Projekt z przedmiotu Eksploracja Danych
Drugi etap: Zrozumienie problemu + Zrozumienie danych
Breast Cancer Wisconsin (Diagnostic) Data Set
Michał Sieczczyński

Charakterystyka zbioru danych

Pochodzenie:
https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Format:
.csv

Liczba przykładów:
569

Ilość zbiorów danych:
1

Cele eksploracji i kryteria sukcesu

Celem eksploracji jest predykcja, który pacjent ma nowotwór złośliwy a który łagodny. Dodatkowym celem jest określenie które atrybuty mają największy wpływ na predykcję. Sukces zostanie osiągnięty, jeżeli model uzyska czułość na poziomie 80% a swoistość na poziomie 60%.

Dyskusja kroków dalszego postępowania

Dobór działania eksploracji
Celem eksploracji jest predykcja, który pacjent ma nowotwór złośliwa, a który łagodny, co odpowiada problemu klasyfikacji binarnej atrybutu diagnosis, przyjmującego wartość “M” (z ang. malignant) dla pacjentów z nowotworem złośliwym oraz “B” (z ang. benign) dla pacjentów z nowotworem łagodnym. Dodatkowo w celu określenia, które atrybuty mają największy wpływ na predykcję zostaną użyte metody cechujące się łatwą wyjaśnialnością.

Dobór algorytmu eksploracji
W celu predykcji atrybutu diagnosis zostanie wykorzystany algorytm drzewa decyzyjnego. Algorytm ten charakteryzuje się łatwą wyjaśnialnością - jesteśmy w stanie określić konkretne warunki jakie sprawdza model, aby dokonać klasyfikacji, a tym samym wskazać cechy, które mają największy wpływ na predykcję.
Dodatkowo w celach eksperymentalnych zostaną wykorzystane algorytmy lasu losowego oraz zbalansowanego lasu losowego. Algorytm lasu losowego to zbiór drzew decyzyjnych trenowanych na podzbiorze losowo wybranych rekordów (losowanie ze zwracaniem - tzw. bootstrap) oraz podzbiorze losowo wybranych cech. Ostateczna predykcja wybiera jest na zasadzie głosowania - najczęściej wskazywana klasa jest ostateczną predykcją. Z kolei zbalansowany las losowy to wersja lasu losowego dostosowana do niezbalansowanych danych. Główną różnicą jest losowanie podzbioru rekordów dla każdego drzewa, w taki sposób, aby ilości próbek poszczególnych klas były identyczne. Są to metody mniej interpretowalne od pojedynczego drzewa decyzyjnego - ścieżka decyzyjna algorytmów jest

dużo bardziej skomplikowana. Jednakże istotność cech dla lasów losowych wyznacza się poprzez znormalizowany, sumaryczny zysk informacyjny danej cechy względem wszystkich drzew, co daje możliwość określenia najbardziej znaczących atrybutów dla predykcji.

Dobór metody testowania wyników
Zbiór danych zawiera 569 rekordów, co wydaje się być wystarczającą liczbą do podziału danych na zbiór testowy i treningowy. Jednakże pojedynczy podział na zbiór testowy i treningowy w sposób losowy może wpływać na wyniki modelu w zależności od wylosowanych próbek. Przykładowo, jeżeli do zbioru testowego zostanie wylosowane więcej próbek prostych w klasyfikacji - wyniki predykcji mogą być zawyżone. Dlatego w celu zwiększenia wiarygodności wyników zostanie użyta kroswalidacja 10-krotna. Trenowanie 10 modeli na 90% danych i testowanie na 10% zapewni zarówno dobrą generalizację ze względu na duży zbiór treningowy jak i wiarygodną ocenę wyników ze względu na zmiany zbioru testowego i uśrednienie wyników.

Przygotowanie danych

Dane brakujące i dane do ujednolicenia
Brak

Zamiana na nominalne/numeryczne
Brak

Podzbiór danych
W analizowanych danych występuje kilka grup atrybutów charakteryzujących się wysoką korelacją między sobą - pierwsza grupa: radius_mean, radius_worst, perimeter_mean, perimeter_worst, area_mean, area_worst, druga grupa: radius_se, area_se, perimeter_se, trzecia grupa: concavity_mean, concavity_worst, concavity_se, concave points_mean, concave points_worst, concave points_se, compactness_worst, compactness_mean, compactness_se. Ze względu na to z każdej grupy pozostawiony zostanie jeden atrybut z największą korelacją do atrybutu celu, czyli diagnosis. Będą to atrybuty odpowiednio: perimeter_worst, radius_se oraz concave points_worst.

Uzupełnienie danych
Dotychczasowe analizy wykazały, że pacjenci z nowotworem łagodnym występują częściej w danych niż pacjenci z nowotworem złośliwym. Zatem klasy nie są zbalansowane. Pierwsze pytanie jakie się nasuwa to czy tak powinno być - czy faktycznie nowotwory łagodne występują częściej niż nowotwory złośliwe. Okazuje się, że tak - szacuje się, że około 60-80% wykrywanych nowotworów to nowotwory łagodne. Zatem lepiej byłoby dla klasyfikatora, aby nauczył się rzeczywistej proporcji w danych, która w tym przypadku jest względnie zachowana (63% rekordów to pacjenci z nowotworem łagodnym). Zatem dane nie zostaną uzupełnione - istotne jest jedynie, aby wagi poszczególnych klas odpowiadające ich liczebności zostały podane do modelu.
Natomiast, gdyby występowanie klas powinno być równoliczne (co nie jest obecne w tym przypadku), wtedy można usunąć część danych lub zastosować augmentację danych (np. algorytm SMOTE), aby faktycznie proporcja klas była taka sama.

