Sprawozdanie z laboratorium: Uczenie maszynowe

Studium przypadku

17 czerwca 2020

Prowadzący: dr hab. inż. Maciej Komosiński

Autor: **Marcin Hradowicz** inf131767 ISWD marcin.hradowicz@student.put.poznan.pl

Zajęcia środowe, 15:10.

Oświadczam/y, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższych autora/ów, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.

1 Analiza danych

1.1 Opis danych

Dane składają się z 303 przypadków. Każdy posiada 14 atrybutów, z czego 13 to cechy, a pozostała kolumna to atrybut decyzyjny. Znaczenie każdego z atrybutów zostało przedstawione w Tabeli 1. Wszystkie atrybuty występują w postaci liczbowej, jednak część z nich jest tak naprawdę wartościami nominalnymi zakodowanymi w postaci liczb. Są to:

- *sex*,
- *cp*,
- *fbs*,
- *restecg*,
- *exang*,
- *slope*,
- *thal*,
- *target*.

Pozostałe są cechami liczbowymi-interwałowymi:

- *age*,
- *trestbps*,
- *chol*,
- *thalach*,
- *oldpeak*,
- *ca*.

Tabela 1: Opis atrybutów zawartych w zbiorze danych.

Nazwa atrybutu w zbiorze	**Rozwinięcie nazwy**	**Opis**
age	age	wiek
sex	sex	płeć 0 - kobieta 1 - mężczyzna
cp	chain pain type 4 values	rodzaj bólu w klatce piersiowej 0 - typowa angina 1 - atypowa angina (częściej u kobiet) 2 - ból nie wywołany przez anginę 3 - bez bólu
trestbps	resting blood pressure	spoczynkowe ciśnienie krwi w mm/Hg
chol	serum cholestoral in mg/dl	cholesterol w surowicy w miligramach na decylitr
fbs	fasting blood sugar >120 mg/dl	czy poziom cukru we krwi jest większy niż 120 mg/dl 0 - nie 1 - tak
restecg	resting electrocardiographic results (values 0,1,2)	spoczynkowe wyniki elektrokardiografii (badanie pracy serca) przyjmuje wartości: 0 - w normie 1 - nieprawidłowość ST-T 2 - prawdopodobny lub pewny przerost lewej komory
thalach	maximum heart rate achieved	maksymalne otrzymane tętno w uderzeniach na minutę
exang	exercise induced angina	występowanie anginy poprzez wysiłek fizyczny 0 - nie 1 - tak
oldpeak	oldpeak = ST depression induced by exercise relative to rest	obniżenie odcinka ST[9] wywołanego wysiłkiem fizycznym w porównaniu do stanu spoczynku
slope	the slope of the peak exercise ST segment	poziom nachylenia wierzchołka odcinka ST, wartości: 0 - wznoszące 1 - płaskie 2 - opadające
ca	number of major vessels (0-3) colored by flourosopy	liczba zabarwionych głównych naczyń krwionośnych metodą fluoroskopii [4]), wartości od 0 do 3
thal	thal: 3 = normal; 6 = fixed defect; 7 = reversable defect	poziom zwężenia ścian komór serca 3 - w normie 6 - stały efekt 7 - odwracalny defekt
target	target	atrybut decyzyjny 0 - brak choroby serca 1 - występuje choroba serca

1.2 Rozkład wartości atrybutów

W Tabeli 2 przedstawiono charakterystyki liczbowe cech. Nie widać w niej żadnej jasnej korelacji między jakimikolwiek atrybutami. Potwierdzają to także Tabele 3 oraz 4 przedstawiające korelację *Pearsona* [12] między atrybutami.

Tabela 2: Charakterystyka wartości atrybutów.

	mean	std	min	max
age	54.366	9.082	29.0	77.0
sex	0.683	0.466	0.0	1.0
cp	0.967	1.032	0.0	3.0
trestbps	131.624	17.538	94.0	200.0
chol	246.264	51.831	126.0	564.0
fbs	0.149	0.356	0.0	1.0
restecg	0.528	0.526	0.0	2.0
thalach	149.647	22.905	71.0	202.0
exang	0.327	0.47	0.0	1.0
oldpeak	1.04	1.161	0.0	6.2
slope	1.399	0.616	0.0	2.0
ca	0.729	1.023	0.0	4.0
thal	2.314	0.612	0.0	3.0

Tabela 3: Korelacja Pearson - część 1.

	age	sex	cp	trestbps	chol	fbs	restecg
age	1.0	-0.098	-0.069	0.279	0.214	0.121	-0.116
sex	-0.098	1.0	-0.049	-0.057	-0.198	0.045	-0.058
cp	-0.069	-0.049	1.0	0.048	-0.077	0.094	0.044
trestbps	0.279	-0.057	0.048	1.0	0.123	0.178	-0.114
chol	0.214	-0.198	-0.077	0.123	1.0	0.013	-0.151
fbs	0.121	0.045	0.094	0.178	0.013	1.0	-0.084
restecg	-0.116	-0.058	0.044	-0.114	-0.151	-0.084	1.0
thalach	-0.399	-0.044	0.296	-0.047	-0.01	-0.009	0.044
exang	0.097	0.142	-0.394	0.068	0.067	0.026	-0.071
oldpeak	0.21	0.096	-0.149	0.193	0.054	0.006	-0.059
slope	-0.169	-0.031	0.12	-0.121	-0.004	-0.06	0.093
ca	0.276	0.118	-0.181	0.101	0.071	0.138	-0.072
thal	0.068	0.21	-0.162	0.062	0.099	-0.032	-0.012

