-null	category
10	stalk-shape	8124 non-null	category
11	stalk-root	5644 non-null	object
12	stalk-surface-above-ring	8124 non-null	category
13	stalk-surface-below-ring	8124 non-null	category
14	stalk-color-above-ring	8124 non-null	category
15	stalk-color-below-ring	8124 non-null	category
16	veil-type	8124 non-null	category
17	veil-color	8124 non-null	category
18	ring-number	8124 non-null	category
19	ring-type	8124 non-null	category
20	spore-print-color	8124 non-null	category
21	population	8124 non-null	category
22	habitat	8124 non-null	category

Liczebność zbioru wynosi 8124 egzemplarze. Wszystkie kolumny mają charakter kategoryczny.

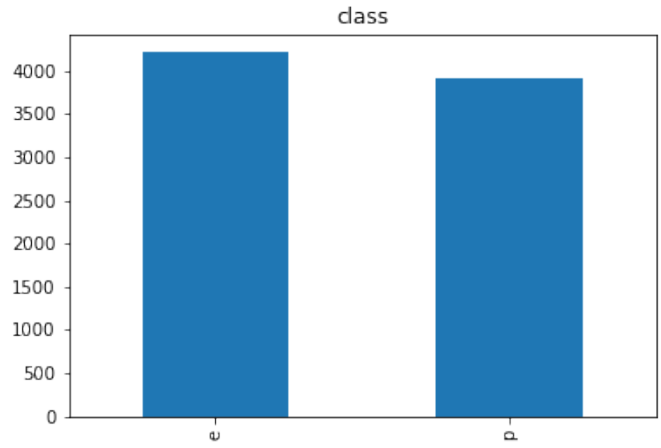
Znaczenie dostępnych kolumn jest następujące:

- class - jadalność (jadalny, niejadalny)

- cap-shape - kształt kapelusza (dzwon, płaski, wcięty)
- cap-surface - powierzchnia kapelusza (włóknista, łuszcząca się, gładka)
- cap-color - kolor kapelusza (żółty, biały...)
- bruises? - czy grzyb jest siniakiem? (tak / nie)
- odor - zapach: (migdałowy, ostry, pikantny, nijaki)
- gill-attachment - sposób przyrastania blaszek do trzonu (przyrośnięte, wolne)
- gill-spacing - odstęp między blaszkami (zbliżone, oddalone)
- gill-size - szerokość blaszki (szeroki, wąski)
- gill-color - kolor blaszki (czarny, brązowy, biały)
- stalk-shape - kształt trzonu (pogrubiający się, zwężający się)
- stalk-root - korzeń trzonu: (bulwiasty, stożkowy)
- stalk-surface-above-ring - typ powierzchni trzonu nad pierścieniem (włóknista, łuskowata, gładka)
- stalk-surface-below-ring -typ powierzchni trzonu poniżej pierścienia(włóknista, łuskowata, gładka)
- stalk-color-above-ring - kolor trzonu nad pierścieniem (brązowy, szary ...)
- stalk-color-below-ring - kolor trzonu poniżej pierścienia (brązowy, szary ...)
- veil-type - typ osłony grzyba (częściowa, całkowita)
- veil-color - kolor osłony grzyba (brązowy, biały . . .)
- ring-number - liczba pierścieni (brak, jeden, dwa)
- ring-type - typ pierścienia (z pajęczyną, zanikający, rozszerzający się, duży, brak, zwisający, ślad pierścienia)
- spore-print-color - kolor wysypu zarodników: (czarny, brązowy, buff, czekoladowy, zielony, pomarańczowy, fioletowy, biały, żółty)
- population - typ populacji (obfita, skupiona, liczna, rozproszona, rzadka, samotna)

- habitat - siedlisko (trawy, liście, łąki, ścieżki, urban, odpady, lasy)

Kolumna class, która oznacza, czy grzyb jest jadalny, będzie stanowiła wyznaczaną w modelach wartość.



Rysunek 1: Kolumna Class

Z jej rozkładu widać, że zbiór jest pod jej względem dobrze zrównoważony i jego dodatkowe balansowanie przy wykorzystaniu jednej z wielu dostępnych metod nie było konieczne.

Zaobserwowano, że spośród pozostałych cech kolumna 'stalk-root' posiadała brakujące wartości i dlatego została usunięta. Sugerowane jest użycie mechanizmów, takie jak maski, które uwzględniają wartości brakujące.