Utworzenie modelu

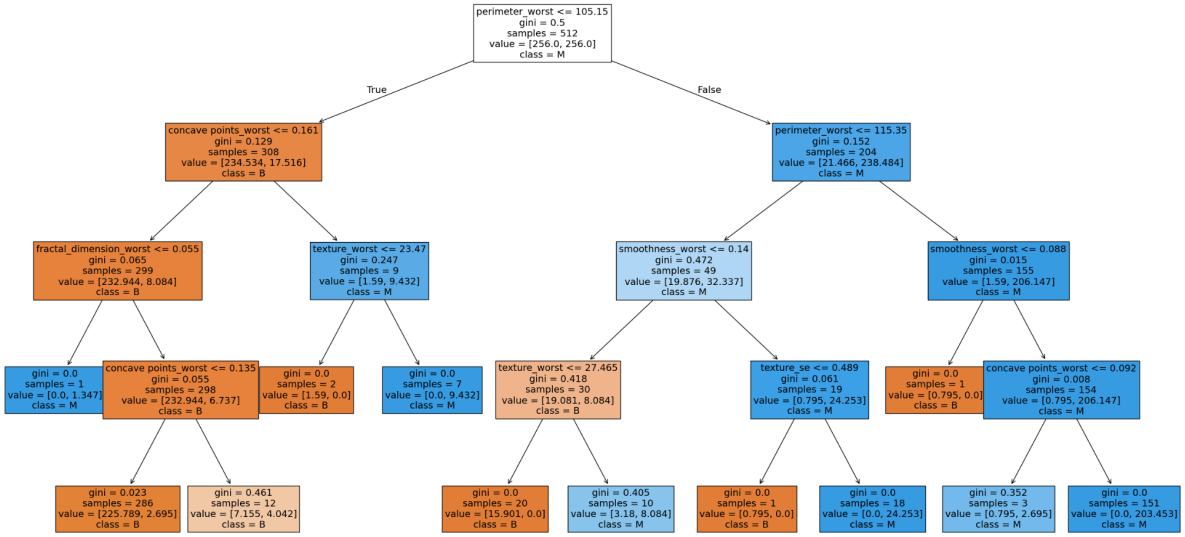
Model drzewa decyzyjnego został utworzony w Pythonie za pomocą biblioteki *scikit-learn* a dokładniej funkcji *DecisionTreeClassifier*. W przypadku hiperparametrów modelu - został ustawione ziarno do losowania (*random_state*) na 0, co zapewnia powtarzalność wyników, także zostały ustawione wagi klas względem ich liczebności (*class_weight="balanced"*). Dodatkowo zostały sprawdzone kilka wartości parametru określającego maksymalną głębokość drzewa - maksymalną liczbę poziomów od korzenia drzewa do najdalszego liścia. *Tabela 1* przedstawia wyniki wstępnych eksperymentów. Model został przetestowany za pomocą 10-krotnej kroswalidacji. Zostały zebrane i uśrednione metryki czułości i swoistości w postaci średnia +- odchylenie standardowe. Niskie wartości miały ograniczać głębokość drzewa i przeciwdziałać nadmiernemu przetrenowaniu modelu. Dodatkowo została sprawdzona wartość None, która oznacza brak ograniczeń w głębokości drzewa. Najwyższa średnia czułość została osiągnięta dla głębokości drzewa równej 4 i 2, jednakże dla wartości 4 osiągnięto wyższą średnią swoistość. Dlatego do dalszej analizy będę wykorzystywał *max_depth*=4.

Tabela 1 Wyniki wstępnych eksperymentów z parametrem określającym maksymalną głębokość drzewa decyzyjnego

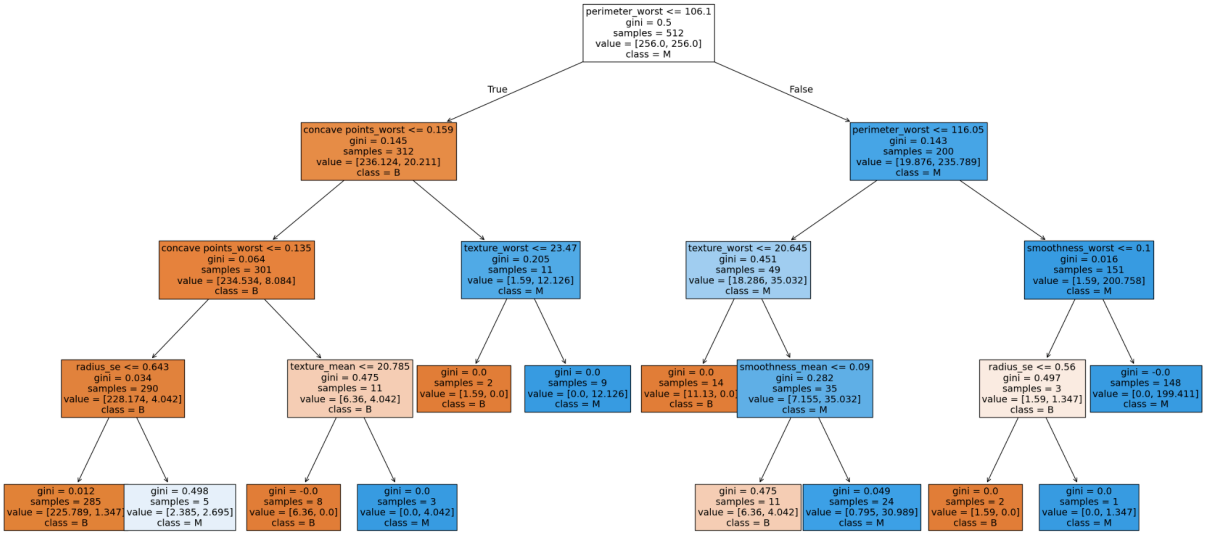
Maksymalna głębokość drzewa (*max_depth*)	Czułość (średnia +- odchylenie standardowe)	Swoistość (średnia +- odchylenie standardowe)
1	0,92 +- 0,05	0,90 +- 0,07
2	0,93 +- 0,07	0,89 +- 0,07
3	0,91 +- 0,06	0,93 +- 0,04
4	0,93 +- 0,05	0,94 +- 0,05
5	0,92 +- 0,05	0,94 +- 0,05
None	0,92 +- 0,04	0,92 +- 0,04

