Heart disease analysis

Wybrany dataset: Heart disease dataset

Znalezienie zestawu danych i analiza

Został znaleziony interesujący nas zestaw danych. Dane pochodziły z czterech konkretnych miejsc (Cleveland, Węgier, Szwajcarii i Long Beach V). Zestaw składał się z 14 kolumn, po 1025 danych w każdej z nich. Zbiór nie posiadał danych tekstowych, wszystkie dane w zbiorze były liczbowe. 1. age

2. sex 1-male 0-female
3. cp - chest pain type (4 values)
4. trestbps - resting blood pressure
5. chol - serum cholestoral in mg/dl
6. fbs - fasting blood sugar > 120 mg/dl
7. restecg - resting electrocardiographic results (values 0,1,2)
8. thalach - maximum heart rate achieved
9. exang - exercise induced angina
10. oldpeak - ST depression induced by exercise relative to rest
11. slope - the slope of the peak exercise ST segment
12. ca - number of major vessels (0-3) colored by flourosopy
13. thal - thal: 0 = normal; 1 = fixed defect; 2 = reversable defect
14. target - refers to the presence of heart disease in the patient (Stosunek zer do jedynek w kolumnie „target” wynosi 499 do 526)

Zbiór danych posiada 4 cechy binarne: sex, fbs, exang, target.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

Część 1

Operowanie na brakujących wartościach (Wartościach NULL) i wypełnianie tych wartości

Zbiór danych był w całości pełny, bez żadnych wartości NULL więc zostało usunięte losowo około 9% danych z każdej z kolumn.

```
      age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
0      52    1  0.0     125.0  212.0  0.0      1.0    168.0    0.0      1.0
1      53    1  0.0     140.0  203.0  1.0      0.0    155.0    1.0      3.1
2      70    1  0.0     145.0  174.0  NaN      1.0    125.0    1.0      2.6
3      61    1  0.0     148.0  203.0  0.0      1.0    161.0    0.0      0.0
4      62    0  0.0     138.0  294.0  1.0      1.0    106.0    0.0      1.9
...   ...  ...  ...       ...    ...  ...      ...      ...    ...      ...
1020   59    1  1.0     140.0  221.0  0.0      1.0    164.0    1.0      0.0
1021   60    1  0.0     125.0  258.0  0.0      0.0    141.0    1.0      2.8
1022   47    1  0.0     110.0  275.0  0.0      0.0    118.0    1.0      1.0
1023   50    0  0.0     110.0  254.0  0.0      0.0    159.0    0.0      0.0
1024   54    1  0.0     120.0  188.0  0.0      NaN      NaN    0.0      1.4

      slope   ca  thal  target
0       2.0  2.0   NaN     0.0
1       0.0  0.0   3.0     0.0
2       NaN  0.0   3.0     0.0
3       2.0  1.0   3.0     0.0
4       1.0  3.0   NaN     0.0
...     ...  ...   ...     ...
1020    2.0  0.0   2.0     1.0
1021    1.0  1.0   3.0     0.0
1022    1.0  1.0   2.0     0.0
1023    2.0  0.0   2.0     1.0
1024    1.0  1.0   3.0     NaN

[1025 rows x 14 columns]
```

Po usunięciu dane zostały wypełnione. Poniżej zostały przedstawione procentowe wartości pustych danych przed wypełnieniem i po wypełnieniu.

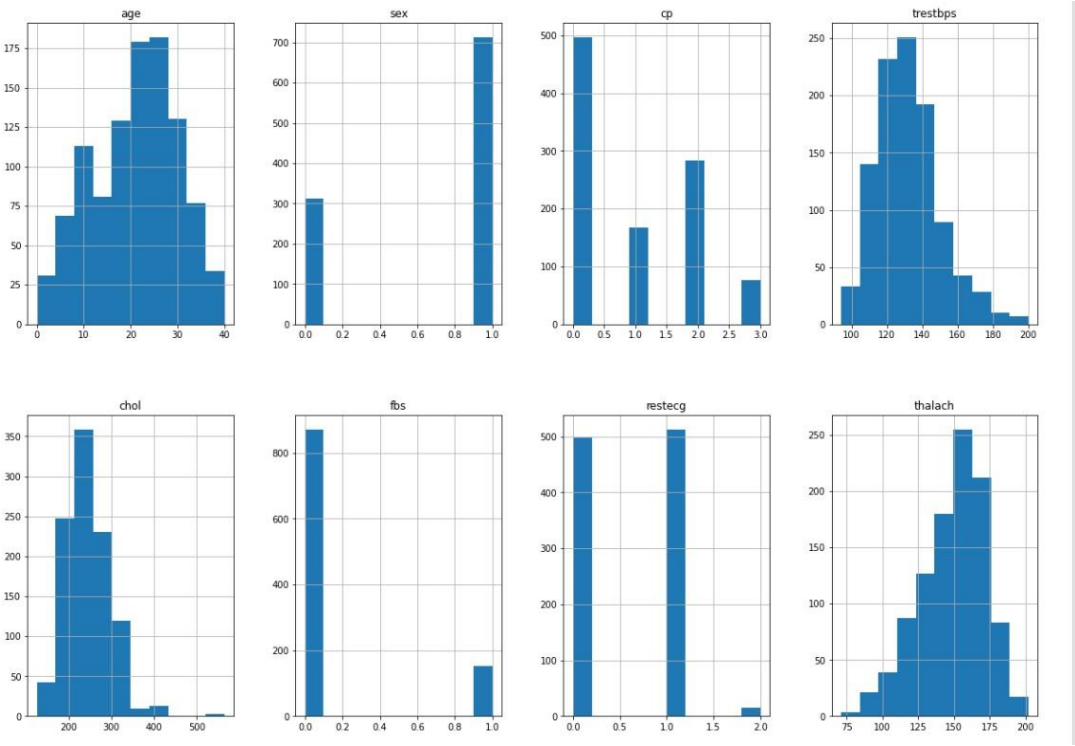
age	0.000000	age	0.0
sex	0.000000	sex	0.0
cp	0.088780	cp	0.0
trestbps	0.087805	trestbps	0.0
chol	0.084878	chol	0.0
fbs	0.086829	fbs	0.0
restecg	0.085854	restecg	0.0
thalach	0.087805	thalach	0.0
exang	0.085854	exang	0.0
oldpeak	0.087805	oldpeak	0.0
slope	0.086829	slope	0.0
ca	0.084878	ca	0.0
thal	0.087805	thal	0.0
target	0.000000	target	0.0
dtype: float64		dtype: float64	

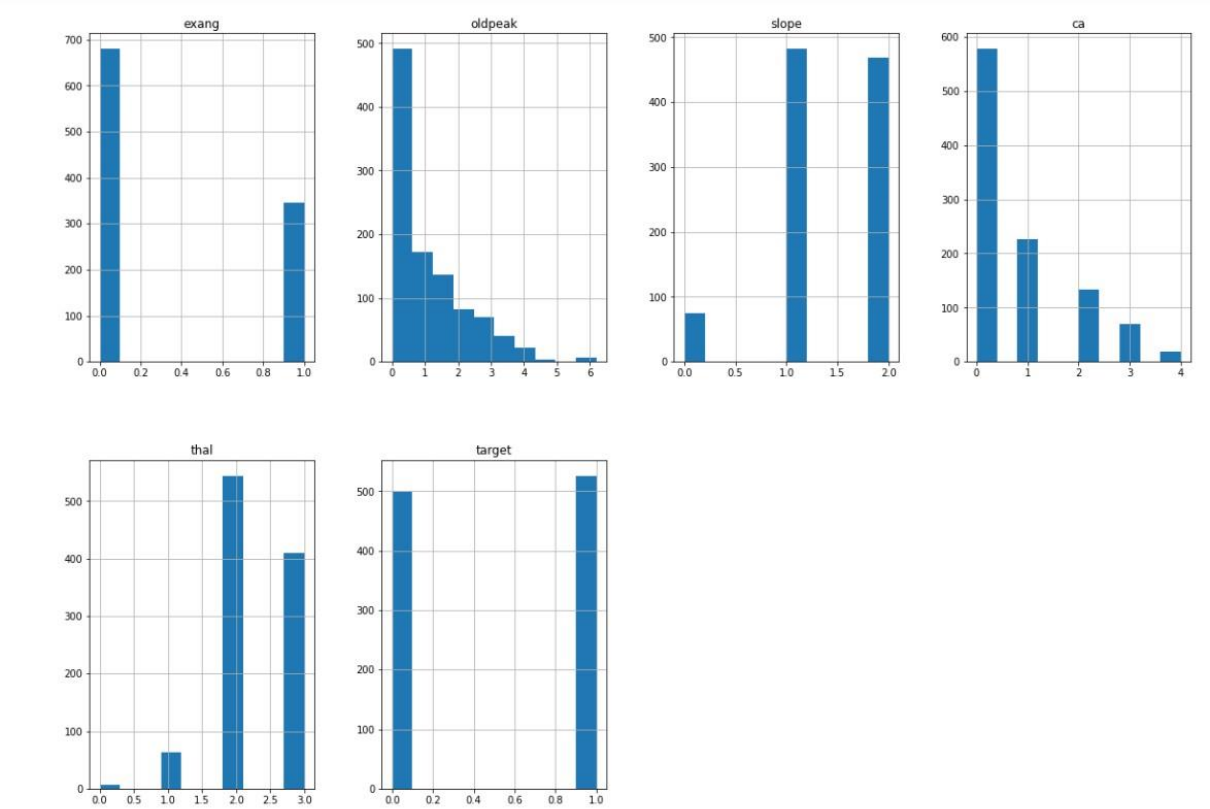
Przed **Po**

Wartości NULL, które pojawiły się po usunięciu danych zostały wypełnione najczęściej pojawiającą się wartością w zbiorze. Metoda wypełniania najczęściej pojawiającą się wartością wydawała się nam najtrafniejsza, ponieważ wypełnianie wartości w zbiorze, w którym są wartości z wieloma miejscami po przecinku poprzez średnią skutkowałoby powstawaniem kolosalnie długiej średniej. Zbiór nie posiadał danych tekstowych więc kodowanie wartości tekstowych nie zostało zastosowane.