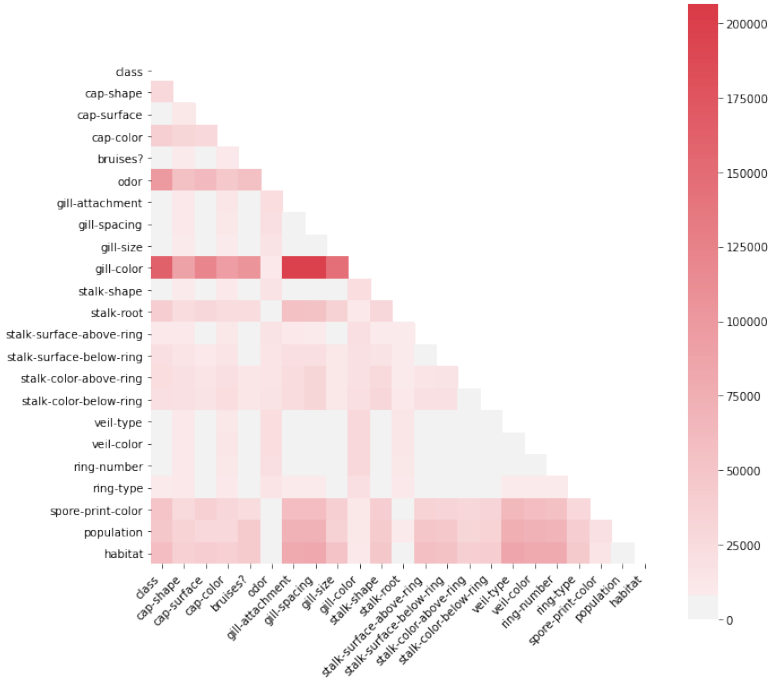
Zauważono także, że kolumna 'veil-type' posiadała tylko jedną wartość, stąd byłaby więc przydatna przy klasyfikacji i ze względu na to również została usunięta.

Żeby zbadać korelację pomiędzy cechami kategorycznymi, został użyty Chi-square test,który służy do ustalenia, czy istnieje statystycznie istotna różnica między oczekiwanymi częstotliwościami a obserwowanymi częstotliwościami w jednej lub więcej kategorii tabeli nieprzewidzianych zdarzeń. Wyraża się on wzorem

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - np_i)^2}{np_i}$$

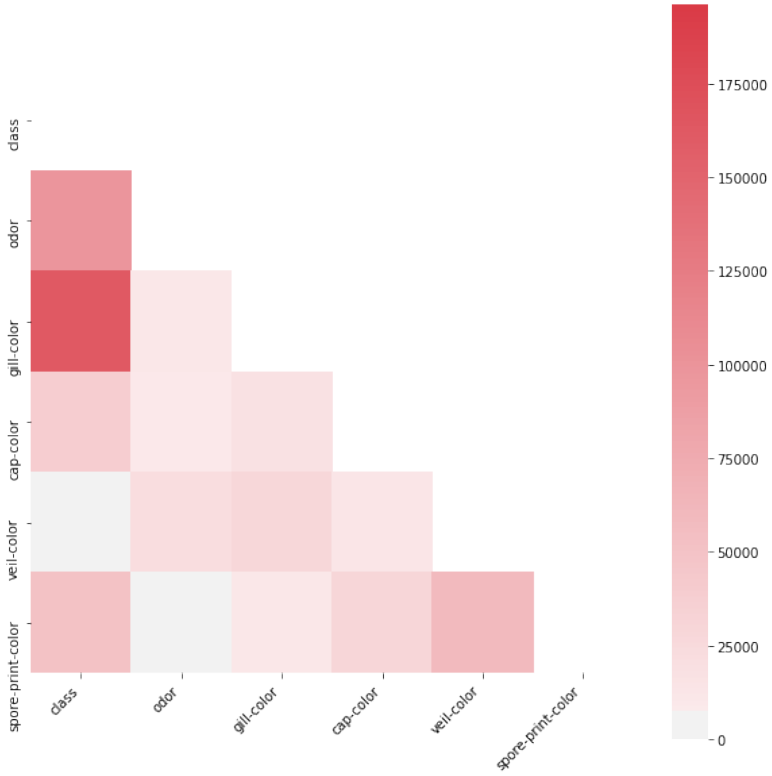
gdzie n - liczba obserwacji, p_i - prawdopodobieństwo występowania kategorii i w zbiorze, x_i - liczba, która odpowiada kategorii i.

Na rys. 3 są przedstawione wyniki przeprowadzenia Chi-square testu dla każdej z kategorii.

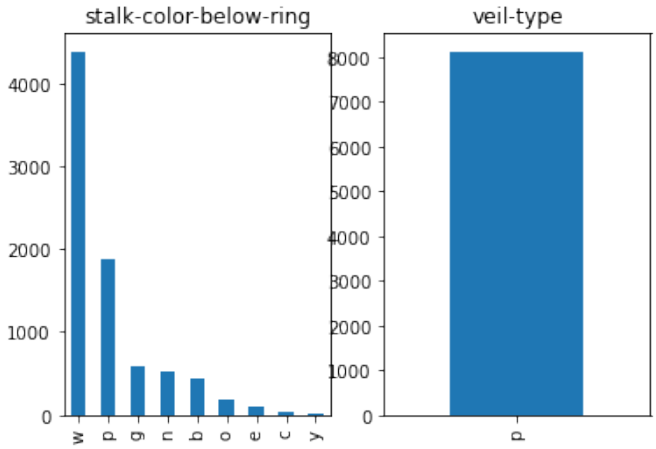


Rysunek 2: Korelacja pomiędzy cechami.