Zatem podstawowa wersja algorytmu ma parametry *random_state=0*, *class_weight="balanced"* oraz *max_depth=4* oraz metryki **czułość=0.93 +- 0,05** i **swoistość=0,94 +- 0,05**. W związku z tym, że model był uruchamiany na 10-krotnej kroswalidacji, zostało utworzone 10 modeli, czyli 10 drzew decyzyjnych. Poniżej prezentuję wykresy danych 10 drzew decyzyjnych.

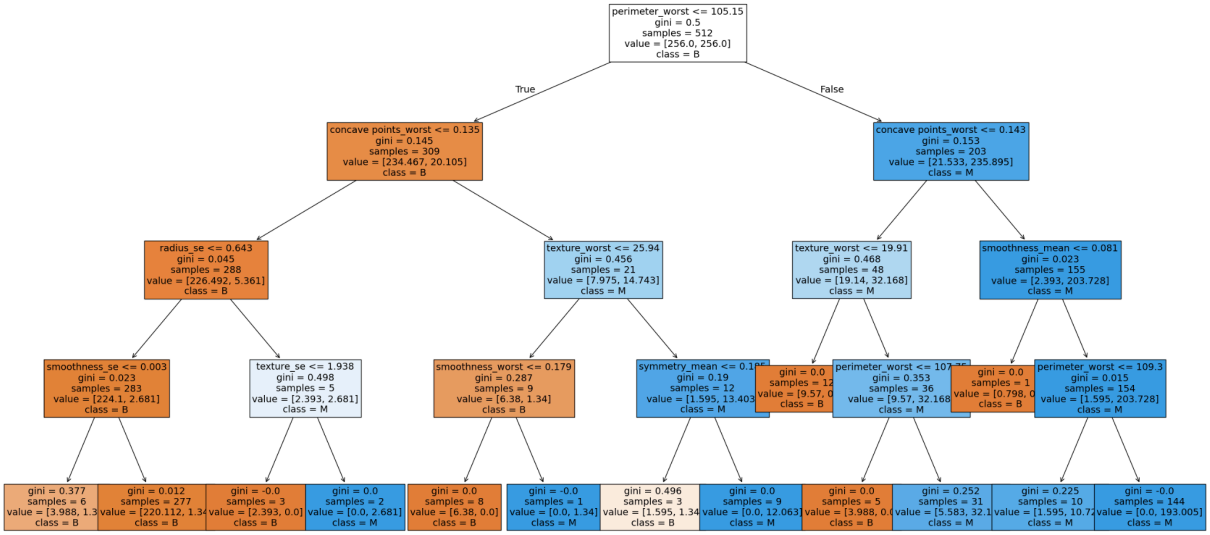
Decision Tree from Fold 1



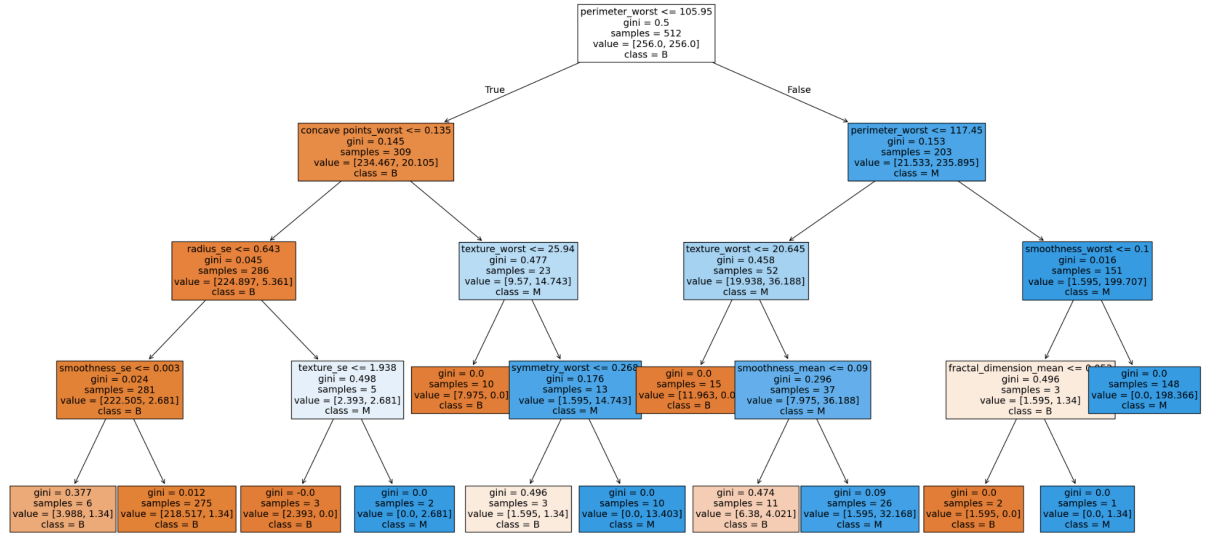
Decision Tree from Fold 2



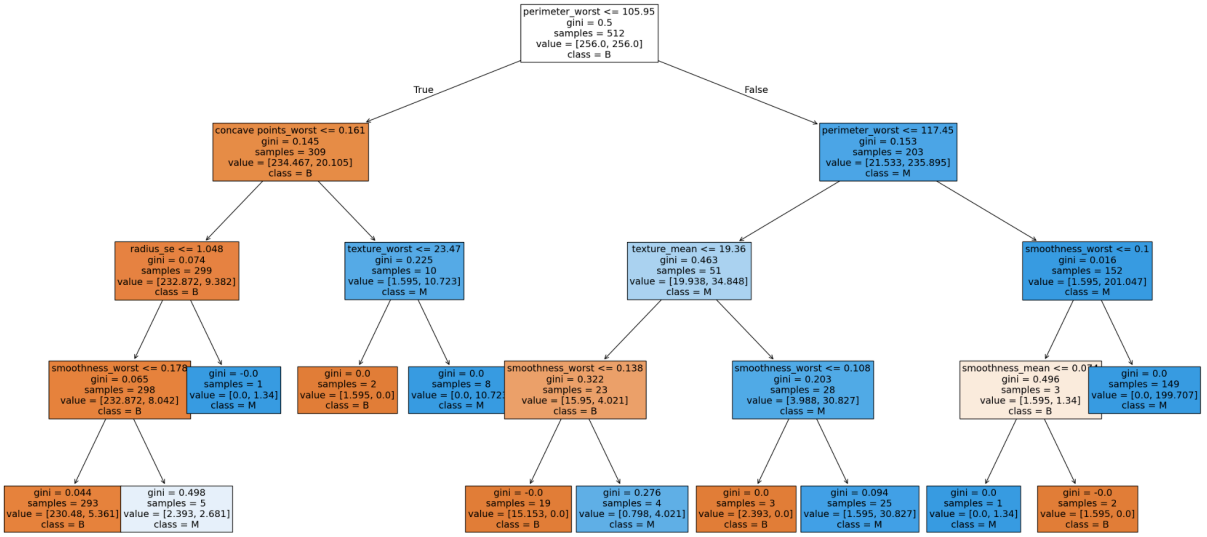
Decision Tree from Fold 3