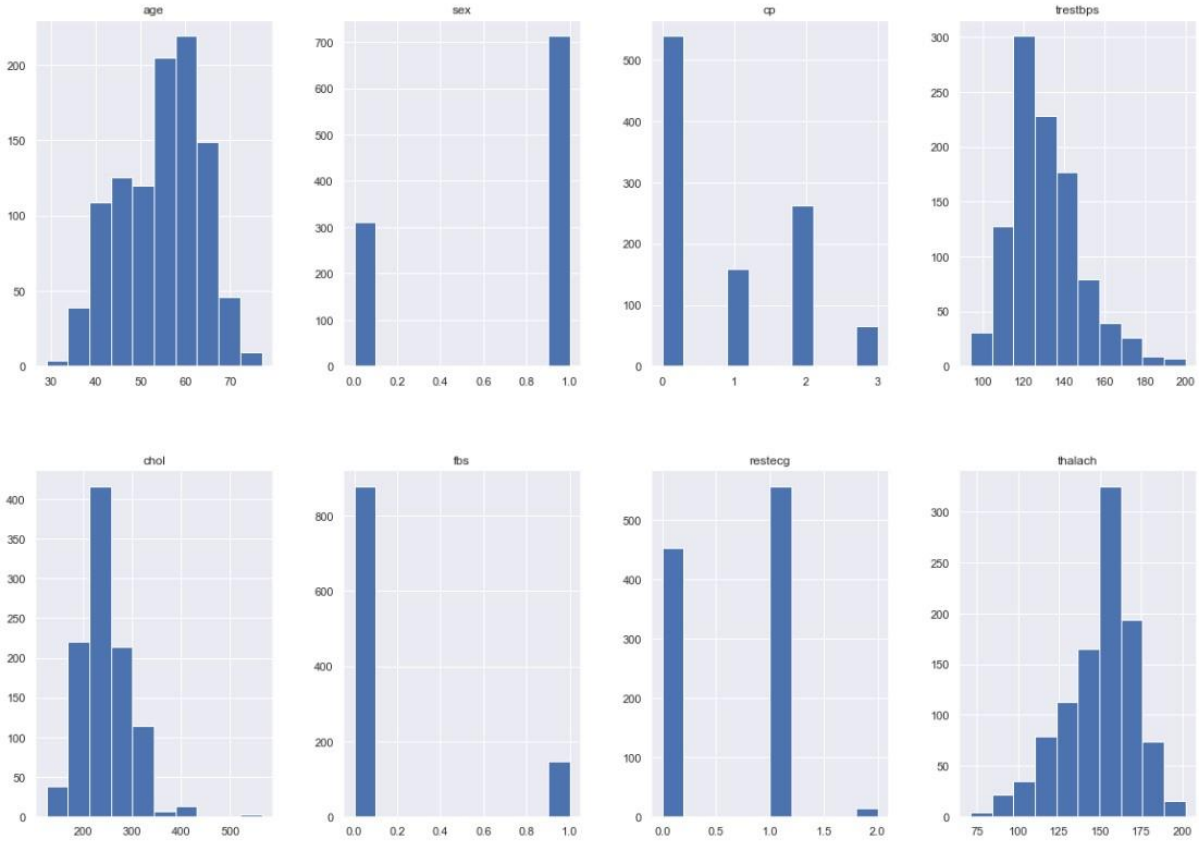
Wizualizacja danych

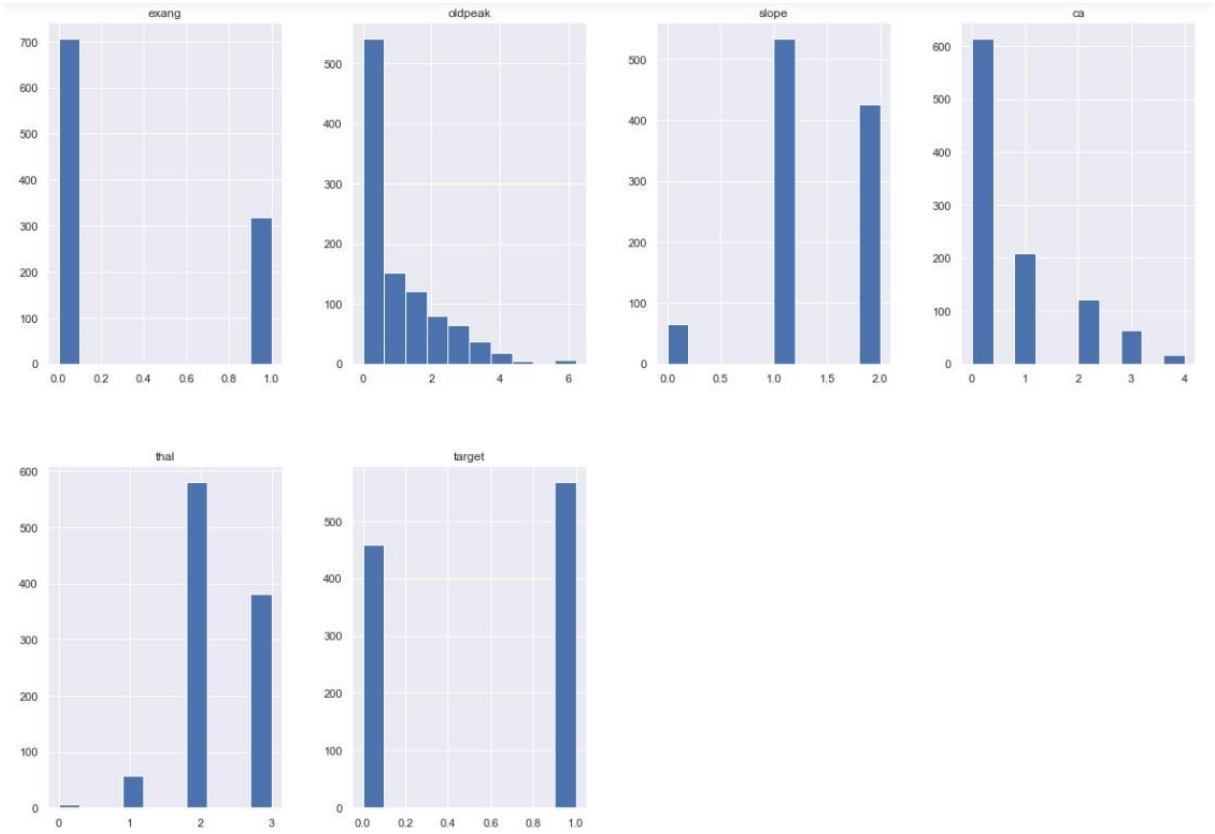
Dane zostały zwizualizowane poprzez: histogram, pairplot, macierz korelacji i boxplot. **Histogram danych początkowych**





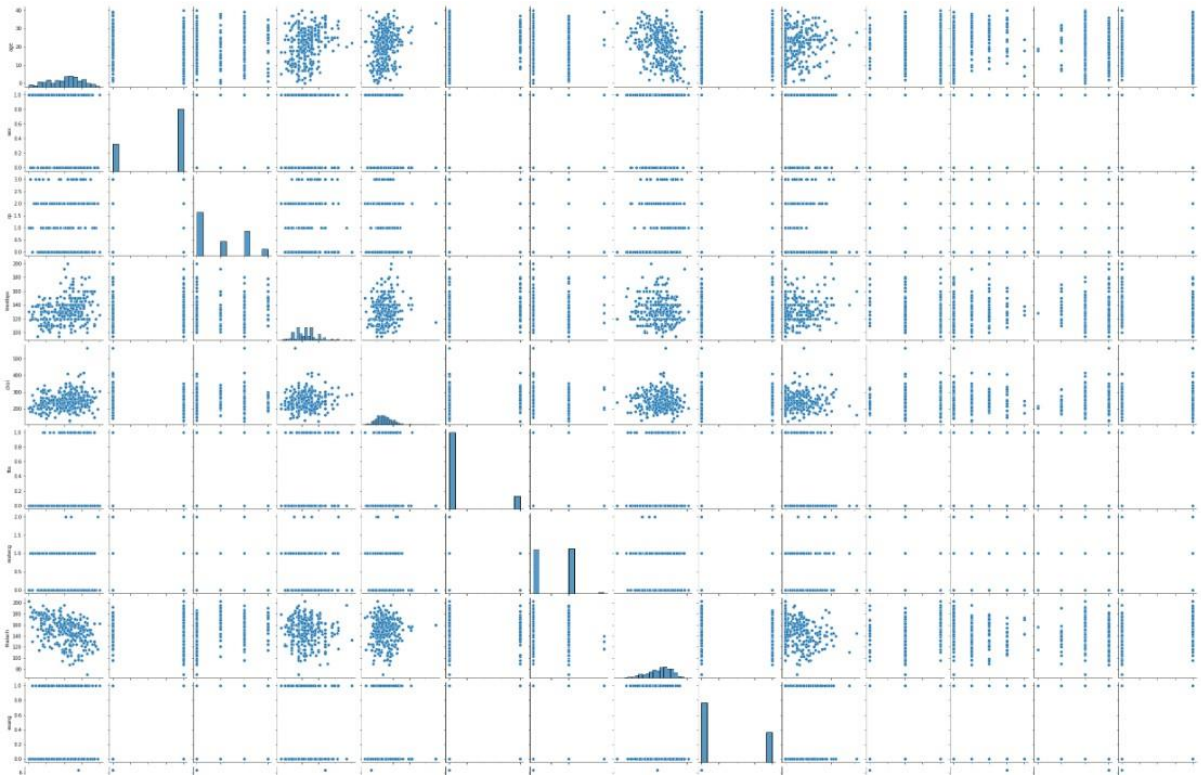
Histogram danych po usunięciu i wypełnieniu