Z wykresu widać, że dużo cech zależą od koloru blaszki. Z kolei jadalność grzyba najwięcej jest skorelowana z kolorem różnych składowych oraz zapachu grzyba.

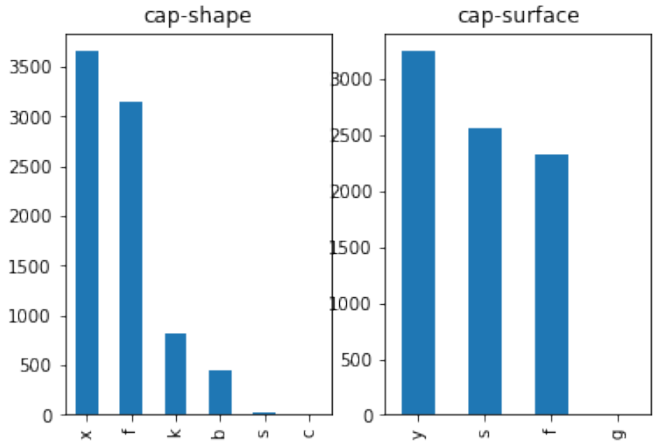


Rysunek 3: Korelacja pomiędzy cechami koloru, jadalności i zapachu.

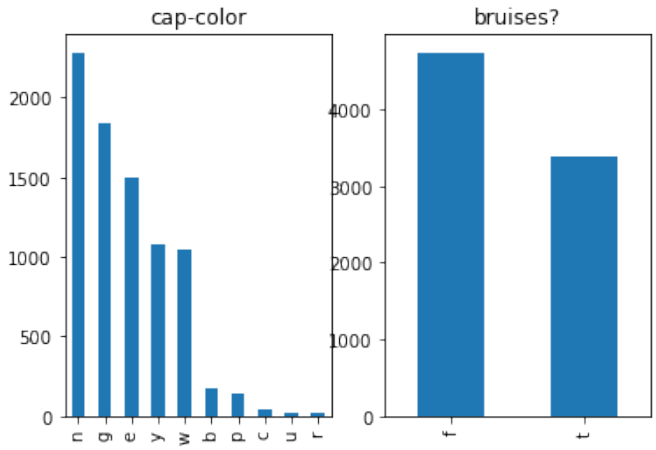


Rysunek 4: Kolumny 'stalk-color-below-ring' oraz 'veil-type'

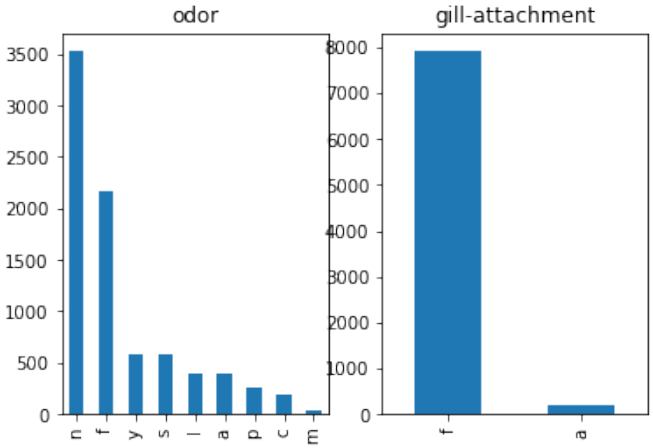
W przypadkach innych cech nie wystąpiły wartości brakujące.



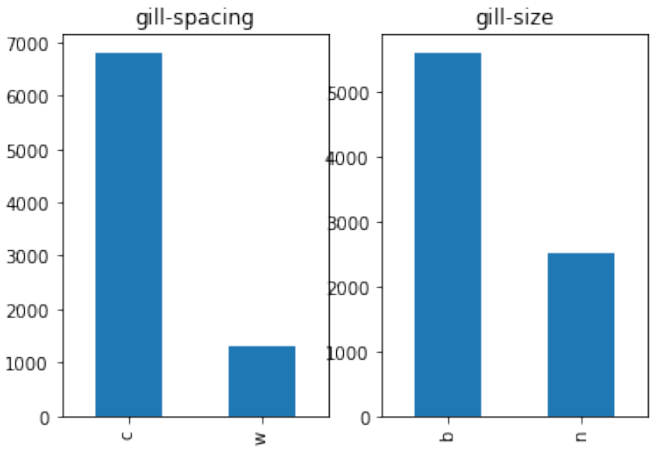
Rysunek 5: Kolumny 'cap-shape' oraz 'cap-surface'



