AKADEMIA GÓRNICZO – HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ZARZĄDZANIA

Katedra: Samodzielna Pracownia Zastosowań Matematyki w Ekonomii

Projekt Dyplomowy

Zastosowanie drzew klasyfikacyjnych i lasów losowych w prognozowaniu chorób serca

Application of classification trees and random forests in predicting heart disease

Autor: *Agata Marta Dratwa*

Kierunek studiów: Informatyka i Ekonometria

Opiekun pracy: *dr hab. Anna Czapkiewicz*

Kraków, 2022

Spis treści

Cel i tematyka pracy

Sztuczna inteligencja to pojęcie bardzo popularne w ostatnich latach ze względu na jej udział w najnowszych osiągnięciach techniki i nauki. Najprościej ujmując sztuczna inteligencja to próba sprawiania, aby komputery myślały i zachowywały się tak jak ludzie. Obecnie ilość danych, która jest generowana, znacznie przewyższa ludzkie zdolności do przyswajania, interpretowania i podejmowania decyzji na podstawie tych informacji. Z kolei sztuczna inteligencja jest w stanie w bardzo efektywny sposób wyszukiwać wzorce w danych. Automatyzacja tych umiejętności pozwala na tworzenie nowych możliwości w wielu sektorach biznesu. Możliwości sztucznej inteligencji zapewniają rozwiązania dla wielu potrzeb sektora opieki zdrowotnej, gdzie wytwarzana jest ogromna ilość danych, jednakże brakuje personelu, który mógłby je przeanalizować i wyciągnąć odpowiednie wnioski. Metody sztucznej inteligencji polegają na przetwarzaniu i analizie danych, dlatego mogą być one zastosowane do wspomagania różnych dziedzin medycyny.

Choroby serca są pierwszą przyczyną zgonów na świecie, zarówno u kobiet, jak i mężczyzn. Szacuje się, że każdego roku 17,9 milionów ludzi umiera z powodu chorób serca, co stanowi 31% wszystkich zgonów na świecie. Jedna trzecia z tych zgonów występuje przedwcześnie, przed 70 rokiem życia. Wysoka śmiertelność i wysokie koszty leczenia sprawiają, że choroby serca są dużym zagrożeniem dla wielu osób w wielu częściach świata — zwłaszcza tych najuboższych. Dlatego tak istotne jest diagnozowanie osób z objawami chorób serca i wczesna interwencja, aby nie dopuścić do pogorszenia się stanu ich zdrowia. Przeanalizowanie zależności między różnymi cechami człowieka i możliwością zachorowania na choroby serca, a następnie stworzenie modelu jest niezwykle istotne w przewidywaniu, które osoby są najbardziej podatne na choroby serca, ponieważ pozwala to na podjęcie odpowiednich działań prewencyjnych lub leczniczych.

Celem niniejszej pracy jest stworzenie modelu uczenia maszynowego, który może przewidzieć obecność lub brak choroby serca u osoby. Problem jest natury medycznej, dlatego celem autora jest stworzenie modelu, który wszystkie trzy mierniki jakości modelu: dokładność, czułość i precyzje będzie miał na wysokim poziomie (minimum 70%). W wielu podobnych badaniach, które przewidują obecność choroby serca, brany pod uwagę jest tylko współczynnik dokładności modelu, który nie jest jednak wystarczający, aby ocenić czy model przewidujący obecność choroby jest dobrej jakości. W niniejszej pracy przeanalizowane zostanie także, jakie czynniki mają największy

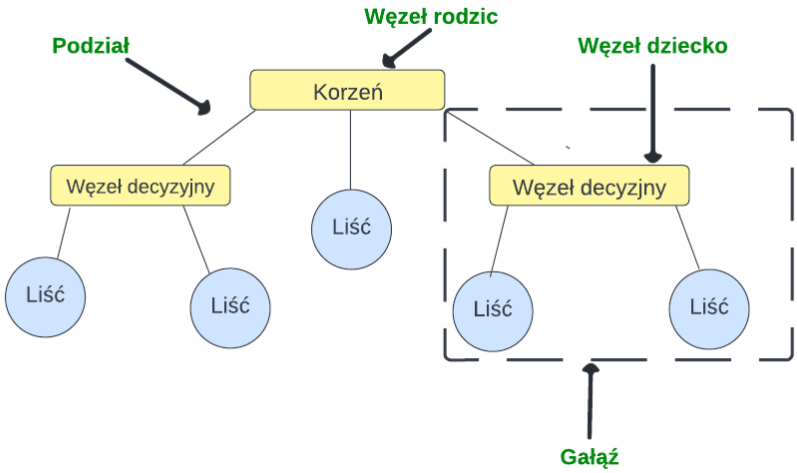
wpływ na choroby serca. Jako algorytmu uczenia maszynowego zostaną wykorzystane drzewa decyzyjne i lasy losowe. Zdecydowano się na wybór tych modeli, gdyż osiągają one wysokie miary jakości, nie wymagają specjalnego przygotowania danych i radzą sobie zarówno z kategorycznymi, jak i numerycznymi danymi co w przypadku tego problemu jest bardzo istotne. Dodatkowo drzewa decyzyjne są bardzo łatwe w interpretacji, można je zwizualizować, co jest wielką zaletą od strony biznesowej. Zbadane zostanie również jak modyfikowanie odpowiednich parametrów, kontrolujących proces uczenia, wpływa na jakość obu modeli i co można zrobić, aby uzyskać jak najwyższe miary jakości modelu. Na końcu oba modele zostaną do siebie porównane.

W pierwszym oraz drugim rozdziale tej pracy zostały poruszone teoretyczne aspekty drzew klasyfikacyjnych i lasów losowych, których znajomość jest niezbędna, aby zrozumieć dalszą część pracy. Następnie w rozdziale trzecim dane dotyczące chorób serca respondentów zostały szczegółowo przeanalizowane i odpowiednio zmodyfikowane. Za pomocą odpowiednich testów statystycznych i wykresów, zostały wybrane zmienne, które najbardziej wpływają na choroby serca. W rozdziale czwartym na podstawie wybranych zmiennych stworzone zostały modele drzewa klasyfikacyjnego i lasu losowego, a następnie odpowiednio zmodyfikowano parametry modelu, aby uzyskać modele o najlepszych miernikach jakości. W ostatnim kroku analizy najlepsze modele zostały szczegółowo podsumowane oraz porównane. Pracę kończy refleksja dotycząca zastosowania wyników niniejszego badania, jego ograniczenia oraz rekomendacje co do dalszych badań i analiz.

1 Drzewa decyzyjne

Algorytm drzew decyzyjnych uznawany jest za najbardziej intuicyjny i łatwy do zwizualizowania model uczenia maszynowego będąc przy tym bardzo skutecznym. Jest to algorytm uczenia nadzorowanego, który dzieli się na dwa główne typy: drzewa klasyfikacyjne i drzewa regresyjne. Celem stosowania drzewa decyzyjnego jest utworzenie modelu, który posłuży do przewidywania klasy (drzewa klasyfikacyjne) lub wartości (drzewa regresyjne) zmiennej docelowej poprzez uczenie się prostych reguł decyzyjnych na podstawie wcześniejszych danych (zwanych zbiorem uczącym). Algorytm ten używany jest w przeróżnych dziedzinach: od medycznej diagnozy do określania ryzyka kredytowego. Ta praca skupia się na opisie drzew decyzyjnych klasyfikacyjnych. Służą one do klasyfikowania obiektów danych do z góry określonego zbioru klas (np. ryzykowny/ nieryzykowny) na podstawie wartości ich atrybutów. Atrybuty to elementy danych wykorzystywane w uczeniu maszynowym. Są one również określane jako cechy, zmienne, pola lub predyktory. Przykład atrybutu to: płeć, wiek.

Drzewo decyzyjne to model, którego graficzna reprezentacja przypomina drzewo odwrócone do góry nogami. Poniższy model przedstawia podstawową strukturę drzewa decyzyjnego wraz z nazwami jego poszczególnych elementów.



Rysunek 1.1 Przykładowy model drzewa decyzyjnego określający czy osoba będzie grała w golfa danego dnia, czy nie
Źródło: Opracowanie własne w Lucidchart

Słownik ważnych pojęć dotyczących drzew decyzyjnych:

- **Korzeń drzewa/ węzeł główny** - węzeł rozpoczynający graf. Reprezentuje cały zestaw danych, który następnie na podstawie jednego z

atrybutów zostaje podzielony na dwa lub więcej jednorodnych podzbiorów

- **Podział** - proces podziału węzła na dwa lub więcej węzłów podrzędnych
- **Węzeł decyzyjny/ węzeł wewnętrzny -** jest to węzeł, który ulega dalszemu podziałowi na kolejne węzły podrzędne na podstawie kolejnego atrybutu
- **Liść/ węzeł zewnętrzny** - węzeł, który nie ulega dalszemu podziałowi, dokonuje się w nich predykcja klasy
- G**ałąź/ poddrzewo** - część drzewa decyzyjnego składająca się z wielu węzłów
- **Węzeł rodzic (nadrzędny) i dziecko (podrzędny**) - węzeł, który jest podzielony na węzły podrzędne jest nazywany węzłem rodzicem węzłów podrzędnych natomiast węzły podrzędne są dziećmi węzła rodzica

Drzewa decyzyjne klasyfikują obserwacje, sortując je w dół drzewa od korzenia do liścia, który określa klasę obserwacji. Każdy węzeł w drzewie reprezentuje test jednego atrybutu obserwacji, a każda gałąź wychodząca od tego węzła odpowiada jednej z możliwych wartości dla tego atrybutu. Obserwacja jest klasyfikowana przez rozpoczęcie od korzenia drzewa, przetestowanie atrybutu, który jest określony przez ten węzeł, a następnie przejście w dół gałęzi drzewa odpowiadającej wartości tego atrybutu w danym przykładzie. Proces ten ma charakter rekurencyjny i jest powtarzany dla każdego poddrzewa zakorzenionego w nowym węźle do momentu, kiedy kolejny węzeł okaże się liściem, ponieważ wtedy dochodzi do klasyfikacji obserwacji [17].

1.1 Miary jakości klasyfikatorów

Tworząc drzewo decyzyjne, bardzo ważnym krokiem jest zdecydowanie, które atrybuty powinny znaleźć się w korzeniu oraz na różnych poziomach drzewa jako węzły wewnętrzne. Jeśli zrobi się to w sposób losowy, istnieje duże ryzyko uzyskania modelu z niskim współczynnikiem dokładności (stosunek dobrze sklasyfikowanych obserwacji do ilości wszystkich sklasyfikowanych obserwacji). Istnieje jednak kilka sposobów, aby w jak najlepszy sposób przyporządkować atrybuty do poszczególnych węzłów w drzewie klasyfikacyjnym. Metody te opierają się na obliczeniu specjalnych miar selekcji dla każdego z podziałów i wybraniu tego, który osiągnął najlepszy wynik. Najpopularniejsze miary selekcji to: współczynnik Giniego, przyrost informacji, współczynnik przyrostu informacji, Chi-kwadrat [8].

1.1.1 Współczynnik Giniego

$$\text{Gini} = 1 - \sum_{j=1}^{n} p_j^2$$

$$p - procentowy\ udział\ klasy\ j\ w\ danym\ węźle$$

Współczynnik Giniego mierzy prawdopodobieństwo zdarzenia, że dana cecha zostanie błędnie sklasyfikowana, gdy zostanie wybrana losowo. Mierzy tak zwaną „nieczystość" atrybutu i przyjmuje zakres od 0 do 1, gdzie 0 oznacza, że węzeł jest „czysty", zatem wszystkie zawarte w nim elementy należą do jednej klasy. Natomiast 1 odpowiada losowemu rozmieszczeniu elementów w różnych klasach. Cecha o najmniejszej wartości współczynnika zostanie wybrana jako korzeń drzewa. Jako że jeden podział skutkuje co najmniej dwoma węzłami podrzędnymi, współczynnik Giniego całego podziału to średnia ważona wskaźnika Gini węzłów podrzędnych. Ten podział, który daje najmniejszą nieczystość, jest najlepszy [8].

1.1.2 Przyrost informacji

$$Przyrost\ informacji = Entropia\ przed\ podziałem - Entropia\ po\ podziale$$

$$Entropia = \sum_{j=1}^{n} -p_j * log_2(p_j)$$

$$p - procentowy\ udział\ klasy\ j\ w\ danym\ węźle$$

Chcąc uzyskać przyrost informacji, na początku należy policzyć entropię. Jest to miara niepewności danych, która w kontekście uczenia maszynowego mierzy zróżnicowanie klas w węźle (nieczystość węzłów). Analogicznie jak w przypadku współczynnika Giniego im mniejsza wartość entropii, tym lepiej - osiąga minimum, kiedy węzeł zawiera obserwacje należące do tylko jednej klasy, a maksimum, kiedy każda klasa ma to samo prawdopodobieństwo znalezienia się w węźle. Podobnie jak dla poprzedniego wskaźnika, entropię całego podziału liczy się jako średnią ważoną entropii węzłów podrzędnych.

Przyrost informacji mierzy efektywność cechy w klasyfikowaniu danych uczących. Jest to oczekiwane zmniejszenie entropii spowodowane przez podział przykładów według tej cechy. Innymi słowy, jest to różnica między nieczystością węzła nadrzędnego a średnią nieczystością węzłów podrzędnych. Im mniejsza entropia węzłów podrzędnych (węzłów po podziale), tym większy przyrost informacji, a więc i większa

efektywność cechy w klasyfikowaniu danych. Atrybut, który będzie miał najwyższy przyrost informacji, zostaje wybierany do danego podziału drzewa [8].

1.1.3 Współczynnik przyrostu informacji

Wadą przyrostu informacji jest to, że faworyzuje on atrybuty o dużej liczbie wartości i wybiera je jako węzły główne, co prowadzi do przeuczenia modelu. Współczynnik przyrostu informacji to modyfikacja przyrostu informacji, która rozwiązuje ten problem, gdyż bierze pod uwagę liczbę gałęzi i ich wielkość. Jest to iloraz przyrostu informacji i informacji wewnętrznej. Informacja wewnętrzna to entropia proporcji podzbiorów danych — to znaczy informacja o tym, jak trudno jest odgadnąć, w której gałęzi znajduje się losowo wybrana próbka. Atrybut z najwyższą wartością współczynnika przyrostu informacji jest wybierany do podziału drzewa.

$$Współczynnik\ przyrostu\ informacji = \frac{Współczynnik\ przyrostu\ informacji}{Informacja\ wewnętrzna}$$

$$Informacja\ wewnętrzna = -\sum \frac{|D_j|}{|D|} * log_2 \frac{|D_j|}{|D|}$$

$$D_j - podzbiór\ danych\ znajdujących\ się\ w\ j - tej\ gałęzi$$

Wadą współczynnika przyrostu informacji jest to, że może wybrać on atrybut tylko dlatego, że jego informacja wewnętrzna jest bardzo niska. Z tego powodu najpierw należy brać pod uwagę atrybuty tylko z odpowiednio dużym przyrostem informacji i dopiero w kolejnym kroku porównywać je na podstawie współczynnika przyrostu informacji [16].

1.1.4 Miara chi-kwadrat

Jedną z najstarszych metod, którą można zastosować, aby znaleźć najlepszy podział drzewa klasyfikacyjnego, jest miara chi-kwadrat. Bazuje on na statystycznej istotności różnic między węzłami potomnymi, a węzłem nadrzędnym.

$$Chi - kwadrat = \sqrt{\frac{(obserwowana - oczekiwana)^2}{oczekiwana}}$$

W powyższym wzorze „oczekiwana” oznacza wartość oczekiwaną dla klasy w węźle dziecka na podstawie rozkładu klas w węźle nadrzędnym. Obserwowana to wartość rzeczywista dla klasy w węźle dziecka. Powyższy wzór pozwala nam obliczyć wartość Chi-kwadrat dla klasy. Aby policzyć Chi-kwadrat dla całego węzła należy zsumować wartości Chi-kwadrat dla wszystkich klas w tym węźle. Wraz ze wzrostem tej wartości zwiększa się różnica między węzłami podrzędnymi i nadrzędnymi. Oznacza to, że im

wyższa wartość tego współczynnika tym dany podział drzewa jest lepszy. W celu podzielenia drzewa decyzyjnego tym sposobem należy podjąć następujące kroki:

1. Oblicz pojedyncze wartości Chi-kwadrat każdego węzła podrzędnego dla wszystkich podziałów. Chi-kwadrat węzła podrzędnego to suma wartości Chi-kwadrat dla każdej klasy w węźle.
2. Dla każdego podziału oblicz wartość Chi-kwadrat, która jest sumą Chi-kwadrat dla wszystkich węzłów podrzędnych.
3. Wybierz podział o najwyższej wartości Chi-kwadrat.
4. Powtarzaj kroki 1-3 aż uzyskasz jednorodne węzły [15].