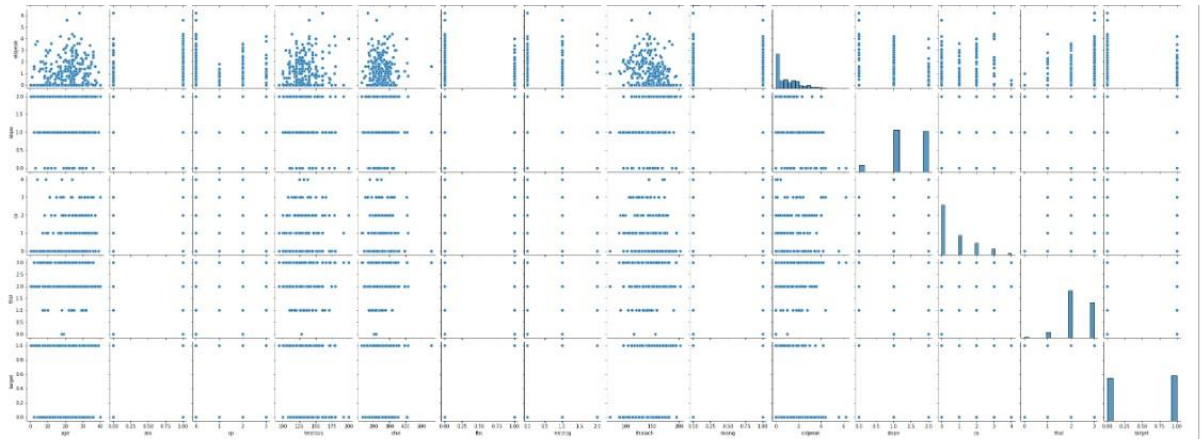




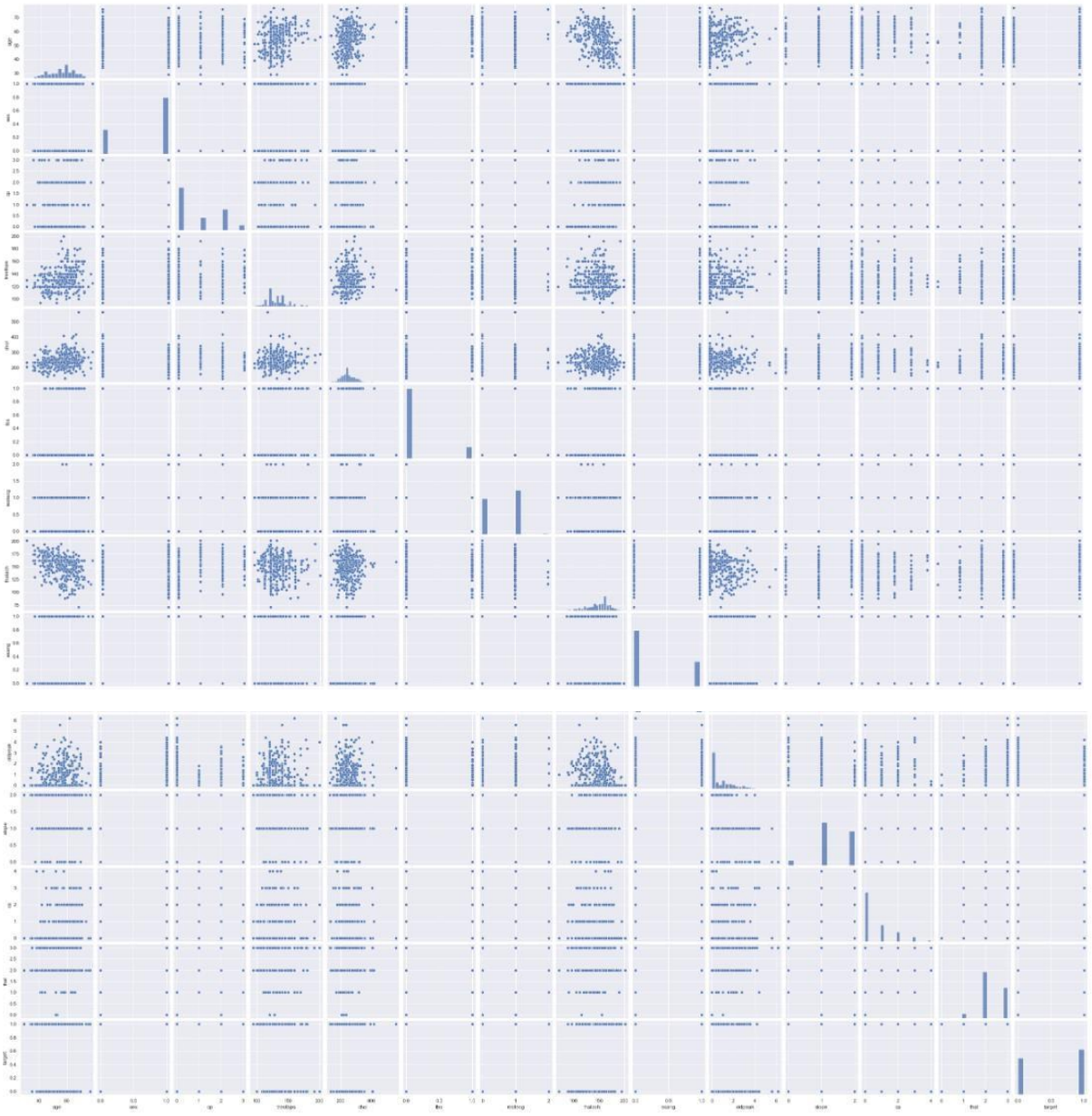
Można wywnioskować, że w każdym z histogramów występuje przynajmniej jedna wartość dominująca, która jest znacząco wyższa od pozostałych wartości.