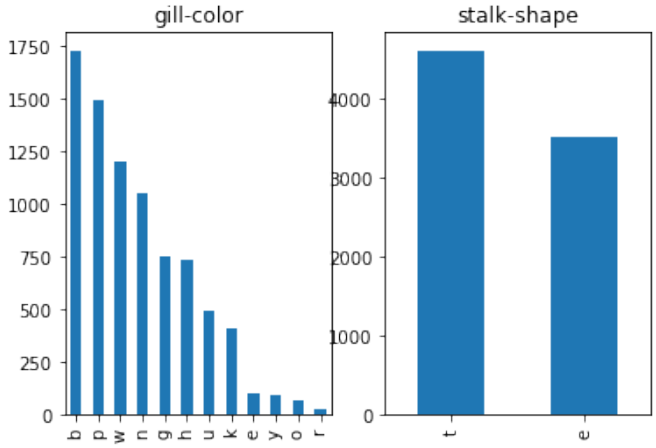
Rysunek 6: Kolumny 'cap-color' oraz 'bruises?'



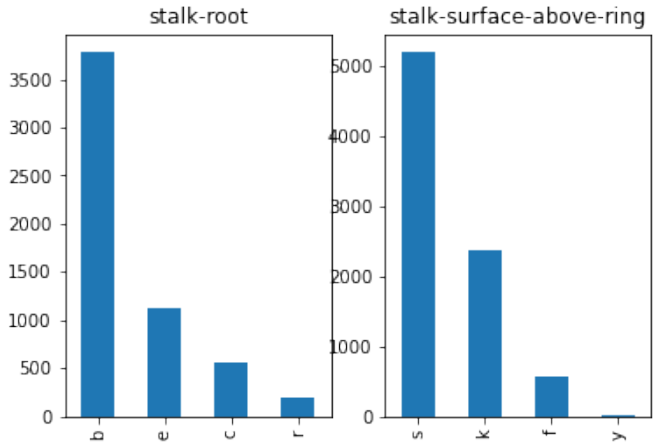
Rysunek 7: Kolumny 'odor' oraz 'grill-attachment'



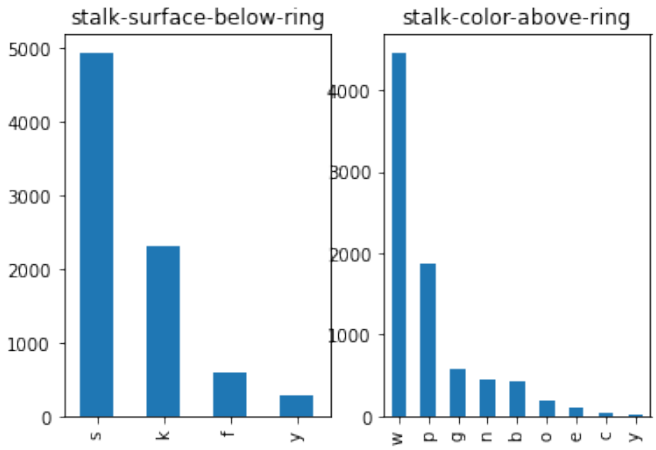
Rysunek 8: Kolumny 'grill-spacing' oraz 'grill-size'



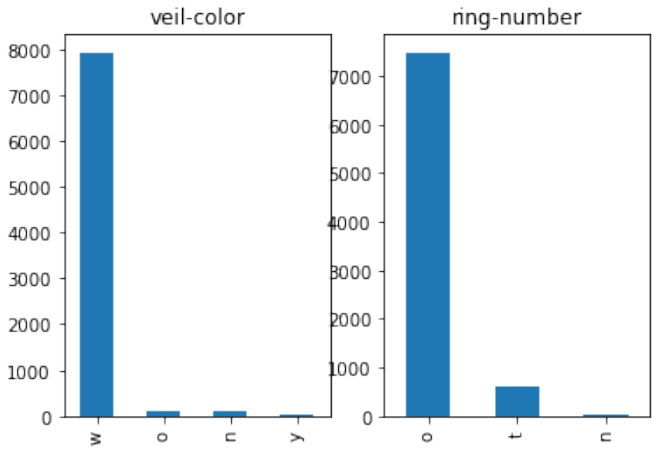
Rysunek 9: Kolumny 'grill-color' oraz 'stalk-shape'



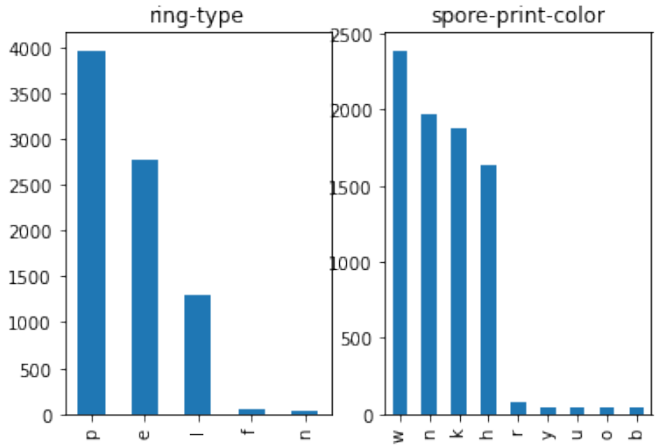
Rysunek 10: Kolumny 'stalk-root' oraz 'stalk-surface-above-ring'