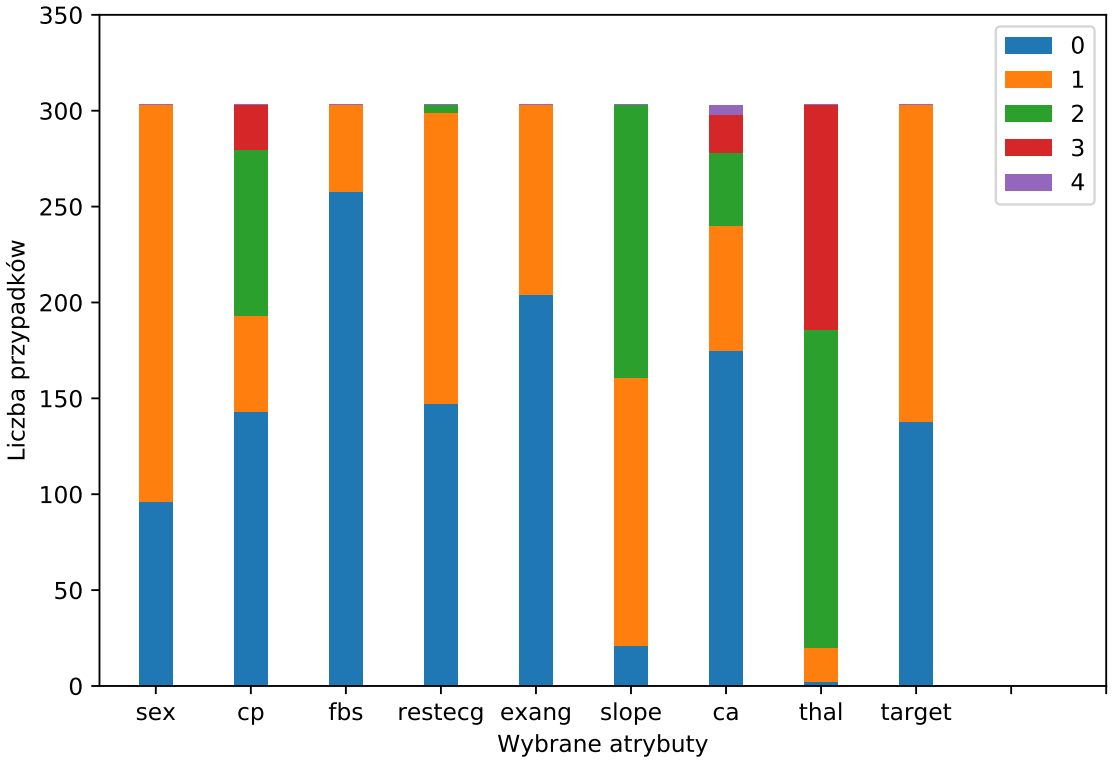
Tabela 4: Korelacja Pearson - część 2.

	thalach	exang	oldpeak	slope	ca	thal
age	-0.399	0.097	0.21	-0.169	0.276	0.068
sex	-0.044	0.142	0.096	-0.031	0.118	0.21
cp	0.296	-0.394	-0.149	0.12	-0.181	-0.162
trestbps	-0.047	0.068	0.193	-0.121	0.101	0.062
chol	-0.01	0.067	0.054	-0.004	0.071	0.099
fbs	-0.009	0.026	0.006	-0.06	0.138	-0.032
restecg	0.044	-0.071	-0.059	0.093	-0.072	-0.012
thalach	1.0	-0.379	-0.344	0.387	-0.213	-0.096
exang	-0.379	1.0	0.288	-0.258	0.116	0.207
oldpeak	-0.344	0.288	1.0	-0.578	0.223	0.21
slope	0.387	-0.258	-0.578	1.0	-0.08	-0.105
ca	-0.213	0.116	0.223	-0.08	1.0	0.152
thal	-0.096	0.207	0.21	-0.105	0.152	1.0

Liczności wartości atrybutów posiadających 5 lub mniej unikalnych wartości zostały przedstawione w Tabeli 5 oraz dla lepszego zwizualizowania rozkładu pomiędzy różne wartości także na Rys. 1.

Tabela 5: Liczności wartości atrybutów poniżej 6 unikalnych wartości.

Nazwa kolumny w zbiorze	**Liczności wartości**
sex	0: 96 1: 207
cp	0: 143 1: 50 2: 87 3: 23
fbs	0: 258 1: 45
restecg	0: 147 1: 152 2: 4
exang	0: 204 1: 99
slope	0: 21 1: 140 2: 142
ca	0: 175 1: 65 2: 38 3: 20 4: 5
thal	0: 2 1: 18 2: 166 3: 117
target	0: 138 1: 165



Rysunek 1: Rozkład wartości atrybutów poniżej 6 różnych wartości.

Po analizie atrybutu *target* można wywnioskować, że zbiór nie jest niezbalansowany, zawiera 138 przypadków klasy 0 (osoba zdrowa), co daje około 46% oraz 165 przypadków klasy 1 (osoba chora), czyli około 54%.

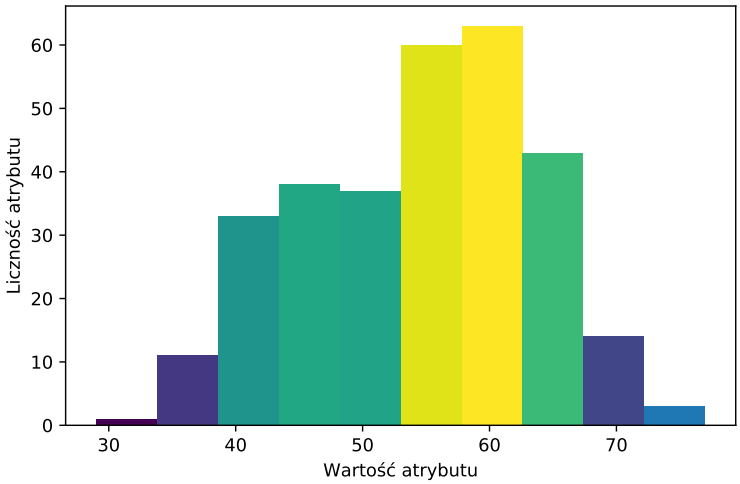
Atrybut *sex* wskazuje na przewagę liczby mężczyzn w zbiorze oznaczanych jako 1, niż kobiet oznaczanych jako 0.

Przy atrybucie *fbs* można zauważyć jeszcze większą dysproporcje wartości. Większość badanych osób ma poziom cukru we krwi mniejszy niż 120 mg/dl.

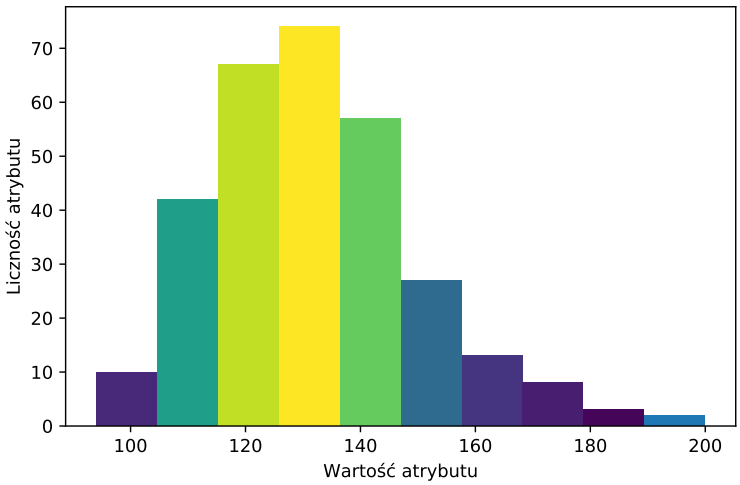
Wartości atrybutu *restecg* jest mniej więcej rozłożony między przypadki 0 oraz 1, przypadek 2 występuje sporadycznie.

Podobnie wartość 0 atrybutu *thal* rzadko występuje.