Pairplot danych początkowych



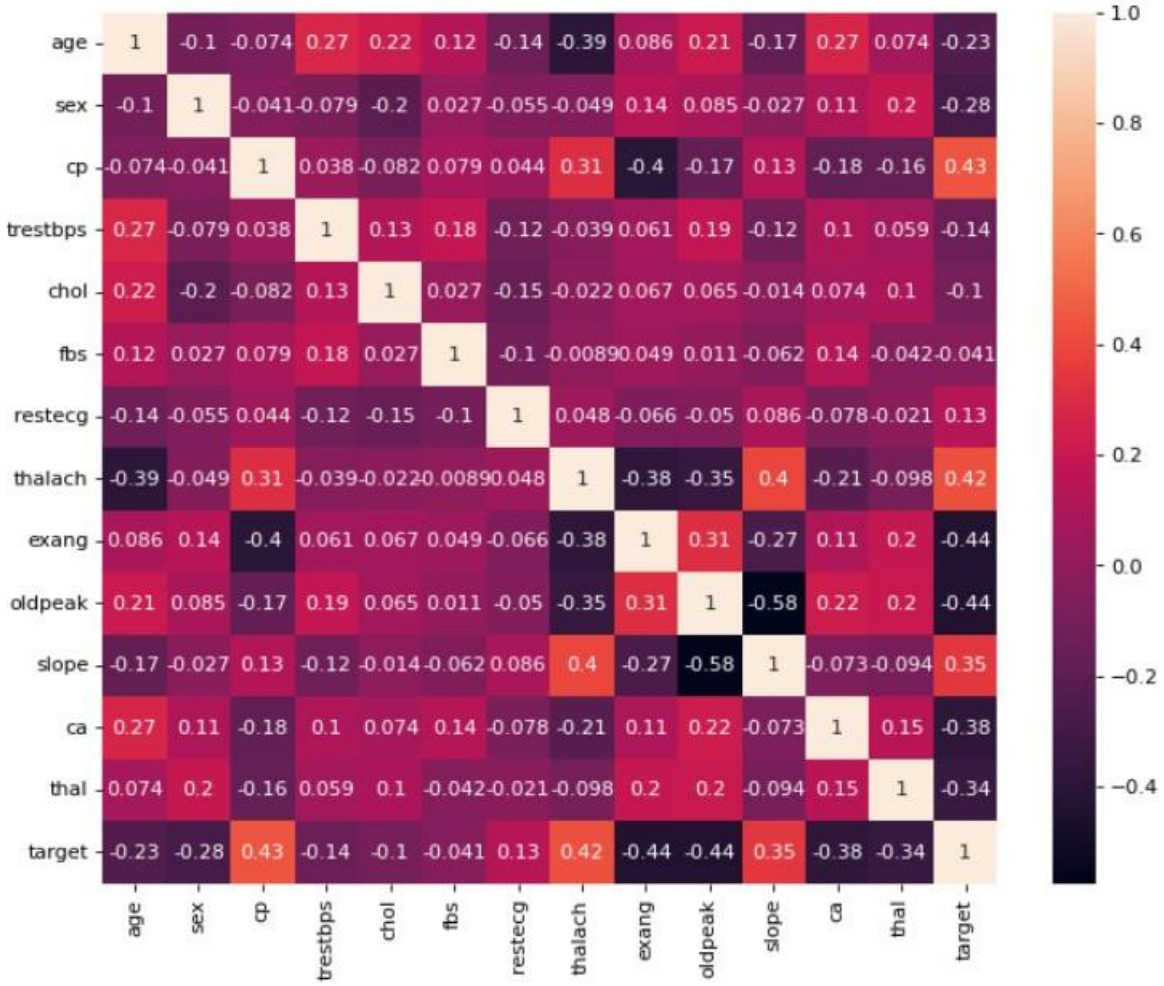


Pairplot danych po usunięciu i wypełnieniu

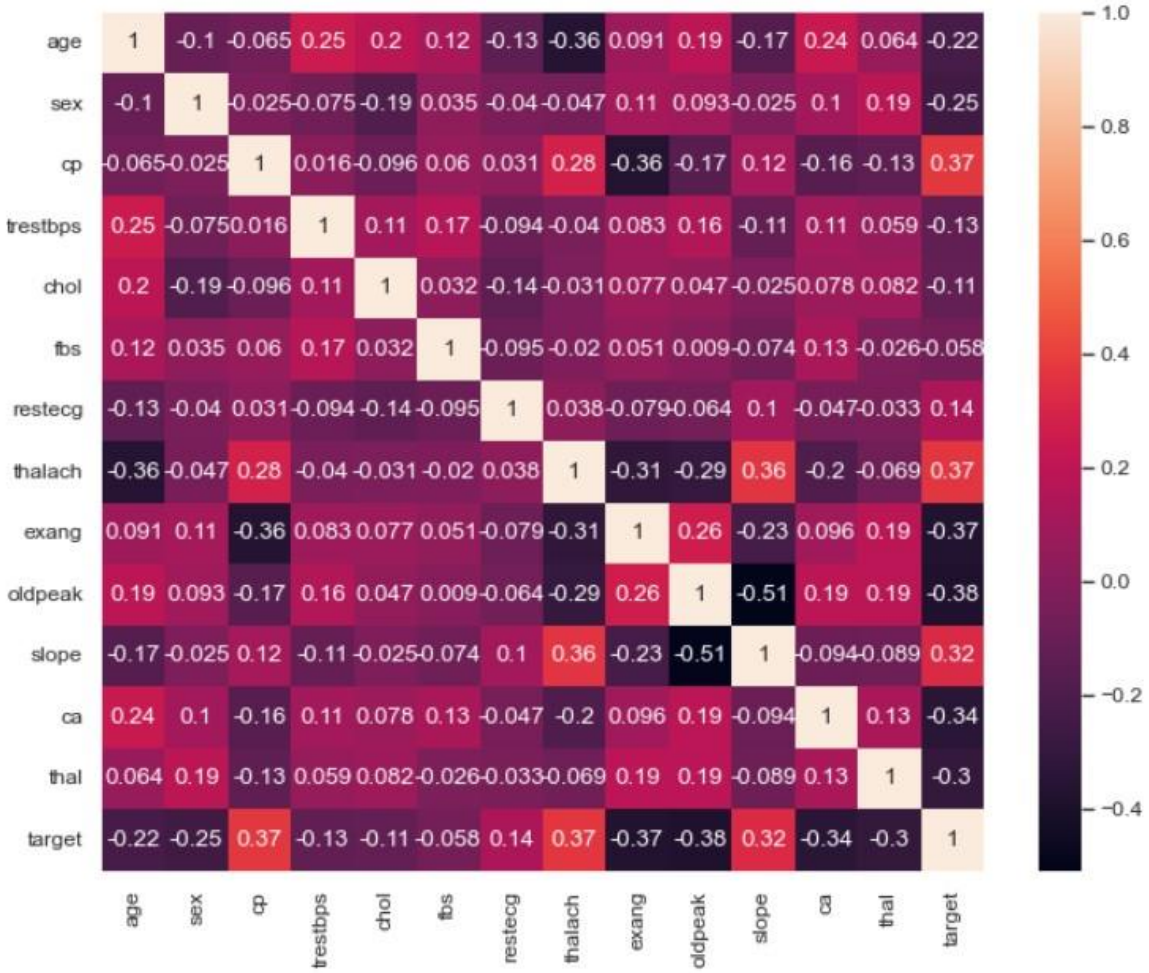


Można zaobserwować, że przedstawione dane są danymi kategorycznymi, a nie ciągłymi.

Macierz korelacji danych początkowych

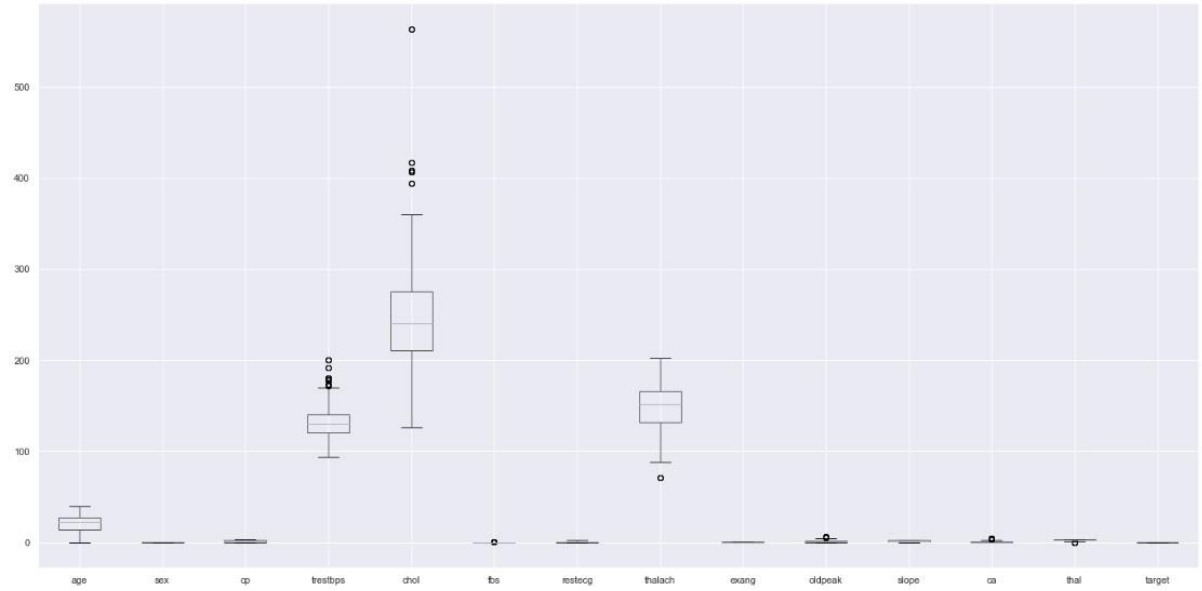


Macierz korelacji danych po usunięciu i wypełnieniu

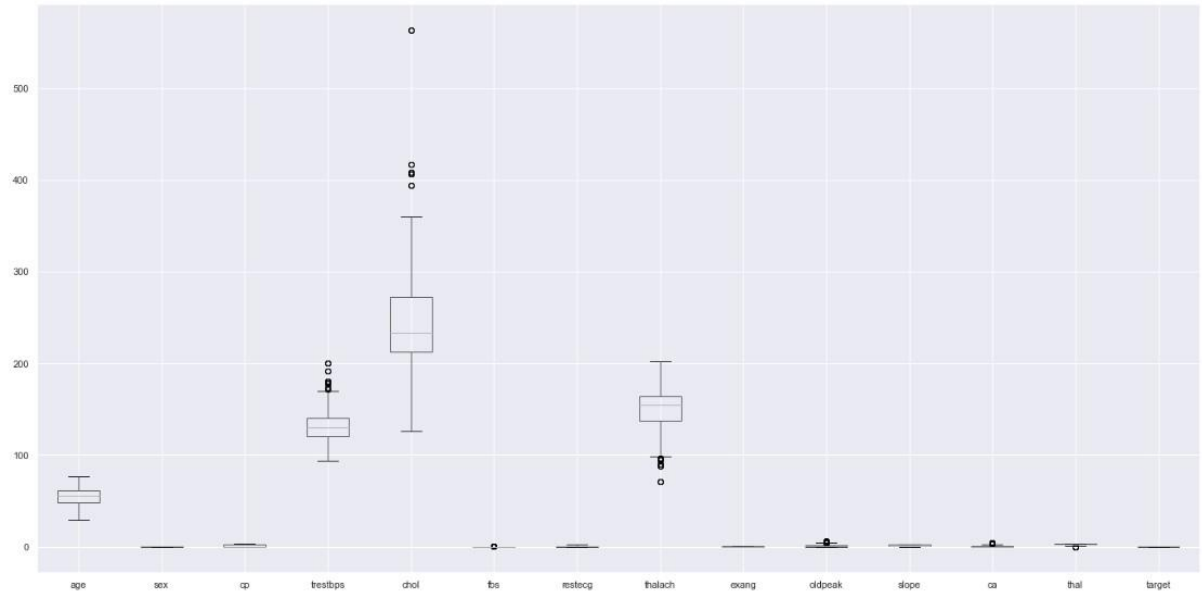


Macierz korelacji określa jak silna jest zależność pomiędzy poszczególnymi cechami. Kolory ciemne wskazują niską zależność, zaś kolory jasne silną. Można zobaczyć znaczącą przewagę kolorów ciemniejszych (Tj. czarny, fioletowy, ciemny róż) co wskazuje na niską zależność cech. Na obu macierzach korelacji można zaobserwować największe zależności pomiędzy cechami: cp do target, thalach do target, slope do target jak i slope do thalach. Oznacza to, że największe oddziaływanie na chorobę serca mają: rodzaj bólu w klatce piersiowej, wysokie tętno jak i odcinek ST.

Boxplot danych początkowych



Boxplot danych po usunięciu i wypełnieniu



Z boxplotów możemy zauważyć obecność danych odstających, z czego cholesterol (chol) ma najbardziej rozległe dane odstające i największy rozrzut pomiędzy kwartylami.

Skalowanie cech

Skalowanie cech polega na sprowadzeniu wartości w kolumnach do wspólnych zakresów wartości. Skalowanie cech można było zrobić za pomocą skalowania min-max albo standaryzacji. W naszym przypadku została wybrana standaryzacja. Poniżej została przedstawiona standaryzacja na danych początkowych jak i na danych po usunięciu i wypełnieniu.

Standaryzacja na danych początkowych

```
array([[-2.74681382e-01,  6.61504088e-01, -9.15755416e-01,
        -3.77635519e-01, -6.59332089e-01, -4.18877924e-01,
         8.91254880e-01,  8.21320521e-01, -7.12287120e-01,
        -6.08883932e-02,  9.95433338e-01,  1.20922066e+00,
         1.08985168e+00],
       [-1.62600006e-01,  6.61504088e-01, -9.15755416e-01,
         4.79107303e-01, -8.33861171e-01,  2.38733039e+00,
        -1.00404855e+00,  2.55967905e-01,  1.40392824e+00,
         1.72713707e+00, -2.24367514e+00, -7.31971475e-01,
         1.08985168e+00],
       [ 1.74278339e+00,  6.61504088e-01, -9.15755416e-01,
         7.64688244e-01, -1.39623266e+00, -4.18877924e-01,
         8.91254880e-01, -1.04869198e+00,  1.40392824e+00,
         1.30141672e+00, -2.24367514e+00, -7.31971475e-01,
         1.08985168e+00],
       [ 7.34051002e-01,  6.61504088e-01, -9.15755416e-01,
         9.36036809e-01, -8.33861171e-01, -4.18877924e-01,
         8.91254880e-01,  5.16899882e-01, -7.12287120e-01,
        -9.12329090e-01,  9.95433338e-01,  2.38624595e-01,
```