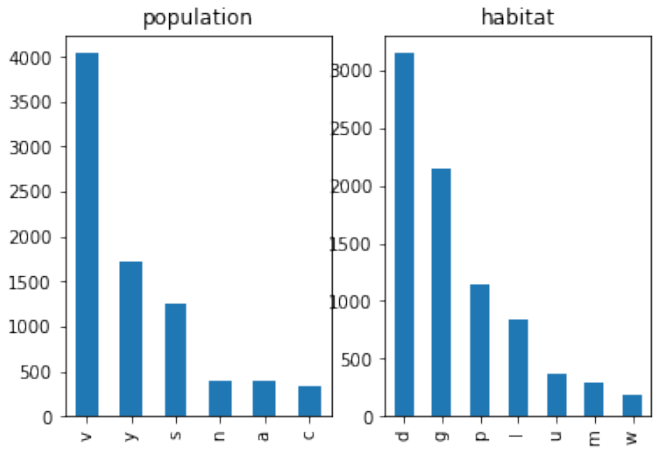
Rysunek 11: Kolumny 'stalk-surface-below-ring' oraz 'stalk-color-above-ring'



Rysunek 12: Kolumny 'veil-color' oraz 'ring-number'



Rysunek 13: Kolumny 'ring-type' oraz 'spore-print-color'



Rysunek 14: Kolumny 'population' oraz 'habitat'

2 Modele

2.1 Naive Bayes

Wybranym prostym modelem probabilistycznym był Naive Bayes. Wykorzystana została jego implementacja w pyro, wykorzystująca mechanizm pyro.infer.SVI oraz TraceEnum_ELBO. Predykcja wykonywana jest za pomocą pyro.infer.Predictive.

Zastosowana implementacja obsługuje wyłącznie kolumny kategoryczne, ponieważ tylko takie występowały w wykorzystywanym zbiorze danych; była więc wystarczająca dla naszych potrzeb.

2.2 Probabilistic Graphical Model - Bayesian Network

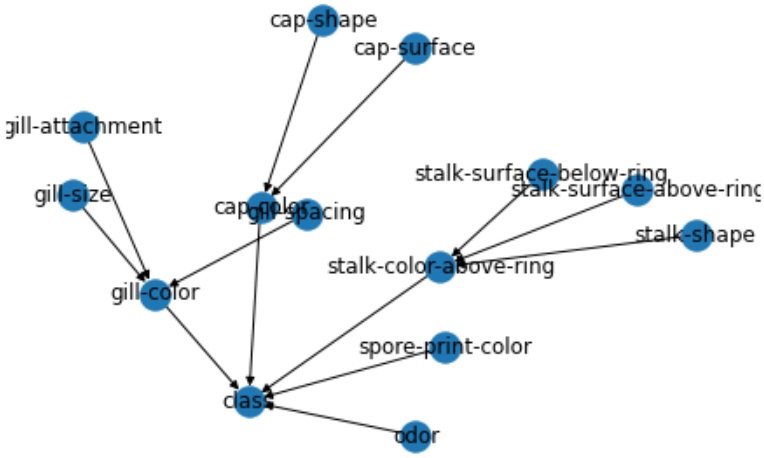
Spośród probabilistycznych modeli graficznych zastosowany został skierowany model sieci Bayesowskiej.

Poznanie struktury sieci bayesowskich może być skomplikowane z dwóch głównych powodów : trudności w wnioskowaniu o przyczynowości i bardzo dużej liczby możliwych skierowanych krawędzi, które mogłyby istnieć w zbiorze danych. Problem pojawia się, gdy algorytm uczenia struktury bierze pod uwagę tylko korelację lub inną miarę współwystępowania w celu ustalenia, czy krawędź powinna istnieć. W części eksploracji danych zostało pokazane, że nie zawsze z tego da się skutecznie takie zależności wywnioskować.

Po rozmowie z ekspertem w dziedzinie grzybów, został sformułowany model, który zdaniem eksperta może być skuteczny w klasyfikacji jadalności grzyba. Zależności pomiędzy cechami zaproponowane przez eksperta są podane na rys. 15

Naiwnym podejściem w przypadku znajdywania takich zależności byłoby przeszukiwanie wszystkich możliwych ukierunkowanych grafów acyklicznych i identyfikowanie tego grafu, który minimalizuje funkcje celu. Naiwna implementacja tego wyszukiwania ma złożoność obliczeniową nadwykładniczą i staje się niemożliwa jeśli weźmie się pod uwagę nawet kilkanaście zmiennych. Jednak programowanie dynamiczne może skutecznie usunąć wiele powtarzanych obliczeń i zredukować je do wykładniczego czasu.