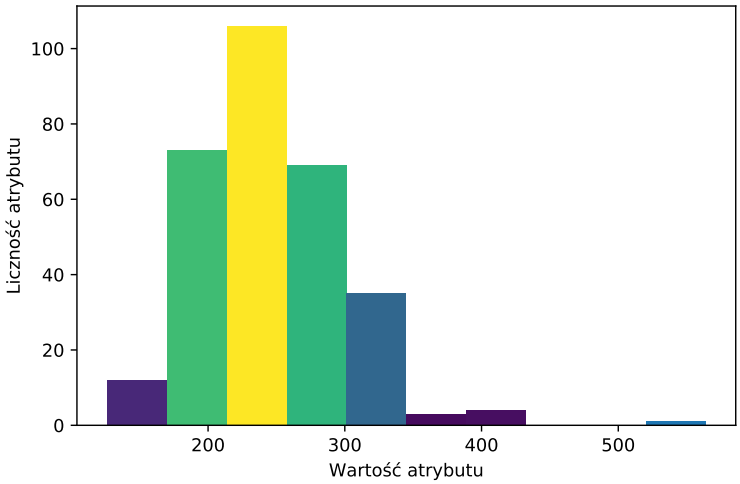
Na Rys. 2-6 przedstawiono rozkład wartości atrybutów liczbowych, których przedział wartości był większy niż 5. Histogramy zostały wykonane z użyciem `matplotlib.pyplot.hist` z ustawioną liczbą pojemników na 10, tzn. parametr `bins=10`. Dzieli on przedział wartości na 10 przedziałów o równych długościach zakresów. Dzięki temu można przyjrzeć się rozkładowi wartości atrybutów, jakie częściej wartości przyjmował, a jakie pojawiały się mniej. Dzięki temu z Rys. 2 można wywnioskować, że najwięcej przebadanych osób to około 60 latkowie, osoby powyżej 70 lat to mała część, a poniżej 25 nie ma żadnej osoby.



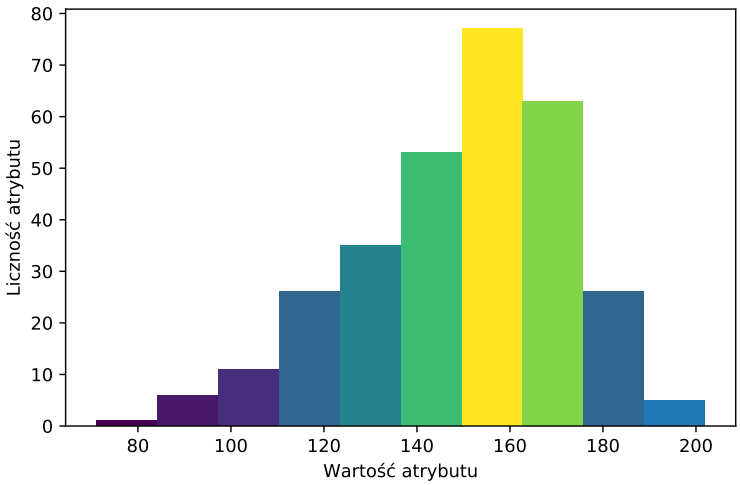
Rysunek 2: Rozkład wartości atrybutu *age*.



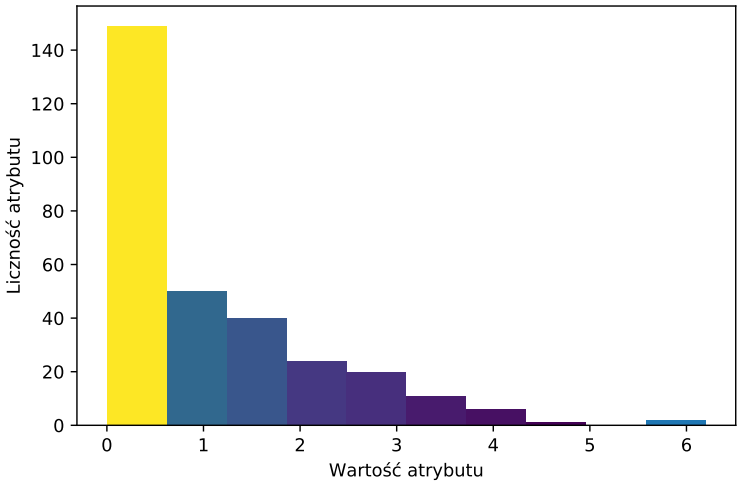
Rysunek 3: Rozkład wartości atrybutu *trestbps*.



Rysunek 4: Rozkład wartości atrybutu *chol*.



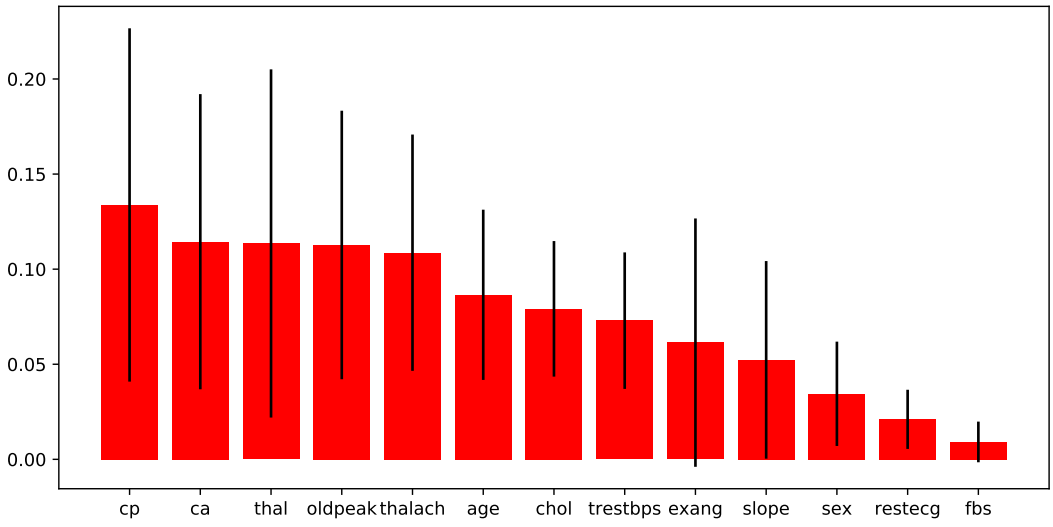
Rysunek 5: Rozkład wartości atrybutu *thalach*.



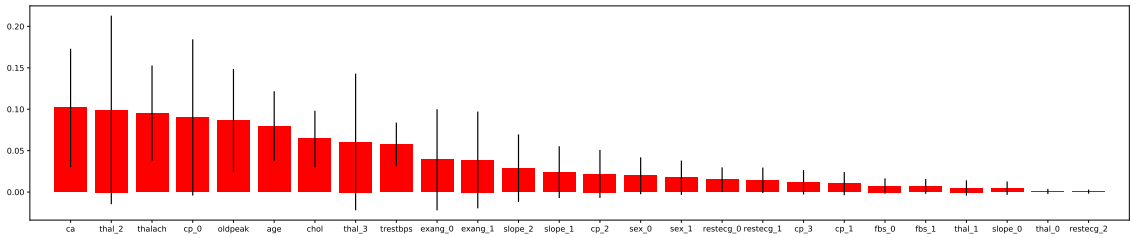
Rysunek 6: Rozkład wartości atrybutu *oldpeak*.