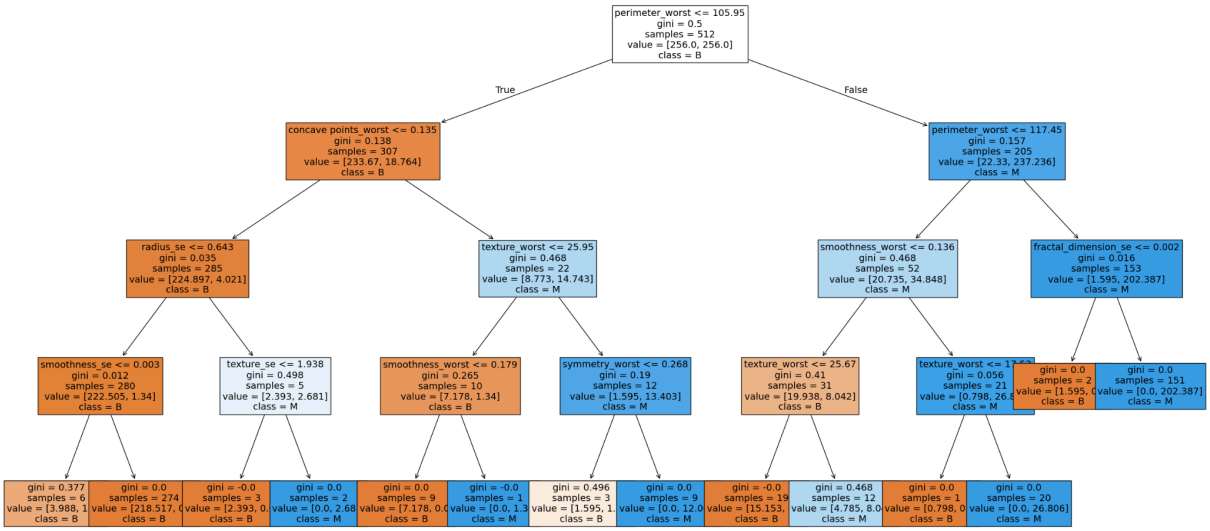
Decision Tree from Fold 4



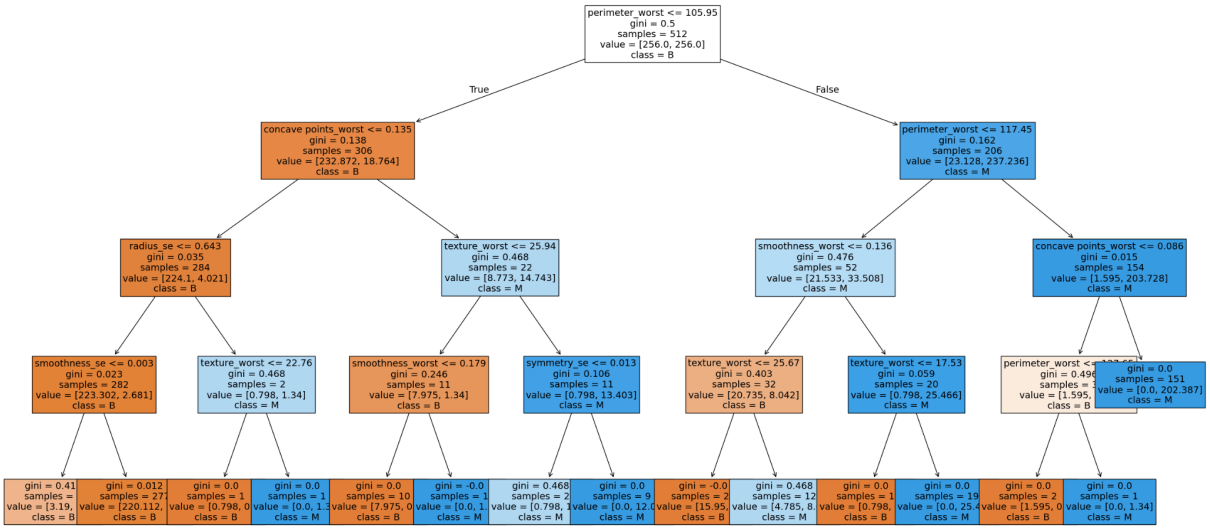
Decision Tree from Fold 5



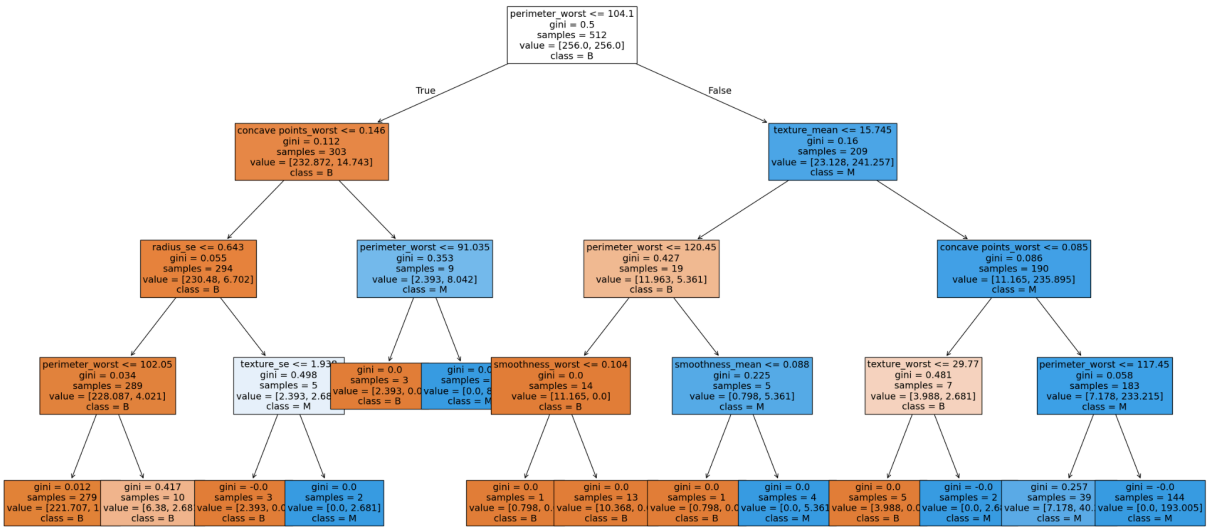
Decision Tree from Fold 6



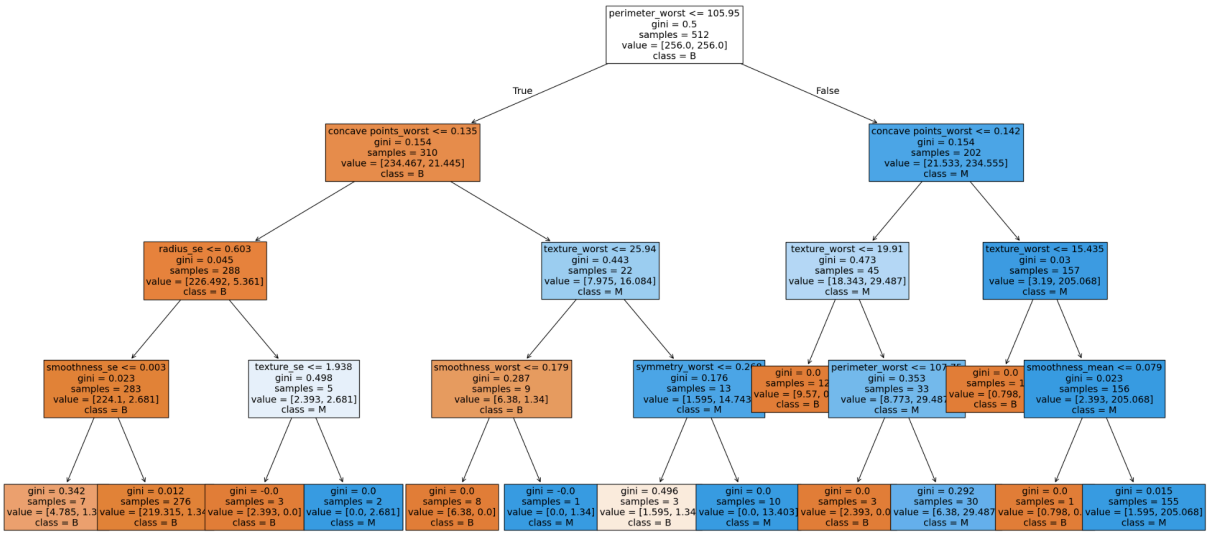
Decision Tree from Fold 7



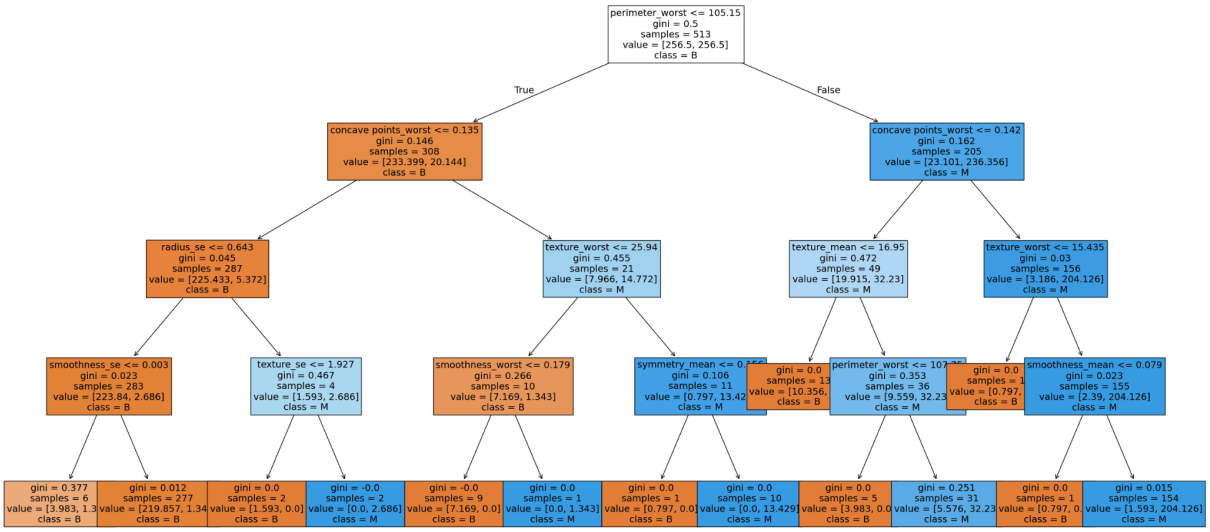
Decision Tree from Fold 8



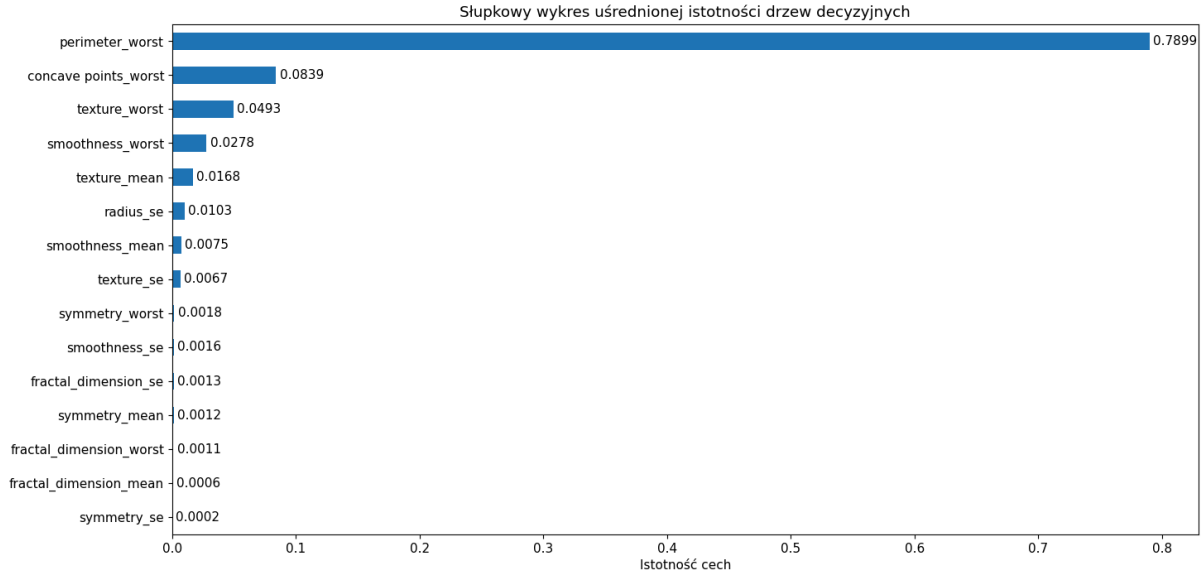
Decision Tree from Fold 9



Decision Tree from Fold 10



Widać, że w korzeniu każdego drzewa znajduje się parametr perimeter_worst. Na pierwszym poziomie drzewa 13 razy wystąpił atrybut