Standaryzacja na danych po usunięciu i wypełnieniu

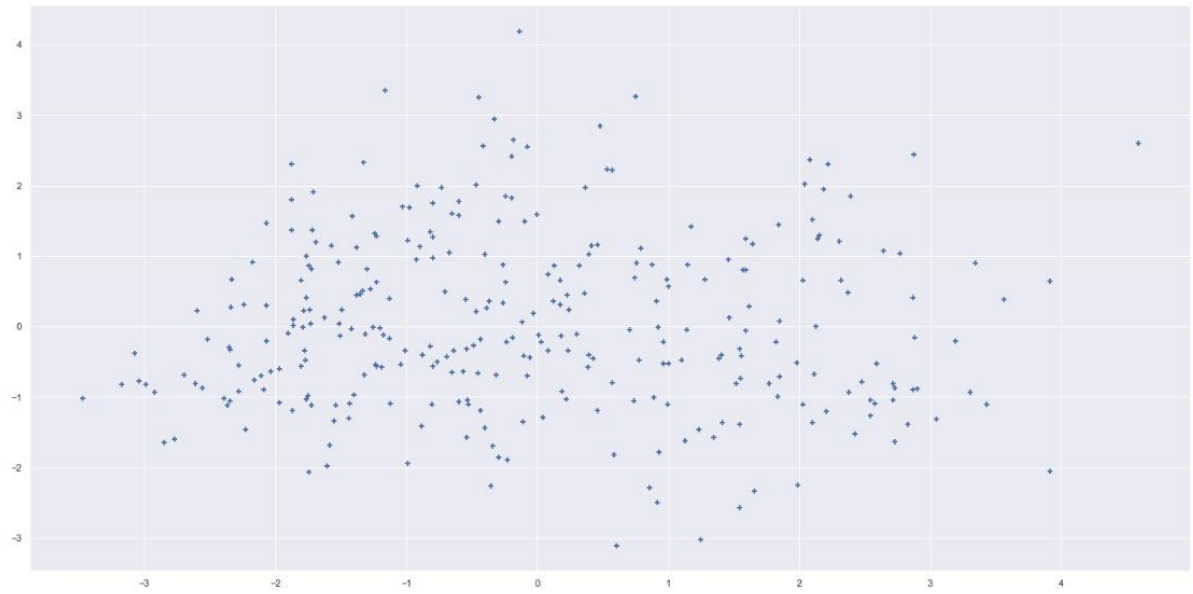
```
array([[-0.26843658,  0.66150409, -0.85048969, -0.32638812, -0.67502019,
        -0.40917718,  0.82260955,  0.79870169, -0.67066224,  0.01784976,
         1.08727761,  1.29524533, -0.50593926],
       [-0.15815703,  0.66150409, -0.85048969,  0.54893324, -0.85635569,
         2.44392903, -1.09369679,  0.21630593,  1.49106353,  1.81325221,
        -2.26952826, -0.68628671,  1.16154543],
       [ 1.71659547,  0.66150409, -0.85048969,  0.84070702, -1.44065899,
        -0.40917718,  0.82260955, -1.12768426,  1.49106353,  1.38577543,
        -0.59112533, -0.68628671,  1.16154543],
       [ 0.72407944,  0.66150409, -0.85048969,  1.0157713 , -0.85635569,
        -0.40917718,  0.82260955,  0.48510397, -0.67066224, -0.83710378,
         1.08727761,  0.30447931,  1.16154543],
       [ 0.834359  , -1.51170646, -0.85048969,  0.43222372,  0.97714775,
         2.44392903,  0.82260955, -1.97887805, -0.67066224,  0.78730795,
        -0.59112533,  2.28601135, -0.50593926],
       [ 0.39324077, -1.51170646, -0.85048969, -1.78525705,  0.05032183,
        -0.40917718, -1.09369679,  0.52990365, -0.67066224,  0.01784976,
        -0.59112533, -0.68628671, -0.50593926],
       [ 0.39324077,  0.66150409, -0.85048969, -0.96829045,  1.46070909,
```

Analiza głównych składowych (PCA)

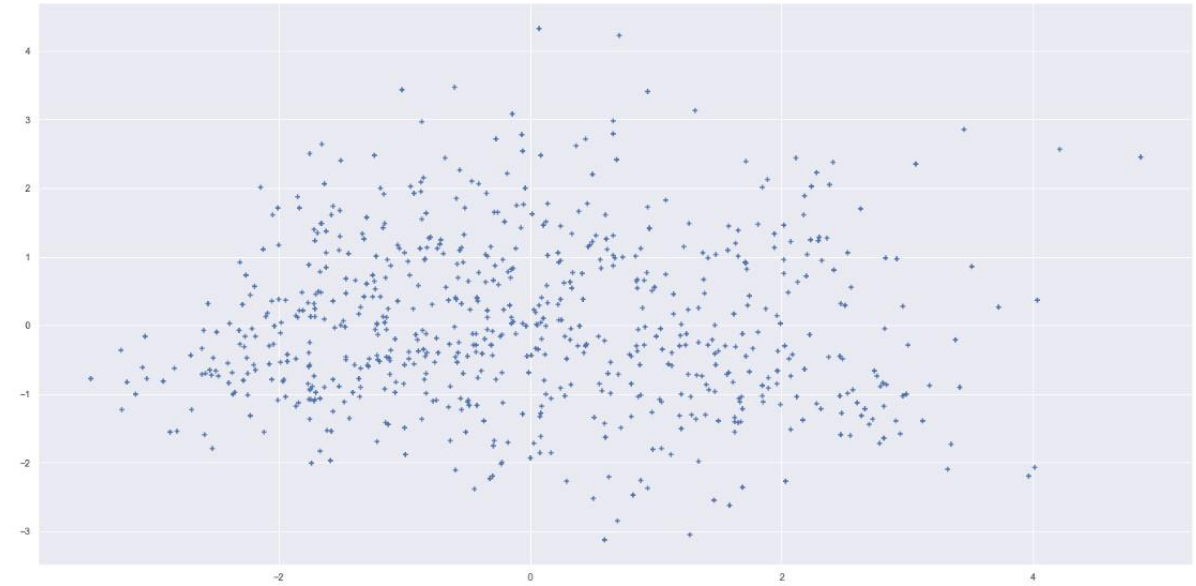
Analiza głównych składowych jest techniką liniowej transformacji, jest najczęściej wykorzystywana w celu redukcji wymiarowości. Analizie głównych składowych zostały poddane oba zbiory (zbiór podstawowy i zbiór po usunięciu i wypełnieniu). Poniżej przedstawione zostały wyniki.

Parametr n_components został ustawiony na wartość 4, co oznacza że w PCA zostały zachowane 4 komponenty.

Zbiór podstawowy z wykorzystaniem PCA



Zbiór po usunięciu i wypełnieniu z wykorzystaniem PCA



Można zauważyć, że zbiór danych po usunięciu i wypełnieniu jest o wiele bardziej skoncentrowany w jednym miejscu.

Część 2

Podział danych na zbiór treningowy i testowy oraz krótkie omówienie algorytmów klasyfikacji

Na samym początku zbiór danych został podzielony na zbiory treningowy i testowy w proporcji 80% do 20% po czym zostało przeprowadzone przetwarzanie danych.

Do klasyfikacji danych użyliśmy kolejno czterech metod klasyfikacji: Support Vector Machine (SVM), regresja logistyczna, drzewo decyzyjne i las losowy. Przetwarzanie danych zostało przeprowadzone na 4 wymienionych wyżej klasyfikatorach i na 4 różnych zbiorach danych (Dane początkowe, dane po usunięciu i wypełnieniu, dane po PCA oraz dane po PCA po usunięciu i wypełnieniu).

Support Vector Machine (SVM) - abstrakcyjny koncept maszyny, która działa jak klasyfikator, a której nauka ma na celu wyznaczenie hiperpłaszczyzny rozdzielającej z maksymalnym marginesem przykłady należące do dwóch klas.

Regresja logistyczna (LR) - Jej celem jest określenie prawdopodobieństwa przynależności próbki do klasy. W tym modelu regresji wykorzystywana jest funkcja logistyczna znana również jako funkcja sigmoidalna.

Drzewo decyzyjne (DT) – Algorytm ten wykorzystywany w uczeniu maszynowym zarówno do rozwiązywania problemu klasyfikacji, jak i regresji. Dzięki prostocie i klarowności w wyborze odpowiedzi, bardzo dobrze nadaje się do zapoznania się z danymi. Na podstawie odpowiedzi na szereg pytań algorytm przypisuje etykietę nowej próbce. Drzewo decyzyjne składa się z węzłów, w których znajdują się pytania, gałęzi, jako wszystkich możliwych odpowiedzi na zadane pytanie, oraz liści, które zawierają ostateczne predykcje. Każde drzewo rozchodzi się w dół od korzenia, czyli pierwszego węzła. Każda gałąź prowadzi do następnego węzła lub do liścia.

Dla drzewa decyzyjnego parametr głębokości drzewa został ustawiony na wartość 3. Przy parametrze głębokości o wartości 9 wynik wynosił 0.9951219512195122, a przy wartości 10 osiągał równe 1.0 dokładności.

Las losowy (RF) – Drzewa decyzyjne są bardzo wrażliwe nawet na niewielkie zmiany w zbiorze treningowym jednak jeśli nie będzie brać się pod uwagę tylko jednego drzewa decyzyjnego, a całą grupę, można dzięki temu osiągnąć lepsze wyniki, taką grupę nazywamy lasem losowym. Koncepcja lasu losowego polega na połączeniu słabych klasyfikatorów (ang. weak learners) w jeden silny klasyfikator (ang. Strong learner) z mniejszą wrażliwością na przetrenowanie, a co za tym idzie, lepszą zdolnością uogólniania. Algorytm lasu losowego rozpoczyna się od stworzenia k nowych zbiorów danych dla k drzew decyzyjnych. Proces ten nazywany jest agregacją (ang. bagging jako skrót bootstrap aggregating) i polega na tworzeniu nowych zbiorów za pomocą losowania ze zwracaniem. Następnym krokiem jest wygenerowanie k drzew decyzyjnych na podstawie nowych zestawów danych.

Dla lasu losowego parametr głębokości został ustawiony na wartość 4. Przy parametrze głębokości o wartości 7 wynik wynosił 0.975609756097561, a przy wartości 8 osiągał równe 1.0 dokładności.

Wyniki dla poszczególnych algorytmów klasyfikacji

	Dane początkowe	Dane po usunięciu i wypełnieniu	Dane po PCA	Dane po PCA po usunięciu i wypełnieniu
SVM	0.7268292682926829	0.6878048780487804	0.8780487804878049	0.8829268292682927
LR	0.8292682926829268	0.8000000000000000	0.8682926829268293	0.8878048780487805
DT	0.824390243902439	0.8390243902439024	0.8878048780487805	0.7658536585365854
RF	0.8731707317073171	0.8682926829268293	0.8439024390243902	0.8243902439024390

Z obserwacji można wywnioskować, że dla danych początkowych oraz danych po usunięciu i wypełnieniu najlepsze wyniki uzyskuje **algorytm lasu losowego**, zaś najgorsze wyniki algorytm **SVM** tak samo w obu przypadkach. Dla danych po PCA wyniki są bardzo podobne, lecz **drzewo decyzyjne** uzyskuje najlepszy wynik, a najsłabszy **regresja logistyczna**. Dla danych po PCA i po usunięciu i wypełnieniu najlepiej wypada **regresja logistyczna** zaś najsłabiej **drzewo decyzyjne**.