Na Rys. 7 oraz 8 przedstawiono także *feature importance* cech obliczonych z wykorzystaniem `sklearn.ensemble.RandomForestClassifier` zbudowanego z 250 drzew (parametr `n_estimators=250`). Zdecydowanie widać, że nie ma jasno określonych kilku cech, które odgrywają najważniejszą rolę. Można zaobserwować raczej tendencje do stopniowego spadku znaczenia atrybutów. Widoczne jest, że przykładowo płeć nie odgrywa znaczącej roli i choroby serca występują niezależnie od niej.

W zbiorze danych, w którym wykorzystano *one hot encoding* widać trochę więcej informacji. Dla danych nominalnych można wywnioskować, która dokładnie wartość cechy ma największe znaczenie. Przykładowo na Rys. 7 widać, że cecha *thal* jest dosyć istotna. Jednak z Rys. 8 możemy nawet powiedzieć, że cechy *thal_2* oraz *thal_3* są tymi ważniejszymi, gdzie *thal_1* i *thal_0* są bardzo mało istotne.



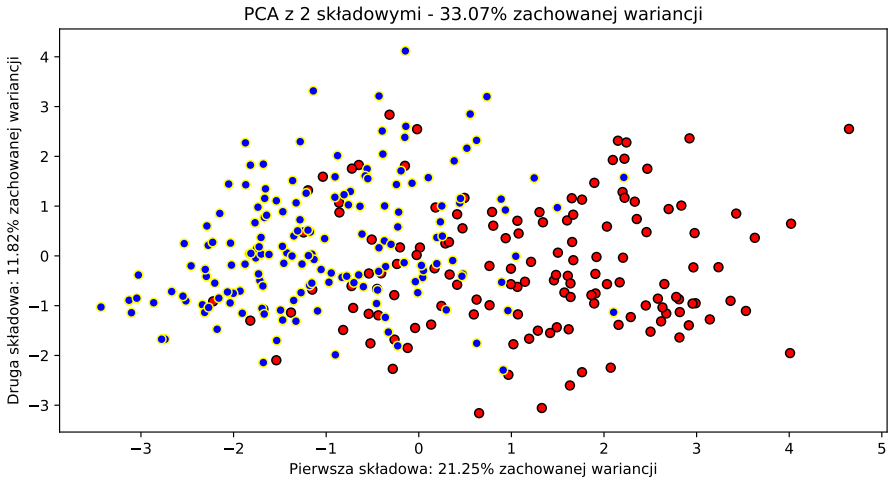
Rysunek 7: *Feature importance* oryginalnych danych.



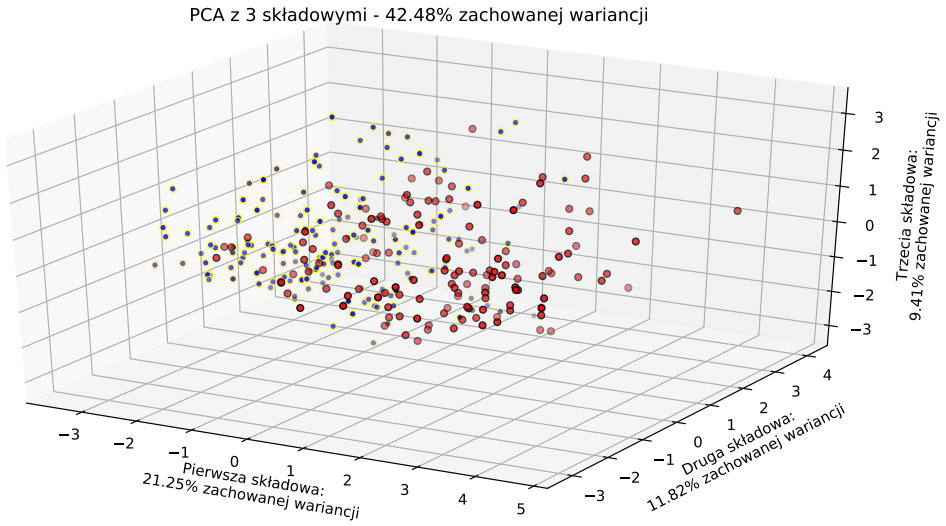
Rysunek 8: *Feature importance* zmodyfikowanych danych.

1.3 Przestrzeń przykładów

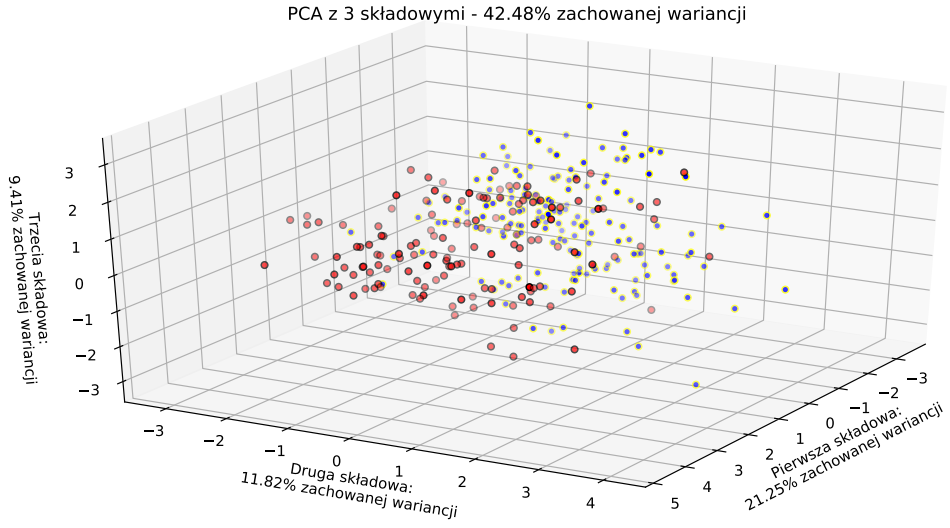
Dokonano przekształcenia przestrzeni przykładów na przestrzeń dwuwymiarową oraz trójwymiarową z wykorzystaniem algorytmu Analizy Głównych Składowych (ang. *Principal Component Analysis*)[11]. Z Rys. 9 można wywnioskować, iż dysponując atrybutami odpowiadającymi pierwszym dwóm składowym można dosyć dobrze za pomocą jednej prostej rozdzielić klasy decyzyjne otrzymując nie najgorsze wyniki. Jednak nie będą to wyniki idealne, ponieważ wiele przykładów nachodzi na siebie. Z Rys. 10 oraz 11 można wyciągnąć podobne wnioski. Jedna hiperpłaszczyzna całkiem dobrze może rozdzielić klasy, jednak poprzez nachodzenie części przykładów na siebie rozdzielenie to nie będzie idealne.