concave points_worst, raz wystąpił atrybut texture_mean i 6 razy wystąpił atrybut primeter_worst. Oznacza to, że te atrybuty te mogą być szczególnie istotne dla klasyfikacji.
Dodatkowo biorąc pod uwagę istotność mierzoną poprzez sumaryczny zysk informacji danej cechy z drzewa decyzyjnego (atrybut feature_impostances_ obiektu wytrenowanego modelu) można ocenić jak ważna jest dana cecha poprzez miarę liczbową. Dana miara istotności została zebrana wobec wszystkich cech dla każdego drzewa decyzyjnego, a następnie uśredniona względem drzew. Poniżej prezentuję słupkowy wykres uśrednionych istotności dla każdego atrybutu.



Jak można było się spodziewać pierwsze dwie najważniejsze cechy to perimeter_worst oraz concave points_worst. Natomiast trzecia najważniejsza cecha to texture_worst - texture_mean jest na piątym miejscu. Wygląda na to, że sumarycznie texture_worst przynosi większy zysk informacyjny niż występujący raz w pierwszym poziomie drzew atrybut texture_mean. Widać również, że atrybut perimeter_worst jest bardzo dominujący - jego istotność jest prawie 10 razy większa od znajdującego się na drugim miejscu atrybutu concave points_worst.

