Akademia Górniczo – Hutnicza im. Stanisława Staszica w Krakowie
Wydział Zarządzania

Uczenie Maszynowe projekt zaliczeniowy

Mateusz Mulka, Karol Kuciński

Spis treści

Wprowadzenie

Choroby układu sercowo-naczyniowego (CVD) stanowią istotny problem zdrowotny na skalę globalną, odpowiadając za znaczną część rocznej liczby zgonów na całym świecie. Ze względu na ich powszechność oraz wpływ na umieralność, wczesne wykrywanie i skuteczne zarządzanie CVD stanowią priorytet. Techniki uczenia maszynowego pojawiły się jako potężne narzędzia w opiece zdrowotnej, oferując potencjał analizy złożonych zbiorów danych oraz pomoc w modelowaniu predykcyjnym w diagnostyce chorób i ocenie ryzyka.

Badany zbiór danych zawiera istotne informacje zebrane z różnych źródeł, konsolidując różnorodne atrybuty związane ze zdrowiem układu sercowo-naczyniowego. Obejmuje 11 kluczowych cech, takich jak wiek, płeć, ciśnienie krwi, poziom cholesterolu czy wyniki elektrokardiogramu. Te atrybuty zostały zebrane z pięciu odrębnych zbiorów danych, w tym obserwacje z takich źródeł jak zbiory danych z Cleveland, Węgier, Szwajcarii, Long Beach VA i Stalog. W wyniku tego analizie poddajemy zbiór danych zawierający 918 obserwacji.

Celem analizy jest wykorzystanie algorytmów uczenia maszynowego do przewidywania prawdopodobieństwa wystąpienia chorób serca na podstawie tych różnorodnych atrybutów. Cechy zbioru danych oferują bogaty wachlarz informacji, które potencjalnie mogą być wykorzystane do opracowania modeli predykcyjnych.

W niniejszym sprawozdaniu zostaną omówione kroki preprocessingu, analizy cech, rozwój modeli oraz metryki oceny wykorzystane do stworzenia solidnych modeli predykcyjnych. Ostatecznym celem jest stworzenie niezawodnego narzędzia predykcyjnego, które będzie możliwie skuteczne w identyfikacji osób narażonych na choroby układu sercowo-naczyniowego.

Analiza oraz przygotowanie danych

Na samym początku dokonaliśmy analizy kolumn - każdej z osobna, skupiając się na znalezieniu wartości odstających oraz sprawdzając liczbę klas dla zmiennych kategorycznych.

```
Analiza kolumny: Age
========================================
Liczba brakujących wartości: 0
Średnia: 53.510893246187365
Minimum: 28
Maksimum: 77
Odchylenie standardowe: 9.43261650673201


Analiza kolumny: Sex
========================================
Liczba brakujących wartości: 0
Unikalne wartości: ['M' 'F']


Analiza kolumny: ChestPainType
========================================
Liczba brakujących wartości: 0
Unikalne wartości: ['ATA' 'NAP' 'ASY' 'TA']


Analiza kolumny: RestingBP
========================================
Liczba brakujących wartości: 0
Średnia: 132.39651416122004
Minimum: 0
Maksimum: 200
Odchylenie standardowe: 18.5141541199078


Analiza kolumny: Cholesterol
========================================
Liczba brakujących wartości: 0
Średnia: 198.7995642701525
Minimum: 0
Maksimum: 603
Odchylenie standardowe: 109.38414455220348


Analiza kolumny: FastingBS
========================================
Liczba brakujących wartości: 0
Średnia: 0.23311546840958605
Minimum: 0
Maksimum: 1
Odchylenie standardowe: 0.423045624739303
```

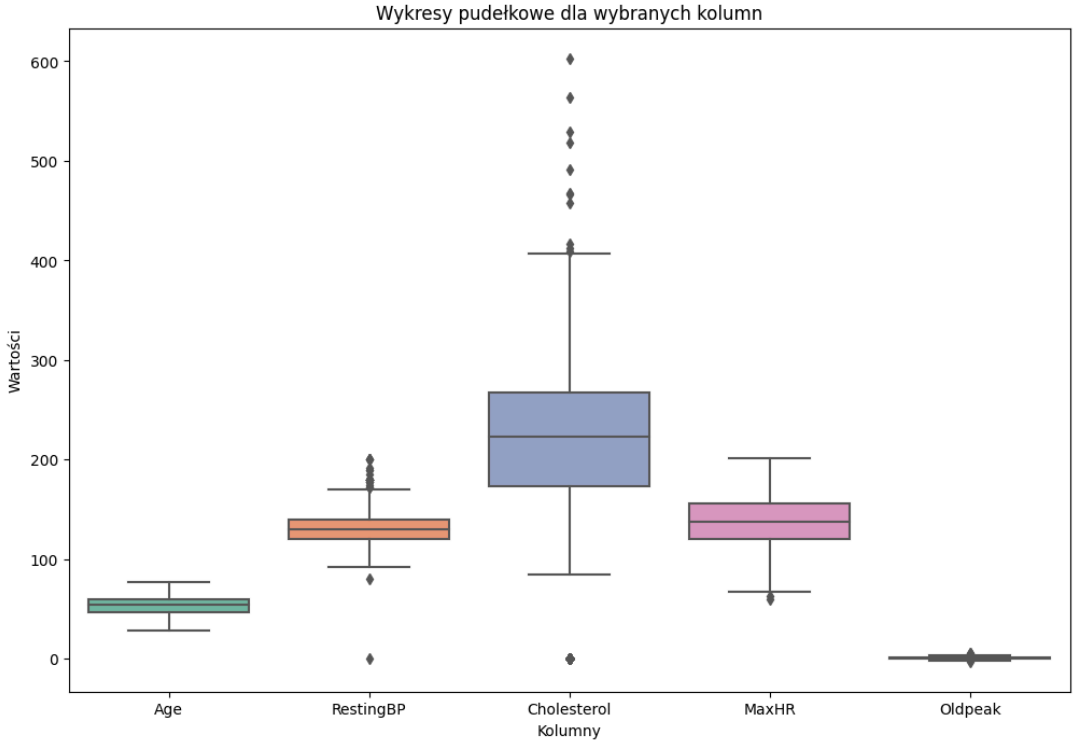
```
Analiza kolumny: FastingBS
========================================
Liczba brakujących wartości: 0
Średnia: 0.23311546840958605
Minimum: 0
Maksimum: 1
Odchylenie standardowe: 0.423045624739303


Analiza kolumny: RestingECG
========================================
Liczba brakujących wartości: 0
Unikalne wartości: ['Normal' 'ST' 'LVH']


Analiza kolumny: MaxHR
========================================
Liczba brakujących wartości: 0
Średnia: 136.80936819172112
Minimum: 60
Maksimum: 202
Odchylenie standardowe: 25.4603341382503


Analiza kolumny: ExerciseAngina
========================================
Liczba brakujących wartości: 0
Unikalne wartości: ['N' 'Y']


Analiza kolumny: Oldpeak
========================================
Liczba brakujących wartości: 0
Średnia: 0.8873638344226579
Minimum: -2.6
Maksimum: 6.2
Odchylenie standardowe: 1.0665701510493257


Analiza kolumny: ST_Slope
========================================
Liczba brakujących wartości: 0
Unikalne wartości: ['Up' 'Flat' 'Down']
```

```
Analiza kolumny: HeartDisease
========================================
Liczba brakujących wartości: 0
Średnia: 0.5533769063180828
Minimum: 0
Maksimum: 1
Odchylenie standardowe: 0.4974137382845968
```

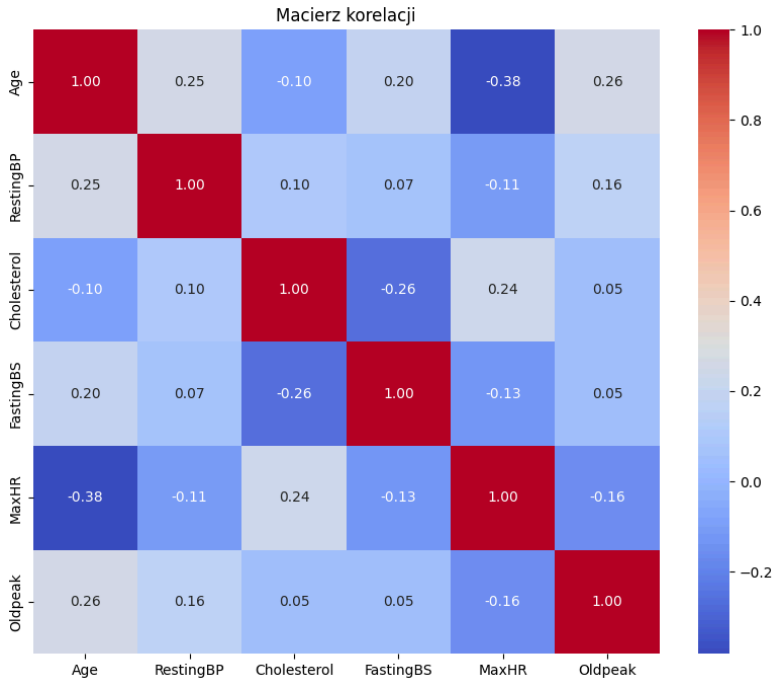
Jak możemy zauważyć, nasze zmienne nie zawierają braków w danych, a zmienne kategoryczne zawierają stosunkowo małą liczbę klas, co może sugerować ich potencjalne zbalansowanie (brak dużej liczby mało licznych kategorii), co do tego ostatniego nie mamy

jednak pewności na tym etapie analizy dlatego w późniejszym etapie przeprowadzimy analizę rozkładów zmiennych i wyciągniemy z niej wnioski.

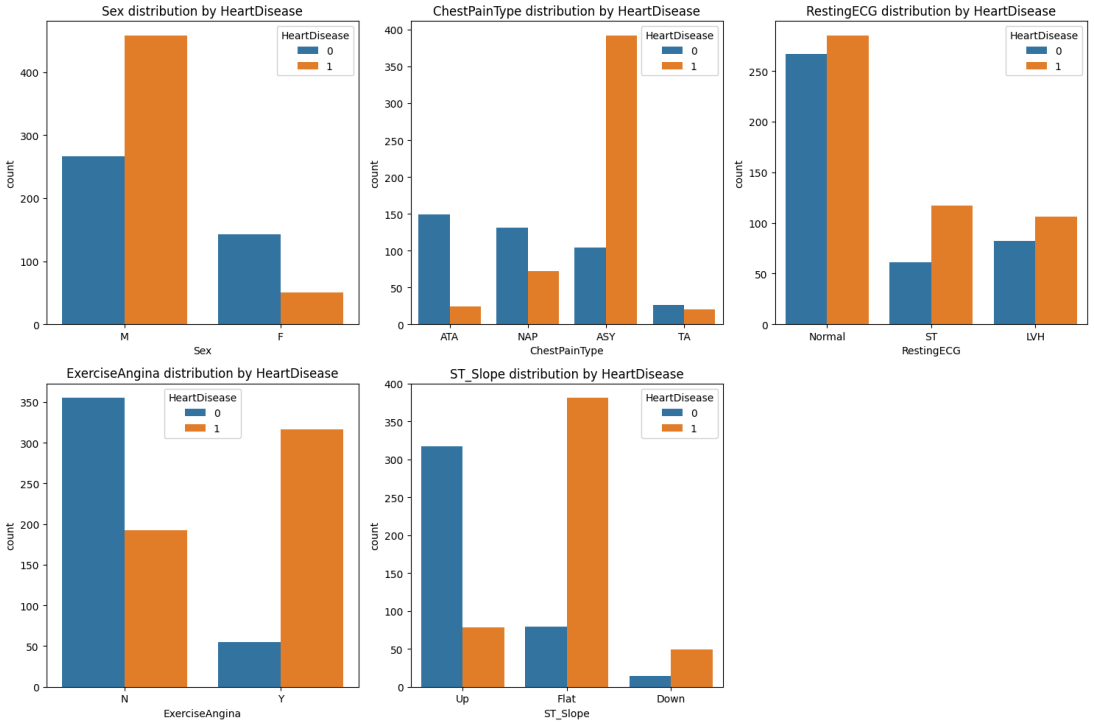
W celu sprawdzenia czy występują dane odstające zastosowaliśmy wykresy pudełkowe, jak łatwo można zauważyć, tego typu dane znajdują się w naszym zbiorze, jednak z uwagi na fakt iż są to dane medyczna, które potencjalnie mogą posiadać cenne informacje o chorych pacjentach, nie będziemy ich usuwać.