1.2 Wady drzew decyzyjnych

Największą wadą drzew decyzyjnych jest ich podatność na przeuczenie. Domyślnie model drzewa decyzyjnego klasyfikacyjnego stworzony w R, Pytonie lub innym języku programowania będzie się tworzył, dopóki każdy pojedynczy obiekt z zestawu danych uczących nie będzie przypasowany do odpowiedniej klasy. Powoduje to, że jakość modelu będzie bardzo wysoka na danych uczących, ale znacznie mniejsza dla danych testowych. Istnieje jednak kilka sposobów, aby uniknąć tego niekorzystnego zjawiska i zostaną one przedstawione w następnym podrozdziale pracy. Jakość modelu odzwierciedlają metryki do oceny jakości klasyfikacji. Im większe są powyższe metryki, tym większa jakość modelu:

- **Dokładność** — stosunek poprawnych predykcji do wszystkich predykcji
- **Czułość** — stosunek prawdziwych pozytywnych identyfikacji do wszystkich obserwacji, które powinny zostać przewidziane jako pozytywne
- **Precyzja** — stosunek prawdziwych pozytywnych identyfikacji do wszystkich obserwacji, które zostały przewidziane jako pozytywne

Należy zwrócić także uwagę, że drzewa decyzyjne, są bardzo wrażliwe na zmiany w zestawie danych uczących. Czasem usuwając nawet jedną obserwację (skrajną) z zestawu danych, można uzyskać znacznie odmienny model po ponownym przetrenowaniu. Co więcej, nawet przy użyciu tego samego zestawu danych uczących przy kolejnym trenowaniu danych nasze wyniki mogą od siebie odbiegać, dlatego że tworzenie modelu np. w języku Python (za pomocą biblioteki Sci-kit learn) jest procesem stochastycznym. Zjawisko to nazywane jest wariancją, która powinna być zredukowana metodami takimi jak: bagowanie i boosting [13].

Kolejną wadą drzew decyzyjnych jest fakt, że nie osiągają one dobrych wyników na niezbalansowanych zestawach danych. Gdy występuje nierównowaga klas, mają one skłonność do przewidywania klasy dominującej. Jest to niepożądane zjawisko podczas tworzenia modeli klasyfikacyjnych, gdyż algorytm będzie ignorować klasę występującą rzadziej, a co za tym idzie, będzie uzyskiwać słabe wyniki podczas predykcji tej klasy. W przypadku zestawu danych dotyczących występowania danej choroby wśród pacjentów lub oceny ryzyka kredytowego jest to szczególnie problematyczne, gdyż model nie będzie skutecznie diagnozować chorych pacjentów czy wskazywać osób z wysokim ryzykiem kredytowym, jeśli klasy te w modelu danych będą mniej liczne. Istnieje jednak kilka sposobów, aby zbalansować zestaw danych, które zostaną przedstawione w dalszych częściach pracy [13].

1.3 Metody zmniejszające przeuczenie modelu

Jak zostało wspomniane wcześniej, drzewa decyzyjne są bardzo podatne na przeuczenie. Oznacza to, że za bardzo dopasowują się do danych uczących, co generuje skomplikowane drzewa o wielu gałęziach i liściach, które osiągają wysoką dokładność na danych uczących, jednakże dużo niższą dla danych testowych. Przycinanie drzewa decyzyjnego pomaga uniknąć tego negatywnego zjawiska. Polega ono na usunięciu poddrzewa, które jest zbędne i nie zwiększa jakości modelu oraz zastąpieniu go węzłem liściowym. Nie tylko proces ten zmniejsza ryzyko przeuczenia modelu, ale również upraszcza jego budowę, dzięki czemu jest bardziej czytelne dla badacza i zwiększa się efektywność obliczeniowa. Przycinanie drzewa decyzyjnego dzieli się na dwie główne kategorie: przycinanie w trakcie wzrostu oraz przycinanie końcowe.

1.3.1 Przycinanie w trakcie wzrostu

Jak sugeruje nazwa, przycinanie w trakcie wzrostu polega na zatrzymaniu rozrostu drzewa przed zakończeniem klasyfikacji zbioru treningowego. Rozrost drzewa powstrzymuje się poprzez ustalenie pewnych ograniczeń i najczęściej jest to przeprowadzane za pomocą modyfikacji hiperparametrów. Hiperparametr jest to parametr, który jest używany do kontrolowania procesu uczącego. Jego wartość jest ustalana przed trenowaniem algorytmu. Hiperparametry można modyfikować i dzięki temu poprawiać jakość modelu lub skracać jego czas kompilacji. Gdy drzewo osiągnie ustaloną wartość dla któregoś z hiperparametrów, przestaje się rozrastać. Zwykle domyślne hiperparametry, które przyjmują algorytmy drzew decyzyjnych w poszczególnych bibliotekach języków programowania, nie są wystarczające i należy je

odpowiednio dostosować do danego problemu. Poniżej znajduje się lista najpopularniejszych hiperparametrów wraz z opisem ich potencjalnego wpływu na model [10].

- **Minimalny spadek nieczystości**

  Regulując ten parametr, określa się jaką minimalną wartość musi mieć współczynnik Giniego, aby można było ponownie podzielić dane. Im mniejsza wartość tego indeksu, tym mniejsza różnorodność klas w liściu. Domyślnie parametr ten ustawiony jest na 0. Oznacza to, że drzewo rozrasta się do momentu, aż każdy liść będzie czysty — będzie w nim tylko jedna klasa. Ustawiając minimalną wartość tego indeksu na większą niż 0, zatrzymuje się proces rozrostu drzewa, mimo że w pojedynczym liściu jest więcej niż jedna klasa. Dzięki temu uzyskany model jest bardziej uniwersalny i uzyska lepsze wyniki na zbiorze testowym [18].

- **Maksymalna głębokość drzewa**

  Domyślnie jest ona ustawiona jako nieskończoność, więc algorytm tworzy drzewo do momentu, aż wszystkie liście będą czyste, co prowadzi do przeuczenia modelu. Ustawiając wartość tego parametru na określoną głębokość, drzewo przestaje się rozrastać, gdy ją uzyska. Parametr ten jest mniej elastyczny niż np. minimalny spadek nieczystości, gdyż czasem zadowalający wynik można uzyskać dla drzewa o głębokości mniejszej niż tej podanej w hiperparametrze. W takiej sytuacji drzewo nadal będzie się rozrastać co mimo modyfikacji hiperparametru doprowadzi do przeuczenia modelu. Z tego powodu przed ostatecznym wyborem maksymalnej głębokości drzewa warto rozważyć kilka różnych głębokości drzewa i zbadać, dla której średnia dokładność modelu z walidacji krzyżowej jest największa. W zależności od rodzaju danych i badanego problemu optymalna maksymalna głębokość drzewa może przyjmować zróżnicowane wartości [10]. Rysunek 1.2. przedstawia, jak wygląda drzewo w zależności od jego głębokości.

Rysunek 1.2 Drzewa decyzyjne o różnej głębokości
Źródło: Opracowanie własne

- **Minimalna liczba obiektów w liściu**

  Wartość ta determinuje, że podział węzła może nastąpić tylko w przypadku, gdy w każdym z jego węzłów podrzędnych pozostanie co najmniej tyle obiektów, ile zostało określone w tym hiperparametrze.
- **Minimalna liczba obiektów w węźle wewnętrznym**

  Jest to minimalna liczba próbek wymagana do podziału węzła wewnętrznego. Jeśli liczba obserwacji w węźle wewnętrznym jest mniejsza niż ten hiperparametr to węzeł ten staje się węzłem liściowym. Domyślna wartość w bibliotece sci-kit learn w Pythonie to 2.
- **Maksymalna liczba cech**

  Parametr ten określa maksymalną liczbę cech braną pod uwagę podczas szukania najlepszego atrybutu do podziału. Domyślnie model używa wszystkich cech podczas każdego podziału. Jest to bardzo kosztowne obliczeniowo, dlatego warto zawęzić zakres cech, szczególnie w przypadku, gdy zestaw danych posiada wiele atrybutów, co bardzo spowalnia proces tworzenia algorytmu. Często używaną wartością jest pierwiastek z wszystkich atrybutów.

W celu uzyskania najlepszego modelu należy bardzo dokładnie przeanalizować w jaki sposób i które hiperparametry wymagają modyfikacji, aby uzyskać drzewo decyzyjne z wysokimi miarami jakości modelu na zestawie danych testowych. Powinno się również zachować ostrożność stosując różne hiperparametry razem, gdyż mogą one negatywnie oddziaływać na siebie nawzajem. Zaletą tego rodzaju przycinania jest to, że jest ono szybkie i wydajne, gdyż pozwala uniknąć generowania zbyt skomplikowanego drzewa. Wyzwaniem związanym z przycinaniem drzewa w trakcie wzrostu jest trudność w dokładnym określeniu kryteriów zatrzymania rozrostu drzewa przed jego budową [6].

1.3.2 Przycinanie po rozroście drzewa

Ten rodzaj przycinania przeprowadzany jest po zbudowaniu drzewa. Na początku budowany jest model, który pozwala na doskonałe sklasyfikowanie zbioru treningowego i z dużym prawdopodobieństwem otrzymanie przeuczonego modelu. Następnie nieistotne gałęzie drzewa są przycinanie. Poniżej opisane są dwa najpopularniejsze sposoby post-przycinania drzew.

W 1986 JR Quinlan zaproponował procedurę przycinania drzew decyzyjnych nazywaną przycinaniem redukującym błąd, które wykonywane jest za pomocą zbioru walidacyjnego. Polega ona na przechodzeniu przez węzły wewnętrzne od dołu do góry i sprawdzaniu każdego węzła wewnętrznego, aby określić, czy jeśli zastąpi się go węzłem liściowym i przyporządkuje mu najczęściej występującą klasę to, czy dokładność modelu się zmniejszy. Jeśli jakość modelu się nie pogorszy, węzeł jest obcinany. Przycinanie węzłów następuje iteracyjnie, kolejno wybierając te węzły, których przycięcie najbardziej zwiększy dokładność drzewa decyzyjnego na zbiorze walidacyjnym. Proces ten jest kontynuowany do momentu, kiedy dalsze przycinanie zmniejszy współczynnik dokładności modelu na zbiorze walidacyjnym [17].

Drugą z metod, bardziej skomplikowaną, jest przycinanie złożoności kosztów. Przebiega ono następująco.:

1. Na podstawie zestawu danych uczących budowana jest sekwencja drzew T_0, T_1, …, T_k, gdzie T_0 oznacza oryginalne drzewo przed przycięciem, natomiast T_k jest korzeniem drzewa. Drzewo T_{i+1} tworzone jest przez zastąpienie co najmniej jednego poddrzewa w drzewie jego poprzednika T_i właściwymi liśćmi. Przycinane są te poddrzewa, które uzyskują najmniejszy wzrost poziomu błędów pozornych na przycięty liść, liczony w następujący sposób.

$$\alpha = \frac{\varepsilon(przycięty(T,t),S) - \varepsilon(T,S)}{|liście(T)| - |liście(przycięty(T,t))|}$$

$\varepsilon(T,S) - błąd\ drzewa\ T\ na\ próbce\ S\ \left(\frac{liczba\ złych\ predykcji}{całkowita\ liczba\ predykcji}\right)$

$|liście(T)| - liczba\ liści\ w\ drzewie\ T$

$przycięty(T,t) - drzewo\ otrzymane\ przez\ zastąpienie\ węzła\ t$
$w\ drzewie\ T\ odpowiednim\ liściem$

2. Szacowany jest błąd generalizacji dla każdego przyciętego drzewa, a następnie wybierane jest najlepsze przycięte drzewo. Jeżeli zbiór danych jest wystarczająco duży, sugerowane jest, aby podzielić go na zbiór uczący, na

którym konstruowane są drzewa oraz na zbiór przycinania, gdzie są one oceniane. Jeżeli zbiór danych nie jest wystarczająco duży, należy skorzystać z walidacji krzyżowej, jednakże należy wziąć pod uwagę, że wiążę się to ze złożonością obliczeniową [13].

1.4 Zalety drzew decyzyjnych

Mimo że tworząc drzewa decyzyjne, należy pamiętać o wadach tego modelu i być czujnym na nadmierne dopasowanie modelu do danych uczących, algorytm ten posiada wiele zalet. Jest odpowiedni zarówno do zmiennych kategorycznych, jak i numerycznych. Daje mu to przewagę nad wieloma modelami takimi jak: najbliżsi sąsiedzi czy Suport Vector Machines, które radzą sobie tylko ze zmiennymi numerycznymi i porządkowymi kategorycznymi, gdyż ich algorytm oparty jest na obliczaniu dystansu między zmiennymi w modelu. Kolejną zaletą drzew decyzyjnych jest fakt, że przygotowanie danych do modelu nie jest z natury procesem złożonym i czasochłonnym. Drzewa decyzyjne nie wymagają standaryzacji danych, dobrze radzą sobie z brakami w danych, dane kategoryczne nie muszą zostać zamienione na zmienne numeryczne (tak zwane „dumny variables") oraz są odporne na błędy w danych. Dodatkowo drzewa decyzyjne mają bardzo mało założeń o zestawie danych uczących w przeciwieństwie do modeli liniowych, gdzie dane muszą spełnić wiele warunków, aby mogły zostać wykorzystane do stworzenia modelu. Co więcej, bardzo dobrze radzą sobie z dużymi zestawami danych i wielkość zestawu danych nie wpływa znacznie na szybkość przewidywania lub utratę współczynnika dokładności. Wynika to z faktu, że dodając nowe zmienne do zestawu danych, czas obliczeń logarytmu zwiększa się w skali logarytmicznej, a nie liniowej lub potęgowej. Jak już zostało wspomniane, na początku drzewa decyzyjne są stosunkowo łatwe w interpretacji, możemy je zwizualizować i przeanalizować, jakie zmienne mają największy wpływ na zmienną wyjściową. Oznacza to, że drzewa decyzyjne są modelami zwanymi „białymi pudełkami". Ta zaleta ma duże znaczenie z perspektywy biznesowej, ponieważ reguły drzewa decyzyjnego można przełożyć na reguły biznesowe [7].

2 Lasy losowe

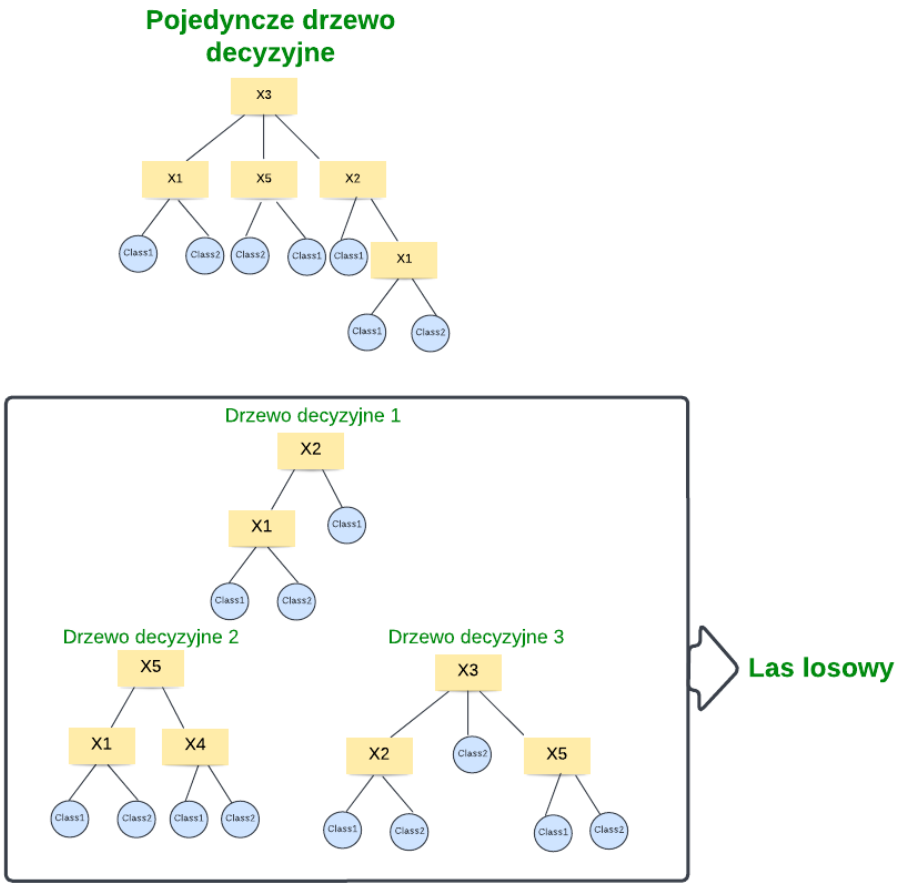
Mądrość tłumów jest to koncepcja zakładająca, że duże grupy ludzi są zbiorowo mądrzejsze niż indywidualni eksperci, jeśli chodzi o podejmowanie decyzji, rozwiązywanie problemów i przewidywanie. Spowodowane jest to tym, że punkt widzenia pojedynczej osoby często jest stronniczy, tymczasem średnia wiedza tłumu zwykle eliminuje stronniczość lub szum, uzyskując bardziej spójny i przejrzysty wynik. Tak samo jest w przypadku algorytmów uczenia maszynowego. Jeśli uśredni się wyniki grupy predyktorów, to zwykle uzyskany wynik będzie lepszy niż nawet najmocniejszy pojedynczy model. Metoda ta nazywana jest uczeniem zespołowym. Jest to rodzaj uczenia maszynowego, który łączy predykcje z wielu modeli w celu uzyskania lepszego wyniku.