Rysunek 9: Przykłady w przestrzeni dwuwymiarowej.



Rysunek 10: Przykłady w przestrzeni trójwymiarowej - perspektywa 1.



Rysunek 11: Przykłady w przestrzeni trójwymiarowej - perspektywa 2.

2 Przekształcenie danych

Z powodu przedstawienia danych nominalnych – takich jak płeć – w postaci liczbowej zdecydowano się na zastosowanie *one hot encode*[21]. Skutkuje to rozłożeniem takiej kolumny odpowiadającej płci na dwie osobne kolumny odpowiadające kobiecie oraz mężczyźnie. Każdy przykład posiada wartość 1 dla odpowiadającej mu płci oraz 0 w pozostałych kolumnach w tym przypadku w kolumnie oznaczającej płeć przeciwną. Przykład przedstawiono w tabeli 6.

Tabela 6: Przykład użycia *one hot encode.*

ID przykładu	**sex_female**	**sex_male**
1	0	1
2	1	0
3	1	0

Stosuje się to, aby nie wprowadzać w błąd algorytmów ucznia maszynowego. Przykładowo mając kolumnę *kolor* z 5 różnymi wartościmai zakodowanymi liczbami od 1 do 5:

1. niebieski,
2. zielony,
3. czerwony,
4. czarny
5. biały.

Algorytm może ze zbioru uczącego wywnioskować, że wartość większa od 4 odpowiadająca kolorowi czarnemu zawsze daje jakąś określoną klasę decyzyjną, co nie musi być prawdą i wynika to tylko z kolejności zakodowaniu kolorów oraz danych w zbiorze uczącym. Wyindukowanie, że wartość 4 jest lepsza od wartości 3 w tym przypadku może być błędną decyzją.

Do utworzenia nowych kolumn na zasadzie opisanej przed chwilą dla atrybutów nominalnych zdecydowano użyć `sklearn.preprocessing.OneHotEncoder`[10]. Dodatkowo użyto `pandas.get_dummies`[5] do utworzenia nowych nazw powstałych kolumn. Atrybuty poddane temu procesowi to:

- *sex*,
- *cp*,
- *fbs*,
- *restecg*,
- *exang*,
- *slope*,
- *thal*.

Po transformacji uzyskujemy 13 nowych kolumn, które dla danego przypadku mogą przyjąć wartości tylko 0 lub 1. Przykładowo kolumna odpowiadająca za atrybut *restecg* przekształciła się w 3 kolumny *restecg_0*, *restecg_1* oraz *restecg_2*.

Ostatecznie zbiór składa się z 27 kolumn, wliczając w to atrybut decyzyjny. Każdy przykład w zbiorze zawiera wartość dla każdej cechy. Nie ma brakujących wartości, więc nie trzeba w tym zakresie wykonywać żadnych kroków.

3 Użycie algorytmów uczenia maszynowego

Wszystkie testy przeprowadzone zostały z użyciem `sklearn.model_selection.GridSearchCV`[6] z zastosowaną kroswalidacją na 10 zbiorach (ang. 10-fold cross-validation)[1]. Do obiektu `GridSearchCV` użyty został obiekt `sklearn.pipeline.Pipeline`[13] składający się z obiektu standaryzującego dane `sklearn.preprocessing.StandardScaler`[17] oraz wykorzystanego klasyfikatora. W celu porównania danych oryginalnych oraz przekształconych za pomocą *one-hot-encodingu* każdy klasyfikator został przetestowany na obu tych zbiorach, lecz należy zwrócić uwagę, że użycie zmienionych danych nie zawsze jest poprawne. Kodowanie odbywa się z wykorzystaniem całego zbioru danych, a nie tylko używając zbioru uczącego. Jest to niepoprawne, ponieważ w rzeczywistej sytuacji, może okazać się, iż w zbiorze testowym znajduje się przykład z całkiem nową wartością dla jakiegoś atrybutu, której nie było w zbiorze uczącym. Jednak w przypadku rozpatrywanego zbioru danych nie ma to większego znaczenia, ponieważ z góry znamy wszystkie wartości cech nominalnych. Jako główną miarę oceny jakości klasyfikatorów wykorzystano trafność (ang. *accuracy*), ponieważ klasy decyzyjne są dobrze zbalansowane. Dla każdego z klasyfikatorów zbiór został podzielony na 80% zbioru uczącego oraz 20% zbioru testowego z użyciem `sklearn.model_selection.train_test_split`[19] z tym samym ziarnem losowości `random_state=42`.

3.1 Drzewa decyzyjne

Użyto klasyfikatora `sklearn.tree.DecisionTreeClassifier`[3]. Strojonymi parametrami były:

- `citerion` - wybór miary wykorzystanej przy ocenie jakości podziału w węźle drzewa,
- `max_depth` - maksymalna liczba poziomów w drzewie.

Wyniki przedstawiono w Tabelach 7 oraz 8, a także w postaci heatmap na Rys. 12 i 13.

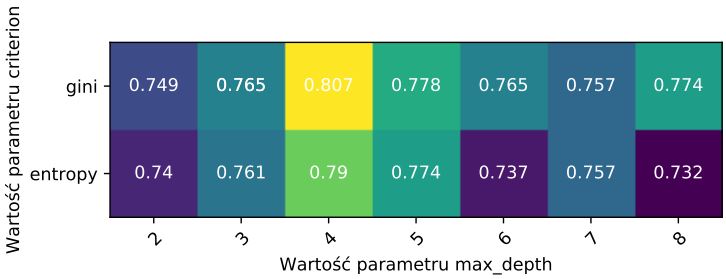
Tabela 7: Wyniki klasyfikatora `DecisionTreeClassifier` na oryginalnych danych.

criterion	max_depth	accuracy_mean	accuracy_std	roc_auc_mean	roc_auc_std	gmean_mean	gmean_std
gini	2	0.749	0.072	0.78	0.081	0.721	0.084
gini	3	0.765	0.099	0.825	0.089	0.738	0.124
gini	4	0.807	0.096	0.816	0.13	0.79	0.105
gini	5	0.778	0.098	0.779	0.114	0.762	0.112
gini	6	0.765	0.101	0.757	0.114	0.752	0.116
gini	7	0.757	0.093	0.762	0.097	0.748	0.107
gini	8	0.774	0.102	0.771	0.104	0.757	0.119
entropy	2	0.74	0.068	0.769	0.085	0.71	0.085
entropy	3	0.761	0.106	0.794	0.104	0.729	0.133
entropy	4	0.79	0.111	0.794	0.112	0.769	0.128
entropy	5	0.774	0.114	0.789	0.124	0.756	0.123
entropy	6	0.737	0.121	0.766	0.114	0.719	0.134
entropy	7	0.757	0.12	0.763	0.127	0.739	0.134
entropy	8	0.732	0.103	0.755	0.118	0.724	0.109