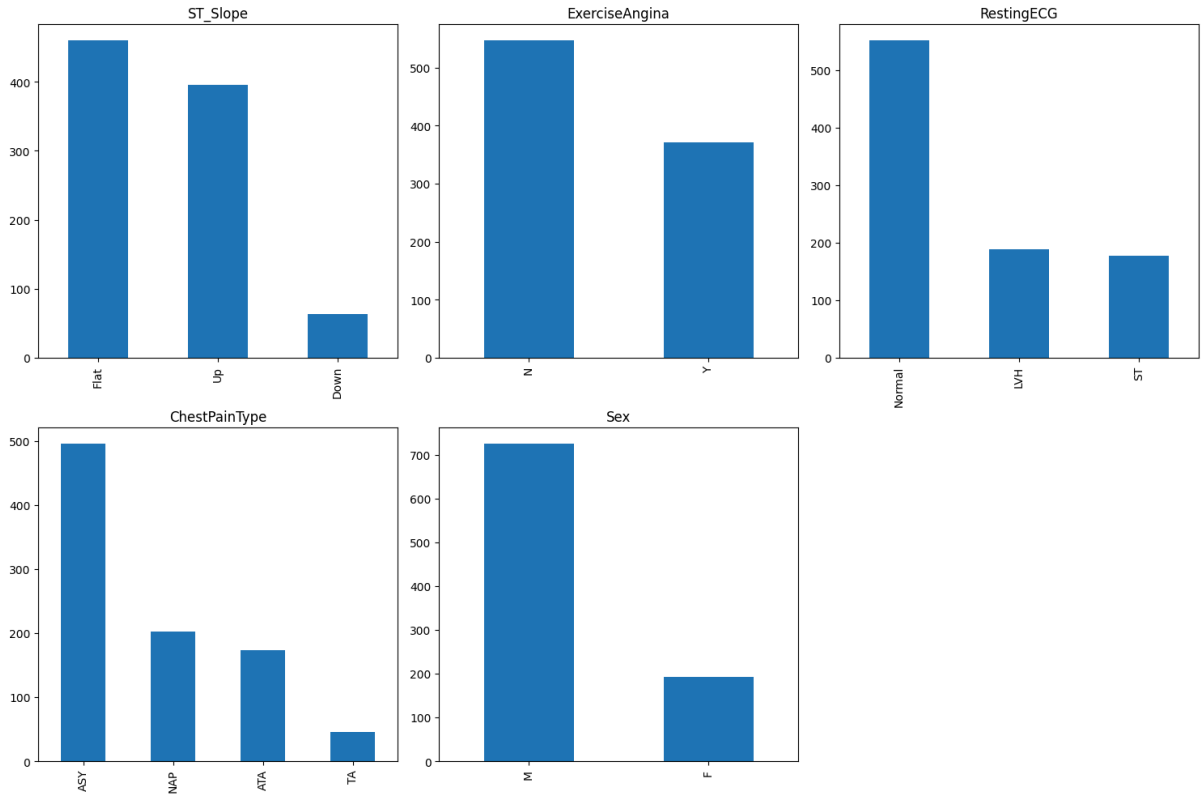
W następnym etapie skupimy się na analizie korelacji między zmiennymi oraz na badaniu rozkładów zmiennych.W tym celu przygotowaliśmy macierz korelacji pomiędzy numerycznymi zmiennymi objaśniającymi.



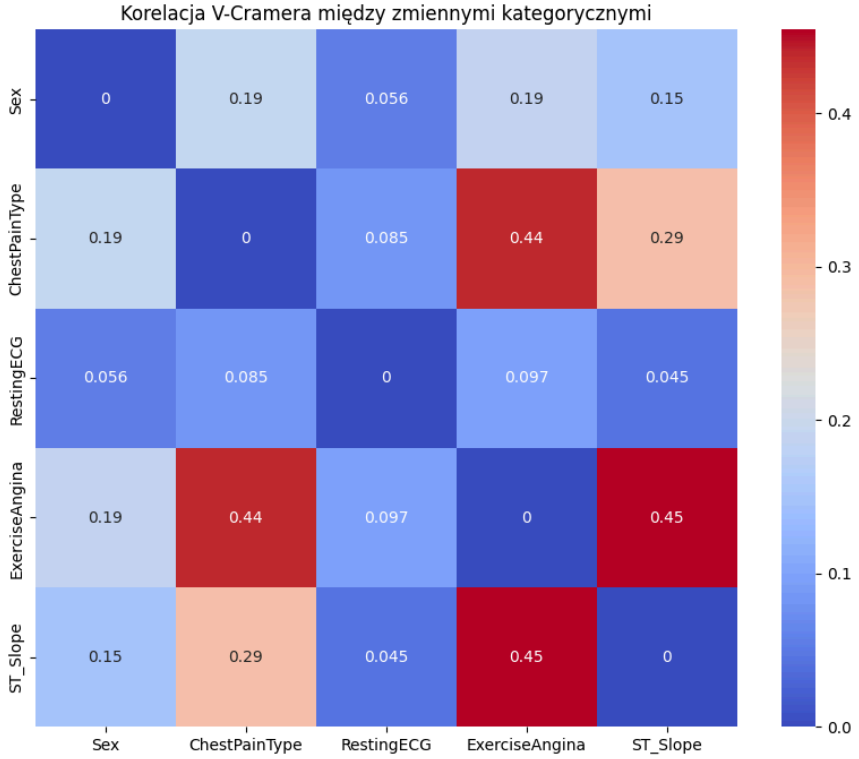
Wszystkie te zmienne cechują się słabą lub bardzo słabą korelacją między sobą, a więc na tym etapie nie odrzucamy żadnej z nich ze zbioru danych. Następnie przechodzimy do analizy rozkładów zmiennych kategorycznych.



Na podstawie analizy rozkładów zmiennych kategorycznych widzimy, że rozkłady te różnią się dla osób chorych i zdrowych, a zatem zmienne te mogą potencjalnie mieć istotny wpływ na przewidywania modelu.



Po zobrazowaniu liczności klas widzimy, że możemy przystąpić do obliczenia współczynnika V-Cramera, ponieważ klasy są wystarczająco liczne.Tym samym ponownie otrzymujemy cieplną mapę korelacji :



Po obliczeniu współczynnika V-Cramera widzimy, że kategoryczne wykazują słabą lub bardzo słabą korelację między sobą. Oznacza to, że wszystkie nasze zmienne możemy

pozostawić w zbiorze danych, a następnie przejść do one-hot encoding oraz do balansowania zbioru względem zmiennej objaśnianej.

```
[47] print(data['HeartDisease'].value_counts())

     1    508
     0    410
     Name: HeartDisease, dtype: int64
```

Jak możemy zaobserwować około 45% wszystkich zmiennych w zbiorze to zmienne reprezentujące osoby zdrowe, a pozostałe wartości reprezentują osoby zdrowe. Taki zbiór uznajemy za wystarczająco zbalansowany i nie musimy używać metod over ani under samplingu.

Podział na zbiór testowy i uczący

Odpowiedni podział zestawu danych na zbiór testowy oraz uczący może być kluczowym elementem do prawidłowej klasyfikacji. Wybierając zbiory różnica pomiędzy liczbą obserwacji dla zmiennej objaśniającej która przyjmuje wartości 0 oraz 1 powinna być jak najmniejsza. Wybrany zbiór danych zawiera 918 obserwacji pośród których 508 osobników miało chorobę serca, natomiast 410 obserwacji nie cierpiało z powodu chorób serca. W celu znalezienia optymalnego random seeda dla którego zbiór uczący i testowy będą jak najbardziej zbliżone pod względem procentowego udziału osób chorych i zdrowych postanowiono przetestować różne random seedy. Dla naszego zestawu danych najlepsze zbalansowanie obu zbiorów jest dla random seeda o wartości równej 5.

```
Najlepszy seed: 5, Najlepsze zbalansowanie (train): 0.0817438692098093
Liczba obserwacji w zbiorze uczącym (klasa 0): 337
Liczba obserwacji w zbiorze uczącym (klasa 1): 397
Liczba obserwacji w zbiorze testowym (klasa 0): 73
Liczba obserwacji w zbiorze testowym (klasa 1): 111
```

Jak widać na wyżej załączonym screenshocie dla random seed równego 5 mamy dla zbioru uczącego odpowiednio 337 obserwacji z wartością zmiennej objaśnianej 0, 397 z wartością 1. Dla zbioru testowego mamy odpowiednio 73 obserwacji z wartościami 0 oraz 111 obserwacji z wartościami 1.

Wybrane metody uczenia maszynowego

Lasy losowe

Lasy losowe, znane również jako lasy przypadkowe, to rodzaj algorytmu stosowanego w uczeniu maszynowym. Są one używane głównie do budowy modeli predykcyjnych i klasyfikacyjnych. Nazwa "las losowy" wynika z faktu, że algorytm ten opiera się na koncepcji wielu drzew decyzyjnych, gdzie każde drzewo jest budowane na podstawie losowego

podzbioru danych treningowych. Ostateczna predykcja lub klasyfikacja jest następnie uzyskiwana poprzez głosowanie lub średnią wyników uzyskanych z poszczególnych drzew, co często prowadzi do bardziej stabilnych i dokładnych wyników niż pojedyncze drzewo decyzyjne.

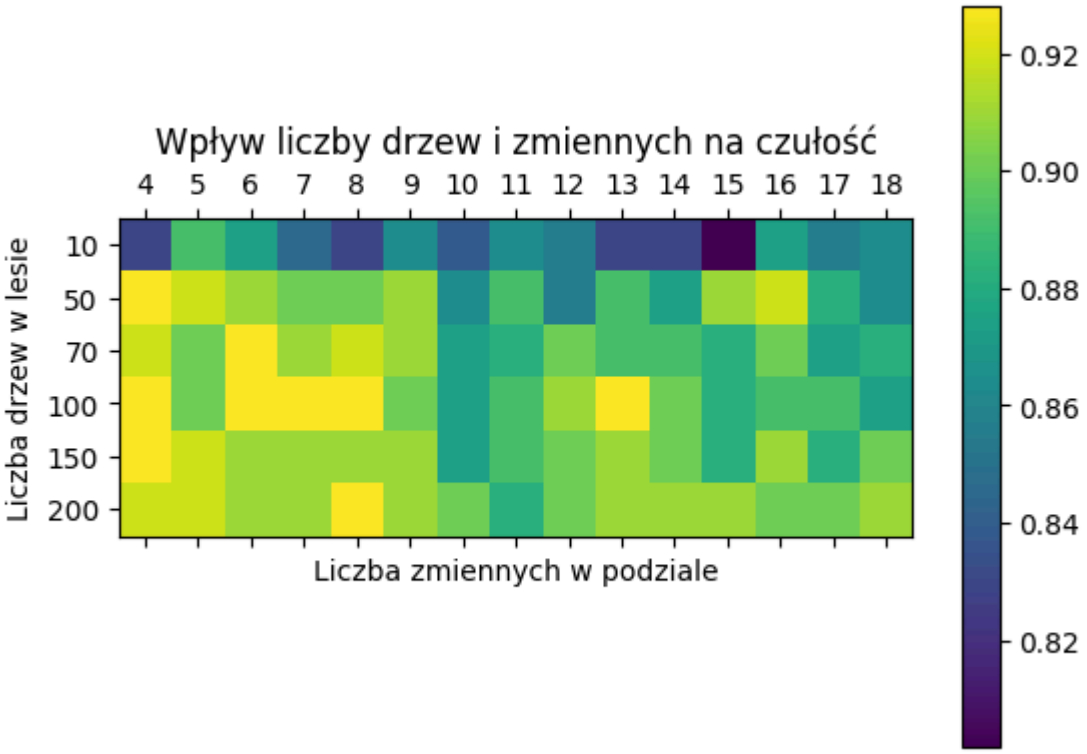
Stosując lasy losowe chcielibyśmy dobrać najlepsze parametry, którymi są liczba drzew w lesie, a także liczba zmiennych stosowanych przy podziale, w tym celu wykonujemy iteracje po tablicach z różnymi wartościami i dla każdej kombinacji tworzymy model.

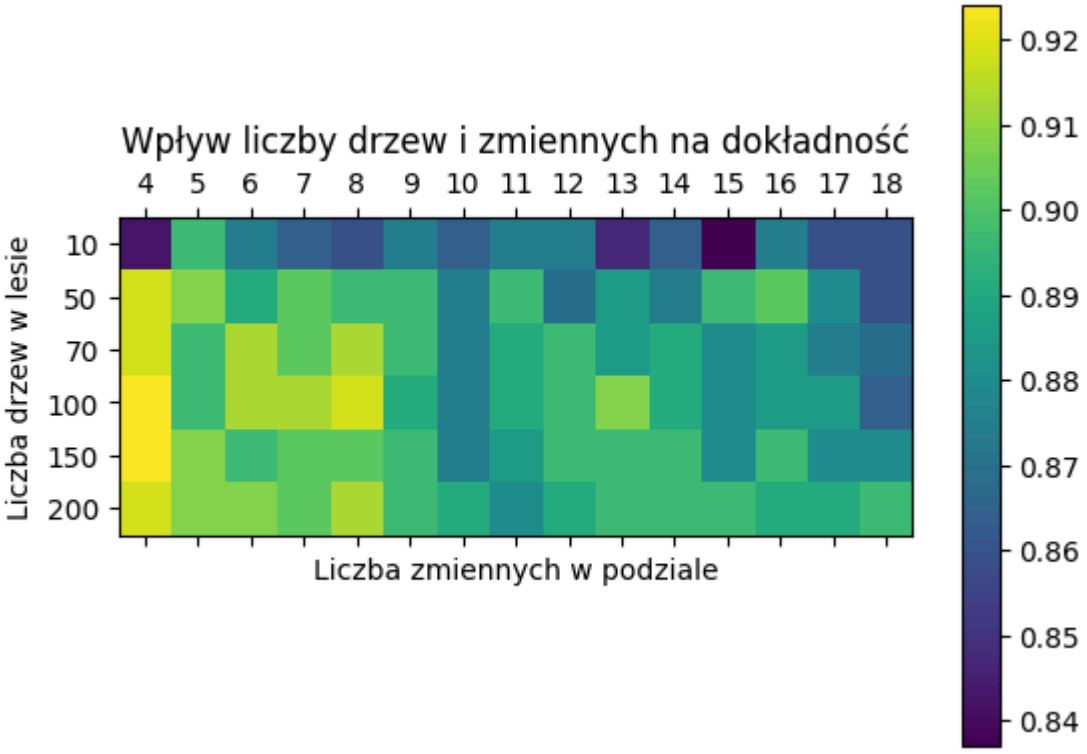
```
Najlepsze parametry dla zbioru testowego:
{'n_estimators': 100, 'max_features': 4}
Najlepsza czułość dla zbioru testowego: 0.9279279279279279
Najlepsza dokładność dla zbioru testowego: 0.9239130434782609

Najlepsze parametry dla zbioru treningowego:
{'n_estimators': 70, 'max_features': 4}
Najlepsza czułość dla zbioru treningowego: 1.0
Najlepsza dokładność dla zbioru treningowego: 1.0
```

Jak można niestety zauważyć, nasz model się przetrenował gdyż działa on idealnie w przypadku zbioru treningowego, a trochę gorzej w przypadku zbioru testowego. Warto tu jednak zaznaczyć, że wyniki dla zbioru testowego są i tak bardzo dobre, choć prawdopodobnie nie tak dobre by w realnym przypadku zastosować ten model w medycynie.

```
Najlepsze parametry dla zbioru treningowego:
{'n_estimators': 200, 'max_features': 5}
Najlepsza dokładność dla zbioru treningowego: 0.8583170254403132
Najlepsza czułość dla zbioru treningowego: 0.8818460050610193

Najlepsze parametry dla zbioru testowego:
{'n_estimators': 100, 'max_features': 4}
Najlepsza dokładność dla zbioru testowego: 0.9239130434782609
Najlepsza czułość dla zbioru testowego: 0.9279279279279279
```

Jak widzimy udało nam się zniwelować problem przeuczenia modelu. W obu przypadkach wyniki nieznacznie się pogorszyły, jednak są dalej względnie zadowalające, a różnica pomiędzy nimi spadła.