Lasy losowe to rodzaj algorytmu uczenia zespołowego. Jest to zbiór pojedynczych drzew decyzyjnych, które zostały stworzone na podstawie losowego podzbioru zestawu danych. Predykcją takiego modelu w przypadku regresji jest średnia, a dla klasyfikacji jest to klasa, która uzyskała najwięcej głosów. Zamiast konkretnego wyniku w przypadku klasyfikacji model może obliczyć prawdopodobieństwo, że obiekt należy do danej klasy. W tej pracy zostanie omówiony, a następnie wykorzystany w badaniach algorytm lasów losowych klasyfikacyjnych.

Drzewa decyzyjne, mimo że są skutecznym algorytmem, to mają pewne ograniczenia. Jak zostało omówione wcześniej, są bardzo podatne na przeuczenie. Co więcej, pojedyncze drzewo decyzyjne nie sprawdza się dobrze na danych, które posiadają anomalia/ wartości odstające, gdyż nie jest w stanie wygładzić tych błędów. Lasy losowe dzięki temu, że używają wielu drzew decyzyjnych i uśredniają wyniki, bardzo dobrze radzą sobie z takimi wyzwaniami.

Najważniejszą zasadą, której należy przestrzegać tworząc las losowy, jest niska korelacja pojedynczych modeli drzew decyzyjnych. Działa to, jak w przypadku tworzenia portfela inwestycyjnego. Papiery wartościowe o niskiej korelacji razem tworzą portfel, który jest bardziej dochodowy, niż gdy korelacja między nimi jest wysoka. Na tej samej zasadzie nisko skorelowane modele mogą tworzyć prognozy zbiorcze o większej dokładności niż jakiekolwiek prognozy indywidualne. Gdyby każde drzewo było generowane na podstawie tego samego zestawu danych, uzyskiwano by cały czas bardzo podobne drzewa decyzyjne i problem przeuczenia modelu nie zostałby rozwiązany. Dlatego algorytm, tworząc każde drzewo, korzysta ze specjalnej techniki wyboru

podzbioru danych do każdego z nich nazywaną bagging. Dodatkowo pojedyncze drzewo w lesie losowym przy każdym podziale korzysta z innego zestawu atrybutów. Na rysunku 2.1 przedstawiony jest model pojedynczego drzewa decyzyjnego, który zbudowany jest na podstawie zestawu danych o 5 atrybutach (X1, X2, X3, X4, X5). Pod nim znajduje się model lasu losowego, składający się z trzech drzew. Oba modele zbudowane są na podstawie tego samego zestawu danych. Poszczególne drzewa decyzyjne w lesie losowym zawierają tylko niektóre atrybuty z całego zestawu danych, w przeciwieństwie do pojedynczego drzewa decyzyjnego. Każde z drzew w lesie losowym wygląda inaczej, gdyż zbudowane są na podstawie innych cech i mogą dawać różne predykcje dla tych samych obserwacji. Dzięki temu drzewa nie są ze sobą skorelowane, a cały model lasu losowego osiąga lepsze wyniki [19].



Rysunek 2.1 Porównanie pojedynczego drzewa decyzyjnego do lasu losowego składającego się z trzech drzew
Źródło: Opracowanie własne w LucidChart

Algorytm lasów losowych jest bardzo szeroko stosowany w różnych branżach: bankowości, handlu elektronicznym, medycynie. Jest to stosunkowo nowy algorytm — w literaturze z końca XX wieku nie ma żadnych wzmianek o tym modelu, mimo tego aktualnie jest jednym z najpopularniejszych algorytmów uczenia maszynowego. Lasy losowe są bardzo uniwersalne, dlatego w przypadku braku pewności jakiego algorytmu użyć, z dużym prawdopodobieństwem będą one dobrym wyborem [4].

2.1 Bagging

W lasach losowych każde drzewo decyzyjne korzysta z innej unikatowej próbki danych, na której jest trenowane. Każdy podzbiór danych jest losowo generowany z całego zestawu danych uczących z powtórzeniem. Metoda ta nazywa się bagging, znana jest także jako agregacja bootstrap. Oznacza to, że pojedynczy obiekt może występować kilka razy w danym podzbiorze. Zwykle podzbiór danych używany przez pojedyncze drzewo jest równy zestawie danych uczących. Jeśli przeprowadzi się to losowe próbkowanie wiele razy, okaże się, że średnio 63.2% oryginalnych danych jest w każdym z drzew, a pozostałe 36.8% to duplikaty z oryginalnego zestawu danych. Lasy losowe zwykle składają się z kilkuset drzew decyzyjnych, więc jest bardzo prawdopodobne, że każdy pojedynczy obiekt danych znajdzie się w co najmniej jednym drzewie. Dzięki tej metodzie zostaje rozwiązany problem anomalii w danych, gdyż część danych, która je powoduje, będzie wykorzystana tylko w niektórych drzewach decyzyjnych, a w większości drzew będą dane bardziej reprezentatywne. Chroni ona także algorytm przed przeuczeniem modelu, czyli redukuje wariancje. Jest to szczególnie przydatne, jeśli operujemy na danych wielowymiarowych, gdyż są one bardziej podatne na to zjawisko. Bagging jest również łatwy w implementacji. Należy jednak pamiętać o jednej z kluczowych wad tej metody — jest kosztowna obliczeniowo, ponieważ wraz ze wzrostem iteracji algorytm staje się coraz wolniejszy, co czyni go nieodpowiednim do zastosowania w czasie rzeczywistym. Z drugiej jednak strony działa on dobrze na algorytmach, które nie są stabilne — dlatego dobrze sprawdza się w przypadku drzew decyzyjnych, ale nie przy np. regresji [5].

2.2 Losowość cech

Kolejnym sposobem, który wykorzystuje algorytm lasów losowych, aby zwiększyć losowość modelu, jest wybór atrybutów przy każdym podziale drzewa. W przypadku klasycznego drzewa decyzyjnego algorytm przy każdym podziale węzła bierze pod

uwagę każdą możliwą cechę i wybiera tę, która powoduje największe rozdzielenie obserwacji w lewym węźle od obserwacji w prawym węźle. Dlatego, jeśli każde drzewo w lesie losowym miałoby do wyboru ten sam zestaw atrybutów, to byłyby one silnie skorelowane ze sobą, co jak zostało wspomniane wcześniej, jest bardzo niepożądanym zjawiskiem podczas tworzenia lasów losowych. Z tego powodu pojedyncze drzewo w lesie losowym przy każdym podziale korzysta z innego zestawu cech. Domyślnie algorytm przy pojedynczym rozgałęzieniu korzysta z podzbioru atrybutów w wielkości pierwiastka z wszystkich atrybutów w danym zestawie danych. Ta technika powoduje, że drzewa w modelu są jeszcze mniej ze sobą skorelowane i lepiej zdywersyfikowane [1].

2.3 Out of bag error

Błąd „out of bag" jest bardzo ważnym zjawiskiem w lasach losowych. Pozwala on oszacować jakość modelu bez konieczności wcześniejszego dzielenia zbioru danych na zestaw uczący i testowy. Do tej pory, aby ocenić jakość modelu, na początku dzielono zbiór na zestaw danych uczących (zwykle 90%) i testowych (zwykle 10%). Następnie trenowano model na danych uczących i szacowano jego dokładność na zbiorze testowym. Metoda ta jest bardzo przydatna, jednakże za każdym razem traci się część danych, gdyż model nie jest trenowany na pełnym zestawie. Jest to szczególnie problematyczne, gdy zestaw danych jest mały, ponieważ niewykorzystanie części danych do trenowania modelu, może mieć duży wpływ na wynik. W przypadku lasów losowych możliwe jest oszacowanie jakości modelu, który trenowany jest na całym zestawie danych. Jak zostało wspomniane wcześniej, w lasach losowych każde drzewo trenowane jest tylko na części oryginalnego zestawu danych uczących (średnio 63.2%), reszta to duplikaty. Cześć danych, która nie jest używana do generowania pojedynczego drzewa, nazywana jest „out of bag". Dla dużej ilości drzew w modelu wszystkie obiekty danych zostaną użyte w co najmniej jednym drzewie. Skoro każde drzewo posiada zestaw danych, na którym nie zostało trenowane, można go użyć, aby obliczyć jakość każdego drzewa. Średnia z błędu wszystkich drzew w lesie decyzyjnym to błąd „out of bag" [8].

2.4 Mierzenie ważności atrybutów

Jedną z zalet lasów losowych jest to, że w łatwy sposób pozwalają obliczyć względną ważność każdej cechy w zestawie danych. Często w prawdziwym życiu korzysta się z danych, które posiadają nawet kilkaset atrybutów. W takim przypadku bardzo przydatnym jest wiedzieć, które cechy mają największy wpływ na zmienną

objaśnianą. Lasy losowe posiadają kilka metod, które pozwalają odpowiedzieć na to pytanie.

Pierwsza z nich to domyślna metoda w Pythonie w bibliotece sklearn., dostępna także w R. Polega ona na zmierzeniu ważności atrybutu jako uśrednionego spadku nieczystości, który obliczony jest ze wszystkich drzew decyzyjnych w lesie losowym. Wartość ta jest obliczana automatycznie dla każdej cechy po utworzeniu modelu i jest skalowana tak, aby wszystkie ważności sumowały się do jedności [12].

Druga, często używana technika jest dostępna w R, ale nie w bibliotece sklearn Pythona. Wykorzystuje ona błąd „out of bag" i działa następująco:

1. Generuje błąd „out of bag" dla drzewa bazowego – jest to bazowy błąd „out of bag"
2. Dla każdego atrybutu, który jest użyty do budowy drzewa, określany jest przedział, który mogą przyjmować te wartości (maximum i minimum dla zmiennych numerycznych, lub lista wartości dla zmiennych kategorycznych).
3. Dla każdej cechy po kolei zmieniana jest losowo jej wartość dla wszystkich obiektów danych (ale tak, żeby znajdowała się w przedziale)
4. Po zmianie pojedynczej cechy obliczany jest błąd „out of bag", który najprawdopodobniej będzie większy niż błąd bazowy.
5. Permutowanym cechom przywracana jest ich oryginalna wartość, a następnie cały proces wykonywany jest od nowa, tym razem wykorzystując inny atrybut. Najważniejsze cechy to te, które mają największy przyrost błędu podczas permutacji [8].

Dzięki tym metodom w łatwy sposób można określić ważność poszczególnych cech i odrzucić te nieistotne. Jest to bardzo potrzebne, gdyż pozwala odpowiedzieć na jedną z ważniejszych zasad w uczeniu maszynowym: „im więcej atrybutów jest w modelu tym większe prawdopodobieństwo, że model będzie przeuczony". Niezależnie od tego, która metoda zostanie użyta, analizując wyniki, należy wziąć pod uwagę normalizację wartości ważności cech. Polega ona na tym, że w przypadku obu metod dostaje się informację, jak relatywnie ważna jest dana cecha tylko w stosunku do innych cech z tych samych danych. Nie można porównywać ważności dwóch atrybutów, które znajdują się w innych zestawach danych. Zazwyczaj im więcej cech w zestawie danych tym mniej ważna będzie pojedyncza cecha. Im bardziej zmienne są ze sobą skorelowane, tym bardziej wartość ważności będzie podzielona między nimi. [8].

2.5 Wybór liczby drzew w lasach losowych

Lasy losowe składają się z wielu drzew, więc nasuwa się pytanie: z ilu dokładnie powinny się składać? Wiele oprogramowań pozwala użytkownikowi modyfikować ich liczbę. Zwykle im więcej drzew tym lepiej, gdyż anomalia w danych są lepiej wygładzone i model jest bardziej odporny na przeuczenie. Zasada ta sprawdza się jednak tylko do pewnego momentu zgodnie ze zjawiskiem „malejących korzyści". Z każdym kolejnym drzewem jakość całego modelu wzrasta mniej niż w przypadku poprzedniego. W końcu korzyści się wyrównają, a kolejne drzewa będą przynosić znikome korzyści. Równocześnie czas potrzebny do wygenerowania modelu wzrasta w sposób liniowy — jeżeli zwiększymy liczbę drzew z 100 do 500, kompilacja algorytmu będzie trwała 5 razy dłużej [8].

Decyzja, ile drzew powinien zawierać model, zależy od zasobów obliczeniowych i od analizowanego problemu. Zwiększenie liczby drzew z 10 do 100, prawdopodobnie znacznie poprawi jakość modelu przy nieznacznym wzroście czasu obliczeniowego. Natomiast przejście z 1 000 do 10 000 drzew nieznacznie zwiększy jakość algorytmu kosztem jego istotnego spowolnienia. Najlepszym sposobem, aby wybrać odpowiednią liczbę, jest rozpoczęcie od małej wartości — np. 100 i stopniowe zwiększanie liczby drzew porównując wyniki za pomocą walidacji krzyżowej. Pozwoli to ocenić, kiedy korzyść ze zwiększenia jakości modelu, nie jest już warta dodatkowego czasu obliczeniowego. Zaleca się, aby na początku najpierw dostosować wszystkie inne hiperparametry modelu, używając do tego małej liczby drzew, które szybko się kompilują, a w ostatnim kroku dostosować liczbę drzew w modelu, gdyż jest to działanie najbardziej czasochłonne obliczeniowo [8].

2.6 Modyfikacja hiperparametrów w lasach losowych

Domyślne hiperparametry lasów losowych dają zazwyczaj dobre wyniki, co jest ich niewątpliwą zaletą. Lasy losowe posiadają niemal wszystkie hiperparametry drzew decyzyjnych (np. maksymalna głębokość drzewa, minimalna liczba obiektów w węźle wewnętrznym, minimalna liczba obiektów w liściu) i działają one analogicznie jak w przypadku pojedynczych drzew decyzyjnych. Niejednokrotnie modyfikacje w domyślanych wartościach tych parametrów w lasach losowych nie przełożą się na większą dokładność modelu, ponieważ lasy losowe nie są tak podatne na przeuczenie jak drzewa decyzyjne i model zespołowy jest odporny na anomalia w pojedynczych

drzewach. Mimo tego warto je jednak zmodyfikować, gdyż zredukowanie głębokości drzew czy minimalnej liczby obiektów w węźle znacząco skróci czas kompilacji modelu, co w przypadku lasów losowych charakteryzujących się większą złożonością obliczeniową jest bardzo istotne. [4]. Poniżej znajduje się lista pozostałych najważniejszych hiperparametrów lasów losowych:

- **Liczba drzew w modelu** — została szczegółowo omówiona w podrozdziale powyżej.
- **Maksymalna liczba atrybutów** — maksymalna liczba cech, których używa drzewo, dokonując każdego następnego podziału — najczęściej używana wartość to pierwiastek z wszystkich atrybutów w zestawie danych. Im mniej cech rozważanych jest przy każdym podziale, tym model szybciej się kompiluje, jednakże, jeśli rozważana jest ich zbyt mała ilość, to model może mieć niską dokładność.
- **Rozmiar próbki bootstrap** — poprzez ten parametr kontrolowany jest „bias-variance trade-off" lasu losowego. Im większa wartość tej próbki, tym mniejsza losowość, co oznacza, że model jest bardziej podatny na przeuczenie. Z drugiej strony wraz ze spadkiem tej wartości znika problem przeuczenia, jednak zwiększa się niedopasowanie modelu do zestawu danych uczących. Zwykle wielkość tej próbki równa jest wielkości zestawu danych uczących.
- **Number of jobs** — informuje komputer ilu procesorów może użyć. Wartość 1 oznacza, że może być użyty tylko jeden procesor, natomiast -1 informuje o tym, że nie ma żadnego ograniczenia [4].