Eksperymenty z modelem i zbiorem danych

Wybranie innego podzbioru atrybutów

Na początku chciałbym wykonać eksperymenty polegające na zredukowaniu ilości cech, które przyjmuje model. Z analizy istotności cech wynika, że większość cech ma znikomą istotność dla modelu. Sprawdziłby zatem jakie są wyniki modelu dla 5 i 3 najważniejszych cech oraz 1 najważniejszej cechy. Dodatkowo warto sprawdzić cechy, które miały najwyższą korelację z atrybutem diagnosis i należały do oddzielnych grup korelacji względem siebie - radius_se, concave points_worst, perimeter_worst. Poniższa tabela prezentuje wyniki

Użyte cechy	Czułość +- odchylenie standardowe	Swoistość +- odchylenie standardowe
wszystkie cechy	0.93 +- 0,05	0,94 +- 0,05
perimeter_worst	0,90 +- 0,07	0,87 +- 0,09
texture_worst, concave points_worst, perimeter_worst	0,95 +- 0,04	0,94 +- 0,04
radius_se, concave points_worst, perimeter_worst	0,89 +- 0,08	0,92 +- 0,04
texture_worst, concave points_worst, perimeter_worst, texture_mean, smoothness_worst	0,93 +- 0,04	0,94 +- 0,05

Najwyższą czułość osiągnięto dla zbioru 3 cech z największą uśrednioną istotnością dla modelu - średni wynik poprawił się o ok. 2 punkty procentowe względem wyniku bazowego (0,93 +- 0,05). Średnia swoistość pozostała bez zmian względem wyniku bazowego. Na uwagę zasługuje również atrybut perimeter_worst - wykorzystując jedynie tą cechę model uzyskał czułość na poziomie średnio około 0,90 oraz swoistość na poziomie średnio około 0,87. Jest to imponujące biorąc pod uwagę tylko jedną cechę i potencjalnie dany atrybutu miałby zastosowanie kliniczne ze względu na prosty algorytm klasyfikacji i relatywnie wysoką skuteczność działania.

Dobór innych parametrów pracy algorytmu

W modelu bazowym sprawdziłem tylko jeden hiperparametr określający maksymalną głębokość drzewa - max_depth. Warto sprawdzić, czy poprzez dokładniejsze przeszukiwanie hiperparametrów nie uda się osiągnąć lepszych wyników. W tym celu wykorzystam metodę przeszukiwania losowego. Metoda ta dla zadanych przedziałów hiperparametrów losuje zadaną liczbę ich kombinacji a następnie trenuje na nich model, w tym przypadku drzewa decyzyjnego. Dokładniej wykorzystam funkcję RandomizedSearchCV z biblioteki scikit-learn, która dodatkowo trenuje model za pomocą techniki kroswalidacji, co umożliwi przeprowadzenie oceny modeli w taki sam sposób jak oceniany był model bazowy. Jako metrykę oceny modeli wybrałem zbalansowaną dokładność, która jest równa średniej arytmetycznej czułości i swoistości. Wybranie samej czułości mogłoby potencjalnie doprowadzić, że model zwracałby tylko klasę pozytywną - nowotwór złośliwy, dlatego niezbędne było również dodanie swoistości do oceny modeli.
Wykorzystałem następujące hiperparametry i ich przedziały do przeszukiwania:

- Miara jakości podziału w węzłach drzewa decyzyjnego (*criterion*) - możliwe wartości to *“gini”* oraz *“entropy”*. W przypadku entropii liczymy średnią ważoną zawartości informacyjnej danej wzorem:

$$I(P) = \sum_{d \in C} -\frac{|P^d|}{|P|} log(\frac{|P^d|}{|P|})$$, gdzie:

P^d - podzbiór tych przykładów ze zbioru przykładów P, które należą do klasy d
C – zbiór klas wyznaczony przez atrybut decyzyjny.
Zatem entropia jest dana wzorem:

$$E_t(P) = \sum_{r \in R_t} \frac{|P_{tr}|}{|P|} I(P_{tr})$$, gdzie:

R_t - zbiór możliwych wyników testu t

P_{tr} - podzbiór tych przykładów ze zbioru P, które dają dla testu t wynik r

Natomiast w przypadku atrybutu *"gini"* stosujemy indeks Gini dany wzorem:

$$G(P) = 1 - \sum_{d \in C} \left(\frac{|P^d|}{|P|}\right)^2,$$

który następnie jest uśredniany dla wszystkich możliwych wyników testu t:

$$G_t(P) = \sum_{r \in R_t} \frac{|P_{tr}|}{|P|} G(P_{tr})$$

- Maksymalna głębokość drzewa, czyli liczba poziomów od korzenia do najdalszego liścia (*max_depth*). W tym przypadku przeszukiwanie będzie odbywać się na wartościach od 1 do 10 oraz z uwzględnieniem wartości None oznaczającej brak limitu głębokości drzewa. Wartość 10 została dobra ze względu na to, aby nie tworzyć zbyt skomplikowanych modeli, które potencjalnie mogłyby być przetrenowane
- Minimalna liczba próbek potrzebna, aby podzielić węzeł (*min_sample_split*) - oznacza, że jeżeli węzeł ma mniej próbek od zadanej wartości to dalej się nie dzieli i staje się liściem. Duże wartości zwiększają regularyzację modelu, ponieważ nie dzieli on wtedy danych zbyt drobno. Przyjąłem wartości od 2 do 60, aby zweryfikować również możliwość większej regularyzacji, biorąc również pod uwagę ilość wszystkich danych.
- Minimalna liczba próbek jaka musi znajdować się w każdym liściu drzewa (min_sample_leaf) - oznacza to, że podziały, które utworzyłyby liście o mniejszej liczbie próbek są niedozwolone. W tym przypadku większe wartości również mogą prowadzić do bardziej uogólnionego drzewa. Wartości do przeszukania wybrałem od 1 do 30 również ze względu na sprawdzenie większej regularyzacji.