Na powyższych wykresach widzimy przedstawioną w formie graficznej czułość i dokładność tego modelu dla różnych parametrów, wykres jest zbudowany na podstawie danych ze zbioru testowego. Wyniki ze zbioru testowego są bardziej miarodajne ponieważ możemy na nich zobaczyć jak model radzi sobie z nowymi danymi. Jak można zauważyć, dokładność najlepsza jest dla 4 zmiennych i 100-150 drzew, natomiast w przypadku czułości ciężko wyróżnić jeden najlepszy obszar. Pomimo trudności w dokładnej interpretacji wyników tylko na podstawie tego wykresu, jesteśmy w stanie stwierdzić, że uzyskane wcześniej wyniki w

formie tekstu wydają się sensowne. Tym samym uznajemy liczbę zmiennych 4 i liczbę drzew 100 za najlepsze parametry dla tego modelu.

Algorytm SVM

Support Vector Machines (SVM) to algorytm uczenia maszynowego wykorzystywany do zadań klasyfikacji i regresji. Działa poprzez znalezienie hiperpłaszczyzny najlepiej separującej punkty danych różnych klas, jednocześnie maksymalizując odległość między nimi. W SVM stosuje się różne rodzaje jąder, takie jak liniowe, wielomianowe i radialne funkcje bazowe (RBF). Jądro liniowe zakłada liniową granicę decyzyjną którą można porównać do “marginesu”, natomiast jądra wielomianowe i RBF pozwalają na obsługę zależności nieliniowych poprzez odwzorowywanie danych do przestrzeni o wyższych wymiarach.

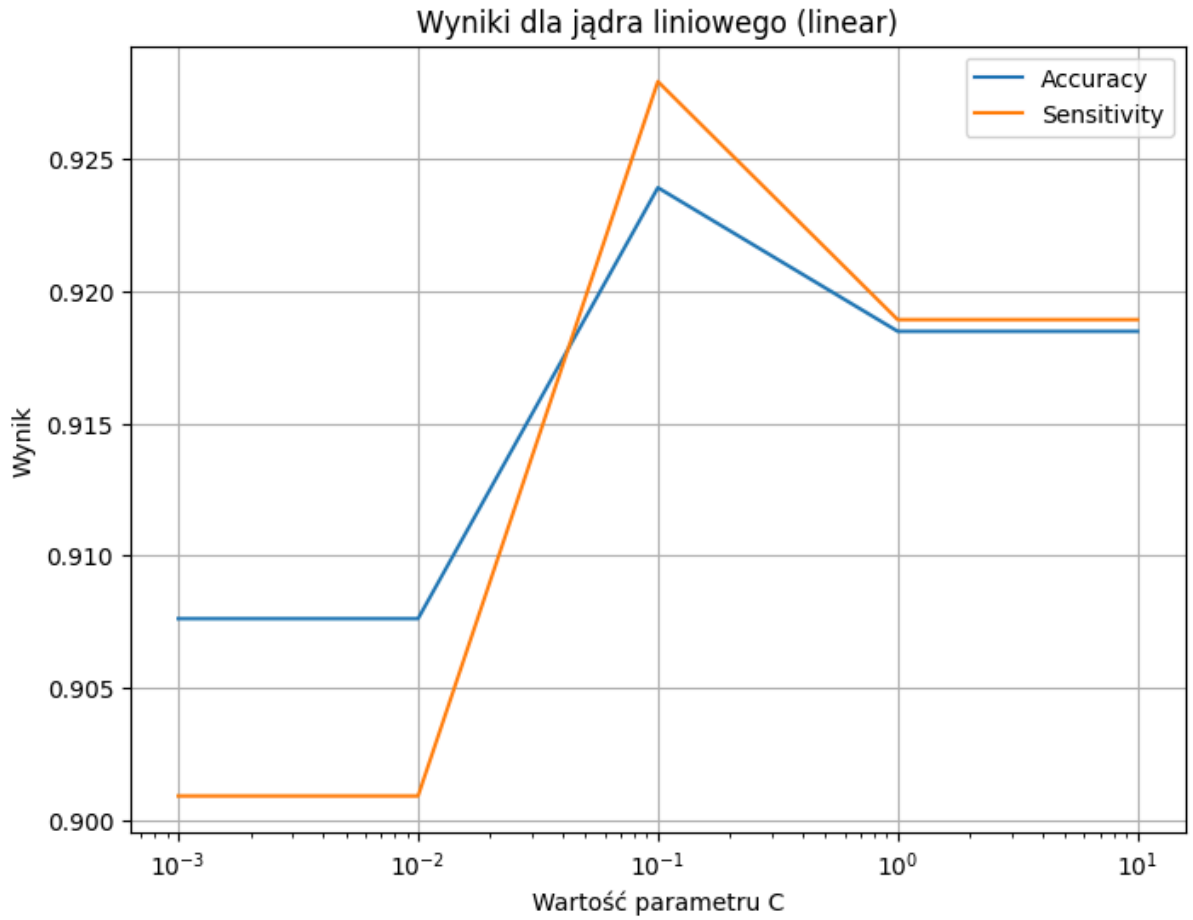
W celu pełnej implementacji i analizy metody SVM przygotowujemy modele ze wszystkimi trzema wcześniej wspomnianymi rodzajami jądra. Jak widać na poniższym wykresie zarówno czułość jak i dokładność posiadają zadowalające wartości dla wszystkich parametrów, jednak swój szczyt osiągają dla wartości 0,1, aby następnie od 0 być na stałym, wysokim i zadowalającym nas poziomie.

Dla jądra liniowego w metodzie SVM można optymalizować hiperparametr C (parametr regulacji). Kontroluje on karę za błędy klasyfikacyjne. Określa wagę, jaką model przywiązuje do minimalizacji marginesu oraz dopuszczenie błędów na zbiorze treningowym. Dla większych wartości C model będzie dążył do minimalizacji błędów na zbiorze treningowym kosztem większego dopasowania do tych danych. Z kolei mniejsze wartości C prowadzą do większego dopuszczenia błędów na zbiorze treningowym, co może prowadzić do lepszej generalizacji modelu na zbiorze testowym.

Optymalizacja tego parametru pozwala na znalezienie równowagi między dopasowaniem modelu do danych treningowych, a jego zdolnością do generalizacji na nowe dane.

W projekcie wybraliśmy przetestowanie następujących wartości parametru C:

```
# Hiperparametry do optymalizacji
param_grid_linear = {'C': [0.001, 0.01, 0.1, 1, 2 , 4 , 10 ]}
```



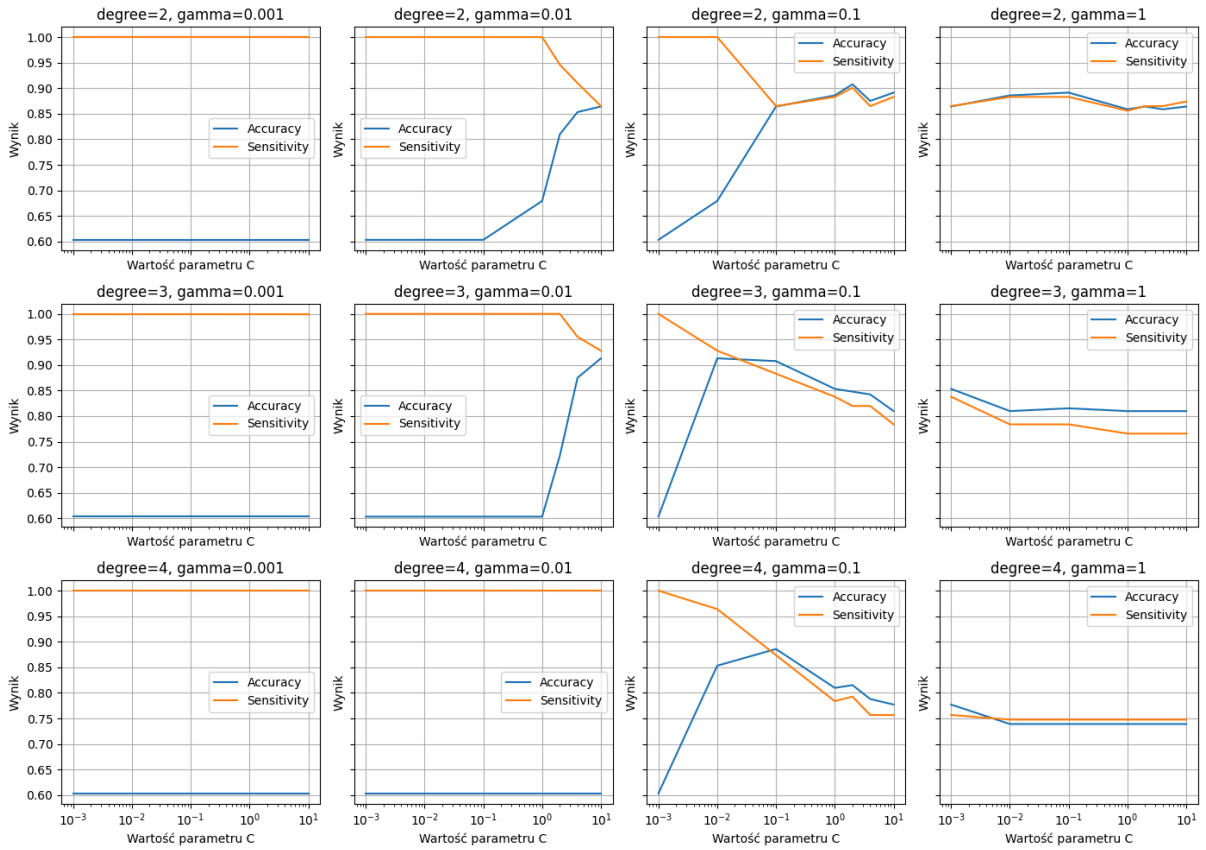
Następnie przeszliśmy do tworzenia jądra wielomianowego.
W tym przypadku optymalizacji mogą podlegać parametry takie jak : parametr C, stopień wielomianu oraz parametr gamma.

Stopień wielomianu ma wpływ na złożoność modelu, niższe stopnie wielomianu są mniej skomplikowane, co może prowadzić do modeli bardziej odpornych na nadmierne dopasowanie, jednak zbyt niski stopień wielomianu może skutkować brakiem wystarczającej złożoności, by model dobrze odwzorował rzeczywisty związek między danymi, a celem predykcyjnym. Wyższe stopnie wielomianu mogą prowadzić do bardziej złożonych modeli, które mogą być bardziej elastyczne, ale mogą również doprowadzić do przeuczenia.

Parametr gamma natomiast kontroluje wpływ pojedynczego przykładu treningowego na obszary decyzyjne. Wysokie wartości gamma mogą prowadzić do modeli o większej złożoności, co może prowadzić do zbytniego dopasowania do danych treningowych (overfitting). Niskie wartości gamma mogą natomiast prowadzić do modeli o mniejszej złożoności ale mogą mieć problemy z dobrą generalizacją na dane testowe.

W projekcie użyliśmy następujących wartości tych parametrów :

```
# Hiperparametry do optymalizacji
param_grid_poly = {'C': [0.001, 0.01, 0.1, 1, 2 , 4 , 10 ], 'degree': [2, 3, 4], 'gamma': [0.001, 0.01, 0.1, 1]}
```



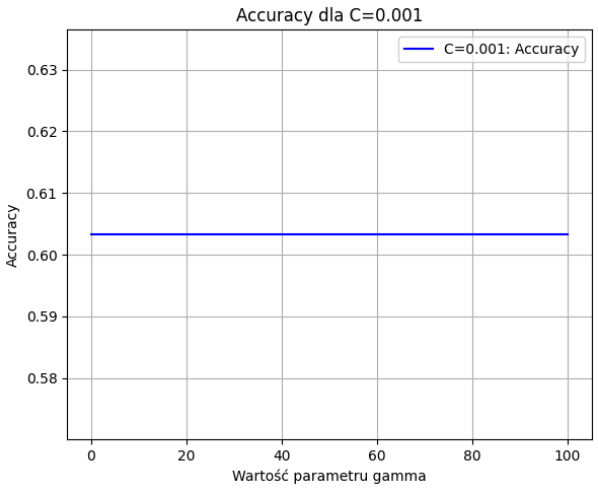
W tym przypadku mamy doczynienia z dość niejednoznacznymi wynikami. Tym razem nie zawsze przy zwiększami parametru C nasze modele się poprawiają. Widzimy również, że modele ze zbyt niskim parametrem gamma, a także zbyt wysokim stopniem wielomianu nie radzą sobie najlepiej, w niektórych przypadkach może dochodzić do trudności z identyfikowaniem jednej z klas (stopień 3,4 dla gamma 0,01 i 0,001). Z samych wykresów ciężko jest wyciągnąć jednoznaczne wnioski lecz potencjalnie najstabilniejszy i możliwe, że najlepszy wydaje się model z gamma = 1 lub 0,1 , a stopniem wielomianu równym 2.