3 Analiza danych

3.1 Dane

Według „Centers for Disease Control and Prevention” (CDC) choroby serca są jednym z najczęstszych powodów śmierci u Amerykanów. Wiele czynników ma wpływ na ich rozwój i według badań najpopularniejsze to: wysokie ciśnienie krwi, wysoki cholesterol, palenie, cukrzyca, otyłość, brak aktywności fizycznej, picie alkoholu. Wykrywanie czynników, które mają największy wpływ na choroby serca, jest kluczowe, aby skutecznie im zapobiegać. To właśnie nowoczesne metody uczenia maszynowego pozwalają tworzyć modele, które z wysoką dokładnością są w stanie przewidywać stan zdrowia pacjenta [11].

Oryginalny zestaw danych wykorzystany w poniższej pracy pochodzi z CDC i stanowi główną część „Behavioral Risk Factor Surveillance System” (BRFSS), które każdego roku przeprowadza badania telefoniczne, aby zebrać informacje na temat stanu zdrowia Amerykanów. BFRSS został założony w 1984 roku i aktualnie przeprowadza badania we wszystkich stanach Ameryki, co rocznie przekłada się na około 400 000 wywiadów z osobami pełnoletnimi. Najnowszy zbiór danych pochodzi z 2020 roku i składa się z 401958 wierszy oraz 279 kolumn. Większość pytań dotyczy stanu zdrowia respondenta. Dane, które będą analizowane to oryginalny zestaw danych, ale zredukowany do 319795 wierszy i 18 kolumn, które potencjalnie mogą udzielić najwięcej informacji związanych z chorobami serca i nadają się do tworzenia modeli uczenia maszynowego. Pochodzą one ze strony Kaggle, która jest internetową platformą dla analityków danych i osób tworzących modele uczenia maszynowego [11].

Wiele analiz i modeli predykcji zostało stworzonych na podstawie tego zestawu danych i dostępne są na stronie Kaggle [11]. Większość z nich osiąga wysoką miarę dokładności, jednakże bardzo niskie miary czułości i precyzji (20% - 50%). Jest to bardzo istotny problem, gdyż czułość na poziomie 30% oznacza, że wśród osób chorych na serce model zdiagnozuje poprawnie tylko 30% respondentów. W przypadku medycznej diagnozy wynik ten jest niezadowalający. Na taki rezultat dotychczas przeprowadzonych analiz wpływ mogło mieć kilka czynników. Prawdopodobnie w żadnej z prac dane nie zostały odpowiednio zbalansowane. Żadna z analiz nie porównuje także zastosowania drzew decyzyjnych i lasów losowych do tego problemu. Nie ma też badań, które sprawdzałyby, jak zmiana hiperparametrów w modelach wpływa na jego jakość. W związku z tym celem tej części pracy jest zbudowanie modeli drzewa decyzyjnego oraz

lasu losowego, które dla każdego z mierników jakości modelu osiągają wysokie rezultaty (minimum 70%). Zbadane również zostanie, jak za pomocą hiperparametrów można ulepszyć model i skrócić jego czas kompilacji. Na końcu dwa najlepsze modele drzewa decyzyjnego i lasu losowego zostaną do siebie porównane.

Przed budową odpowiednich modeli zostanie przeprowadzone czyszczenie danych oraz ich analiza, aby znaleźć zmienne, które będą mieć największy wpływ na choroby serca. Zostaną one także odpowiednio zmodyfikowane, aby lepiej nadawały się do tworzenia modeli.

3.2 Opis zmiennych

W tabeli 3.1 znajdują się wszystkie zmienne z zestawu danych wraz z ich opisem oraz określeniem typu zmiennej.

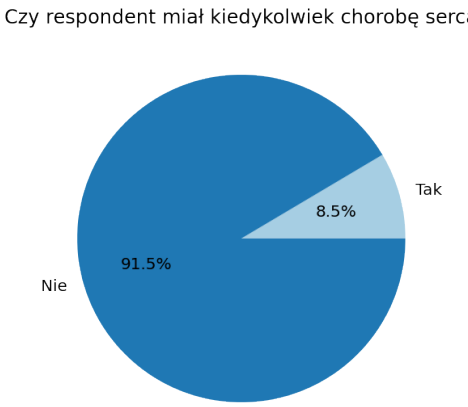
Tabela 3.1 Opis zmiennych
Źródło: Opracowanie własne

Nazwa zmiennej	**Opis zmiennej**	**Typ zmiennej**
HeartDisease	Respondenci, którzy kiedykolwiek zgłosili wystąpienie choroby wieńcowej serca (CHD) lub zawału mięśnia sercowego (MI).	Kategoryczna
BMI	Indeks masy ciała	Numeryczna
Smoking	Czy respondent wypalił co najmniej 100 papierosów w ciągu całego życia	Kategoryczna
AlcoholDrinking	Dorośli mężczyźni, którzy spożywają co najmniej 14 drinków tygodniowo lub kobiety spożywające ponad 7 drinków tygodniowo	Kategoryczna
Stroke	Czy respondent kiedykolwiek miał udar?	Kategoryczna
PhysicalHealth	Ile dni w ciągu ostatnich 30 dni zdrowie fizyczne (obejmuje choroby fizyczne i urazy) respondenta było złe	Numeryczna
MentalHealth	Ile dni w ciągu ostatnich 30 dni zdrowie psychiczne respondenta było złe?	Numeryczna
DiffWalking	Czy respondent ma poważne problemy w chodzeniu lub wspinaniu się po schodach?	Kategoryczna
Sex	Płeć respondenta	Kategoryczna

AgeCategory	14 stopniowa kategoria wiekowa respondenta	Kategoryczna
Race	Rasa/ pochodzenie etniczne respondenta	Kategoryczna
Diabetic	Czy respondent kiedykolwiek miał cukrzyce	Kategoryczna
PhysicalActivity	Czy respondent uprawiał jakąś aktywność fizyczną w ciągu ostatnich 30 dni?	Kategoryczna
GenHealth	W jakim stanie jest ogólne zdrowie respondenta (5 kategorii)	Kategoryczna
SleepTime	Ile średnio respondent śpi w ciągu 24 godzin	Numeryczna
Asthma	Czy kiedykolwiek respondent miał astmę?	Kategoryczna
KidneyDisease	Czy kiedykolwiek respondent miał chorobę nerek?	Kategoryczna
SkinCancer	Czy kiedykolwiek respondent miał raka skóry?	Kategoryczna

3.3 Zmienna objaśniana

Jak zostano wspomniane powyżej zmienna objaśniana to HeartDisease, czyli odpowiedź na pytanie: „Czy kiedykolwiek respondent zgłosili wystąpienie choroby wieńcowej serca (CHD) lub zawału mięśnia sercowego (MI)?”



Wykres 3.1 Rozkład chorób serca wśród respondentów
Źródło: opracowanie własne w Pythonie

Dane są nieproporcjonalne — tylko 8.5% wśród ankietowanych miało chorobę serca. Zestaw danych jest niezbalansowany, więc przed stworzeniem modelu należy zbalansować zestaw danych, aby dokładność modelu była większa i miał on większą szansę na nauczenie się cech, które określają ankietowanych z chorobą serca.

3.4 Zmienne objaśniające

W zestawie danych występują dwa typy zmiennych: numeryczne i kategoryczne. W celu wybrania zmiennych, które najlepiej opisują zmienną y, należy je szczegółowo przeanalizować oraz oszacować ich wpływ na chorobę serca. Zostanie to przeprowadzone za pomocą wykresów oraz odpowiednich testów statystycznych.

3.4.1 Zmienne numeryczne

W celu zbadania czy istnieje związek pomiędzy poszczególnymi zmiennymi numerycznymi a zmienną choroby serca, zostaną przeanalizowane wykresy pudełkowe oraz wyniki testu U Manna-Whitneya. Test ten porównuje rozkłady dwóch zmiennych. Hipoteza zerowa mówi o tym, że zmienne mają ten sam rozkład, natomiast hipoteza alternatywna, że rozkłady zmiennych istotnie różnią się między sobą. Jeżeli wartość p w teście jest < 0,05, należy odrzucić hipotezę zerową na rzecz hipotezy alternatywnej.

Wykres 3.2 Rozkład zmiennych numerycznych w zależności od chorób serca
Źródło: Opracowanie własne w Pythonie

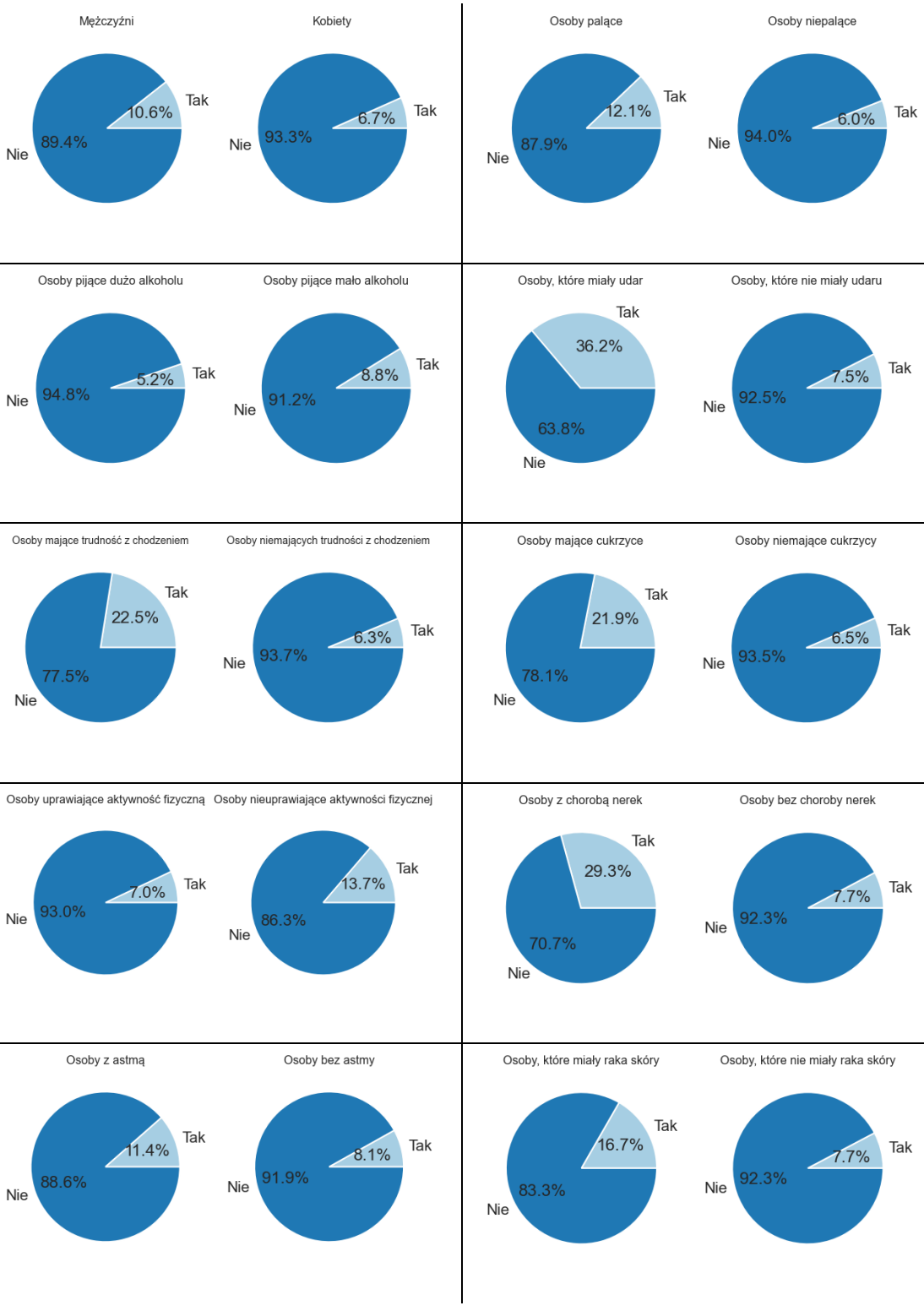
- **BMI –** zmienna posiadała dużo wartości odstających. Usunięto wszystkich respondentów, którzy mieli BMI poniżej 13 lub powyżej 85, co prawdopodobnie było błędem przy zapisie danych, gdyż jest to bardzo rzadko spotykane. Na podstawie wykresu 3.2 widać, że powyższe rozkłady nieznacznie różnią się od siebie, co może oznaczać, że BMI ma wpływ na choroby serca. Wartość p dla Testu U Manna-Whitneya to $2{,}44e^{-232}$, a więc odrzuca się hipotezę zerową. Oznacza to, że rozkład BMI osób o chorym sercu różni się od rozkładu BMI osób o zdrowym sercu, co oznacza, że prawdopodobnie indeks masy ciała ma wpływ na choroby serca.
- **Wiek** – zmienna wiek jest kategoryczna, ale ma aż 13 kategorii wiec zostanie zmieniona na numeryczną, aby można było zaprezentować ją na wykresie pudełkowym. Jako wiek dla każdej z kategorii przyjęto średnią z danej kategorii. Na wykresie 3.2 widać, że w tym przypadku rozkłady znacznie różnią się od

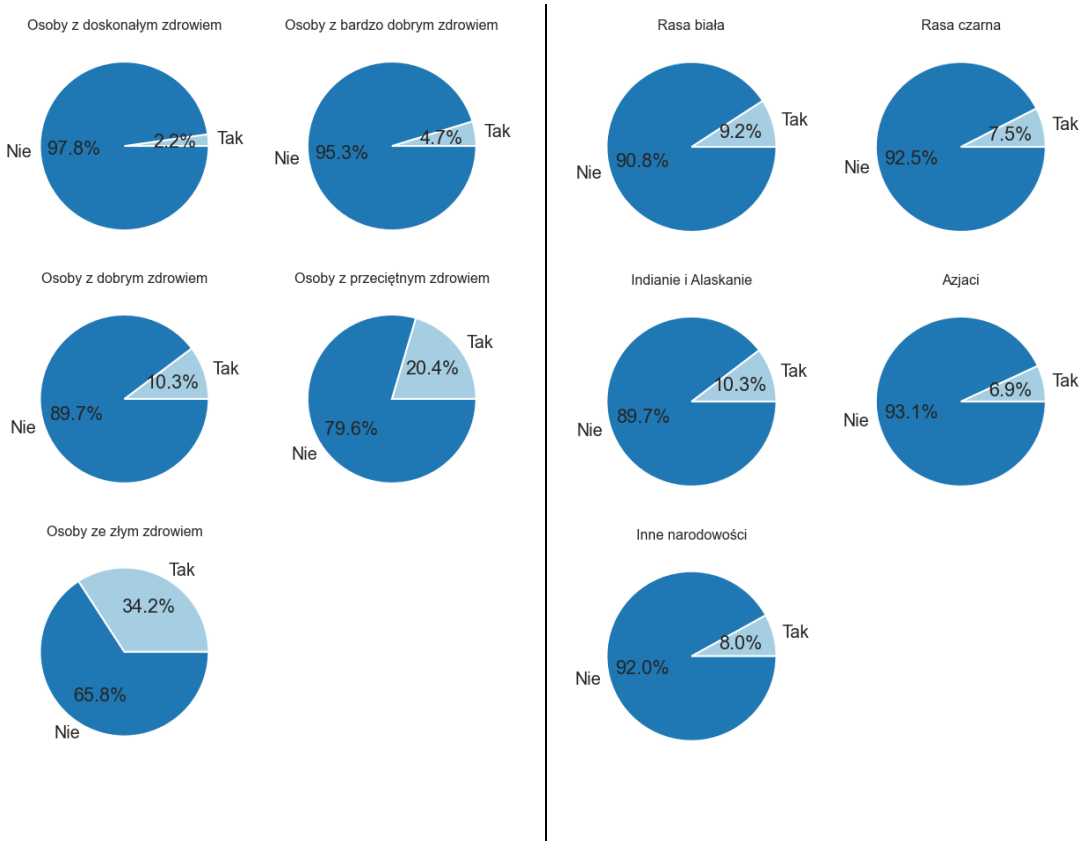
siebie. Osoby młode, które mają choroby serca to wartości odstające, a więc jest to bardzo rzadki przypadek. Wartość p testu U Manna-Whitneya wynosi 0, a więc rozkłady istotnie różnią się od siebie - należy zawrzeć tę zmienną w modelu.