Macierze pomyłek

Tablica pomyłek (macierz błędów) – stosuje się ją w celu oceny jakości klasyfikacji binarnej (na dwie klasy). Dane oznaczone etykietami: pozytywną i negatywną poddawane są klasyfikacji, która przypisuje im predykowaną klasę pozytywną albo predykowaną klasę negatywną. Możliwa jest sytuacja, że dana oryginalnie oznaczona jako pozytywna zostanie omyłkowo zaklasyfikowana jako negatywną. Każda z takich sytuacji jest przedstawiona w tablicy pomyłek.

	Pozytywna	Negatywna
Pozytywna	Prawdziwie pozytywna (TP)	Fałszywie pozytywna (FP)
Negatywna	Fałszywie Negatywna (FN)	Prawdziwie negatywna (TN)

Macierz pomyłek, posiada 4 wartości:

1) **TP - prawdziwie pozytywna** **–** przewidywanie pozytywne przypadku choroby serca, które zostało sklasyfikowane pozytywnie.

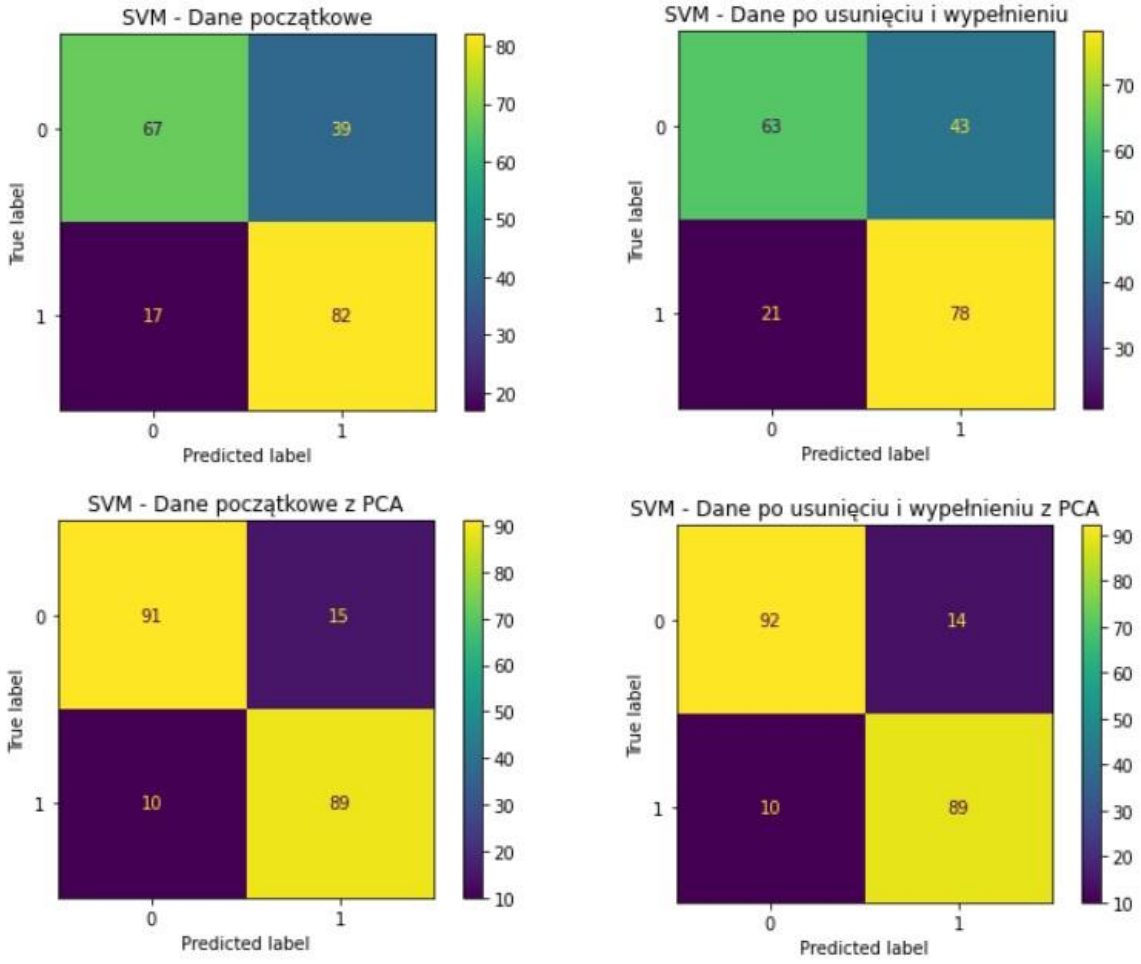
2) **FP - fałszywie pozytywna** **-** przewidywanie pozytywne przypadku choroby serca, które zostało sklasyfikowane negatywnie.

3) **FN - fałszywie negatywna** **-** przewidywanie negatywne przypadku choroby serca, które zostało sklasyfikowane pozytywnie.

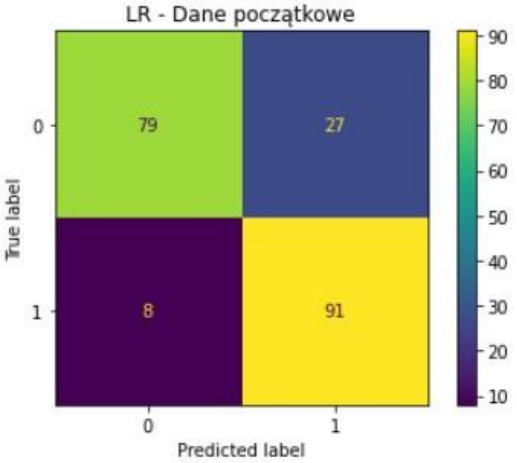
4) **TN - prawdziwie negatywna -** przewidywanie negatywne przypadku choroby serca, które zostało sklasyfikowane negatywnie.

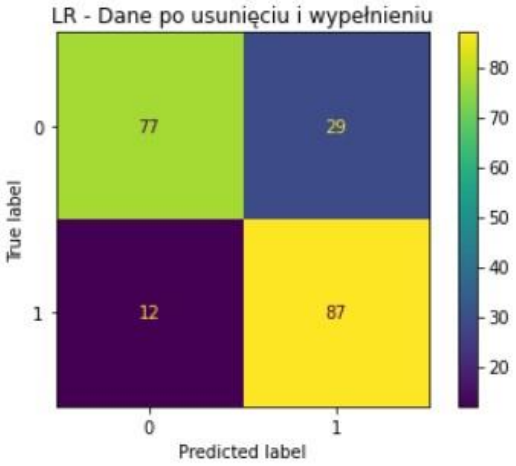
W naszym przypadku najbardziej zależy nam na uzyskaniu jak najmniejszych wyników w wartości **fałszywie negatywnej**, ponieważ wartość ta oznacza, że nie zostały stwierdzone choroby serca u pacjenta, który taką chorobę posiadał.

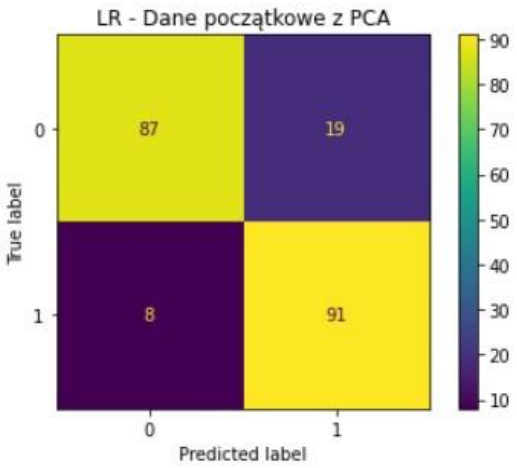
Support Vector Machine (SVM)

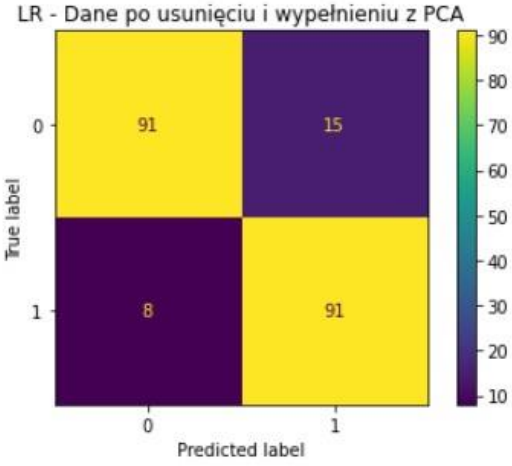


Regresja logistyczna (LR)