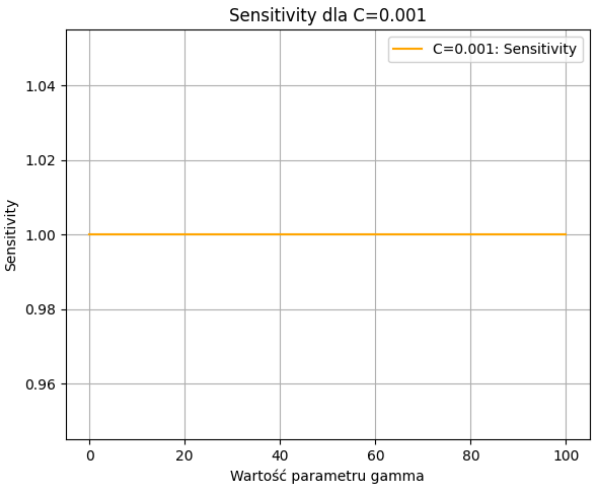
W przypadku jądra radialnego możemy analizować parametr gamma oraz parametr C, które mają takie same interpretacje jak w przypadku wcześniejszych podpunktów.

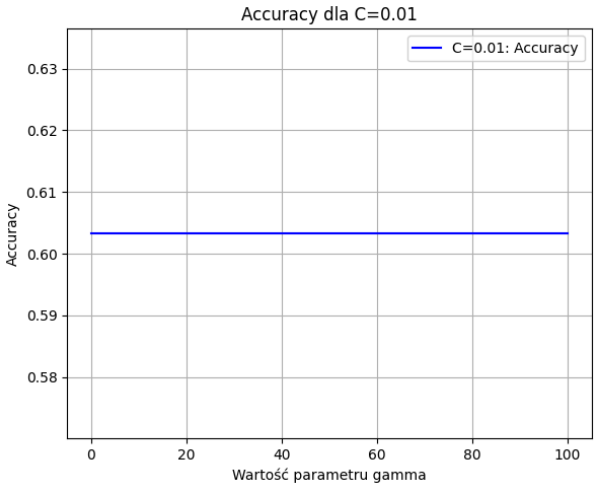
parametry jakie użyliśmy do analizy to :

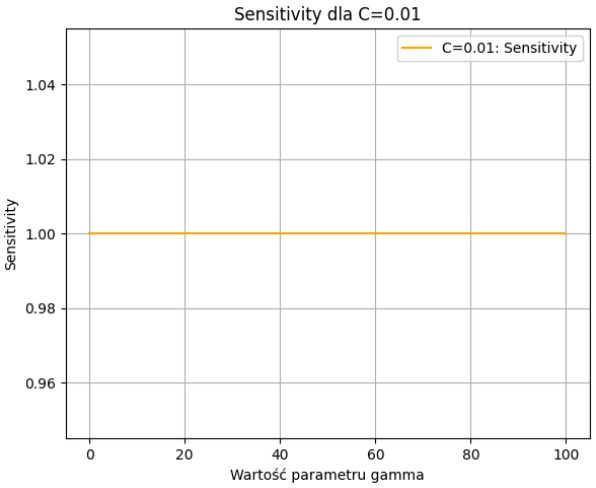
```
# Hiperparametry do optymalizacji
param_grid_rbf = {'C': [0.001, 0.01, 0.1, 1, 2 , 4 , 10 ], 'gamma': [0.001, 0.01, 0.1, 1, 10, 100]}
```

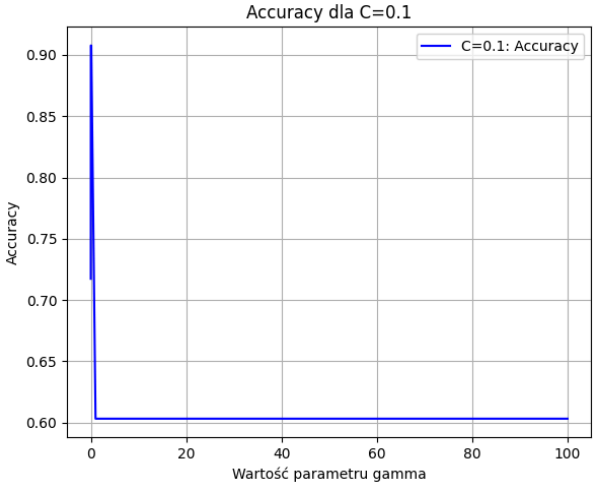
Po wytrenowaniu modeli i naniesieniu wyników na wykresy otrzymaliśmy następujące rezultaty:

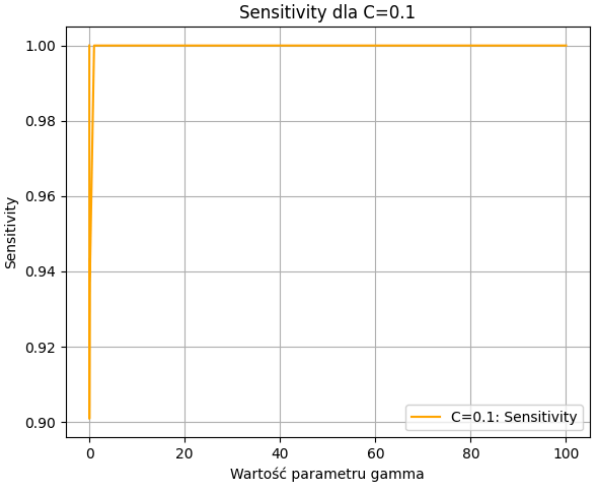

Accuracy dla C=0.001
C=0.001: Accuracy
Accuracy
Wartość parametru gamma

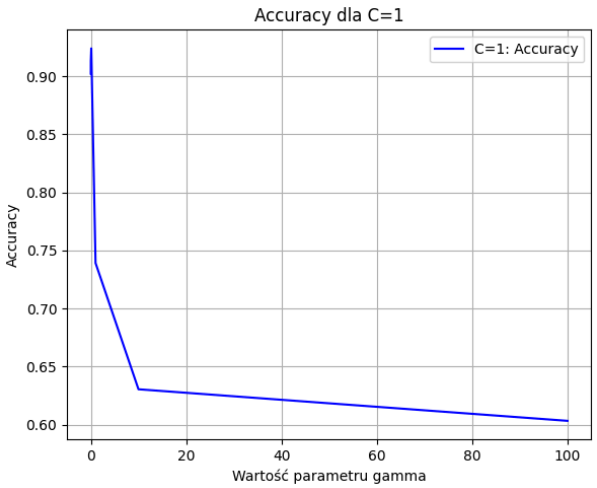

Sensitivity dla C=0.001
C=0.001: Sensitivity
Sensitivity
Wartość parametru gamma

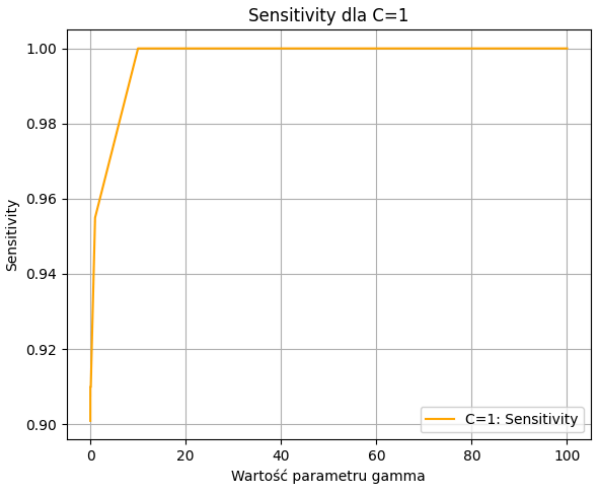

Accuracy dla C=0.01
C=0.01: Accuracy
Accuracy
Wartość parametru gamma

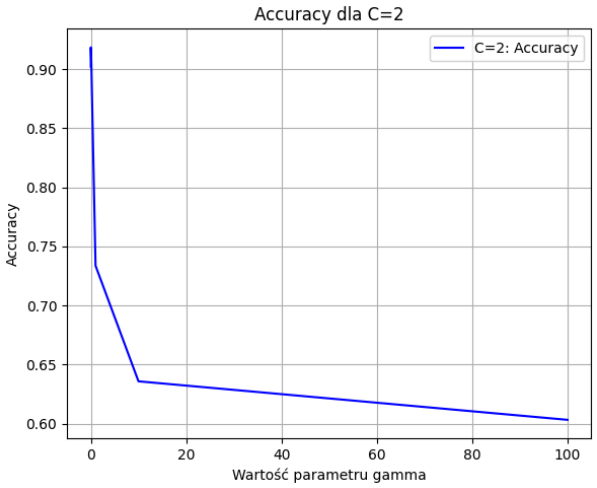

Sensitivity dla C=0.01
C=0.01: Sensitivity
Sensitivity
Wartość parametru gamma

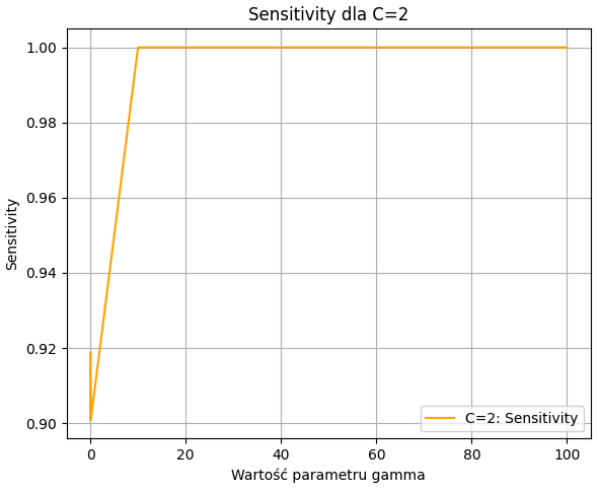

Accuracy dla C=0.1
C=0.1: Accuracy
Accuracy
Wartość parametru gamma

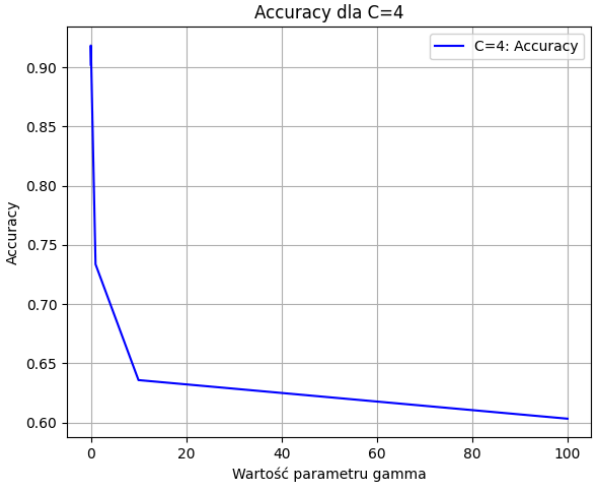

Sensitivity dla C=0.1
C=0.1: Sensitivity
Sensitivity
Wartość parametru gamma

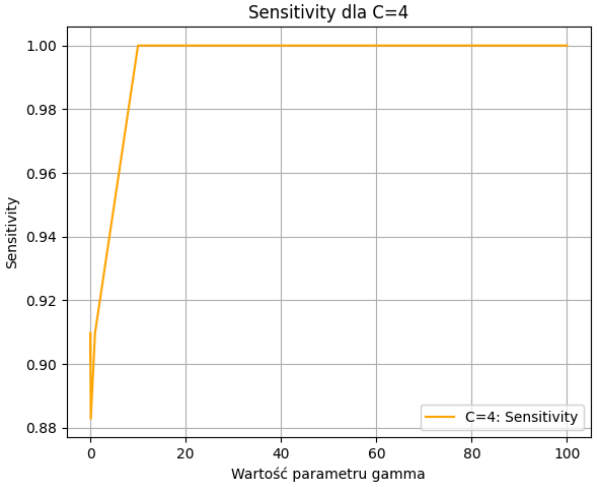

Accuracy dla C=1
C=1: Accuracy
Accuracy
Wartość parametru gamma

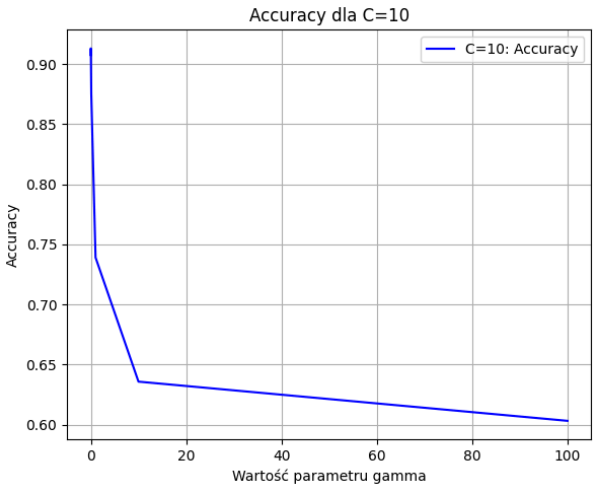

Sensitivity dla C=1
C=1: Sensitivity
Sensitivity
Wartość parametru gamma

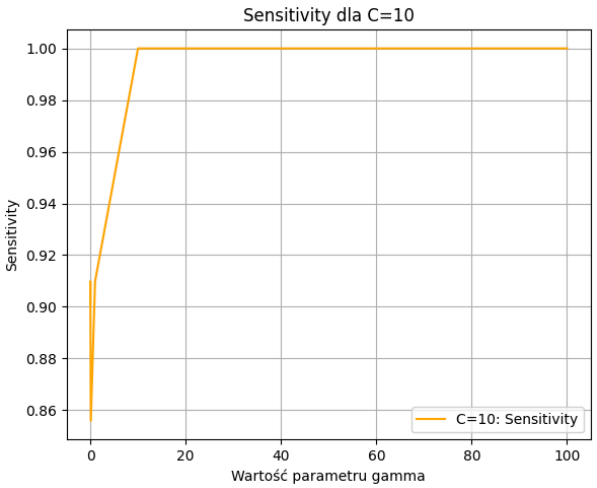

Accuracy dla C=2
C=2: Accuracy
Accuracy
Wartość parametru gamma


Sensitivity dla C=2
C=2: Sensitivity
Sensitivity
Wartość parametru gamma


Accuracy dla C=4
C=4: Accuracy
Accuracy
Wartość parametru gamma


Sensitivity dla C=4
C=4: Sensitivity
Sensitivity
Wartość parametru gamma





W przypadku tych wykresów niestety jeszcze trudniej o rzetelną ocenę wyników, z tego powodu dokonaliśmy bezpośredniego porównania statystyk. Wyniki prezentują się następująco:

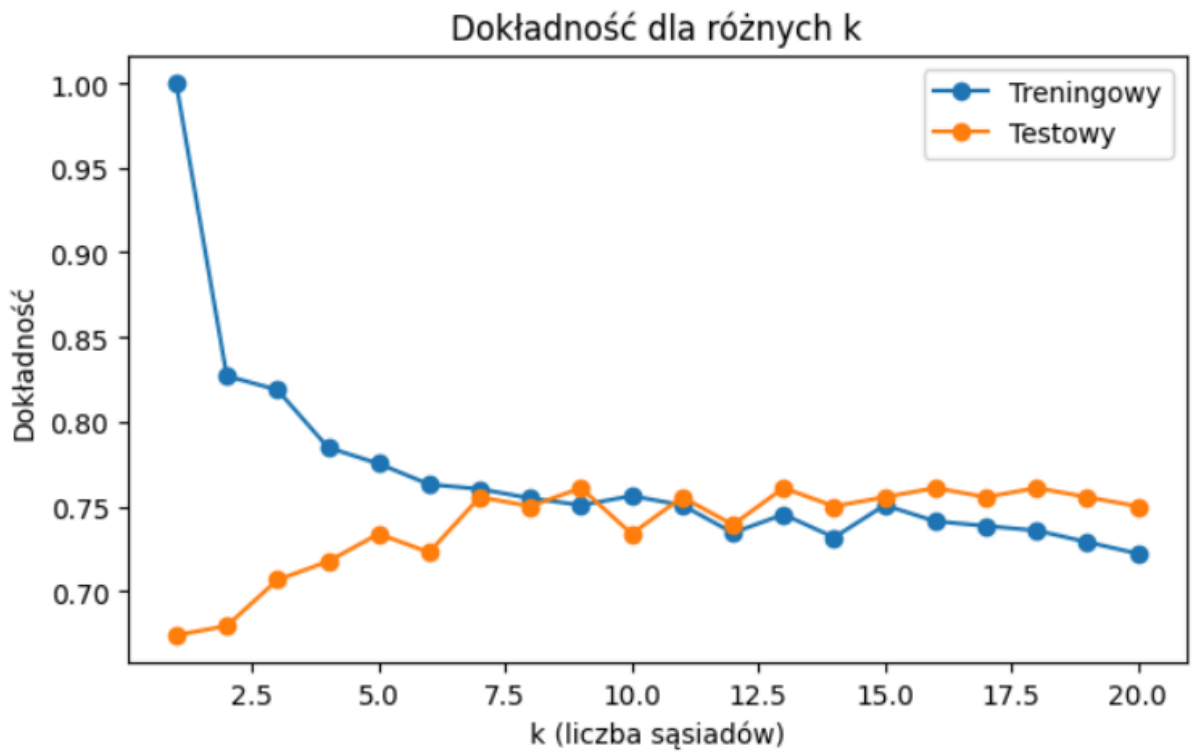
```
Najlepsze wyniki dla różnych jąder:
Jądro: Linear
 - Parametry: {'C': 0.1, 'Accuracy': 0.9239130434782609, 'Sensitivity': 0.9279279279279279}
Jądro: Poly
 - Parametry: {'C': 0.01, 'Degree': 3, 'Gamma': 0.1, 'Accuracy': 0.9130434782608695, 'Sensitivity': 1.0}
Jądro: RBF
 - Parametry: {'C': 1, 'Gamma': 0.1, 'Accuracy': 0.9239130434782609, 'Sensitivity': 1.0}
```

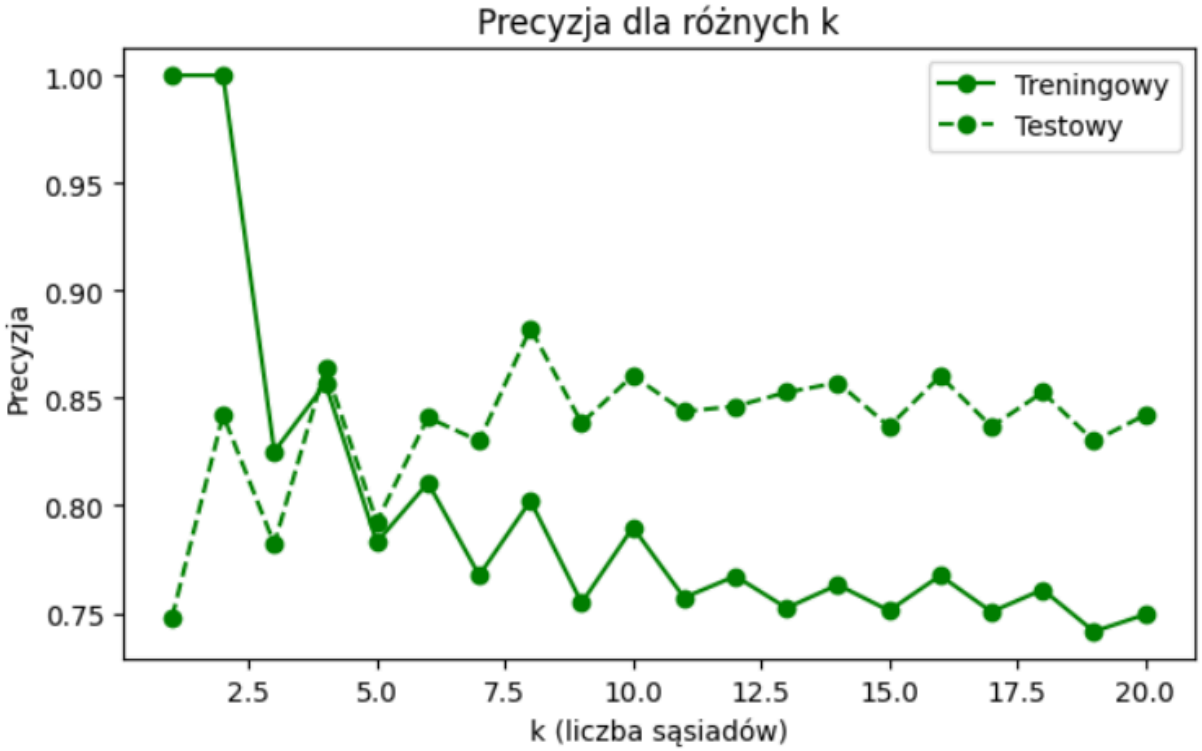
Wszystkie modele uzyskały wysokie i względnie podobne wyniki, jednak z uwagi na fakt, że są to dane medyczne, najważniejszą miarą jest sensitivity. Napotkaliśmy tutaj prawdopodobnie jednak problem przeuczenia i błędnej klasyfikacji zmiennych gdyż w dwóch przypadkach wynik ten jest równy 1.0 . Z tego powodu za najlepszy z modeli uznajemy wskazany model z jądrem liniowym.

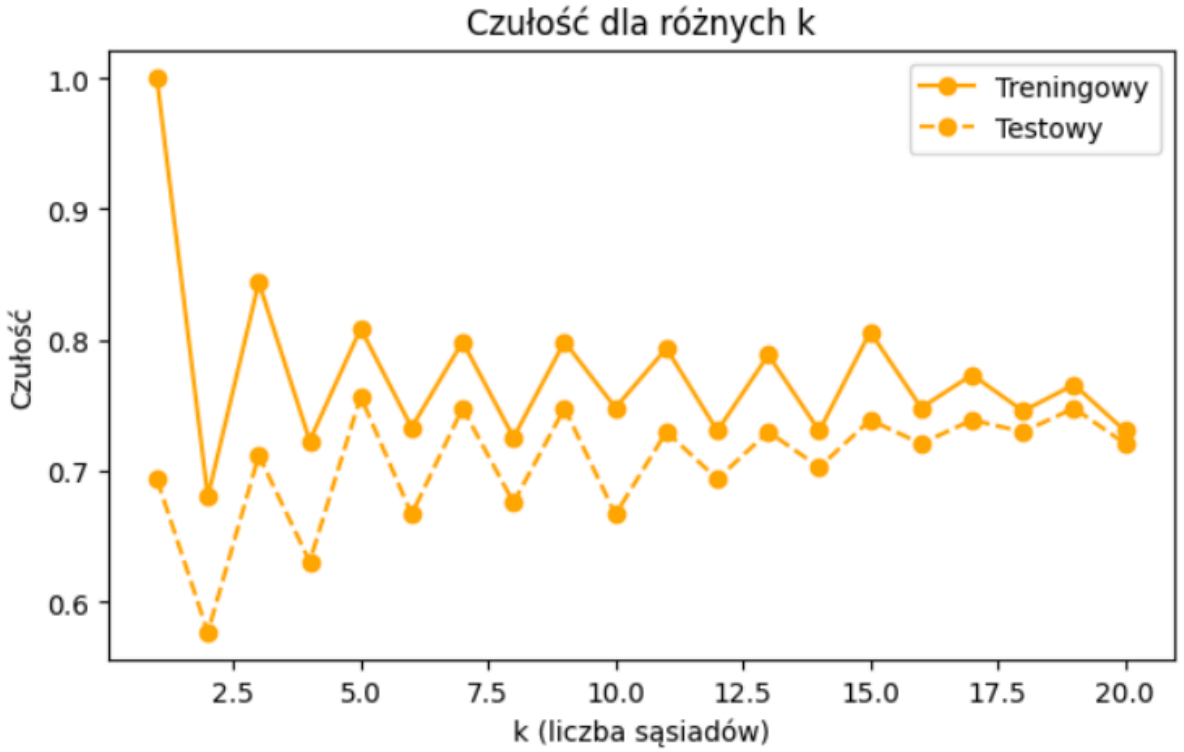
K-Nearest Neighbors (KNN)

KNN to algorytm uczenia maszynowego, który działa na zasadzie klasyfikacji przez większość. W skrócie, dla danego punktu danych, algorytm identyfikuje k najbliższych sąsiadów w przestrzeni cech na podstawie pewnej miary odległości, najczęściej euklidesowej. Klasa danego punktu jest przewidywana na podstawie etykiet k sąsiadów, przy czym często stosuje się głosowanie większościowe. Algorytm KNN jest prosty w implementacji i stosunkowo elastyczny, ale jego wydajność może zależeć od właściwego doboru parametru k oraz właściwego przeskalowania zmiennych.

Poniższy wykres przedstawiają wpływ liczby sąsiadów na zbiorze testowym oraz treningowym dla dokładności, czułości oraz precyzyjności.


Dokładność dla różnych k
Treningowy
Testowy
Dokładność
1.00
0.95
0.90
0.85
0.80
0.75
0.70
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
k (liczba sąsiadów)


Precyzja dla różnych k
Treningowy
Testowy
Precyzja
1.00
0.95
0.90
0.85
0.80
0.75
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
k (liczba sąsiadów)



Wybór najlepszego K jest zależny od zbioru danych z jakim mamy do czynienia. W przypadku wybranego medycznego zestawu danych największe znaczenie ma czułość. Zarówno zbiór testowy jak i treningowy powinny mieć podobne współczynniki dokładności, czułości oraz precyzji aby nie doprowadzić do sytuacji w której mamy problem z nadmiernym dopasowaniem jednego z zbiorów.

k	AccTrain	AccTest	PrecTrain	PrecTest	RecallTrain	RecallTest
1	1.000000	0.673913	1.000000	0.747573	1.000000	0.693694
2	0.826975	0.679348	1.000000	0.842105	0.680101	0.576577
3	0.818801	0.706522	0.825123	0.782178	0.843829	0.711712
4	0.784741	0.717391	0.856716	0.864198	0.722922	0.630631
5	0.775204	0.733696	0.782927	0.792453	0.808564	0.756757
6	0.762943	0.722826	0.810585	0.840909	0.732997	0.666667
7	0.760218	0.755435	0.767554	0.830000	0.798489	0.747748
8	0.754768	0.750000	0.802228	0.882353	0.725441	0.675676
9	0.750681	0.760870	0.754762	0.838384	0.798489	0.747748
10	0.756131	0.733696	0.789894	0.860465	0.748111	0.666667
11	0.750681	0.755435	0.757212	0.843750	0.793451	0.729730
12	0.734332	0.739130	0.767196	0.846154	0.730479	0.693694
13	0.745232	0.760870	0.752404	0.852632	0.788413	0.729730
14	0.731608	0.750000	0.763158	0.857143	0.730479	0.702703
15	0.750681	0.755435	0.751174	0.836735	0.806045	0.738739
16	0.741144	0.760870	0.767442	0.860215	0.748111	0.720721
17	0.738420	0.755435	0.750611	0.836735	0.773300	0.738739
18	0.735695	0.760870	0.760925	0.852632	0.745592	0.729730
19	0.728883	0.755435	0.741463	0.830000	0.765743	0.747748
20	0.722071	0.750000	0.749354	0.842105	0.730479	0.720721

Powyższa tabela przedstawia zebrane wyniki dla różnych wartości parametru K dla zbioru testowego oraz treningowego. Zdecydowano się dobrać parametr k równy 7 ze względu na stosunkowo podobną i wysoką czułość dokładność precyzyjność zarówno na zbiorze testowym i uczącym. Dla wartości k równego 7, analiza klasyfikatora k-najbliższych sąsiadów (KNN) przedstawia następujące wyniki. Dokładność modelu na zbiorze treningowym wynosi 76.02%, podczas gdy na zbiorze testowym jest nieco niższa i wynosi 75.54%. Precyzja na zbiorze treningowym wynosi 76.76%, a na zbiorze testowym 75.74%. Czułość (recall) na zbiorze treningowym jest wyższa, osiągając 79.85%, natomiast na zbiorze testowym wynosi 74.77%. Warto zauważyć, że model wykazuje dobre wyniki, a bliskość wyników między zbiorem treningowym a testowym sugeruje brak nadmiernej dopasowania. Niemniej jednak istnieje pewne niezrównoważenie, zauważalne zwłaszcza w kontekście nieco niższej czułości na zbiorze testowym w porównaniu do precyzji.