- **Zdrowie fizyczne -** Zmienna przyjmuje wartości od 0 do 30. Mówi o tym, ile dni w ciągu ostatniego miesiąca osoba czuła się źle. Na wykresie 4.2 można zauważyć, że rozkłady zmiennych różnią się od siebie. Potwierdza to test U Manna-Whitneya, w którym wartość p wynosi 0, a więc rozkłady zmiennych istotnie różnią się od siebie
- **Zdrowie psychiczne** - Co ciekawe wykresy pudełkowe wyglądają tak samo w obu przypadkach. Wartość p testu U Manna-Whitneya wynosi 0,065, co świadczy o braku podstaw do odrzucenia hipotezy zerowej, a więc rozkłady zmiennych nie różnią się od siebie istotnie. Oznacza to, że prawdopodobnie zmienna nie ma wpływu na choroby serca i nie należy uwzględniać jej w modelu.
- **Sen** – Zmienna posiadała dużo wartości odstających, które usunięto, gdyż niemożliwe jest, aby osoba dorosła spała średnio mniej niż 3h lub więcej niż 20h. Mimo że wykresy wydają się niemal identyczne, to wartość p dla testu U Manna-Whitneya wynosi $3{,}95e^{-07}$, a więc rozkłady zmiennych różnią się od siebie, co świadczy o tym, że ilość snu może mieć wpływ na występowanie chorób serca.

3.4.2 Zmienne kategoryczne

W celu zbadania czy poszczególne zmienne kategoryczne są zależne od zmiennej objaśnianej „choroby serca”, zostaną wykorzystane wykresy kołowe oraz test Fishera, który bada niezależność dwóch zmiennych binarnych. Hipoteza zerowa mówi, że nie istnieje zależność pomiędzy badanymi cechami populacji, natomiast hipoteza alternatywna mówi, że istnieje zależność pomiędzy badanymi cechami populacji. Jeżeli wartość $p < 0{,}05$, odrzuca się hipotezę zerową na rzecz hipotezy alternatywnej, co oznacza, że zmienne są od siebie zależne. Dla zmiennej kategorycznej, która przyjmuje więcej niż dwie wartości, zostanie użyty test zależności Chi- kwadrat.


Mężczyźni
Tak
10.6%
Nie
89.4%
Kobiety
Tak
6.7%
Nie
93.3%
Osoby palące
Tak
12.1%
Nie
87.9%
Osoby niepalące
Tak
6.0%
Nie
94.0%
Osoby pijące dużo alkoholu
Tak
5.2%
Nie
94.8%
Osoby pijące mało alkoholu
Tak
8.8%
Nie
91.2%
Osoby, które miały udar
Tak
36.2%
Nie
63.8%
Osoby, które nie miały udaru
Tak
7.5%
Nie
92.5%
Osoby mające trudność z chodzeniem
Tak
22.5%
Nie
77.5%
Osoby niemających trudności z chodzeniem
Tak
6.3%
Nie
93.7%
Osoby mające cukrzyce
Tak
21.9%
Nie
78.1%
Osoby niemające cukrzycy
Tak
6.5%
Nie
93.5%
Osoby uprawiające aktywność fizyczną
Tak
7.0%
Nie
93.0%
Osoby nieuprawiające aktywności fizycznej
Tak
13.7%
Nie
86.3%
Osoby z chorobą nerek
Tak
29.3%
Nie
70.7%
Osoby bez choroby nerek
Tak
7.7%
Nie
92.3%
Osoby z astmą
Tak
11.4%
Nie
88.6%
Osoby bez astmy
Tak
8.1%
Nie
91.9%
Osoby, które miały raka skóry
Tak
16.7%
Nie
83.3%
Osoby, które nie miały raka skóry
Tak
7.7%
Nie
92.3%



Wykres 3.3 Obecność chorób serca u respondentów w zależności od poszczególnych zmiennych kategorycznych
Źródło: Opracowanie własne w Pythonie

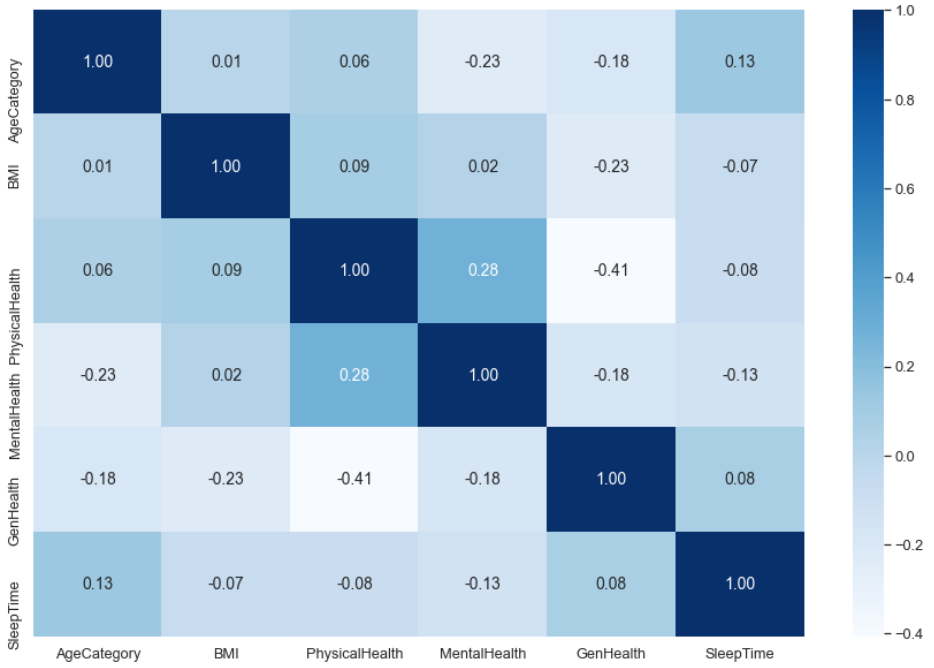
Wykres 3.3 Przedstawia obecność chorób serca u badanych w zależności od poszczególnych zmiennych kategorycznych. Na jasnoniebiesko zaznaczony jest odsetek osób, które nie mają choroby serca, natomiast na ciemnoniebiesko ci, którzy mają choroby serca. Dla każdej ze zmiennych binarnych dodatkowo wykonano test Fishera. Wartość p dla każdej zmiennej binarnej wynosi mniej niż 0,05. Oznacza to, że zmienne: płeć, palenie, picie alkoholu, udar, cukrzyca, trudność w chodzeniu, choroba nerek, astma, rak skóry, aktywność fizyczna („Czy osoba uprawiała aktywność fizyczną w przeciągu ostatnich 30 dni?”) i zmienna choroba serca są od siebie zależne, dlatego należy każdą z nich zawrzeć w modelu. Na podstawie wykresów można stwierdzić, że prawdopodobnie najsilniejsza zależność jest dla zmiennych: udar, trudność z chodzeniem i cukrzyca.

Zmienna rasa nie jest binarna, dlatego zostanie wykorzystany test niezależności Chi kwadrat. Hipoteza zerowa, analogicznie jak w przypadku testu Fishera mówi o niezależności zmiennych, natomiast alternatywna o tym, że zmienna są zależne. Wartość p dla tego testu wynosi 0,07, więc brak podstaw, aby odrzucić hipotezę zerową. Nie można stwierdzić, że zmienne są zależne, prawdopodobnie nie należy zawierać zmiennej

w modelu. Zmienna „ogólne zdrowie" jest kategoryczna, jednakże można ją uporządkować i nadać rangi odpowiednim wartościom. Im lepsze zdrowie respondenta, tym wyższa ranga (od 1 do 5). Z tego powodu w tym przypadku, analogicznie jak dla zmiennych numerycznych zostanie przeprowadzony test U Manna-Whitneya. Wartość p wynosi 0,00 co oznacza, że zależność istnieje - pacjenci z chorobami serca gorzej oceniają swoje zdrowie niż Ci bez chorób serca.

3.5 Korelacja

Zarówno zmienna objaśniana, jak i większość zmiennych objaśniających jest kategoryczna, dlatego nie można policzyć dla nich korelacji Pearsona czy Spearmana. Korelację Spearmana można jednak wykorzystać dla części zmiennych, które są numeryczne lub kategoryczne rangowe. Oprócz tego, że korelacja między zmiennymi objaśniającymi a zmienną objaśnianą powinna być wysoka, należy również pamiętać, że korelacja między zmiennymi objaśniającymi powinna być jak najmniejsza. W tym kroku zostanie sprawdzone, czy korelacja między zmiennymi objaśniającymi nie jest zbyt wysoka.



Wykres 3.4 Tablica korelacji zmiennych
Źródło: Opracowanie własne w Pythonie

Zmienne objaśniające, które są ze sobą najbardziej skorelowane to:

- Zdrowie ogólne ze zdrowiem fizycznym (-0,48)

- Zdrowie psychiczne ze zdrowiem fizycznym (0,28)

Wybierając zmienne do modelu, należy pamiętać, że obie zmienne z każdej z par nie powinny razem znaleźć się w modelu. Oznacza to, że zdrowie ogólne i zdrowie fizyczne nie powinny być w jednym modelu. Jak wiadomo z wcześniejszej analizy zdrowie psychiczne i choroby serca nie są zmiennymi zależnymi, więc zmienna zdrowie psychiczne na pewno nie zostanie uwzględniona w modelu.

4 Tworzenie modeli

Po szczegółowej analizie danych należy wybrać te zmienne, które najbardziej wpływają na chorobę serca, pamiętając, że zmienne objaśniające nie powinny być ze sobą mocno skorelowane. Atrybuty, które znajdą się w modelu to:

- BMI
- Wiek
- Ilość snu
- Ogólne zdrowie
- Płeć
- Palenie
- Alkohol
- Udar
- Cukrzyca
- Aktywność fizyczna
- Choroba nerek
- Problemy z chodzeniem
- Astma
- Rak skóry

Mimo wysokiej zależności pominięta została zmienna „zdrowie fizyczne”, gdyż była ona wysoko skorelowana ze zmienną „ogólne zdrowie”, co mogłoby negatywnie wpłynąć na model.

4.1 Kodowanie danych kategorycznych i zbalansowanie danych

Lasy losowe oraz drzewa decyzyjne nie wymagają skalowania zmiennych numerycznych, więc można pominąć ten krok i nie wpłynie to negatywnie na wyniki. Aby można było stworzyć odpowiednie modele w Pythonie należy zakodować dane kategoryczne. Zmienne, które przyjmowały wartości „tak" i „nie” zostają zamienione na zmienne binarne - odpowiednio 1 i 0. Atrybut „rasa”, który przyjmuję więcej niż 2 kategorie (5), zostanie zakodowany jako zmienne numeryczne za pomocą metody polegającej na przekształceniu kategorii do postaci binarnego wektora (ang. One-Hot Encoding). Zmienna kategoryczna „zdrowie” jest zmienną porządkową więc zostanie użyte kodowanie porządkowe. Kategorie przyjmą wartości od 1 do 5, gdzie 1 oznacza najgorsze zdrowie respondenta, a 5 najlepsze.

Jak zostało wspomniane wcześniej, zestaw danych jest niezbalansowany. Oznacza to, że ilość osób z chorobami serca (8.5% respondentów) znacząco różni się od tych, którzy nie mają chorób serca (91,5%). Jest to niepożądane zjawisko podczas tworzenia modeli klasyfikacyjnych, gdyż algorytm będzie ignorować klasę, występującą rzadziej co przełoży się na słabe wyniki podczas predykcji tej klasy. W przypadku tych danych jest to szczególnie duży problem, gdyż to właśnie wysoka skuteczność wykrywania chorób serca wśród ludzi chorych jest najważniejsza. Zdecydowanie gorszym błędem w tym przypadku jest niewykrycie choroby serca u osoby chorej niż zdiagnozowanie osoby zdrowej jako chorą. Istnieje kilka sposobów, aby poradzić sobie z niezbalansowanym zestawem danych. Analizowany zestaw danych posiada aż 319794 wierszy, więc można skorzystać z próbkowania danych polegającego na usuwaniu danych z klasy większościowej. Jest to szybka i łatwa metoda, rekomendowana tylko w przypadku dużych zbiorów danych, gdyż część z nich zostaje usunięta, więc w przypadku mniejszych zbiorów danych ryzyko usunięcia istotnych informacji jest zbyt duże. W tym przypadku ryzyko to jest niewielkie przez to, jak szeroki jest zestaw danych.

Za pomocą funkcji „RandomUnderSampler” z biblioteki imblearn w Pythonie dane zostały zbalansowane. Nowy zestaw składa się z 54316 obserwacji, z czego27158 respondentów ma chorobę serca, a 27158 osób nie ma choroby serca, a więc zestaw danych jest idealnie zbalansowany.

4.2 Podział modelu na zestaw uczący i testowy

W celu utworzenia modeli na początku dane zostały podzielone na zestaw uczący i testowy (w stosunku 70% do 30%). Na zestawie uczącym za każdym razem, kiedy będzie tworzony nowy model, zostanie przeprowadzona walidacja krzyżowa. Jest to metoda wielokrotnego próbkowania, która do trenowania i testowania modelu wykorzystuje różne części danych w różnych iteracjach. Ta technika posiada parametr k, który określa liczbę grup, na jakie ma zostać podzielony zestaw danych. Najpopularniejszym wyborem jest k =10, który zostanie zastosowany w tej pracy. Dzięki metodzie walidacji krzyżowej będzie można znaleźć model o takich hiperparametrach, które dają najlepszą jakość. Po dobraniu odpowiednich hiperparametrów najlepszy model drzewa decyzyjnego i lasu losowego zostanie przetestowany na zestawie testowym.

4.3 Drzewa Klasyfikacyjne

Wszystkie modele drzew decyzyjnych w niniejszej pracy zostaną utworzone za pomocą funkcji „DecisionTreeClassifier” z biblioteki sklearn w Pythonie. Tak jak wspomniano wcześniej drzewa decyzyjne posiadają wiele hiperparametrów, które można modyfikować, aby poprawić jakość modelu. W tym kroku zostaną zbadane poszczególne parametry i ich wpływ na jakość modelu. Drzewa decyzyjne są bardzo podatne na przeuczenie, dlatego zostaną wybrane parametry, które najbardziej ograniczają to zjawisko.

Tabela 4.1 Wyniki modelu drzewa decyzyjnego bez modyfikacji hiperparametrów
Źródło: Opracowanie własne

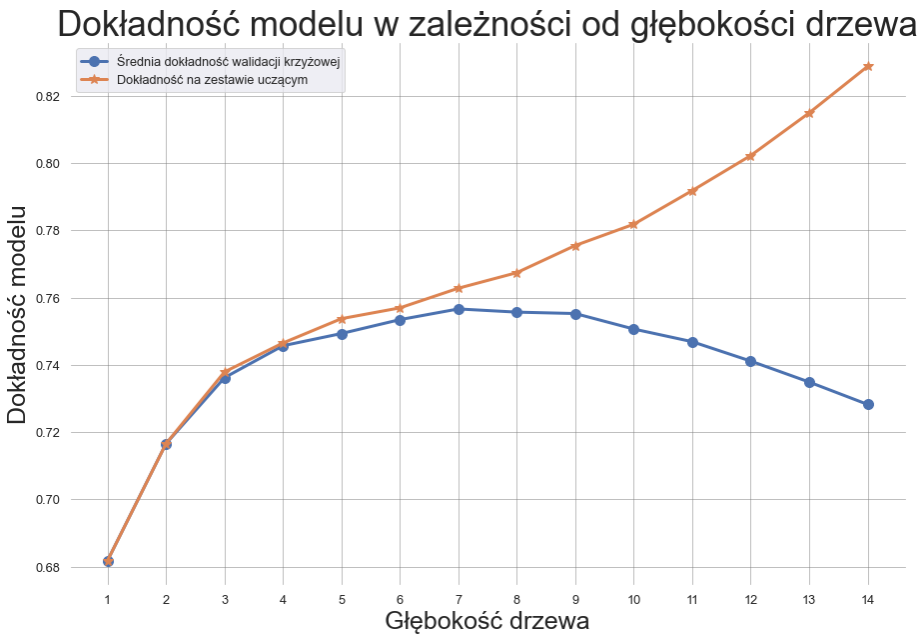
	Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
Zestaw uczący	98,03%	96,48%	99,56%	7,18s
Walidacja krzyżowa	67,42%	65,78%	67,93%	58,28s

Z poniższej tabeli wynika, że domyślne drzewo decyzyjne bez żadnej zmiany hiperparametrów osiąga bardzo dobre wyniki na zbiorze uczącym (niemal 100% dla każdego rodzaju wskaźnika). Równocześnie jednak wyniki uzyskane dla walidacji krzyżowej są znacznie gorsze — o około 30 pp. niższe dla każdego wskaźnika. Zachodzi zjawisko przeuczenia modelu, gdyż model za bardzo dopasował się do danych uczących. W następnych krokach poprzez modyfikacje hiperparametrów postarano się ograniczyć to negatywne zjawisko i zwiększyć każdą z miar jakości modelu. Czas kompilacji modelu uzyskanego za pomocą walidacji krzyżowej jest znacznie dłuższy, gdyż uzyskiwanie modelu w ten sposób jest bardziej skomplikowane obliczeniowo. Nadal jest jednak poniżej minuty, co jest zadowalającym wynikiem, biorąc pod uwagę, że zestaw danych jest bardzo obszerny.