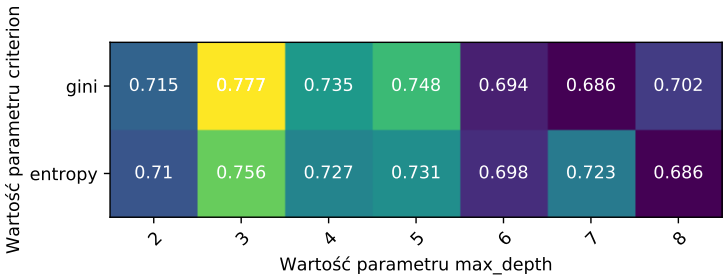
Tabela 8: Wyniki klasyfikatora `DecisionTreeClassifier` na zmienionych danych.

criterion	max_depth	accuracy_mean	accuracy_std	roc_auc_mean	roc_auc_std	gmean_mean	gmean_std
gini	2	0.715	0.048	0.799	0.066	0.689	0.07
gini	3	0.777	0.06	0.811	0.097	0.764	0.063
gini	4	0.735	0.078	0.754	0.08	0.721	0.082
gini	5	0.748	0.047	0.756	0.067	0.74	0.046
gini	6	0.694	0.059	0.713	0.066	0.688	0.056
gini	7	0.686	0.046	0.682	0.045	0.677	0.044
gini	8	0.702	0.048	0.699	0.051	0.692	0.057
entropy	2	0.71	0.054	0.793	0.084	0.682	0.078
entropy	3	0.756	0.049	0.795	0.07	0.74	0.05
entropy	4	0.727	0.068	0.757	0.04	0.709	0.071
entropy	5	0.731	0.044	0.74	0.056	0.719	0.047
entropy	6	0.698	0.038	0.723	0.044	0.692	0.039
entropy	7	0.723	0.053	0.73	0.04	0.716	0.052
entropy	8	0.686	0.08	0.697	0.065	0.677	0.079

Rysunek 12: Wyniki klasyfikatora `DecisionTreeClassifier` w postaci heatmapy na oryginalnych danych.



Rysunek 13: Wyniki klasyfikatora `DecisionTreeClassifier` w postaci heatmapy na zmienionych danych.



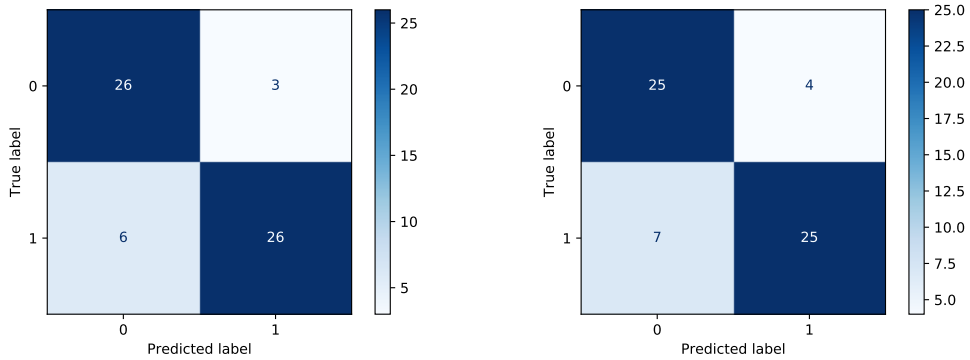
Najlepsze wyniki zostały osiągnięte dla klasyfikatorów, w których parametr `max_depth` został ustawiony na 3 lub 4. Zdecydowanie można uznać, że zbyt niska wartość tego parametru – jak 2 – lub zbyt wysoka powoduje pogorszenie wyników. Oznacza to, że klasyfikator prawdopodobnie ulega przeuczeniu w skutek zbyt dużego rozmiaru drzewa.

Tabela 9: Wyniki trafności na zbiorze testowym dla najlepszego uzyskanego modelu `DecisionTreeClassifier`.

Oryginalne dane		Zmienione dane	
Uczący	Testowy	Uczący	Testowy
0.807	0.852	0.777	0.819

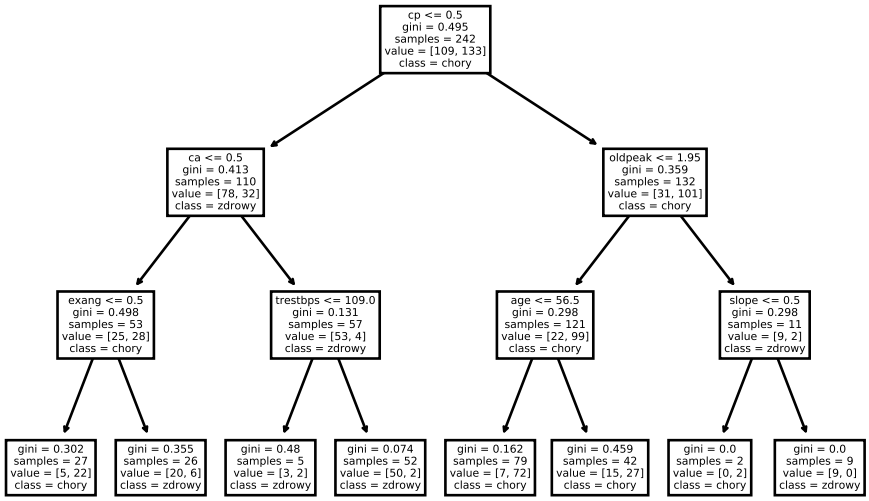
Z Tabeli 9 można wywnioskować, że na zmienionych danych drzewa decyzyjne radzą sobie trochę gorzej.

Rysunek 14: Macierz pomyłek dla najlepszego modelu klasyfikatora `DecisionTreeClassifier` na oryginalnym (lewo) i zmodyfikowanym (prawo) zbiorze.



Na Rys. 15 przedstawiono także przykładowe drzewo dla oryginalnego zbioru danych dla parametrów `criterion='gini'` i `max_depth='3'` zwizualizowane za pomocą `sklearn.tree.plot_tree`[14]. Dzięki temu można sprawdzić w jaki sposób drzewo decyzyjne klasyfikuje przykłady. Przykładowo jeżeli `cp > 0.5 i oldpeak > 1.95 i slope <= 0.5` to drzewo wnioskuje, iż osoba jest chora. Widać także, że wybrane przez drzewo cechy odpowiadają tym z największą wartością *feature importance* ukazaną na Rys. 7.

Rysunek 15: Wizualizacja drzewa.



3.2 Metoda wektorów wspierających - SVC

Użyto klasyfikatora `sklearn.svm.SVC`[18]. Strojonymi parametrami były:

- `C` - parametry regularyzacyjny, im mniejsza wartość tym większy margines, a także mniejsza szansa przeuczenia, co skutkuje gorszym dopasowaniem do danych uczących,
- `kernel` - funkcja jądrowa użyta w algorytmie.

Wyniki przedstawiono w Tabelach 10 oraz 11, a także w postaci heatmap na Rys. 16 i 17.