Dla wybranego parametru k=7 przeprowadzono standaryzację danych w celu sprawdzenia pozytywnego wpływu na model.

```
Dokładność modelu ze standaryzacją (Test): 89.67%
Czułość modelu ze standaryzacją (Test): 88.29%
Dokładność modelu bez standaryzacji (Test): 75.54%
Czułość modelu bez standaryzacji (Test): 74.77%

Dokładność modelu ze standaryzacją (Trening): 86.92%
Czułość modelu ze standaryzacją (Trening): 90.43%
Dokładność modelu bez standaryzacji (Trening): 76.02%
Czułość modelu bez standaryzacji (Trening): 79.85%
```

Wyniki analizy modelu KNN sugerują, że standaryzacja danych znacząco poprawiła jego skuteczność. Na zbiorze testowym, dokładność modelu wzrosła z 75.54% do 89.67%, a czułość z 74.77% do 88.29% po zastosowaniu standaryzacji. Podobne korzyści z standaryzacji były obserwowane na zbiorze treningowym, choć różnice w dokładności były mniejsze. Standaryzacja danych pozwoliła modelowi lepiej radzić sobie z niestandardowymi skalami cech, co przełożyło się na poprawę jego zdolności do identyfikacji pozytywnych przypadków.

Regresja logistyczna

Regresja logistyczna to technika statystyczna używana do modelowania relacji między zmiennymi niezależnymi, a binarną zmienną zależną, która przyjmuje jedną z dwóch możliwych kategorii. Jest stosowana głównie w zadaniach klasyfikacji binarnej, gdzie celem jest przewidzenie, czy obserwacja należy do jednej z dwóch klas. Regresja logistyczna wykorzystuje funkcję logistyczną do transformacji liniowej kombinacji zmiennych niezależnych, co umożliwia oszacowanie prawdopodobieństwa przynależności danej obserwacji do danej klasy. Ostateczna decyzja klasyfikacyjna jest podejmowana na podstawie ustalonego progu prawdopodobieństwa.

Dla prostego modelu regresji logistycznej sprawdzono dokładność oraz czułość na zbiorach testowych oraz uczących.

```
Dokładność modelu ze standaryzacją (Test): 92.39%
Czułość modelu ze standaryzacją (Test): 92.79%
Dokładność modelu bez standaryzacji (Test): 92.39%
Czułość modelu bez standaryzacji (Test): 92.79%

Dokładność modelu ze standaryzacją (Trening): 85.29%
Czułość modelu ze standaryzacją (Trening): 87.91%
Dokładność modelu bez standaryzacji (Trening): 85.83%
Czułość modelu bez standaryzacji (Trening): 88.92%
```

Model regresji logistycznej osiągnął wysoką skuteczność na zbiorze testowym, gdzie dokładność wynosi 92.39%. Czułość tego modelu, czyli jego zdolność do identyfikowania pozytywnych przypadków, również była wysoka i wyniosła 92.79%. Interesujący jest fakt, że standaryzacja danych nie miała znaczącego wpływu na wyniki klasyfikacji, ponieważ modele ze standaryzacją i bez standaryzacji osiągnęły podobne wyniki. W przypadku zbioru testowego model uzyskał takie same wyniki dla danych ze standaryzacją oraz dla danych bez standaryzacji.
Na zbiorze treningowym modele również uzyskały dobre wyniki. Model ze standaryzacją osiągnął dokładność na poziomie 85.29%, a jego czułość wyniosła 87.91%. W przypadku modelu bez standaryzacji, dokładność wyniosła 85.83%, a czułość była na poziomie 88.92%. Oba modele wykazały się skutecznością w identyfikowaniu pozytywnych przypadków na zbiorze treningowym. Warto zauważyć, że różnice między modelami ze standaryzacją a bez standaryzacji nie są znaczące mimo to dla zbioru uczącego lepsze wyniki są dla zbioru danych bez standaryzacji.

Parametr C w regresji logistycznej kontroluje siłę regularyzacji.
Regularyzacja w regresji logistycznej pomaga zapobiegać przeuczeniu (overfitting) poprzez kary nałożone na współczynniki regresji. Parametr C jest odwrotnością siły regularyzacji: im mniejsza wartość C, tym silniejsza regularyzacja, a im większa wartość C, tym słabsza regularyzacja.
Niska wartość C (np. 0.001) oznacza silną regularyzację, co może prowadzić do prostszych modeli, bardziej odporne na przeuczenie, ale jednocześnie mniej elastycznych.
Wysoka wartość C (np. 1000) oznacza słabą regularyzację, co pozwala modelowi na dopasowanie się do danych treningowych bardziej dokładnie, ale zwiększa ryzyko przeuczenia. Sprawdzono dla jakiej wartości parametru C mamy najlepsze wyniki modelu.

```
# Definicja siatki parametrów do przetestowania
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
```

Dla zestawu danych ze standaryzacja:

```
Najlepsze parametry: {'C': 0.01}
Dokładność modelu ze standaryzacją (Test): 90.76%
Czułość modelu ze standaryzacją (Test): 90.09%

Dokładność modelu ze standaryzacją (Trening): 85.97%
Czułość modelu ze standaryzacją (Trening): 89.17%
```

Dla zestawu danych bez standaryzacji :

```
Najlepsze parametry: {'C': 0.1}
Dokładność modelu (Test): 90.76%
Czułość modelu (Test): 90.09%

Dokładność modelu (Trening): 85.56%
Czułość modelu (Trening): 88.16%
```

Dla danych ze standaryzacją na zbiorze testowym model osiągnął dokładność na poziomie 90.76%, co mierzy skuteczność ogólną w poprawnym przewidywaniu klas. Czułość modelu na zbiorze testowym wyniosła 90.09%, co oznacza, że model skutecznie wykrył 90.09% rzeczywistych pozytywnych przypadków.
Na zbiorze treningowym model uzyskał dokładność wynoszącą 85.97%, co świadczy o jego skuteczności na danych, na których był trenowany. Czułość modelu na zbiorze treningowym wyniosła 89.17%. .Model regresji logistycznej dla zbioru bez standaryzacji osiągnął dokładność na poziomie 90.76% na zbiorze testowym, co oznacza poprawne zaklasyfikowanie 90.76% obserwacji spośród wszystkich. Jednocześnie czułość modelu na poziomie 90.09% wskazuje na zdolność do identyfikowania pozytywnych przypadków. Na zbiorze treningowym model osiągnął dokładność 85.56%, a czułość wyniosła 88.16%

Co ciekawe najlepsze parametry pojawiły się dla modelu regresji logistycznej z domyślnymi parametrami, bez standaryzacji i wyglądają one następująco

Testowy		Uczący	
Dokładność	Czułość	Dokładność	Czułość
92,39%	92,79%	87,91%	88,92%

Porównanie wyników uzyskanych za pomocą różnych metod

W celu porównania najlepszych wyników stworzono tabelę w której umieszczono najlepsze wyniki dla wybranych metod z wskazanymi dobranymi parametrami.

		Testowy	
	Dobrane parametry	Dokładność	Czułość
Regresja logistyczna	domyślne, bez standaryzacji	92,39%	92,79%
KNN	k=7, standaryzacja danych	89,67%	88,29%
Lasy losowe	liczba zmiennych =4, liczba drzew=100	92,39%	92,79%
SVM	jądro liniowe, 'c'=0.1	92,39%	92,79%

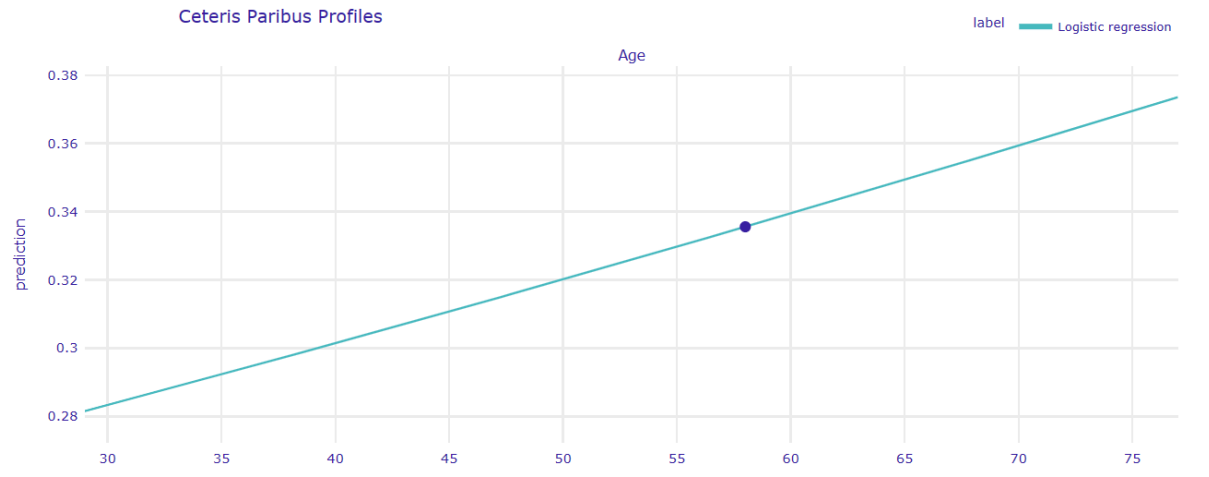
Spośród wybranych metod najgorsze wyniki dał model KNN. Najlepsze wyniki dla dokładności oraz czułości są takie same dla 3 metod(regresja logistyczna, lasy losowe, SVM) co może oznaczać że dla wybranych metod zostały dobrane odpowiednie hiperparametry. Wyniki są zadowalające.

Analiza interpretowalności

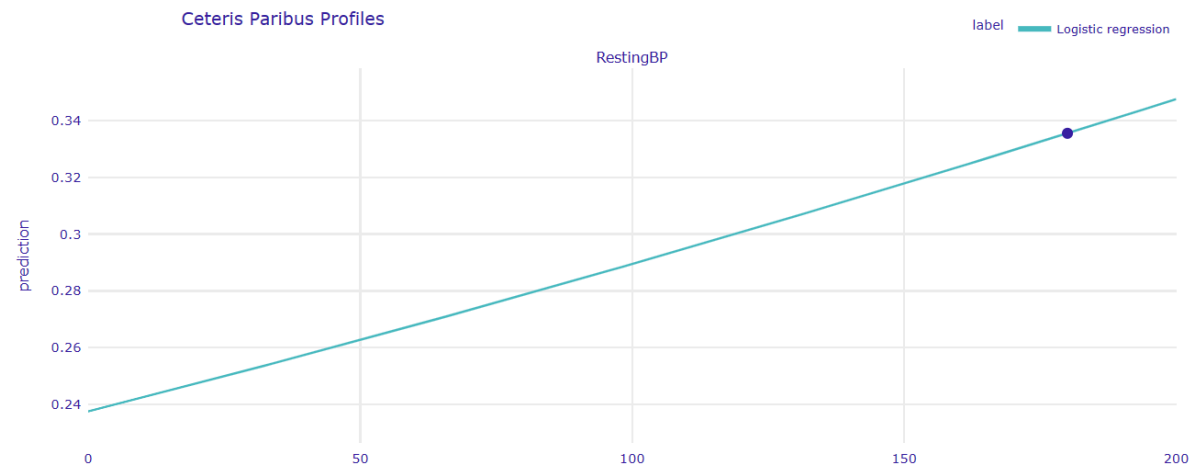
Postanowiono przebadać interpretowalność metody Regresji logistycznej dla której wyniki były najbardziej zadowalające.