4.3.1 Modyfikowanie hiperparametrów

- **Głębokość drzewa**

Poniższy wykres przedstawia wpływ głębokości drzewa na jakość modelu. Im większa głębokość drzewa, tym większa dokładność modelu dla danych uczących, jednakże zwiększa się również ryzyko przeuczenia modelu.



Wykres 4.1 Wykres miary dokładności modelu w zależności od głębokości drzewa
Źródło: Opracowanie własne w Pythonie

Średnia z dokładności walidacji krzyżowej przyjmuje największy wynik dla głębokości drzewa równej 8. Powyżej tej głębokości zachodzi zjawisko przeuczenia modelu. Dodatkowo im większa głębokość drzewa tym algorytm dłużej się kompiluje. W tabeli 4.2 znajdują się mierniki jakości modelu o głębokości wynoszącej 8.

Tabela 4.2 Miary jakości i czas kompilacji modelu drzewa decyzyjnego o głębokości wynoszącej 8
Źródło: Opracowanie własne w Pythonie

Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
75,69%	79,2%	74,15%	56,56s

Każdy z mierników znacznie się poprawił w porównaniu z poprzednim modelem. W 75,69% (o 8 pp. lepiej niż poprzedni) przypadków model prawidłowo zdiagnozował pacjenta. Spośród osób, które mają chorobę serca model w 79,2% (o 14 pp. lepiej niż poprzedni) przypadków ją zdiagnozował. Wynik ten jest bardzo satysfakcjonujący, a zmieniono tylko jeden parametr. Czas kompilacji skrócił się o 2 sekundy. W dalszej części pracy każdy nowy model drzewa decyzyjnego będzie miał maksymalną głębokość równą 8.

- **Gini vs. Entropia**

Wybór najlepszego atrybutu do podziału drzewa na poszczególnych etapach jego rozrostu jest dokonywany na kilka sposobów. Najpopularniejszymi metodami jest liczenie współczynnika Giniego lub Entropii. W tabeli 4.3 przedstawione są współczynniki pomiaru modelu w zależności od użytego kryterium.

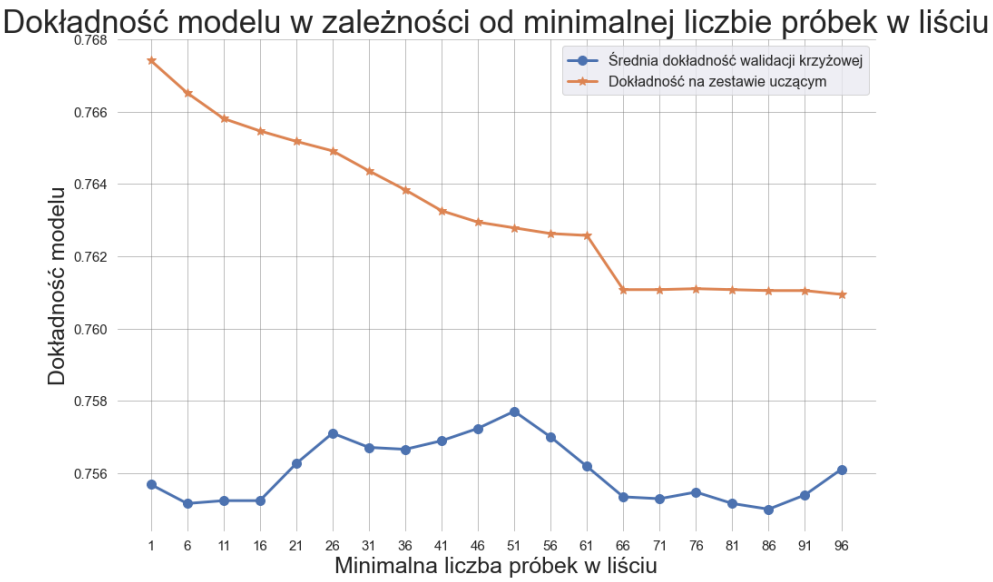
Tabela 4.3 Mierniki jakości modelu drzewa decyzyjnego w zależności od użytego kryterium do podziału drzewa
Źródło: Opracowanie własne w Pythonie

	Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
Współczynnik Giniego	75,69%	79,2%	74,15%	56,56s
Entropia	75,68%	79,16%	74,13%	56,21s

Niezależnie od użytej metody zarówno mierniki jakości modelu, jak i czas kompilacji są do siebie bardzo zbliżone, więc wybór metody w tym przypadku nie ma znaczenia.

- **Minimalna liczba próbek w liściu**

Hiperparametr ogranicza liczbę próbek w liściu w drzewie decyzyjnym, przez co gałęzie drzew mogą przyjmować różną głębokość.



Wykres 4.2 Wykres miernika dokładności modelu drzewa decyzyjnego w zależności od minimalnej liczby próbek w liściu
Źródło: Opracowanie własne w Pythonie

Niezależnie od liczby próbek w liściu dokładność modelu jest bardzo zbliżona. Gdy model posiada minimalnie od 26 do 51 próbek w liściu, średnio jest o 0,15 pp. lepszy, niż gdy hiperparametr ma domyślną wartość równą 1. Przyjęto, że najlepszy model ma minimum 51 próbek w liściu.

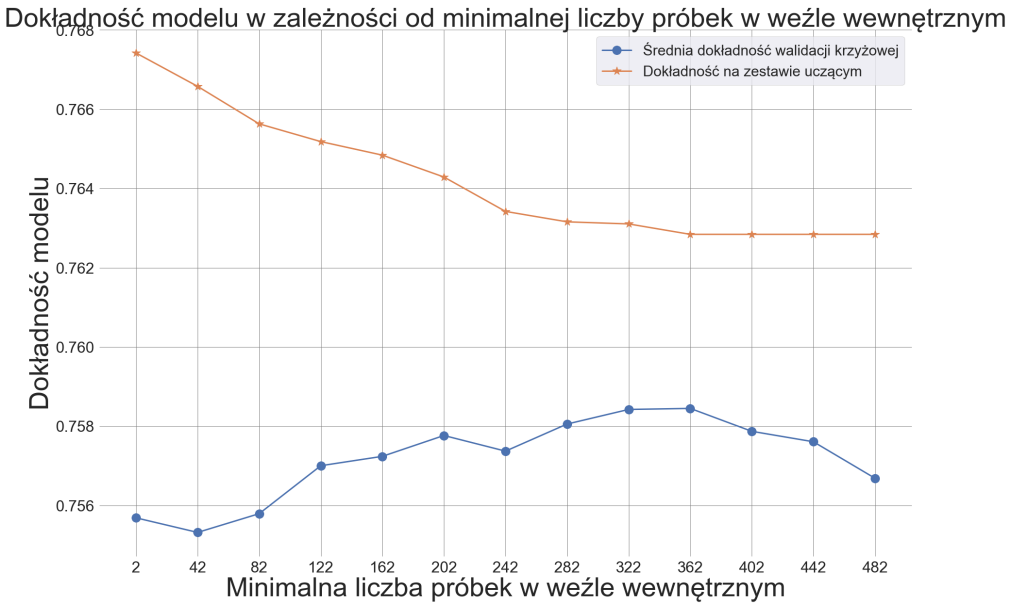
Tabela 4.4 Mierniki jakości modelu drzewa decyzyjnego o minimum 51 próbek w liściu i maksymalnej głębokości równej 8
Źródło: Opracowanie własne w Pythonie

Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
75,78%	80,77%	73,42%	56,31s

Zarówno mierniki jakości modelu, jak i czas kompilacji, gdy drzewo ma minimum 51 próbek w liściu różną się nieznacznie od modelu, który przyjmuje domyślną wartość parametru równą 1. W dalszej części pracy użyto domyślnej wartości tego hiperparametru.

- **Minimalna liczba próbek w węźle wewnętrznym**

Ten hiperparametr brzmi podobnie do poprzedniego, jednak należy pamiętać, że liść to węzeł zewnętrzny, który nie ma dzieci, natomiast ten parametr mówi o węźle wewnętrznym, który może mieć dalszy podział.



Wykres 4.3 Mierniki jakości modelu drzewa decyzyjnego dla minimum 322 próbek w węźle wewnętrznym
Źródło: Opracowanie własne w Pythonie

W tym przypadku największa jakość modelu jest dla około 322 próbek w węźle wewnętrznym. Należy jednak zwrócić uwagę, że różnice są niewielkie i dla 42 próbek jakość modelu jest najgorsza, ale tak naprawdę przekłada się to na około 0,3 pp. niższą dokładność modelu i o 2s szybszą kompilację.

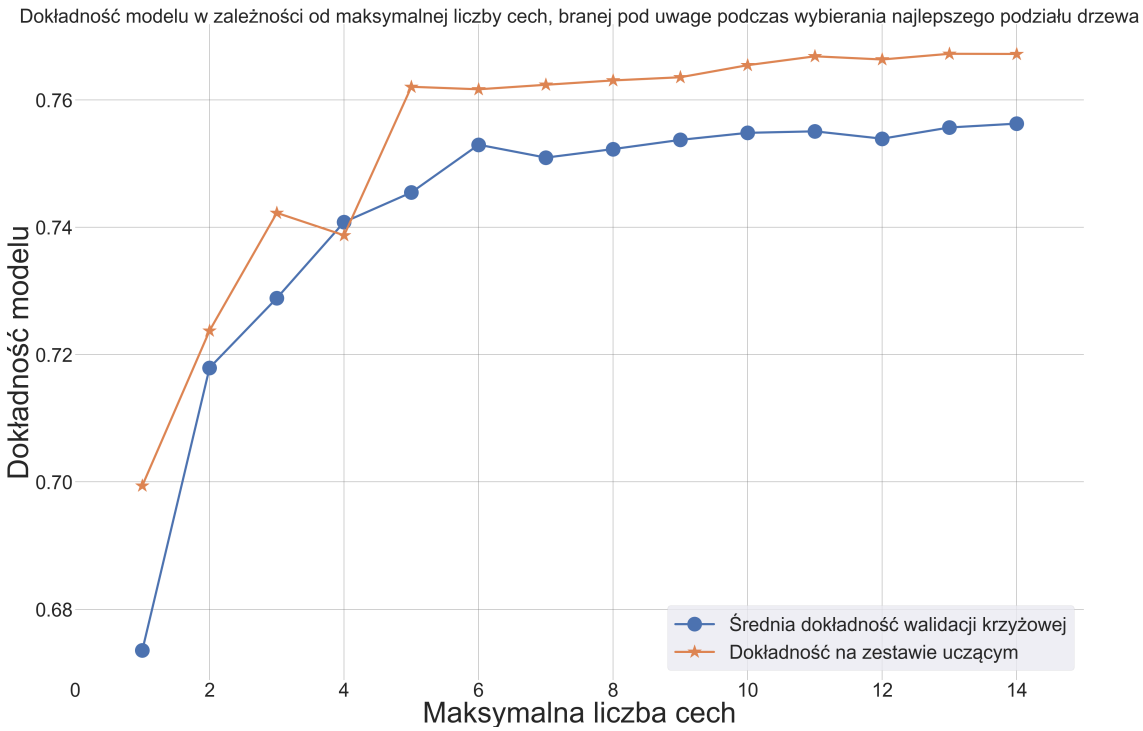
Tabela 4.5 Statystyki opisowe modelu drzewa decyzyjnego dla minimum 322 próbek w węźle wewnętrznym
Źródło: Opracowanie własne w Pythonie

Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
75,84%	80,26%	73,73%	56,4s

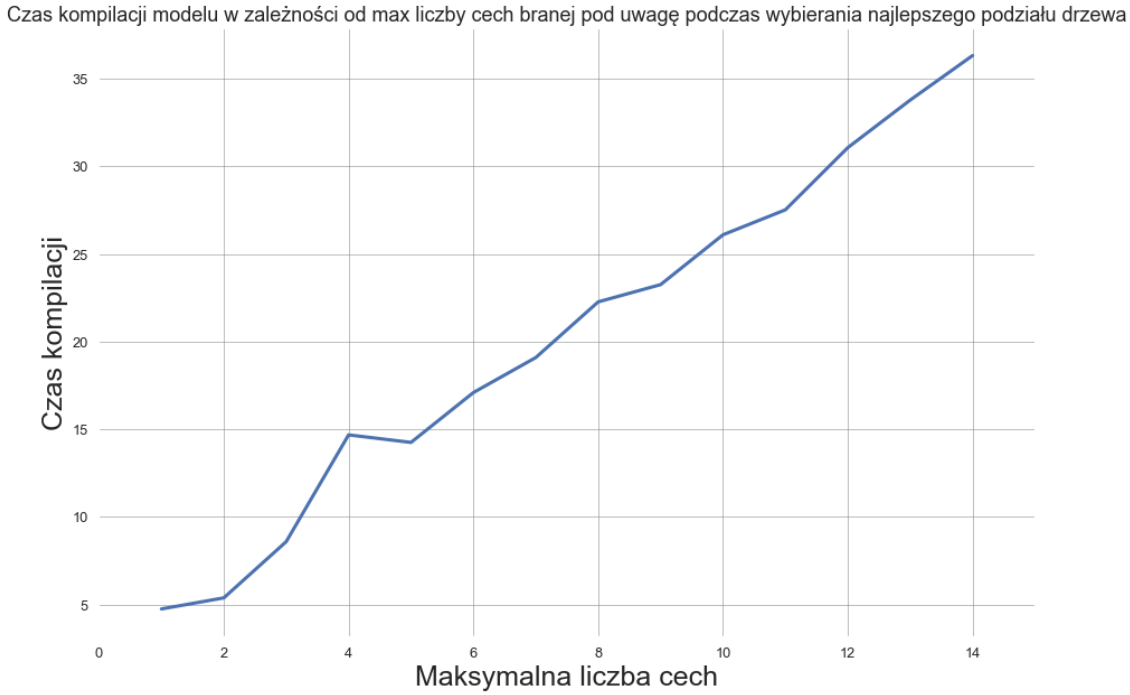
Zarówno mierniki jakości modelu, jak i czas kompilacji, gdy drzewo ma minimum 322 próbek w węźle wewnętrznym różną się nieznacznie od modelu, który przyjmuje domyślną wartość parametru równą 2. W dalszej części pracy użyto domyślnej wartości tego hiperparametru.

- **Maksymalna liczba cech**

Gdy model przy każdym podziale bierze pod uwagę wszystkie cechy, jest to bardzo czasochłonne dla algorytmu, więc istnieje hiperparametr, który sprawia, że algorytm przy każdym podziale bierze pod uwagę tylko część cech (np. pierwiastek z wszystkich cech czy logarytm o podstawie dwa z wszystkich cech). W tym przypadku model może mieć maksymalnie 14 zmiennych, więc zostanie zbadana dokładność modeli, które przyjmują maksymalnie od 1 do 14 cech.



Wykres 4.4 Dokładność modelu drzewa decyzyjnego w zależności od maksymalnej liczby cech, branej pod uwagę podczas wybierania najlepszego podziału drzewa



Wykres 4.5 Czas kompilacji modelu drzewa decyzyjnego w zależności od maksymalnej liczby cech branej pod uwagę podczas wybierania najlepszego podziału drzewa
Źródło: Opracowanie własne w Pythonie

Na wykresie 4.4 można zauważyć, że od dziesięciu atrybutów w górę nie ma znaczącej różnicy w jakości modelu, jednakże znacząco na niekorzyść zmienia się czas

kompilacji. Oznacza to, że optymalna liczba cech wynosi 10, gdyż gdy model bierze pod uwagę wszystkie zmienne, jakość modelu jest minimalnie większa, a czas kompilacji dłuższy o około 12 sekund.