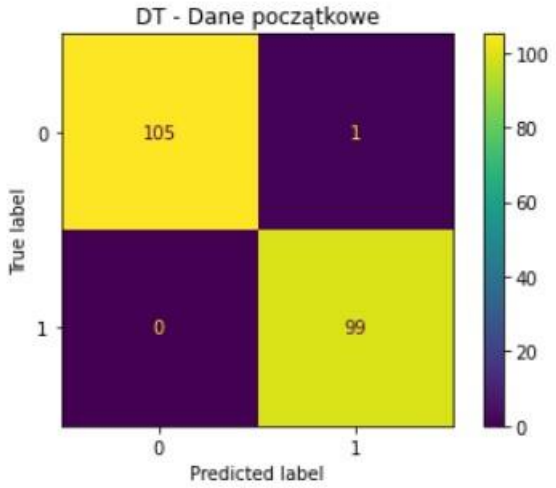


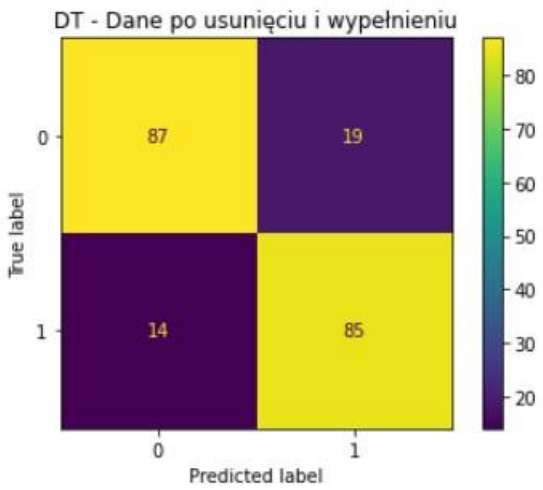


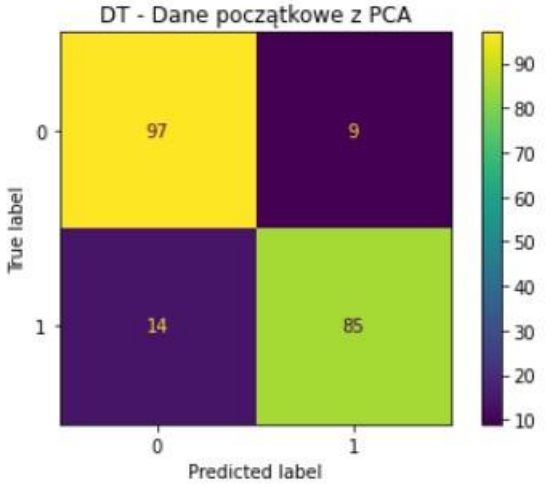


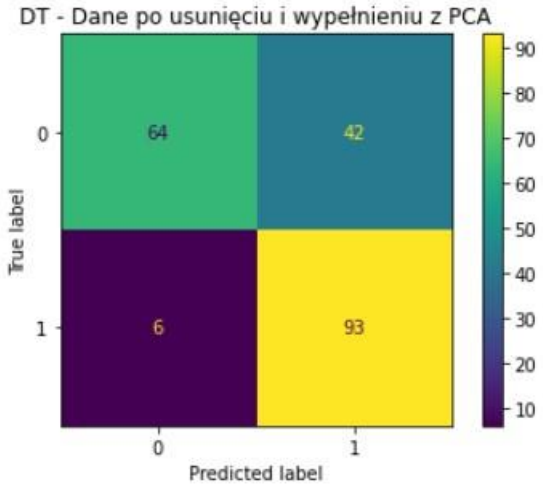


Drzewo decyzyjne (DT)

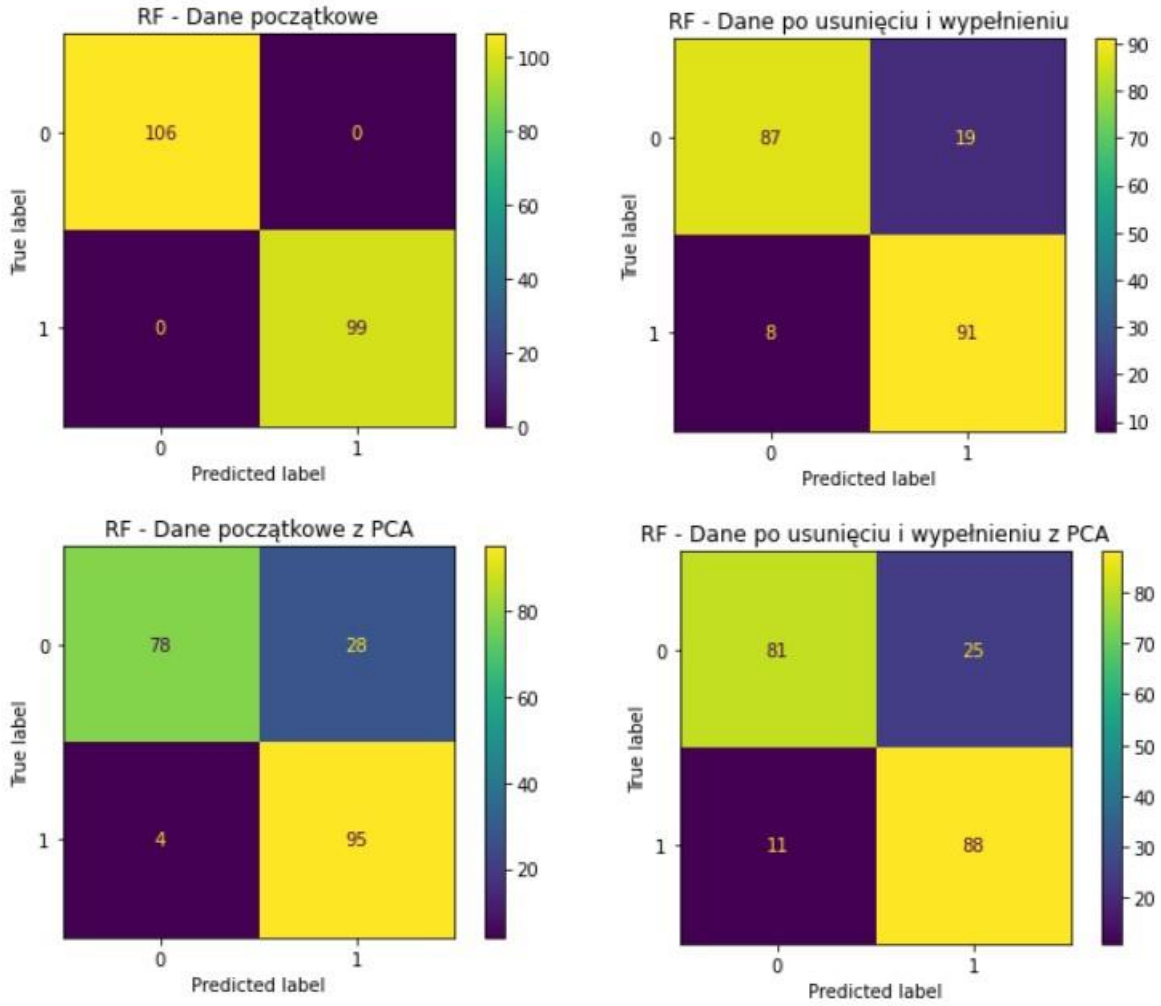








Las losowy (RF)



Najlepsze rezultaty - Najwięcej prawdziwie pozytywnych (TP) i prawdziwie negatywnych (TN)

Najsłabsze rezultaty - Najwięcej fałszywie pozytywnych (FP) i fałszywie negatywnych (FN)

TP TN : FP FN

Dane początkowe:

Najlepsze rezultaty: RF 205 : 0

Najsłabsze rezultaty: SVM 56 : 149 **Dane**

po usunięciu i wypełnieniu:

Najlepsze rezultaty: RF 178 : 27

Najsłabsze rezultaty: SVM 141 : 64 **Dane**

początkowe z PCA:

Najlepsze rezultaty: DT 182 : 23

Najsłabsze rezultaty: RF 173 : 32

Dane po usunięciu i wypełnieniu z PCA:

Najlepsze rezultaty: LR 182 : 23

Najsłabsze rezultaty: DT 157 : 48

Porównanie macierzy pomyłek

	Dane początkowe	Dane po usunięciu i wypełnieniu	Dane początkowe z PCA	Dane po usunięciu i wypełnieniu z PCA
Support Vector Machine (SVM)	**TP** 67 **FP** 39 **FN** 17 **TN** 82	**TP** 63 **FP** 43 **FN** 21 **TN** 78	**TP** 91 **FP** 15 **FN** 10 **TN** 89	**TP** 92 **FP** 14 **FN** 10 **TN** 89
Regresja logistyczna (LR)	**TP** 79 **FP** 27 **FN** 8 **TN** 91	**TP** 77 **FP** 29 **FN** 12 **TN** 87	**TP** 87 **FP** 19 **FN** 8 **TN** 91	**TP** 91 **FP** 15 **FN** 8 **TN** 91
Drzewo decyzyjne (DT)	**TP** 105 **FP** 1 **FN** 0 **TN** 99	**TP** 87 **FP** 19 **FN** 14 **TN** 85	**TP** 97 **FP** 9 **FN** 14 **TN** 85	**TP** 64 **FP** 42 **FN** 6 **TN** 93
Las losowy (RF)	**TP** 106 **FP** 0 **FN** 0 **TN** 99	**TP** 87 **FP** 19 **FN** 8 **TN** 91	**TP** 78 **FP** 28 **FN** 4 **TN** 95	**TP** 81 **FP** 25 **FN** 11 **TN** 88

Walidacja krzyżowa

Walidacja krzyżowa – polega na podziale danych na podzbiory, a następnie przeprowadza analizy na danych podzbiorach.

N_jobs – Zmienna ta oznacza liczbę zadań do równoległego uruchomienia. Wartość zmiennej została ustawiona na -1, co oznacza pracę na wszystkich wątkach.

RepeatedKFold – powtarza zgięcia n razy (Zależne od n_splits) z różną losowością w każdym powtórzeniu. n_repeats określa Ile razy należy powtórzyć walidację krzyżową.

Wykorzystana konfiguracja:

n_splits=10 n_repeats=3

Wyniki dla poszczególnych algorytmów klasyfikacji z walidacją krzyżową

	Dane początkowe	Dane po usunięciu i wypełnieniu	Dane po PCA	Dane po PCA po usunięciu i wypełnieniu
SVM	0.7008249254394314	0.6790691033695031	0.8399739831207564	0.8380449267085476
LR	0.8413224189352115	0.8338378069674471	0.8383368234025	0.8351100958182625
DT	0.8234818199124309	0.8140332508407895	0.8347706072720349	0.7795577130528587
RF	0.8835935021257693	0.8738371724094167	0.8685830319182688	0.8396789136366520

Z obserwacji można wywnioskować, że dla danych początkowych oraz danych po usunięciu sytuacja się nie zmienia (najgorzej wypada **SVM**, a najlepiej **algorytm lasu losowego**). Dla danych po PCA także najlepiej wypada **algorytm lasu losowego**, zaś najsłabiej **drzewo decyzyjne**. Dla danych po PCA i po usunięciu i wypełnieniu znowu wygrywa **algorytm lasu losowego**, a najsłabiej wypada **drzewo decyzyjne**. We wszystkich z wymienionych przypadków najlepiej wypada **algorytm lasu losowego**.