Profile ceteris paribus (PCP)

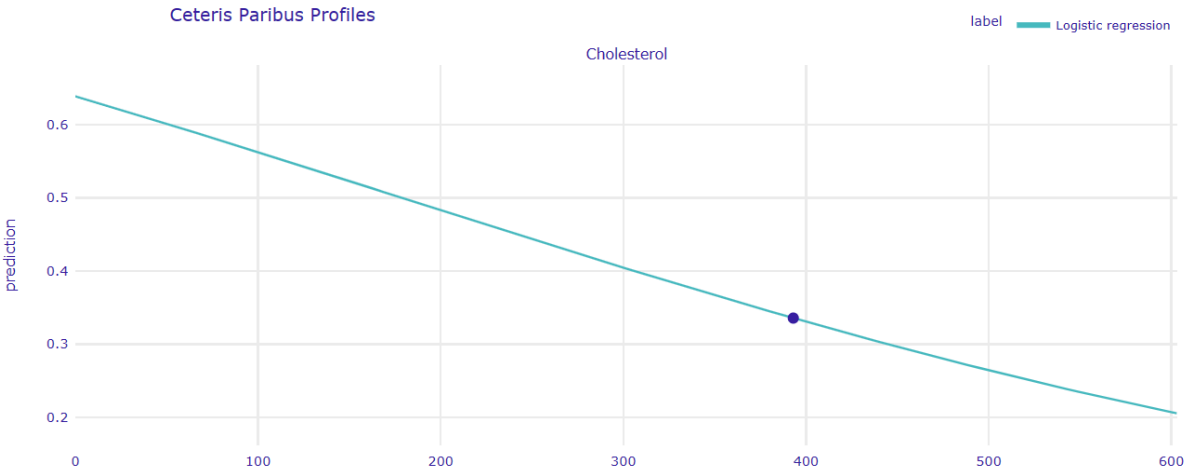
Profile ceteris paribus to technika, która pokazuje, jak zmiana jednej zmiennej wpływa na wynik modelu, przy jednoczesnym utrzymaniu pozostałych zmiennych na stałym poziomie. Ze względu na ograniczone możliwość pythona tworzenia wykresów PCP zdecydowano się na stworzenie wykresów jedynie dla zmiennych ilościowych.



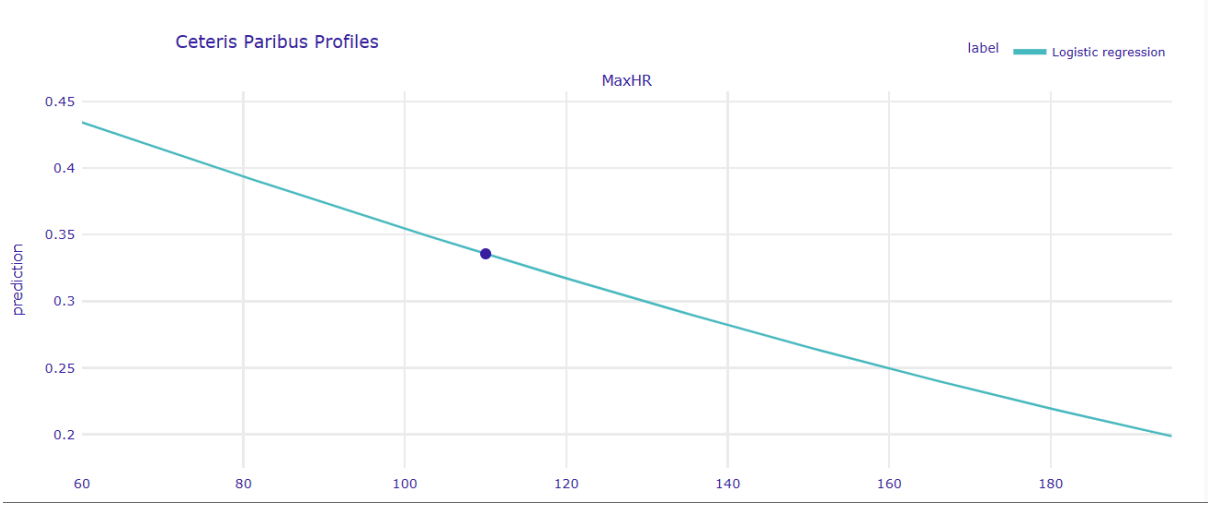
Powyższy wykres przedstawia, jak zmienia się przewidywane prawdopodobieństwo choroby serca w miarę zmiany wieku, przy zachowaniu innych zmiennych na stałym poziomie. Zwiększenie wartości zmiennej age prowadzi do wzrostu predykcji choroby serca. Kropka na wykresie oznacza obserwacje na podstawie której był tworzony wykres PCP.



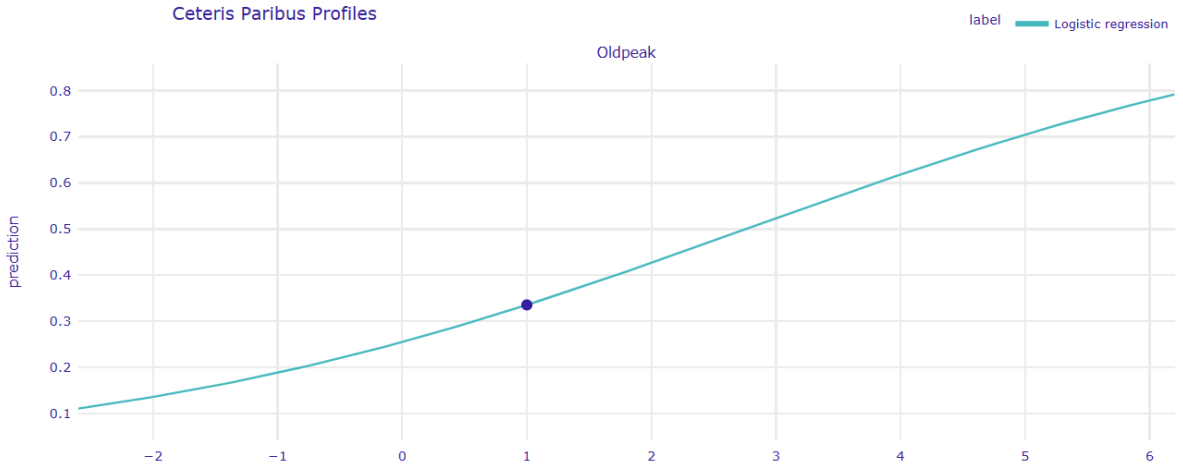
Podobnie jak w poprzednim wykresie wraz ze wzrostem wartości RestingBP można zaobserwować wzrost prawdopodobieństwa choroby serca. Obserwacja na podstawie której został utworzony wykres to mężczyzna w wieku 58 lat.



Na podstawie powyższego wykresu można wywnioskować że jeżeli Cholesterol rośnie to prawdopodobieństwo choroby serca zmniejsza się co jest dosyć zaskakującę. Dla osoby której Cholesterol wynosi 300 prawdopodobieństwo choroby serca zwiększa się aż o 40 punktów procentowych.



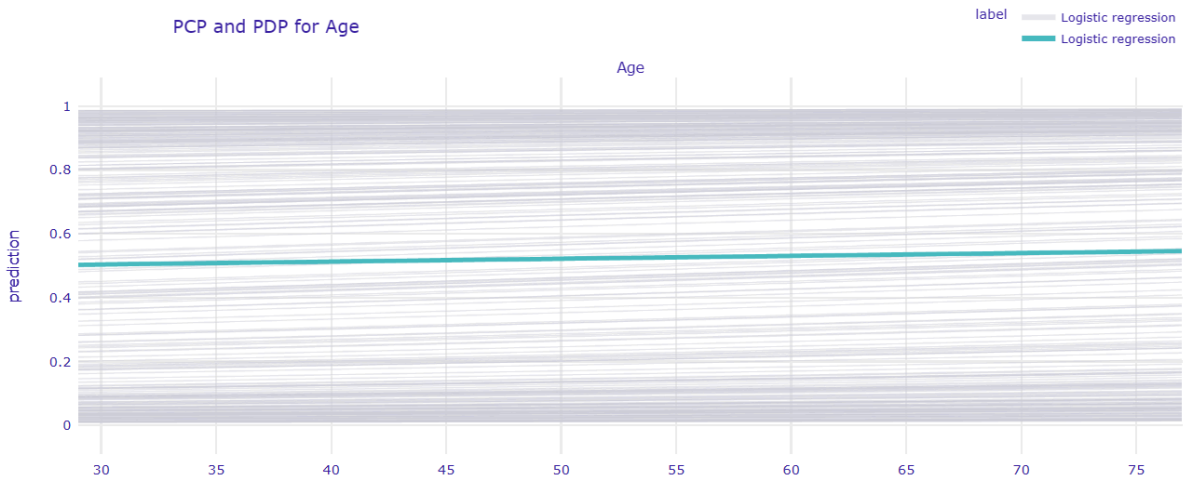
Podobnie jak w poprzednim wykresie wraz ze wzrostem wartości MaxHr można zaobserwować spadek prawdopodobieństwa choroby serca. Dla osoby której MaxHR wynosi 160 prawdopodobieństwo choroby serca zwiększa się o 25 punktów procentowych.



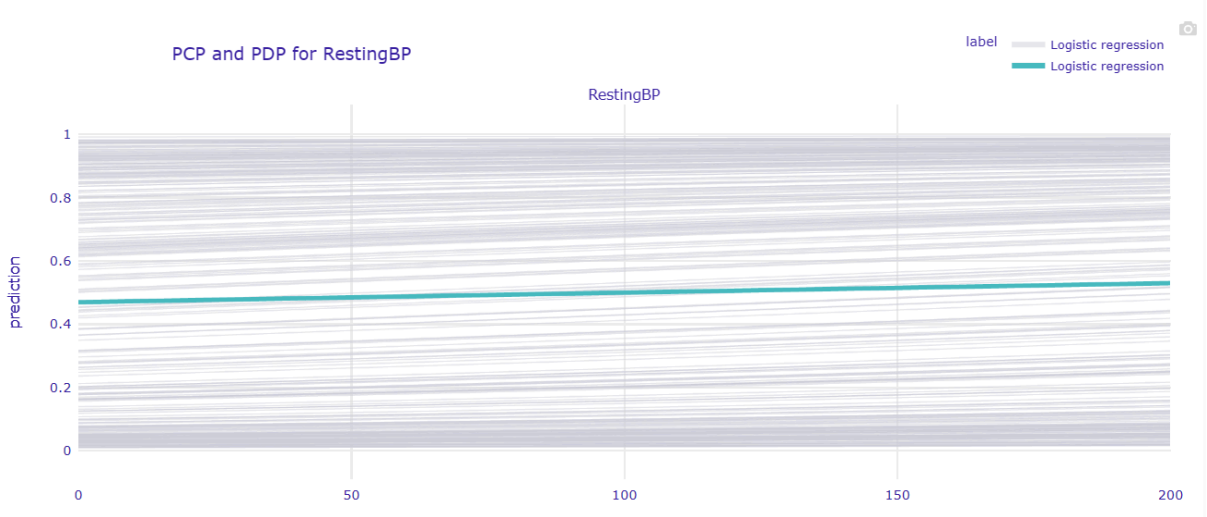
Powyższy wykres przedstawia, jak zmienia się przewidywane prawdopodobieństwo choroby serca w miarę zmiany zmiennej Oldpeak, przy zachowaniu innych zmiennych na stałym poziomie. Zwiększenie wartości zmiennej Oldpeak prowadzi do wzrostu predykcji choroby serca. Dla osoby której Oldpeak wynosi 5 prawdopodobieństwo choroby serca zwiększa się aż o 70 punktów procentowych.

Wykresy częściowej zależności (PDP)

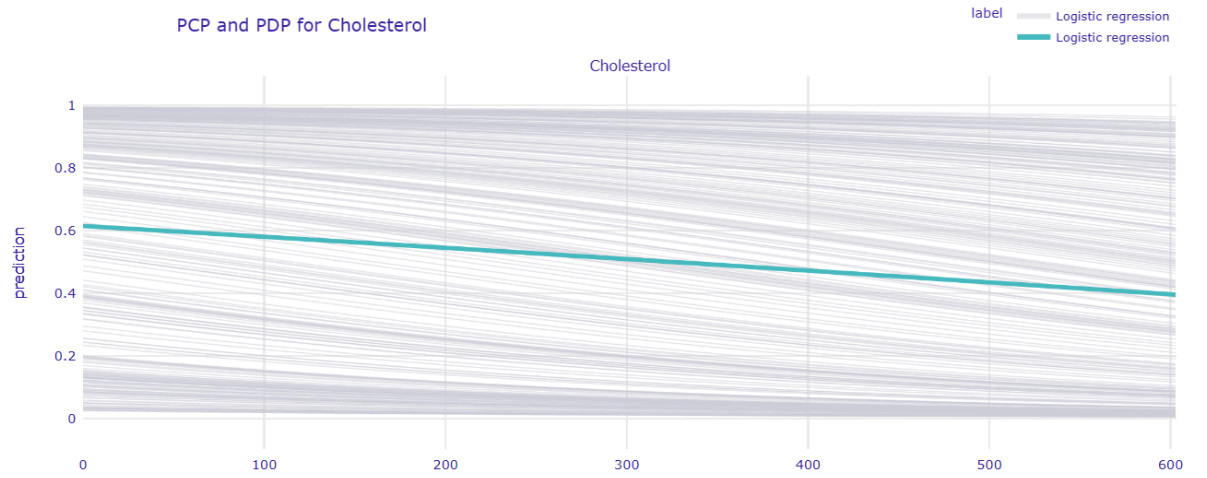
Wykres Partial Dependence Plot to narzędzie wizualizacyjne używane w dziedzinie uczenia maszynowego do zrozumienia wpływu pojedynczej zmiennej niezależnej na wynik modelu. Pozwala ono na zobaczenie, w jaki sposób prognozy modelu zmieniają się w odpowiedzi na zmiennej zmiany danej cechy, jednocześnie kontrolując wpływ innych zmiennych. W celu interpretacji częściowej zależność posłużono się jedynie zmiennymi ilościowymi ponieważ możliwość tworzenia takich wykresów dla innych rodzajów zmiennych są mocno ograniczone.