Przeszukiwanie losowe wykonałem przez 100 iteracji i model z najlepszą zbalansowaną dokładnością miał następujące parametry: random_state=0, *class_weight="balanced"*, *min_samples_split=14*, *min_samples_leaf=7*, *max_depth=7*, *criterion='gini'*. Model uzyskał czułość na poziomie 0,93 +- 0,05 oraz swoistość równą 0,96 +- 0.03. W prawdzie osiągnięto większą swoistość od modelu bazowego (0,94 +- 0,05), jednakże czułość pozostała bez zmian.

Sprawdzenie algorytmów alternatywnych
Dodatkowo zostały sprawdzone dwa algorytmy alternatywne - las losowy oraz zbalansowany las losowy. Do ich implementacji posłużyłem się funkcjami

RandomForestClassifier z biblioteki *scikit-learn* oraz funkcji *BalancedRandomForestClassifier* z biblioteki imbalance-learn. Oba te algorytmy zostały przetestowane metodą 10-krotnej walidacji krzyżowej. Są to algorytmy będące złożeniem modeli drzew decyzyjnych. Zastosowałem domyślne hiperparametry modeli z wyjątkiem tych użytych w modelu bazowym drzewa decyzyjnego - *random_state=0*, *class_weight="balanced"*, *max_depth=4*. Dane algorytmy otrzymały następujące wyniki:

Model	Czułość - średnia +- odchylenie standardowe	Swoistość - średnia +- odchylenie standardowe
Drzewo decyzyjne	0,93 +- 0,05	0,94 +- 0,05
Las losowy	0,94 +- 0,04	0,98 +- 0,03
Zbalansowany las losowy	0,96 +- 0,04	0,95 +- 0,04

Okazuje się, że złożenie modeli daje lepsze wyniki od bazowego modelu drzewa decyzyjnego. W przypadku każdego modelu dla obu metryk nastąpiła poprawa. Las losowy poprawił średnią swoistość o ok. 4 punkty procentowe a średnią czułość o ok. 1 punkt procentowy. Natomiast zbalansowany las losowy poprawił średnią czułość o ok. 3 punkty procentowe a średnią swoistość o ok. punkt procentowy. Ze względu na większą wagę czułość względem swoistości w tej analizie wybrany zostały model zbalansowanego lasu losowego.

Podsumowanie

Przegląd wykonanego procesu
Na początku został zaimplementowany model drzewa decyzyjnego, dla którego ręcznie sprawdziłem niewielką liczbę parametrów. Model ten osiągnął średnią czułość równą 0.93 +- 0,05 oraz średnią swoistość równą 0,94 +- 0,05. Następnie zostały wybrane zbiory kilku cech. Zastosowanie trzech najważniejszych cech względem istotności drzewa decyzyjnego podniosło średnią czułość do 0,95 +- 0,04. Kolejnym krokiem było przeprowadzenie przeszukiwania losowego, które miało na celu dokładniejsze sprawdzenie optymalnych wartości hiperparametrów. Jednakże metoda spowodowała wzrost średniej swoistości do 0,96 +- 0.03, lecz nastąpił także spadek czułości do 0,93 +- 0,05, co ze względu na większy nacisk na czułość nie jest korzystne dla celu eksploracji. Ostatecznie sprawdzono złożenie drzew decyzyjnych w postaci lasu losowego oraz zbalansowanego lasu losowego. Oba algorytmy osiągnęły większą czułość i swoistość od bazowego modelu drzewa decyzyjnego, a w szczególności zbalansowany las losowy miał średnią czułość na poziomie 0,96 +- 0,04 oraz swoistość na poziomie 0,95 +- 0,04.
Najlepszym modelem ze względu na cel eksploracji wydaje się być model lasu losowego. Jednakże również bardzo korzystnie wyszedł model drzewa decyzyjnego dla trzech najważniejszych cech, który osiągnął czułość zaledwie o 1 punkt procentowy mniej, cechując się znacznie prostszą architekturą. Zapewne w zastosowaniach praktycznych prostszy model okazałby się bardziej przydatny.
W przypadku przeszukiwania losowego, które nie zakończyło się sukcesem, powodem może być zbyt mało wyczerpujące przeszukanie. Możliwe, że zastosowanie większej liczby iteracji

przyniosłoby lepsze rezultaty. Inną opcją jest zastosowanie sprytniejszego algorytmu przeszukującego przestrzeń hiperparametrów np. TPE (z ang. Tree-structured Parzen Estimator), który w uproszczeniu na podstawie analizy prawdopodobieństwa uczy się z poprzednich wyników i celuje w bardziej obiecujące regiony, przez co w mniejszej liczbie iteracji można znaleźć lepsze wyniki od przeszukiwania losowego.

Stopień pokrycia celów
Celami eksploracji było osiągnięcie predykcji atrybutu diagnosis z czułością wynoszącą co najmniej 80% i swoistością co najmniej 60%. Dodatkowym celem było określenie jakie atrybuty mają największy wpływ na predykcję.
Cel został jak najbardziej osiągnięty. Najlepszy model lasu losowego osiągnął czułość równą ok. 96% a swoistość ok. 95%. Innym modelem godnym wspomnienia jest model drzewa decyzyjnego, który bazuje na trzech najlepszych cechach, który osiągnął czułość równą ok. 95% a swoistość ok. 94%. Na podstawie analizy istotności cech możemy dojść do wnioski, że najważniejszą cechą decydującą o predykcji złośliwości nowotworu jest perimeter_worst. Jego istotność okazuje się być ok. 10 razy większa od pozostałych cech, a wykorzystując tylko tą cechę można stworzyć model drzewa decyzyjnego o czułości ok. 90% oraz swoistości ok. 87%.