Porównanie wyników algorytmów klasyfikacji z wynikami algorytmów klasyfikacji i walidacją krzyżową

	Wyniki bez walidacji krzyżowej	Wyniki z walidacją krzyżową	Odchylenie standardowe	Różnica
SVM dane początkowe	0.7268292682926829	0.7008249254394314	0.047148513986350070	0.02600434285325148
SVM dane po usunięciu i wypełnieniu	0.6878048780487804	0.6790691033695031	0.042840507704366500	0.00873577467927733
SVM dane po PCA	0.8780487804878049	0.8399739831207564	0.030758160965539714	0.03807479736704844
SVM dane po PCA po usunięciu i wypełnieniu	0.8829268292682927	0.8380449267085476	0.033573167250241700	0.04488190255974511
LR dane początkowe	0.8292682926829268	0.8413224189352115	0.029483746533544363	0.01205412625228474
LR dane po usunięciu i wypełnieniu	0.8000000000000000	0.8338378069674471	0.043525862343727434	0.03383780696744709
LR dane po PCA	0.8682926829268293	0.8383368234025	0.028994240527609315	0.02995585952432922
LR dane po PCA po usunięciu i wypełnieniu	0.8878048780487805	0.8351100958182625	0.034339833502902390	0.05269478223051804
DT dane początkowe	0.824390243902439	0.8234818199124309	0.032103030948679200	0.00090842399000812
DT dane po usunięciu i wypełnieniu	0.8390243902439024	0.8140332508407895	0.038532112071474230	0.02499113940311292
DT dane po PCA	0.8878048780487805	0.8347706072720349	0.037796883387411300	0.05303427077674561
DT dane po PCA po usunięciu i wypełnieniu	0.7658536585365854	0.7795577130528587	0.036308231215436824	0.01370405451627332
RF dane początkowe	0.8731707317073171	0.8835935021257693	0.031676830684455065	0.01042277041845218

RF dane po usunięciu i wypełnieniu	0.8682926829268293	0.8738371724094167	0.036092423534410080	0.00554448948258745
RF dane po PCA	0.8439024390243902	0.8685830319182688	0.032402554951390630	0.02468059289387858
RF dane po PCA po usunięciu i wypełnieniu	0.8243902439024390	0.8396789136366520	0.032018514858490416	0.01528866973421305

W 9 na 16 przypadków wyniki bez walidacji krzyżowej okazały się lepsze, zaś z zastosowaniem walidacji krzyżowej 7 na 16 przypadków było dokładniejszych.

Zespołowa klasyfikacja

Zespołowa klasyfikacja – Jest to połączenie kilku klasyfikatorów i stworzenia jednego metaklasyfikatora z większą zdolnością uogólniania i stabilnością. Stosuje się ją, ponieważ dla pewnego rodzaju danych jeden klasyfikator może sobie radzić lepiej a inny gorzej.

Zespołowa klasyfikacja została przeprowadzona na czterech klasyfikatorach: Support Vector Machine (SVM), Regresja logistyczna (LR), Drzewo decyzyjne (DT) i Las losowy (RF).

Voting – Jeżeli opcja jest ustawiona na „hard" używa przewidywanych etykiet klas do głosowania według zasady większości. Jeśli jednak opcja jest ustawiona na „soft", przewiduje etykietę klasy na podstawie argumentów sum przewidywanych prawdopodobieństw.

Opcja „voting" w VotingClassifier została ustawiona na wartość „hard".

Weights – Określa wagę głosu dla każdego z algorytmów klasyfikacji.

Wyniki klasyfikacji zespołowej

```
          Bez walidacji      | Z walidacja krzyżowa|
-------------------------------------------------------------
VC data
Średnia: 0.8634146341463415 | 0.86803096643188 | +/-0.03649086134143738 | Wieksza o: 0.0046163322855385935
                              ^
VC deleted data
Średnia: 0.8195121951219512 | 0.8485151342090234 | +/-0.03468473601941711 | Wieksza o: 0.029002939087072188
                              ^
VC data with PCA
Średnia: 0.8926829268292683 | 0.8572022336442667 | +/-0.03256619894255366 | Wieksza o: 0.03548069318500158
^
VC deleted data with PCA
Średnia: 0.8097560975609757 | 0.8217812043911416 | +/-0.036247520085196924 | Wieksza o: 0.012025106830165888
                              ^
```

W trzech na cztery przypadki wyniki wychodzą lepsze z zastosowaniem walidacji krzyżowej.

Porównanie klasyfikatorów do klasyfikacji zespołowej (Bez walidacji krzyżowej)

```
Tabelka bez walidacji
Standardowe dane   | Usunięte dane      | Dane z PCA         | Usunięte dane z PCA
-------------------------------------------------------------------------------
SVM
0.7268292682926829 | 0.6780487804878049 | 0.8780487804878049 | 0.8
LR
0.8292682926829268 | 0.7804878048780488 | 0.8682926829268293 | 0.8048780487804879
DT
0.824390243902439 | 0.7902439024390244 | 0.8878048780487805 | 0.7463414634146341
RF
0.8731707317073171 | 0.8341463414634146 | 0.8439024390243902 | 0.824390243902439
=========
VC
0.8634146341463415 | 0.8195121951219512 | 0.8926829268292683 | 0.8097560975609757
```

Z tabelki można wywnioskować, że klasyfikacja zespołowa uzyskuje lepszy wynik w jednym na cztery przypadki, gdzie w trzech pozostałych wciąż dominuje algorytm lasu losowego.

Porównanie klasyfikatorów do klasyfikacji zespołowej (Z walidacji krzyżowej)

```
Tabelka z walidacją krzyżową
Standardowe dane   | Usunięte dane      | Dane z PCA         | Usunięte dane z PCA
-------------------------------------------------------------------------------
SVM
0.7008249254394314 | 0.6820039342597881 | 0.8399739831207564 | 0.8155974363855573
LR
0.8413319373056666 | 0.8211783742623261 | 0.8383368234025 | 0.8188336823402499
DT
0.8234818199124309 | 0.799076718065867 | 0.8347706072720349 | 0.7954692556634305
RF
0.8835935021257693 | 0.8680182752712735 | 0.8685830319182688 | 0.8230471476616538
=========
VC
0.86803096643188 | 0.8485151342090234 | 0.8572022336442667 | 0.8217812043911416
```

We wszystkich przypadkach algorytm lasu losowego okazał się najprecyzyjniejszy.

Część 3

Optymalizacja

Algorytmem, który został wybrany do optymalizacji jest Grid Search. Algorytm ten został wykorzystany na dwóch modelach uczenia maszynowego jakimi były Support Vector Machine (SVM) i Regresja Logistyczna (RL).

Grid Search – Jest to algorytm oparty o wyczerpujące wyszukiwanie określonych wartości parametrów dla estymatora. W GridSearchCV parametr scoring został ustawiony na celność (accuracy).

Dla Support Vector Machine parametrami były:

Kernel o wartościach:

- linear
- poly
- rbf

C o wartościach:

- 0.01
- 0.1
- 1
- 10
- 20

Dla Regresji Logistycznej parametrami były:

Solver o wartościach:

- newton-cg
- lbfgs
- liblinear
- sag
- saga

Penalty o wartościach:

- l1
- l2

C o wartościach:

- 0.01
- 0.1
- 1
- 10
- 20

W regresji logistycznej parametry „solver” i „penalty” nie współgrały ze sobą we wszystkich przypadkach przez co niektóre parametry „penalty” mogły nie działać z niektórymi parametrami „solver”.

Wybór algorytmu zależy od wybranego parametru „penalty”. Parametry „penalty”, które są obsługiwane przez parametry „solver”:

‘newton-cg’ - [‘l2’, ‘none’]

‘lbfgs’ - [‘l2’, ‘none’]

‘liblinear’ - [‘l1’, ‘l2’]

‘sag’ - [‘l2’, ‘none’]

‘saga’ - [‘elasticnet’, ‘l1’, ‘l2’, ‘none’]

Najlepsze wyniki i parametry

Support Vector Machine (SVM)

```
SVM
==Best parameters==
{'C': 0.01, 'kernel': 'linear'}
==Score==
0.8341463414634147
```

Regresja logistyczna (RL)

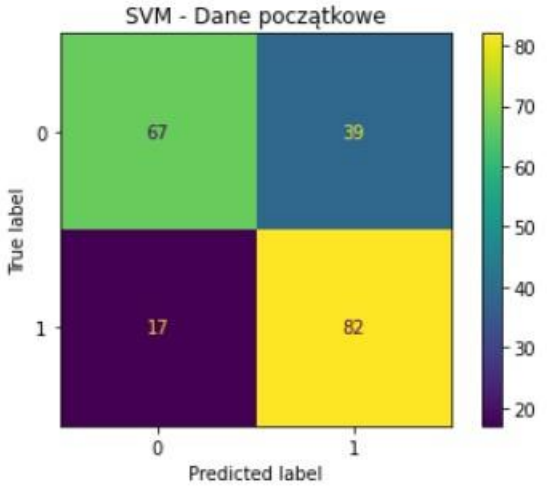
```
LR
==Best parameters==
{'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
==Score==
0.8512195121951219
```

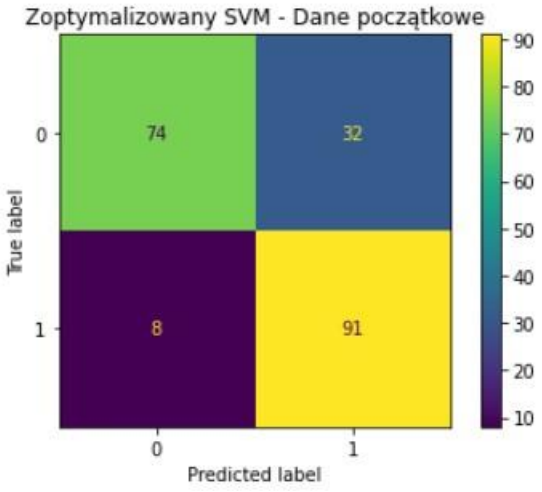
Tabela z wynikami przed i po optymalizacji

Standardowy	Zoptymalizowany	Poprawa wyniku o	Procent poprawy
SVM			
0.7268292682926829	0.8341463414634147	0.1073170731707318	14.7651 %
LR			
0.8292682926829268	0.8512195121951219	0.02195121951219514	2.64706 %

Macierz pomyłek na przykładzie SVM

Jako że w SVM procent poprawy wyszedł dużo lepszy niż w regresji logistycznej, przeanalizowaliśmy jego macierze pomyłek.





	SVM – Dane początkowe	Zoptymalizowany SVM – Dane początkowe
TP - prawdziwie pozytywna	67	74
FP - fałszywie pozytywna	39	32
FN - fałszywie negatywna	17	8
TN - prawdziwie negatywna	82	91

W naszym przypadku najbardziej zależało nam na uzyskaniu jak najmniejszych wyników w wartości **fałszywie negatywnej**. Z tabelki powyżej można wyczytać, że wartości **fałszywie negatywne** po optymalizacji się zmniejszyły, a co za tym idzie algorytm skutecznie zmniejszył ilość przewidywanych przypadków, w których u pacjenta nie została stwierdzona choroba serca, a taką chorobę posiadał. Przypadki spadły z **17** aż do **8**, co daje wynik aż **9** przypadków mniej.