Tabela 4.6 Mierniki jakości modelu drzewa decyzyjnego, gdy model bierze pod uwagę maksymalnie 8 atrybutów podczas wybierania najlepszego podziału drzewa
Źródło: Opracowanie własne w Pythonie

Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
75,58%	80,26%	73,38%	30,85s

4.4 Lasy losowe

Modele lasów losowych będą tworzone za pomocą funkcji „RandomForestClassifier" z biblioteki „sklearn" Pythona. W pierwszym kroku stworzono model z domyślnymi hiperparametrami i mierniki jakości modelu widoczne są w tabeli poniżej.

Tabela 4.7 Mierniki jakości modelu drzewa decyzyjnego z domyślnymi hiperparametrami
Źródło: Opracowanie własne w Pythonie

	Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
Zestaw uczący	98,29%	96,48%	99,56%	76,77s
Walidacja krzyżowa	71,6%	72,83%	71,03%	623,97s

Wszystkie mierniki jakości modelu lasu losowego są lepsze niż w przypadku domyślnego modelu drzewa decyzyjnego. Nie jest to zaskoczeniem, gdyż ten typ modelu jest mniej podatny na przeuczenie niż drzewa decyzyjne, co przełożyło się na mniejsze różnice w jakości modelu pomiędzy zestawem uczącym i średnią z walidacji krzyżowej. Minusem jest znacznie dłuższy czas kompilacji modelu, co jednak można zredukować, modyfikując odpowiednio parametry.

4.4.1 Modyfikowanie hiperparametrów

Analogicznie jak w przypadku drzew decyzyjnych, przeanalizowano kilka hiperparametrów oraz ich wpływ na jakość i prędkość kompilacji modelu. Ze względu na to, że lasy losowe są dużo mniej podatne na przeuczenie prawdopodobnie modyfikacja parametrów nie usprawni tak modeli jak w przypadku drzew decyzyjnych. Domyślne

parametry, dają zadowalające wyniki, co oczywiście nie oznacza, że nie można ich poprawić.

- **Maksymalna głębokość drzewa w lesie losowym**

Gdy wartość maksymalnej głębokości drzewa nie jest określona, nie tylko model jest bardziej podatny na przeuczenie, ale również algorytm dłużej się kompiluje, co w przypadku lasów losowych ma szczególne znaczenie, gdyż algorytm nie buduje jednego drzewa, a kilkadziesiąt lub kilkaset. W tym przypadku nie trzeba tworzyć kolejnego wykresu, wystarczy przeanalizować wykres 4.1, gdzie widać, że już dla maksymalnej głębokości równej 8, dokładność modelu jest wysoka.

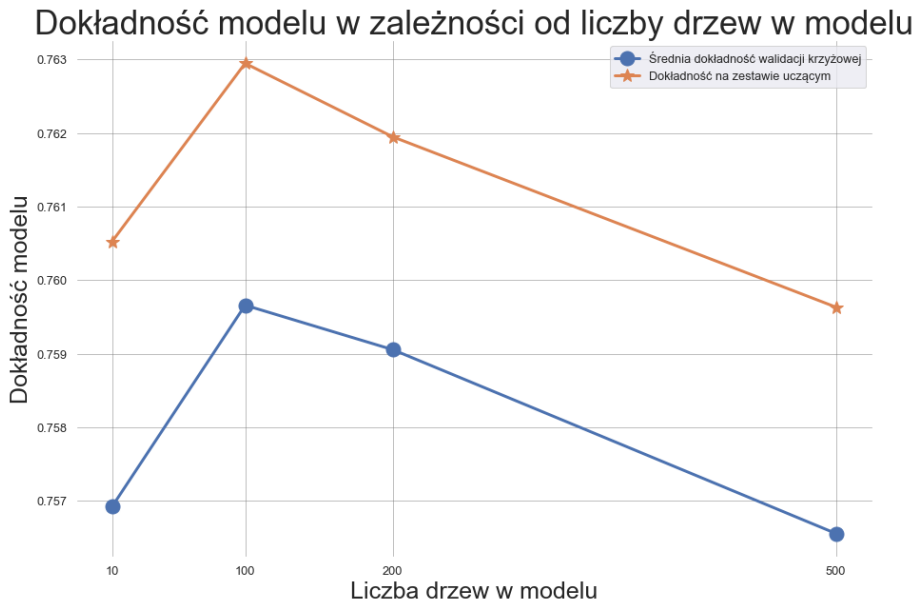
Tabela 4.8 Mierniki jakości modelu lasu losowego o maksymalnej głębokości drzewa równej 8
Źródło: Opracowanie własne w Pythonie

Dokładność	**Czułość**	**Precyzja**	**Czas kompilacji**
76,1%	81,15%	73,66%	550,72s

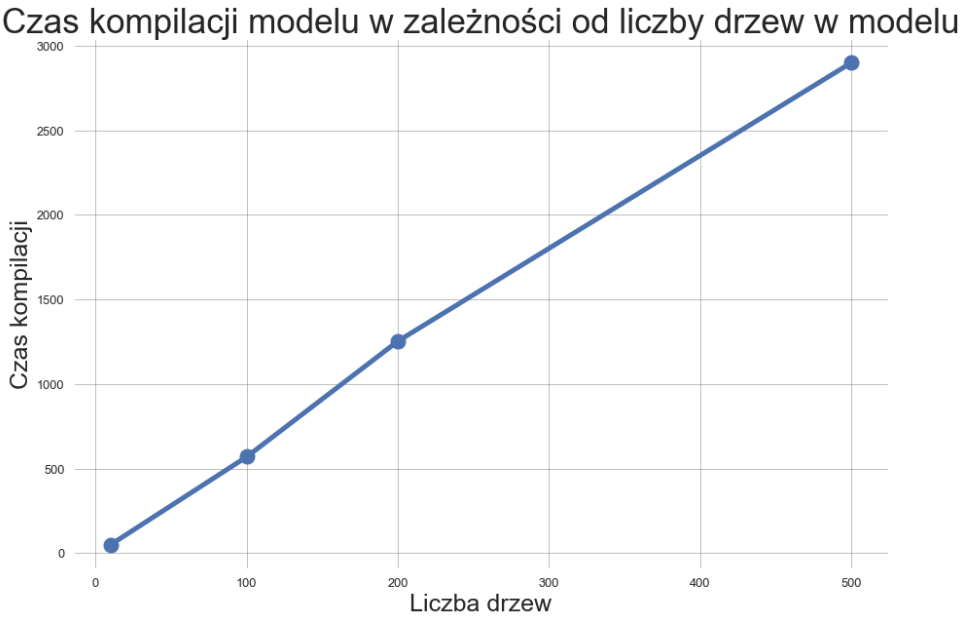
Nie tylko czas kompilacji zmniejszył się o 73 sekundy, ale również wszystkie mierniki jakości modelu znacznie się poprawiły, w szczególności czułość, która wynosi aż 81,15% (o prawie 10 pp. więcej). Jest to bardzo zadowalającym wynikiem, gdyż oznacza, że jeśli pacjent ma chorobę serca, to model na 81,15% ją zdiagnozuje. Dokładność wynosi 76,1%, czyli model myli się z diagnozą mniej więcej przy co czwartym pacjencie. W dalszej części badania każdy model lasu losowego będzie przyjmował maksymalną głębokość równą 8.

- **Liczba drzew**

Zwykle im więcej drzew tym lepsza dokładność modelu, jednakże istotnie zwiększa się również wtedy czas kompilacji. Należy znaleźć moment, kiedy korzyść ze zwiększenia jakości modelu, nie jest już warta dodatkowego czasu obliczeniowego.



Wykres 4.6 Dokładność modelu lasu losowego w zależności od liczby drzew w modelu
Źródło: Opracowanie własne w Pythonie

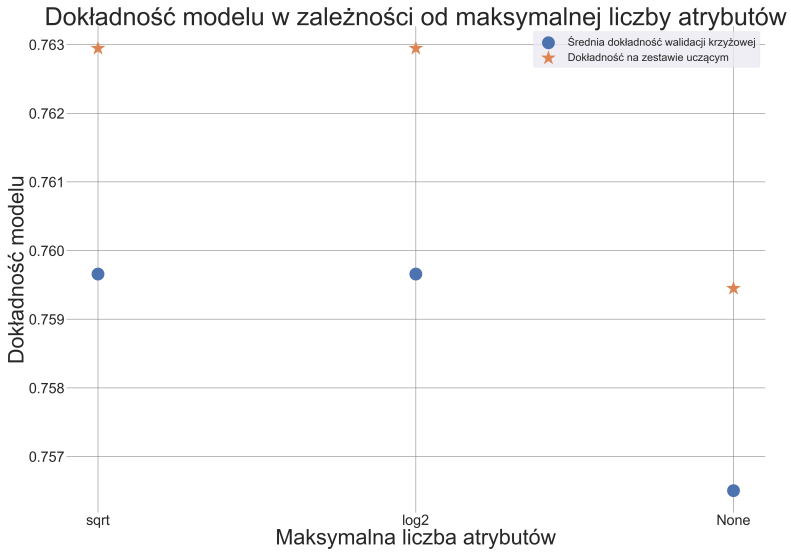


Wykres 4.7 Czas kompilacji w zależności od liczby drzew w modelu lasu losowego
Źródło: Opracowanie własne w Pythonie

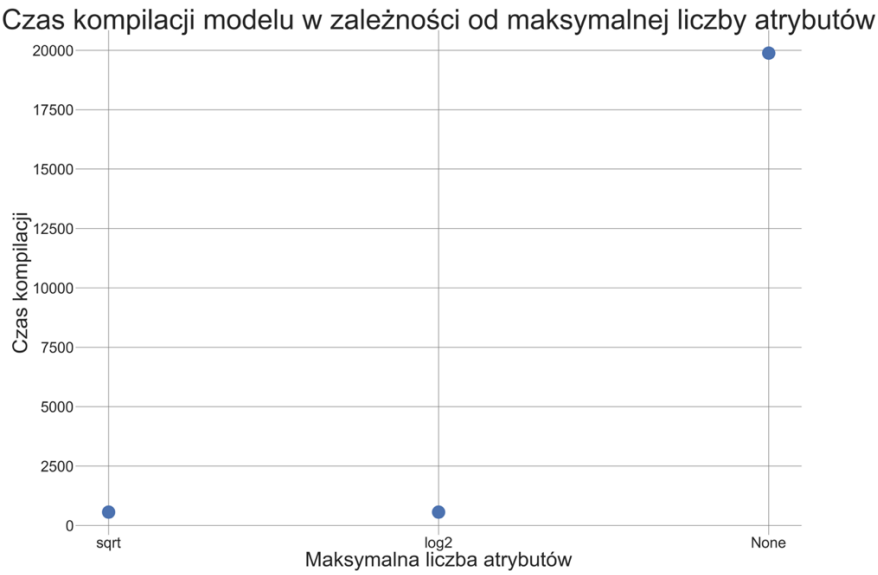
Wykres 4.6 pokazuje, że model jest najdokładniejszy dla 100 drzew. Wraz ze wzrostem liczby drzew powyżej tej wartości dokładność zaczyna się pogarszać, a czas kompilacji znacznie wydłuża (wykres 4.7). W związku z tym domyślna wartość hiperparametru ilości drzew równa 100 jest najlepsza i nie trzeba jej zmieniać. Mierniki jakości modelu i czas kompilacji modelu o 100 drzewach zawarte są w tabeli 4.8.

- **Maksymalna liczba atrybutów**

Jest to maksymalna liczba cech, której używa drzewo, dokonując każdego następnego podziału — najczęściej używana wartość i równocześnie domyślna wartość dla funkcji „RandomForestClassifier” w Pythonie to pierwiastek z wszystkich atrybutów w zestawie danych (inaczej niż w przypadku drzew decyzyjnych, gdzie równała się ona wszystkim atrybutom). Pozostałe, często używane wartości to $\log_2$(liczba wszystkich atrybutów) lub ‘None’, który oznacza, że wszystkie atrybuty są brane pod uwagę.



Wykres 4.8 Dokładność modelu lasu losowego w zależności od maksymalnej liczby atrybutów branej pod uwagę podczas każdego podziału drzew
Źródło: Opracowanie własne w Pythonie



Wykres 4.9 Czas kompilacji w zależności od maksymalnej liczby atrybutów branej pod uwagę podczas każdego podziału drzew
Źródło: Opracowanie własne w Pythonie

Analizując wykres 4.8 i 4.9 widać, że również w tym przypadku model uzyskuje najlepsze wyniki dla domyślnego hiperparametru, który w tym przypadku wynosi pierwiastek z wszystkich atrybutów w zestawie danych. Wraz ze wzrostem liczby atrybutów nie tylko jakość modelu się nie zwiększa, ale również czas kompilacji jest znacznie dłuższy. Mierniki jakości modelu z maksymalną liczbą atrybutów równą pierwiastkowi z wszystkich atrybutów znajdują się w tabeli 4.8.

- **Liczba zadań, które mają być wykonywane równolegle**

Mówi komputerowi, ile procesorów może on używać. Jeśli wartość wynosi 'None' oznacza to, że może używać tylko jednego procesora, wartość -1 oznacza, że nie ma żadnego ograniczenia.

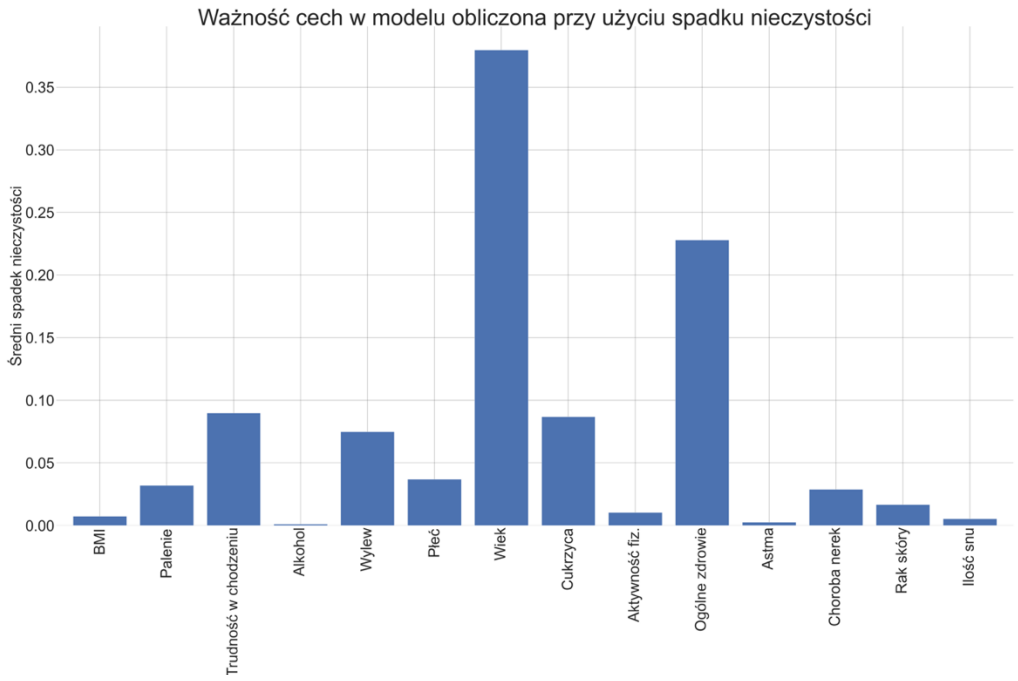
Tabela 4.9 Mierniki jakości modelu lasu losowego i czas kompilacji w zależności od liczby procesorów, które może komputer używać
Źródło: Opracowanie własne

N_jobs	Dokładność	Czułość	Precyzja	Czas kompilacji
-1	76,085	81,15%	73,66%	127,29s
None	76,1%	81,15%	73,66%	550,72s

Różnica w czasie kompilacji jest znacząca, jeśli zmienimy hiperparametr z domyślnego na -1. Model kompiluje się aż 4 razy szybciej i przyjmuje niemal identyczne wyniki.

4.4.2 Ważność cech

Lasów losowych w przeciwieństwie do drzew decyzyjnych nie można zwizualizować. Jednakże możliwe jest określenie ważności każdej ze zmiennej zawartej w modelu. Jest to bardzo przydatne, gdyż w prosty sposób można zwizualizować, w jaki sposób poszczególne atrybuty wpływają na choroby serca. Ważność atrybutów została policzona na podstawie średniego spadku nieczystości. Sposób działania tej metody został przedstawiony w części teoretycznej.