Biblioteka **pomegranate** daje możliwość przeprowadzenie takiego wyszukiwania za pomocą algorytmu A* oraz programowania dynamicznego, jednak w przypadku tego zbioru danych to jest bardzo czasochłonne, wyszukiwanie struktury zajmuje godziny czasu.



Rysunek 15: Struktura zaproponowana przez eksperta

Następnym typem wyszukiwania struktury jest podejście zachłanne. Metoda zachłannego wyszukiwania iteracyjnie znajduje najlepszą zmienną, którą można dodać do rosnąco posortowanego porządku topologicznego, umożliwiając nowej zmiennej rysowanie tylko ze zmiennych znajdujących się już w porządku topologicznym. Ma dobrą równowagę między tworzeniem dobrych (często optymalnych) wykresów a niewielkim kosztem obliczeniowym i powierzchnią pamięci. Nie ma jednak gwarancji, że dzięki temu powstanie optymalny globalnie wykres.

Jednak nawet takie względnie szybkie podejście jest nie odpowiednie w przypadku tego zbioru danych. Algorytm znajduje dużo połączeń pomiędzy zmiennymi, co powoduje dużą ilość parametrów, które się nie mieszczą w pamięci komputerowej

Finalnie, aby uzyskać dobre struktury w rozsądnym czasie został użyty algorytm budowy drzewa Chow-Liu [1], która uczy się optymalnego drzewa na podstawie danych. Zasadniczo algorytm oblicza wzajemną informację między wszystkimi parami zmiennych, a następnie znajduje maksymalne drzewo rozpinające. Algorytmem jest o złożoności $O(d^2)$ (d - liczba atrybutów) i praktycznie jest szybki i efektywny pod względem pamięci, choć tworzy struktury o gorszym prawdopodobieństwie warunkowym.

Model sieci Bayesowskich został zaimplementowany za pomocą pyro w klasie PGM. Do tej klasy jest przekazywany graf skierowany (musi być być acykliczny) z

zależnościami między zmiennymi. Na podstawie tego grafu są inicjalizowane parametry w pyro, które są uczone iteracyjnie przy pomocy SVI. Podczas trenowania wszystkie zmienne są obserwowane, natomiast podczas testowania, cecha klasy nie jest obserwowana oraz jest wyliczana za pomocą klasy Predictive na podstawie wytrenowanych parametrów.

3 Eksperymenty

Scenariusz testowy został przygotowany w ten sposób, że 80 procent danych przeznaczonych zostało do przeprowadzenia uczenia, a pozostałe 20 procent do testów uzyskanego modelu.

Wykorzystane zostały miary accuracy, precision, recall oraz F1 .

Każda z nich liczona było pięciokrotnie w procesie stratyfikowanej krosswalidacji, a następnie uzyskane wyniki zostały uśrednione.

3.1 Naive Bayes

Algorytm Naive Bayes sprawdził się w tym przypadku bardzo dobrze uzyskując wyniki poszczególnych miar na poziomie 96 procent.

Wyniki na danych treningowych:

	precision	recall	f1-score	support
0	0.97	0.96	0.96	3133
1	0.96	0.97	0.97	3366
accuracy			0.96	6499
macro avg	0.96	0.96	0.96	6499
weighted avg	0.96	0.96	0.96	6499

Wyniki na danych testowych:

	precision	recall	f1-score	support
0	0.96	0.95	0.95	783
1	0.95	0.96	0.96	842
accuracy			0.95	1625
macro avg	0.95	0.95	0.95	1625
weighted avg	0.95	0.95	0.95	1625

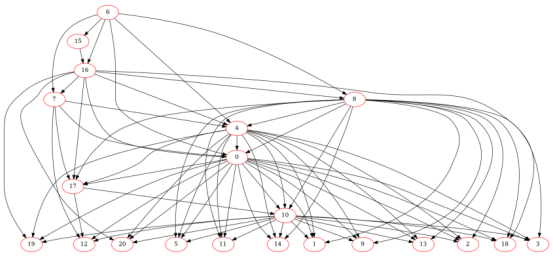
Wyniki uzyskane na danych testowych są na tym samym poziomie, co dla danych treningowych, co sugeruje że uzyskany model dobrze klasyfikuje przydatność wskazanego grzyba do spożycia.

3.2 Bayesian Network

Zależności pomiędzy atrybutami zostały wyliczone za pomocą biblioteki **pomegranate**.

3.2.1 Podejście zachłanne

Na początku zostało użyte podejście zachłanne, które wygenerowało strukturę zwizualizowaną na rysunku 16.