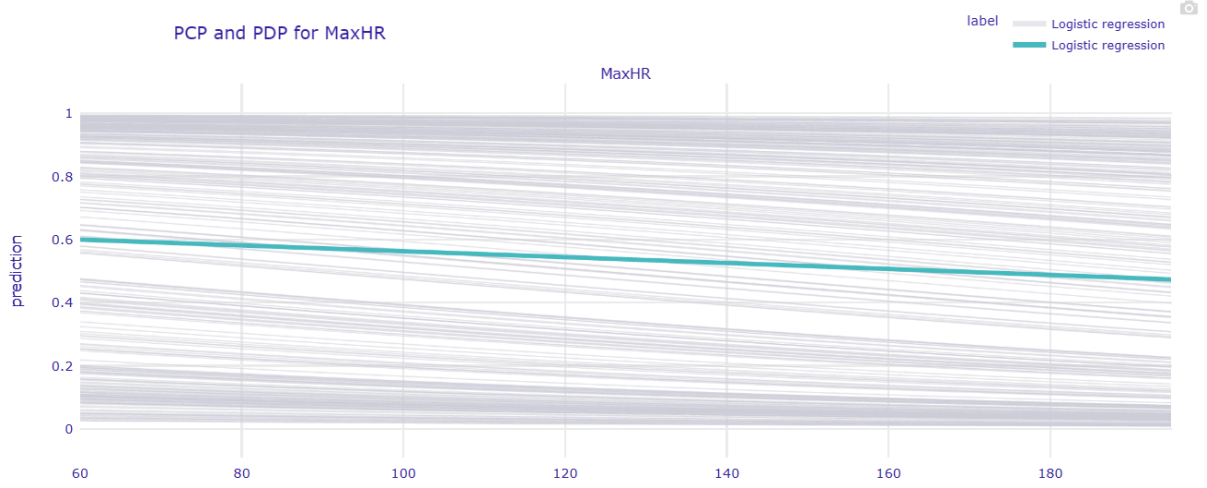


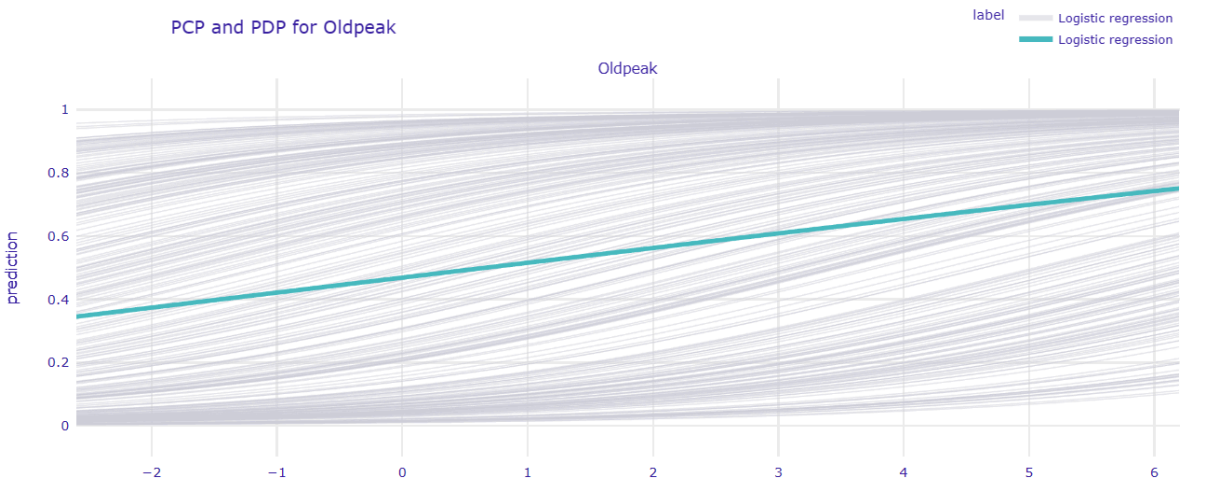
Powyższy wykres PDP przedstawia wpływ zmiany zmiennej age na predykcje modelu przy zachowaniu niezmienności pozostałych zmiennych. Krzywa zaznaczona niebieskim kolorem jest delikatnie rosnąca co oznacza że większy wiek idzie w parze z zwiększonym ryzykiem choroby serca. Na wykresie można zauważyć że dla różnych obserwacji wiek może w większym bądź mniejszym stopniu wpływać na predykcje.



Wykres przedstawia wpływ zmiennej RestingBP na predykcje modelu przy zachowaniu pozostałych zmiennych niezmiennych. Wszystkie obserwacje przedstawione na wykresie za pomocą szarych oraz niebieskiej lini są delikatnie rosnące co oznacza że wzrost RestingBM skutkuje wzrostem predykcji. Podobnie jak w poprzednim przypadku na wykresie znajduje się wiele obserwacji dla których nie da się wyciągnąć konkretnych wniosków jak bardzo wzrośnie predykcja biorąc pod uwagę tylko wybraną zmienną.





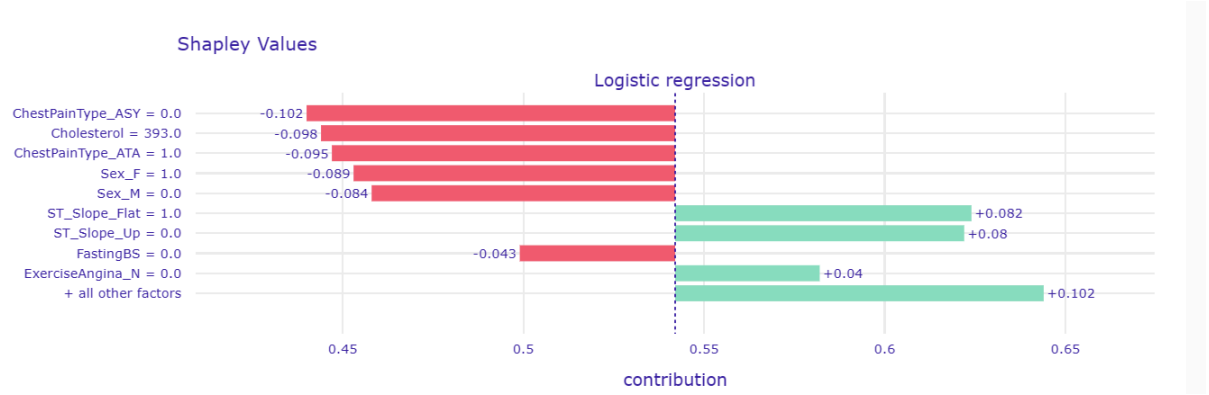


W przypadku wykresów PDP dla zmiennych Cholesterol, MaxHR oraz Oldpeak występuje ten sam problem co w pozostałych wykresach. Ciężko zauważyć pewien wzór w jakim stopniu mogłaby wzrosnąć bądź spaść predykcja. Natomiast wnioski które możemy wyciągnąć to wraz z wzrostem zmiennej Oldpeak zwiększa się ryzyko choroby serca.

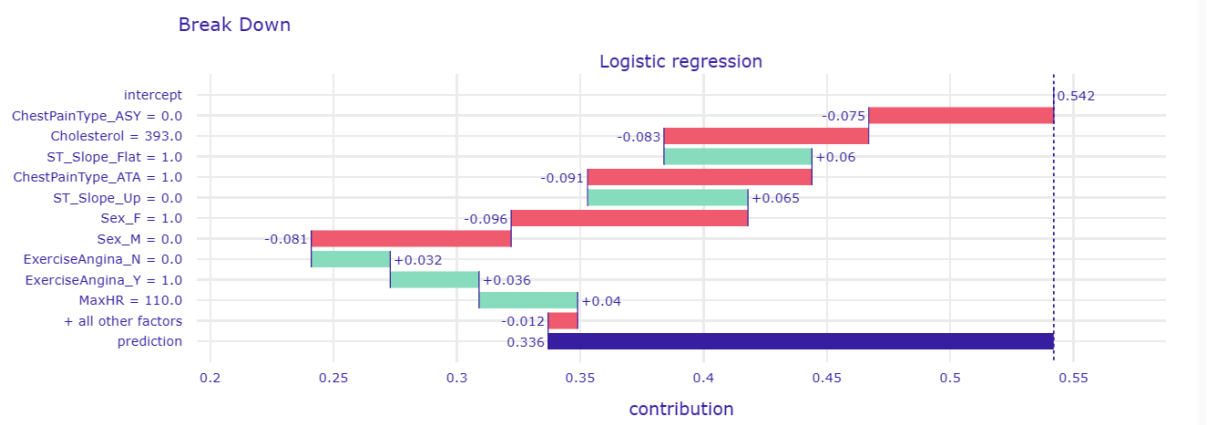
Wzrost zmiennych Cholesterol oraz MaxHR może delikatnie zmniejszyć ryzyko choroby serca.

Wartości SHAP

Na wykresie SHAP, każda cecha ma swój pasek, a kolor tego paska odpowiada wartościom SHAP dla danej obserwacji i danej cechy. Wykres ten pozwala na jednoczesne porównanie wpływu wielu cech. Wykres SHAP zaczyna się od linii bazowej, reprezentującej średnią wartość prognozy modelu dla całego zestawu danych. Jest to punkt, od którego zaczynamy dodawać i odejmować wpływ poszczególnych cech. Na samej górze wykresu znajdują się zmienne które w największym stopniu wpływają na model.



Dla danej obserwacji największy wpływ na model ma zmienna ChestPainType_ASY równa 0, w takim przypadku ryzyko choroby serca zmniejsza się o 10,02 punkta procentowego. Zmienne Cholesterol=304, ChestPainType_ATA=1, Sex_F=1,Sex_M =0, FastingBS przyczyniają się do zmniejszenia prognozy modelu dla danej obserwacji. Zmienne takie jak ST_Slope_Flat =1, ST_Slope_Up=0, ExerciseAngina_N =0,wpływają na zwiększenie się prognozy modelu.



Na powyższym wykresie każda cecha jest przedstawiona na osobnym pasku, a wartości skumulowane wpływów dodawane są na osi Y. W wykresie Break down ułożenie kolejności zmiennych może mieć duży wpływ na rzeczywiste wartość predykcji. Przy zaproponowanej kolejności zmiennych dla uśrednionej obserwacji największy pozytywny wpływ na zwiększenie predykcji mają zmienne ST_Slope_Up=0 oraz ST_Slope_Flat=1, natomiast największy wpływ na zmniejszenie predykcji mają ChestpainType_ASY=0, Cholesterol=393,

ChestPainType_ATA=1, Sex_F=1. Ostateczna wartość predykcji otrzymania choroby serca przez uśrednioną obserwacje wynosi 0,336.

Podsumowanie

Praca projektowa skoncentrowała się na analizie danych dotyczących chorób układu sercowo-naczyniowego oraz zastosowaniu różnych metod uczenia maszynowego w celu przewidywania prawdopodobieństwa ich wystąpienia. Wstęp do pracy jasno określił cel projektu, którym było wykorzystanie technik uczenia maszynowego do predykcji chorób układu sercowo-naczyniowego na podstawie różnorodnych danych zdrowotnych. Projekt skupił się na analizie danych, przygotowaniu ich do modelowania, podziale na zbiór uczący i testowy, a także zastosowaniu czterech różnych metod uczenia maszynowego: Lasy Losowe, SVM (Support Vector Machines), KNN (K-Nearest Neighbors), oraz Regresja logistyczna.

Analiza danych obejmowała sprawdzenie braków danych, wartości odstających oraz analizę korelacji między zmiennymi. Zdecydowano się na przyjęcie zbalansowanego zbioru danych, co umożliwiło uniknięcie konieczności stosowania metod oversamplingu lub undersamplingu. Podział na zbiór uczący i testowy został odpowiednio dostosowany, a także przetestowano różne random seedy w celu uzyskania jak najbardziej zbalansowanego podziału. W implementacji modeli Lasy Losowe, SVM, KNN, oraz Regresja logistyczna przeprowadzono optymalizację hiperparametrów, a analiza wyników wskazywała na zadowalające rezultaty. Modele Lasy Losowe,SVM oraz Regresja logistyczna uzyskały najlepsze wyniki, a analiza interpretowalności wybranego modelu(Regresja logistyczna), w tym wykresy PCP oraz PDP, pozwoliły na lepsze zrozumienie wpływu poszczególnych zmiennych na predykcję.

Projekt ukazał, że techniki uczenia maszynowego, takie jak lasy losowe, SVM, KNN oraz regresja logistyczna, mają duży potencjał w przewidywaniu prawdopodobieństwa wystąpienia chorób układu sercowo-naczyniowego. Skuteczność tych modeli w identyfikowaniu osób narażonych na tego typu schorzenia jest obiecująca. Przed rozpoczęciem procesu uczenia maszynowego istotne jest dokładne zrozumienie i analiza dostępnych danych. Projekt uwypuklił znaczenie analizy rozkładów zmiennych, korelacji, oraz braków danych. Dodatkowo, równowaga zbioru danych, standaryzacja i inne techniki przygotowania danych mają wpływ na skuteczność modeli. W trakcie projektu eksperymentowano z różnymi parametrami dla poszczególnych algorytmów, takich jak liczba drzew w lasach losowych, parametry jądra w SVM, liczba sąsiadów w KNN, czy siła regularyzacji w regresji logistycznej. Dobór odpowiednich parametrów ma istotny wpływ na skuteczność modeli. Stosowanie standaryzacji danych miało znaczący wpływ na skuteczność modelu KNN, podczas gdy w przypadku innych algorytmów efekt ten był mniej istotny. Wniosek ten sugeruje, że niektóre modele są bardziej wrażliwe na różnice w skali danych niż inne.