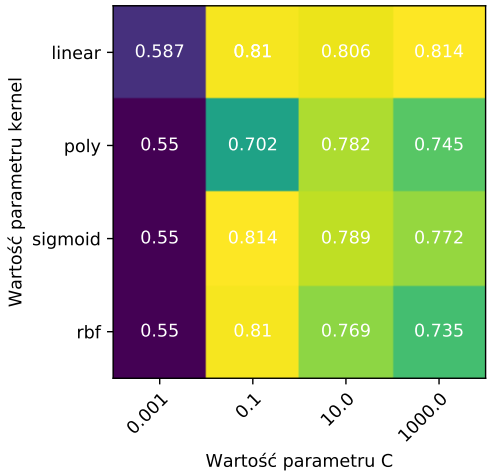
Tabela 10: Wyniki klasyfikatora `SVC` na oryginalnych danych.

C	kernel	accuracy_mean	accuracy_std	roc_auc_mean	roc_auc_std	gmean_mean	gmean_std
0.001	linear	0.587	0.031	0.904	0.061	0.218	0.185
0.001	poly	0.55	0.013	0.898	0.056	0.0	0.0
0.001	sigmoid	0.55	0.013	0.905	0.064	0.0	0.0
0.001	rbf	0.55	0.013	0.886	0.072	0.0	0.0
0.1	linear	0.81	0.05	0.89	0.066	0.794	0.051
0.1	poly	0.702	0.036	0.901	0.054	0.58	0.062
0.1	sigmoid	0.814	0.063	0.89	0.064	0.787	0.072
0.1	rbf	0.81	0.06	0.892	0.074	0.78	0.072
10.0	linear	0.806	0.072	0.889	0.062	0.796	0.072
10.0	poly	0.782	0.092	0.858	0.072	0.763	0.108
10.0	sigmoid	0.789	0.059	0.865	0.069	0.787	0.056
10.0	rbf	0.769	0.088	0.867	0.066	0.755	0.085
1000.0	linear	0.814	0.07	0.89	0.063	0.806	0.068
1000.0	poly	0.745	0.112	0.807	0.098	0.738	0.119
1000.0	sigmoid	0.772	0.065	0.863	0.077	0.768	0.069
1000.0	rbf	0.735	0.098	0.839	0.079	0.728	0.098

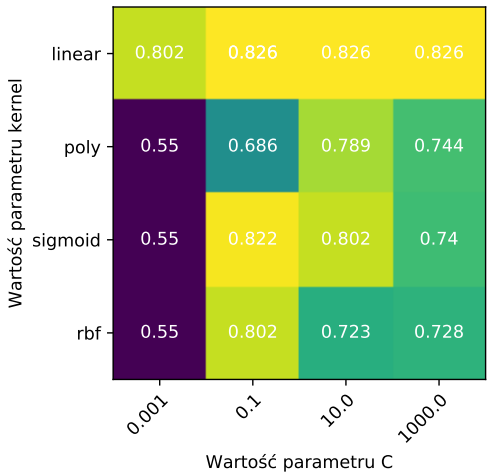
Tabela 11: Wyniki klasyfikatora `SVC` na zmienionych danych.

C	kernel	accuracy_mean	accuracy_std	roc_auc_mean	roc_auc_std	gmean_mean	gmean_std
0.001	linear	0.802	0.093	0.906	0.059	0.761	0.128
0.001	poly	0.55	0.013	0.88	0.073	0.0	0.0
0.001	sigmoid	0.55	0.013	0.907	0.059	0.0	0.0
0.001	rbf	0.55	0.013	0.89	0.066	0.0	0.0
0.1	linear	0.826	0.085	0.898	0.071	0.816	0.091
0.1	poly	0.686	0.053	0.886	0.074	0.518	0.193
0.1	sigmoid	0.822	0.08	0.902	0.066	0.812	0.085
0.1	rbf	0.802	0.095	0.891	0.067	0.772	0.115
10.0	linear	0.826	0.086	0.899	0.077	0.814	0.098
10.0	poly	0.789	0.058	0.856	0.074	0.782	0.059
10.0	sigmoid	0.802	0.084	0.87	0.07	0.797	0.088
10.0	rbf	0.723	0.104	0.843	0.08	0.71	0.115
1000.0	linear	0.826	0.086	0.899	0.077	0.814	0.098
1000.0	poly	0.744	0.045	0.844	0.046	0.738	0.046
1000.0	sigmoid	0.74	0.064	0.834	0.066	0.735	0.073
1000.0	rbf	0.728	0.063	0.848	0.057	0.712	0.081

Rysunek 16: Wyniki klasyfikatora `SVC` w postaci heatmapy na oryginalnych danych.



Rysunek 17: Wyniki klasyfikatora `SVC` w postaci heatmapy na zmienionych danych.



Jak widać dla obu przypadków danych najlepsze wyniki uzyskano dla liniowej funkcji jądrowej. Parametr regularyzacyjny nie miał, aż tak dużego wpływu na wyniki, jednak można zauważyć, że zbyt niska wartość prowadząca do powiększenia marginesu skutkuje zdecydowanym zwiększeniem liczby błędów.

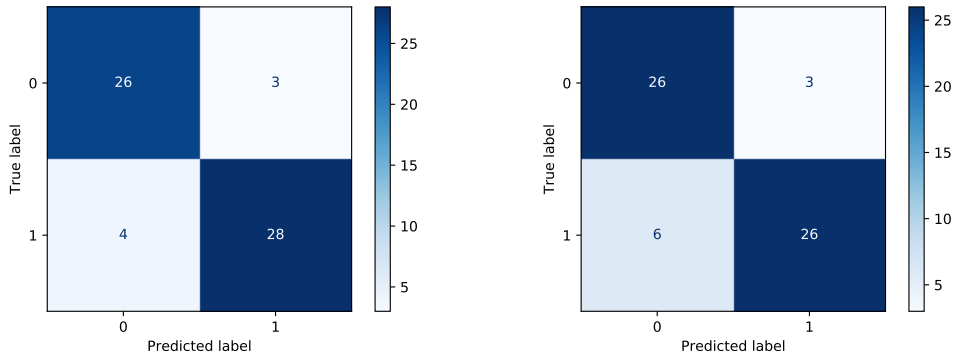
Tabela 12: Wyniki trafności dla najlepszego uzyskanego modelu `SVC`.

Oryginalne dane		Zmienione dane	
Uczący	Testowy	Uczący	Testowy
0.814	0.885	0.826	0.852

Wyniki w Tabeli 12 są lepsze na zbiorze testowym niż na uczącym. Tutaj znowu model

wyuczony na danych oryginalnych lepiej uogólnia nowe dane.

Rysunek 18: Macierz pomyłek dla najlepszego modelu klasyfikatora `SVC` na oryginalnym (lewo) i zmodyfikowanym (prawo) zbiorze.