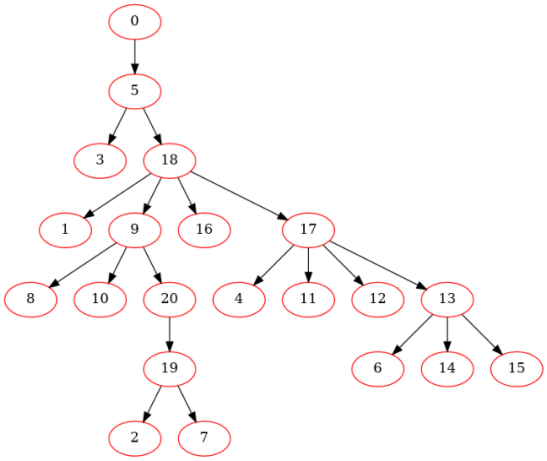


Rysunek 16: Zachłanne podejście

Policzenie tej struktury zajęło 25 min, oraz model nie nadaję się na uczenie ze względu na to, że parametry dla takiego modelu nie mieszczą się w posiadanej pamięci komputerowej.

3.2.2 Podejście aproksymacyjne

W tym eksperymencie został użyty algorytm Chow Liu. Policzenie struktury modelu zajmuje od 2 do 10 sekund. Zależności, które są pozyskiwane, są rzadkie i łatwo czytelne. Wizualizacja struktury jest podana na rys. **??**. Wierzchołek 0 odpowiada za klasę (jadalny/niejadalny), natomiast atrybut 5 odpowiada za zapach.



Rysunek 17: Algorytm Chow Liu

Wyniki na danych treningowych:

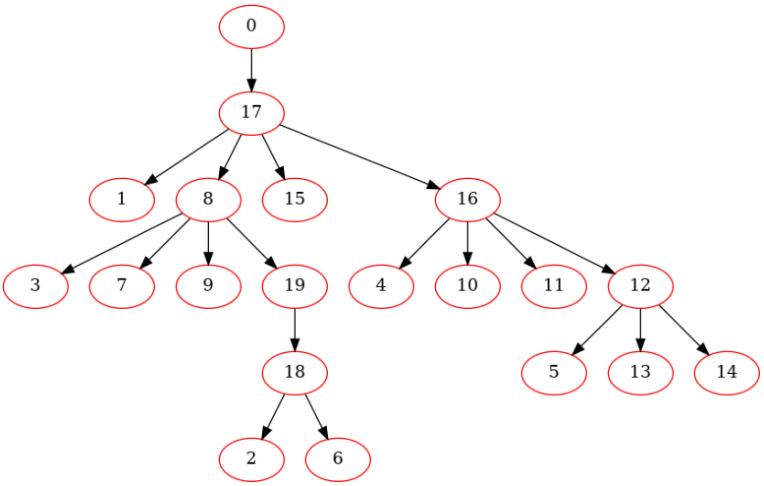
	precision	recall	f1-score	support
0	0.96	0.93	0.94	3147
1	0.93	0.96	0.95	3378
accuracy			0.94	6525
macro avg	0.95	0.94	0.94	6525
weighted avg	0.94	0.94	0.94	6525

Wyniki na danych testowych:

	precision	recall	f1-score	support
0	0.95	0.93	0.94	769
1	0.94	0.96	0.95	830
accuracy			0.94	1599
macro avg	0.94	0.94	0.94	1599
weighted avg	0.94	0.94	0.94	1599

3.2.3 Podejście aproksymacyjne dla modelu bez atrybutu kluczowego

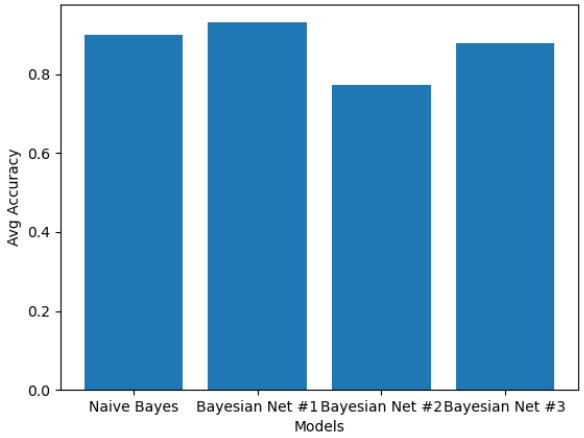
Z powyższego wynika, że jeden z atrybutów jest kluczowy, czyli taki, na podstawie którego można bardzo precyzyjnie wyznaczyć jadalność grzyba. Jeżeli policzyć tym samym algorytmem zależności bez tego atrybutu, to atrybut spore-print-color (17) jest uznawany za algorytm jako jedyny, od którego zależy klasa jadalności. Przy takim uproszczeniu taki model będzie miał gorsze wyniki, jednak wykazuje to, że algorytm wykrył tylko najbardziej istotne zależności i taki model jest tylko przybliżeniem rzeczywistości.