Wykres 4.10 Ważność cech w modelu obliczana na podstawie średnio spadku nieczystości
Źródło: Opracowanie własne w Pythonie

Z wykresu 4.10 jasno wynika, że spośród cech użytych w modelu, najważniejsze to: wiek respondenta, ogólne zdrowie, czy respondent ma problemy z chodzeniem i czy miał kiedykolwiek cukrzyce lub wylew. Oznacza to, że oceniając prawdopodobieństwo choroby serca u pacjenta, przede wszystkim powinniśmy zwracać uwagę na te cechy. Co ciekawe niewielki wpływ na chorobę ma to, czy pacjent spożywa dużo alkoholu, kiedykolwiek chorował na astmę oraz średnia ilość snu badanego. Prawdopodobnie wykluczenie tych zmiennych z modelu nie miałoby znacznego wpływu na jego dokładność, a mogłoby przyspieszyć czas kompilacji.

4.5 Porównanie dwóch najlepszych modeli

W ostatniej części zostanie przeprowadzone porównanie modelu drzewa decyzyjnego i lasu losowego, które osiągnęły najlepsze wyniki. Modele zostaną przetestowane na danych testowych, które do tej pory nie były używane.

Tabela 4.10 Porównanie modelu drzewa decyzyjnego o głębokości = 8, maksymalnej liczbie atrybutów = 10 oraz lasu losowego o maksymalnej głębokości drzew = 8, n_jobs=-1, 100 drzewach i max liczbie atrybutów = pierwiastek z wszystkich atrybutów.
Źródło: Opracowanie własne

Rodzaj modelu	**Dokładność**	**Czułość**	**Precyzja**	**Czas kompilacji**
Drzewo decyzyjne	76,54%	82,48%	73,71%	3,09s
Las losowy	76,32%	80,39%	73,98%	9,43s

Oba modele przyjmują bardzo podobne mierniki jakości. Najważniejszy cel tego badania został spełniony, gdyż w obu przypadkach każdy z mierników jakości modelu osiąga wysokie wartości — powyżej 70%. Różnice między dokładnością i precyzją są minimalne, większą różnicę można zauważyć w przypadku czułości — dla lasów losowych jest ona o 2 punkty procentowe mniejsza. Tak jak zostało wspomniane wcześniej, jest to bardzo ważny miernik jakości w przypadku tego zestawu danych, o ile nie najważniejszy. Lasy losowe w 80,39% przypadkach poprawnie diagnozują chore osoby, a drzewa decyzyjne w 82,48%. Dla obu modeli czas kompilacji skrócił się istotnie, gdyż nie zastosowano walidacji krzyżowej, która znacznie wydłuża ten proces. Przy ostatecznym trenowaniu i testowaniu najlepszych modeli, nie korzysta się z walidacji krzyżowej. Algorytm lasów losowych nie tylko osiągnął trochę gorsze wyniki, ale również kompiluje się 3 razy dłużej niż algorytm lasu decyzyjnego, co jest istotną różnicą. Jednakże należy pamiętać, że model lasu losowego bez żadnych zmian osiągał bardzo dobre wyniki (tabela 4.7), natomiast hiperparametry drzewa decyzyjnego trzeba było odpowiednio dostosować, aby uzyskać satysfakcjonujące rezultaty. Hiperparametr, który w obu przypadkach miał największy wpływ na jakość modelu to maksymalna głębokość drzewa. Nie tylko poprawił on każdy miernik jakości modelu, ale także znacznie skrócił czas kompilacji modelu, szczególnie w przypadku lasów losowych, gdzie generuje się aż 100 drzew, więc każda sekunda zaoszczędzona na pojedynczym modelu drzewa jest istotna.

Zakończenie

Podsumowanie badania

Celem niniejszej pracy było zastosowanie drzew klasyfikacyjnych i lasów losowych w predykcji chorób serca. Stworzono modele drzew decyzyjnych i lasów losowych i za pomocą modyfikacji hiperparametrów starano się poprawić ich jakość. Model drzewa klasyfikacyjnego osiągnął lepsze mierniki jakości modelu oraz krótszy czas kompilacji, jednakże różnice we współczynnikach jakości modelu wahały się na poziomie jedynie kilku dziesiątych pp. dla dokładności i precyzji oraz około 2 pp. dla czułości.

Najlepszy model drzewa decyzyjnego, który zbudowano, miał maksymalną głębokość wynoszącą 8 i maksymalną liczbę atrybutów równą 10. Współczynnik dokładności tego modelu wyniósł 76,54%, czułość 82,48%, a precyzja 73,71%, czas kompilacji modelu to 3,09s. Powyższe wyniki pokazują, że udało się osiągnąć jeden z celów badania, którym było uzyskanie wszystkich trzech miar jakości na wysokim poziomie (minimum 70%). Według najlepszej wiedzy autora na podstawie przeanalizowanych badań historycznych przeprowadzonych na tym zbiorze danych, do tej pory większość modeli osiągała wysoki współczynnik dokładności, jednakże czułość i precyzja pozostawały na bardzo niskim poziomie (około 20% - 50%). Takie modele nie nadawały się do medycznej diagnozy, gdyż w tym przypadku współczynnik czułości, mówiący o prawidłowej diagnostyce osób chorych, jest bardzo ważną miarą. Uzyskanie drzewa klasyfikacyjnego o wysokiej jakości było możliwe nie tylko dzięki zbalansowaniu zbioru danych, ale także poprzez modyfikację hiperparametrów modelu. Parametr, który najbardziej wpłynął na jakość modelu drzewa klasyfikacyjnego i czas kompilacji to maksymalna głębokość drzewa. Za pomocą walidacji krzyżowej zbadano, jak poszczególne głębokości drzewa wpływają na jego jakość i zauważono, że najwyższą osiąga on dla maksymalnej głębokości równej 8. Dzięki modyfikacji tego hiperparametru zwiększono dokładność modelu o prawie 10 pp., a czułość aż o 17 pp., neutralizując w ten sposób zjawisko przeuczenia modelu. Modyfikacje hiperparametrów: liczba próbek w węźle wewnętrznym, minimalna liczba próbek w liściu oraz metoda podziału drzewa, nie wpłynęły znacząco na jakość modelu, ani na czas kompilacji, dlatego pozostawiono domyślne wartości tych hiperparametrów. Ostatni hiperparametr – liczba atrybutów do podziału, została zmieniona z domyślnej wartości równej wszystkim atrybutom do 10.

Zauważono, że powyżej 10 atrybutów jakość modelu się nie zmienia, ale czas kompilacji jest dużo dłuższy, co tłumaczy zasadność tej modyfikacji.

Lasy losowe mimo tego, że powszechnie uważane są za bardziej skuteczny algorytm niż drzewa decyzyjne, osiągnęły gorsze wyniki. Dokładność modelu wynosiła 76,32%, czułość 80,39% a precyzja 73,89% przy czasie kompilacji równym 9,43 sekundy. Jednakże nie sposób pominąć, że bez zmiany hiperparametrów osiągnął on znacznie lepsze wyniki niż domyślny model drzewa decyzyjnego – współczynnik czułości na poziomie 72,83%, gdzie dla drzewa było to 65,78%. Również w przypadku lasu losowego największy wpływ na jego jakość miała zmiana parametru maksymalnej głębokości drzewa. Ustalenie tego parametru na poziomie 8 przełożyło się na zwiększenie dokładności o 5 pp., a czułości o 8 pp. Zbadano również, że dla hiperparametrów ilość drzew oraz maksymalna liczba atrybutów do podziału model uzyskuje najlepsze wyniki dla domyślnych wartości tych hiperparametrów, czyli kolejno 100 i pierwiastek z wszystkich atrybutów. Ostatni parametr, który zbadano i został zmodyfikowany to liczba procesorów, z których może korzystać komputer. Gdy komputer korzystał ze wszystkich procesorów, zamiast tylko 1, model kompilował się 4 razy szybciej.

W pracy sprawdzono również, które cechy mają największy wpływ na choroby serca. Wiek respondenta, ogólne zdrowie, obecność cukrzycy i trudność z chodzeniem okazały się czynnikami, które najbardziej determinowały zachorowalność. Zostały one wyselekcjonowane na podstawie średniego spadku nieczystości obliczonego w modelu lasu losowego. Te wyniki są spójne z zależnościami zidentyfikowanymi na podstawie wykresów i testów statystycznych opisanych w rozdziale czwartym.

Powyższe wyniki pozwalają na przedstawienie kilku rekomendacji profilaktyki chorób serca. Osoby starsze powinny regularnie korzystać z konsultacji kardiologicznych, ponieważ wraz z wiekiem narażenie na choroby serca wzrasta. W sytuacji zaobserwowania u siebie zadyszki podczas chodzenia również należy skonsultować się z lekarzem, gdyż jest to jeden z istotnych objawów choroby serca. Aby uniknąć cukrzycy, a co za tym idzie nie zwiększyć ryzyka zapadnięcia na choroby serca, należy nie spożywać słodkich rzeczy w nadmiarze, regularnie badać poziom cukru we krwi na czczo, a w przypadku zachorowania pamiętać o kontrolnych wizytach u kardiologa. Czynnikami o najmniejszym wpływie na rozwój chorób serca okazało się spożywanie alkoholu oraz występowanie astmy u pacjenta. Prawdopodobnie usunięcie ich z modelu nie wpłynęłoby negatywnie na jego jakość.

Ograniczenia przeprowadzonego badania

Każde przeprowadzone badanie posiada pewne ograniczenia, które wpływają na przebieg analizy oraz jej wynik. Najważniejsze jest jednak, aby w ostatecznym rozrachunku nie zaburzały one w znaczącym stopniu wiarygodności przeprowadzonego badania. Celem tego podrozdziału jest krótka charakterystyka ograniczeń niniejszego badania, których świadomość pozwoli odbiorcy tej pracy na jak najbardziej efektywne jej wykorzystanie.

Dane, które zostały wykorzystane w tej pracy, pochodzą z wiarygodnego i powszechnie uznawanego źródła, jakim jest „Centers do Fisease Control and Prevention" w USA i zostały skrupulatnie przygotowane do badania. Jednakże mimo tego należy wziąć pod uwagę potencjalne defekty tych danych. Powszechnie wiadome jest, że w przypadku ankiet, szczególnie tych telefonicznych nie można być pewnym prawdomówności respondentów. Istnieje wiele badań, które potwierdzają, że jako ludzie mamy tendencje do „idealizowania" swojej osoby. Telefoniczny charakter ankiety mógł skłaniać niektórych respondentów do przeinaczania faktów, gdyż po drugiej stronie słuchawki siedział człowiek, co sprawiało, że nie czuli się w pełni anonimowi. W przypadku danych wykorzystanych do tego badania, szczególnie kilka zmiennych było na to narażonych. Jednym z nich niewątpliwie mogło być BMI pacjenta, gdyż dla wielu osób otyłość jest chorobą niezwykle wstydliwą i zaniżającą ich samoocenę. Niezależne badania niejednokrotnie pokazały, że respondenci zaniżają wartość tego parametru, aby wypaść lepiej w oczach ankietera lub sami przed sobą nie chcą przyznać jak poważna jest ich choroba. Pytanie o palenie papierosów i picie alkoholu również mogło być narażone na fałszywe odpowiedzi, gdyż wiele osób wstydzi się swoich nałogów. W zbiorze danych było też kilka zmiennych, które podlegały bardzo subiektywnej opinii, takie jak na przykład ogólne zdrowie pacjenta (wyrażone w skali od 1 do 5) oraz ilość dni w ciągu ostatnich 30 dni, w których pacjent źle się czuł. W tym przypadku ciężko jest porównać jedną odpowiedź do drugiej, gdyż niektóre osoby mają tendencje do bagatelizowania wszelkich objawów i nigdy nie narzekają na swoje zdrowie, natomiast inni posiadają skłonności hipochondryczne i wyolbrzymiają swoje negatywne samopoczucie, co sprawia, że sprowadzenie wszystkich respondentów do jednego mianownika na podstawie zadeklarowanych wartości może nie być w pełni miarodajne.

Rekomendacje dotyczące dalszych badań

Badania dotyczące przewidywania chorób serca (czy też jakiejkolwiek innej choroby) dostarczają wielu cennych informacji, które mogą uratować wiele ludzkich istnień. Z tego powodu tak ważne jest, aby ciągle rozszerzać ich zakres i analizować nowe zestawy danych co pozwoli uzyskiwać jak najdokładniejsze modele i analizy.

Dalsza praca nad modelami drzewa klasyfikacyjnego i lasu losowego przewidujących choroby serca może obejmować kolejne metody ulepszania modeli. W tej pracy w części empirycznej zastosowano tylko przycinanie w trakcie wzrostu. Dobrym rozwiązaniem mogłoby okazać się również przycinanie po rozroście drzewa, które często daje lepsze wyniki niż poprzednia metoda, gdyż pozwala na lepsze dostosowanie do danego problemu, co mogłoby się przełożyć na uzyskanie jeszcze lepszych wyników modelu. Podobne modelowanie można również przeprowadzić na innych zestawach danych, które posiadają informacje o poziomie ciśnienia i cholesterolu respondenta, gdyż według wielu badań te czynniki mają bardzo duży wpływ na choroby serca. Wracając na moment do opisanych wyżej ograniczeń niniejszego badania, podobną analizę można powtórzyć dla danych zbieranych ze stuprocentowym zachowaniem anonimowości respondenta i zweryfikować czy wówczas wpływ niektórych zmiennych takich jak np. spożywanie alkoholu, czy BMI okaże się istotnie większy. Dodatkowo zalecane jest stworzenie innych modeli uczenia maszynowego takich jak na przykład sieci neuronowe czy regresja logistyczna i porównanie ich z drzewami decyzyjnymi oraz lasami.

Bibliografia

1. Leo Breiman, “Random Forests”, Wydział Statystyki Uniwersytet Kalifornii Berkley 2011
2. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, Charles J.Stone, “Classification and Regression Trees”, Chapman and Hall/CRC, Nowy Jork 2017
3. Andriy Burkov, “The Hundred Page Machine Learning Book”, Andriy Burkov, 2019
4. Niklas Donges, 22 lipca 2021, “Random Forest Algorithm: A Complete Guide” https://builtin.com/data-science/random-forest-algorithm, dostęp: 10.05.2022
5. Aurélien Géron, „Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems”,Wyd. II, O,Reilly Media, Sebastopol 2019
6. Himani Gulati, 11 lutego 2022 „Hyperparameter Tuning in Decision Trees and Random Forests” https://www.section.io/engineering-education/hyperparmeter-tuning/, dostęp: 6.05.2022
7. Prashant Gupta, 17 maja 2017, „Drzewa decyzyjne w uczeniu maszynowym” https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052, dostęp: 6.05.2022
8. Scott Hartshorn, “Machine Learning with Random Forest and Decision Trees A Visual Guide For Beginners”, 2016
9. Tom M. Mitchell, “Machine Learning”, Wyd. I, McGraw-Hill Education, Nowy Jork 1997
10. Mukuesh Mithrakumar, 11 listopad, 2019 “How to tune a decision tree?” https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680, dostęp: 6.05.2022
11. Kamil Pytlak, 2022 „Indywidualne kluczowe wskaźniki chorób serca” https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease dostęp: 15.04.2022
12. Sebastian Raschka, “Python Machine Learning”, Pack Publishing, Brimingham 2015
13. Lior Rokach, Oded Maimon “Data mining with decision trees Theory and applications”,World Scientific, Izrael 2014

14. Stuart Russell, Peter Norvig, “Artificial Intelligence A Modern Approach”, Wyd. III, Pearson Education, New Jersey 2010
15. Abhishek Sharma 30 czerwca 2020, “4 Simple Ways to Split a Decision Tree in Machine Learning”, https://www.analyticsvidhya.com/blog/2020/06/4-ways-split-decision-tree/ dostęp: 20.04.2022
16. Neelam Tyagi 22.03.2021 “What is Information Gain and Gini Index in Decision Trees?”, 20.04.2022
17. Ian H. Witten, Eibe Frank, Mark A.Hall “Data mining Practical Machine Learning Tools and Techniques”, Wyd. III, Elsevier, Burlington 2011
18. Soner Yildirim, 28 lipca 2020, „Hyperparameters of Decision Trees explained with visualisation” https://towardsdatascience.com/hyperparameters-of-decision-trees-explained-with-visualizations-1a6ef2f67edf, dostęp: 30.04.2022
19. Tony Yiu, 12 czerwiec 2019, „Rozumienie lasu losowego: Jak działa algorytm i dlaczego jest tak skuteczny?” https://towardsdatascience.com/understanding-random-forest-58381e0602d2m dostęp: 10.05.2022
20. Joaquin Vanschoren, Rafael Gomes Mantovani, Tomáš Horváth, Ricardo Cerri, Sylvio Barbon Junior, André Carlos Ponce de Leon Ferreira de Carvalho, “An empirical study on hyperparameter tuning of decision trees” https://arxiv.org/abs/1812.02207

Spis tabel, rysunków i wykresów

Spis tabel

Spis wykresów

Spis rysunków