Rysunek 18: Model uzyskany przez algorytm Chow Liu bez cechy zapachu

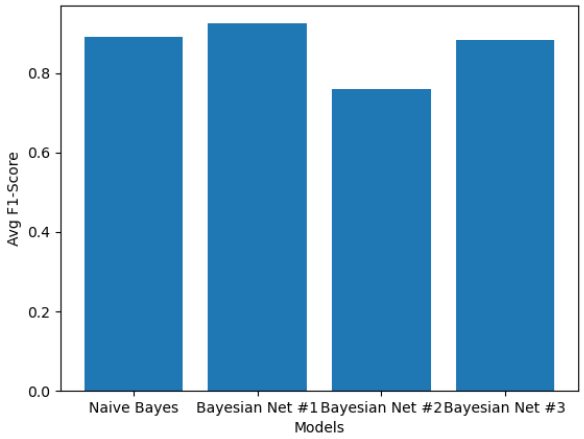
3.3 Porównanie modeli

Każdy z modeli został przetestowany ujednoliconym procesem polegającym na wykonaniu stratyfikowanej pięciokrotnej kroswalidacji. Wartości metryk zostały zebrane i uśrednione, a następnie zwizualizowane na diagramach.



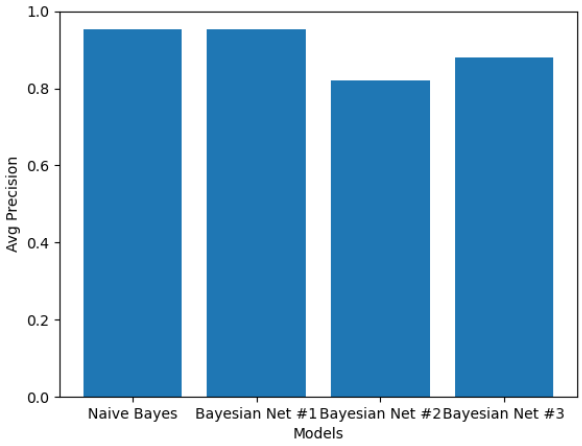
Rysunek 19: Porównanie metryk Accuracy dla poszczególnych modeli

Pod względem dokładności najlepiej wypadła sieć uzyskana za pomocą pomegrenade, za nią uplasował się algorytm Naive Bayes. Pozostałe sieci, w tym uzyskana na podstawie wskazówek od naszego eksperta nie były aż tak dobre.



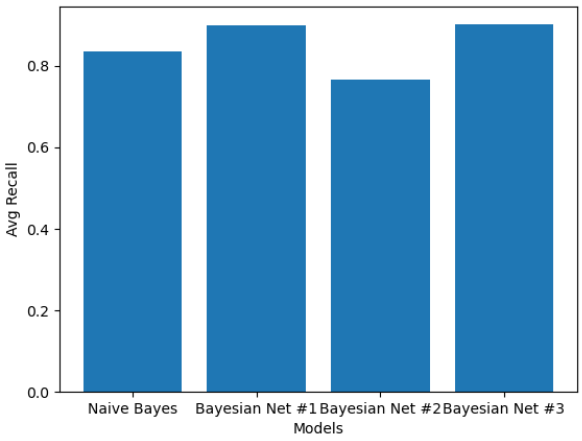
Rysunek 20: Porównanie metryk F1-Score dla poszczególnych modeli

Porównanie za pomocą metryki F1-score wypada podobnie jak dla dokładności. Ponownie najlepsza okazała się sieć uzyskana za pomocą pomegranade.



Rysunek 21: Porównanie metryk Precision dla poszczególnych modeli

Pod względem precyzji nasze dwa pierwsze modele uzyskały bardzo zbliżone wyniki.



Rysunek 22: Porównanie metryk Recall dla poszczególnych modeli

Przy porównaniu pod względem metryki Recall zauważamy, że w tym przypadku niedoceniana dotychczas sieć zbudowana zgodnie ze wskazówkami naszego eksperta uzyskuje wyniki lepsze od modelu naiwnego Bayesa i zbliżone do sieci

uzyskanej za pomocą pomegrenade.

4 Wnioski

Podsumowując możemy uznać, że najlepszy model udało nam się uzyskać wykorzystując probailistyczny model graficzny zbudowany według grafu zasugerowanego przez bibliotekę pomegrenade za pomocą algorytmu 'chow-liu'. Jest to najprostszy z oferowanych algorytmów, ale mimo to okazał się w tym przypadku bardzo skuteczny.

Jeśli chodzi o przygotowanie danych wejściowych dla modeli, to w retrospekcji zauważamy, że amiast usuwać kolumnę z brakującymi wartościami można było przyjąć inne rozwiązanie polegające na zastosowaniu maskowania brakującej wartości za pomocą poutine.mask. Takie podejście zastosujemy prawdopodobnie w kolejnych budowanych modelach.

Literatura

[1] https://en.wikipedia.org/wiki/Chow%E2%80%93Liu_tree.

[2] https://archive.ics.uci.edu/ml/datasets/Mushroom.