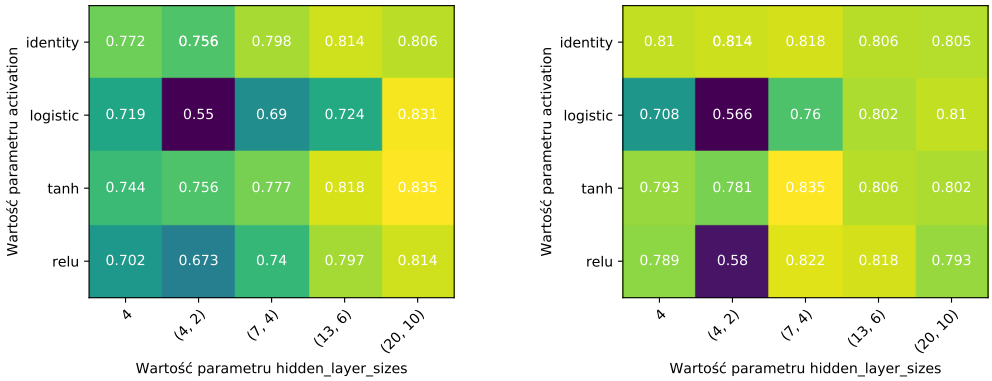
3.3 Sieci neuronowe - MLPClassifier

Użyto klasyfikatora `sklearn.neural_network.MLPClassifier`[8]. Strojonymi parametrami były:

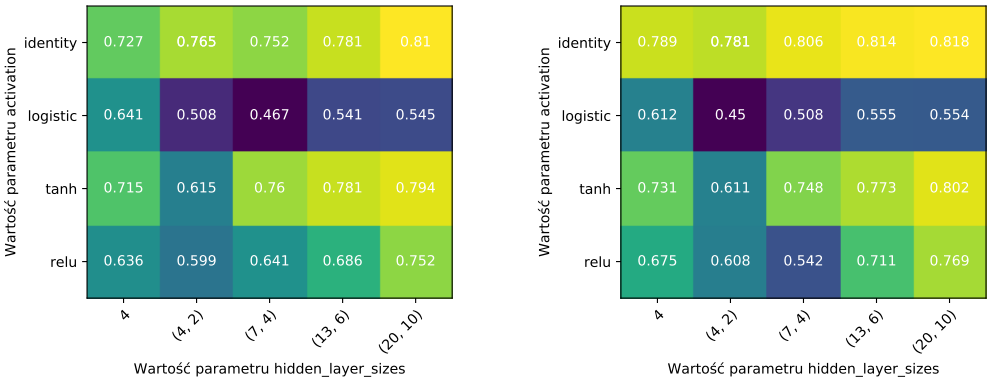
- `hidden_layer_sizes` - ustalenie liczby warstw i neuronów warstw ukrytych,
- `activation` - funkcja aktywacji,
- `solver` - metoda optymalizacji wag.

Wyniki przedstawiono w postaci heatmap na Rys. 19, 20 oraz 21.

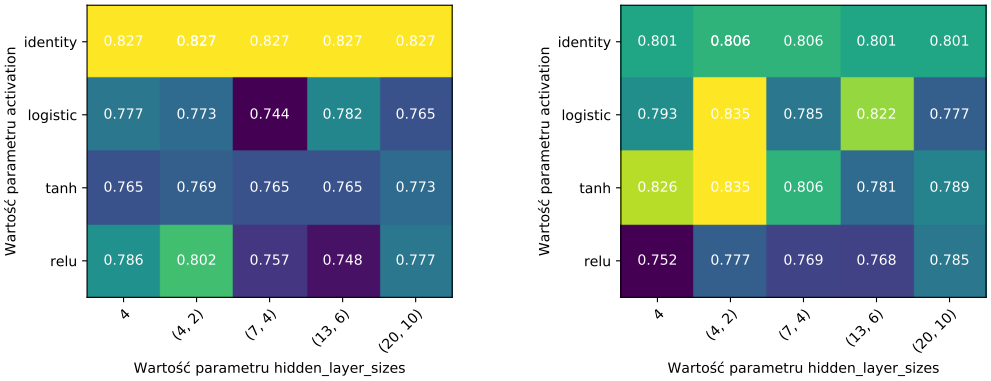
Rysunek 19: Wyniki klasyfikatora `SVC` w postaci heatmapy na oryginalnych (lewo) i zmienionych danych (prawo) dla `solver='adam'`.



Rysunek 20: Wyniki klasyfikatora `SVC` w postaci heatmapy na oryginalnych (lewo) i zmienionych danych (prawo) dla `solver='sgd'`.



Rysunek 21: Wyniki klasyfikatora `SVC` w postaci heatmapy na oryginalnych (lewo) i zmienionych danych (prawo) dla `solver='lbfgs'`.



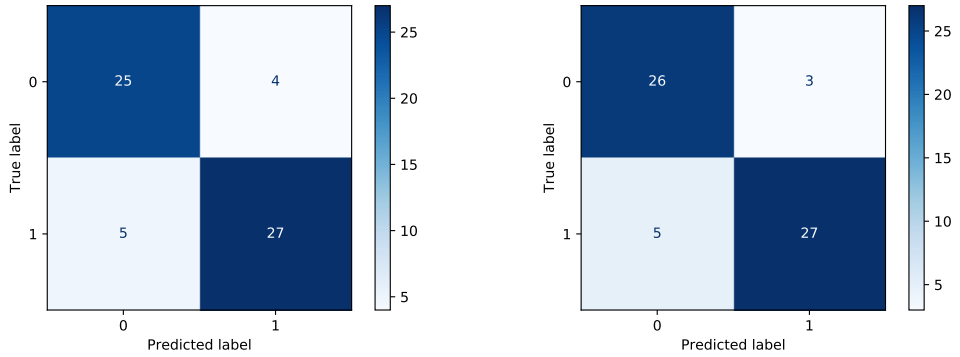
Można zauważyć, iż w większości przedstawionych heatmap dominują wyniki uzyskane za pomocą funkcji aktywacji *'identity'*, co oznacza, że sieć uprasza problem, ze względu na jego liniową charakterystykę. Jednym wyjątkiem jest funkcja aktywacji *'adam'*, która uzyskuje zbliżone, a niekiedy nawet lepsze wyniki. Jeżeli chodzi o wykorzystany rodzaj zbioru danych, to nie widać tutaj znaczących różnic.

Tabela 13: Wyniki trafności dla najlepszego uzyskanego modelu `MLPClassifier`.

Oryginalne dane		Zmienione dane	
Uczący	Testowy	Uczący	Testowy
0.835	0.852	0.835	0.869

Co do wyników z Tabeli 13 sytuacja wygląda podobnie dla obu zbiorów danych, jednak model utworzony ze zmodyfikowanego zbioru trochę lepiej poradził sobie ze zbiorem testowym, popełniając jeden błąd mniej.

Rysunek 22: Macierz pomyłek dla najlepszego modelu klasyfikatora `MLPClassifier` na oryginalnym (lewo) i zmodyfikowanym (prawo) zbiorze.



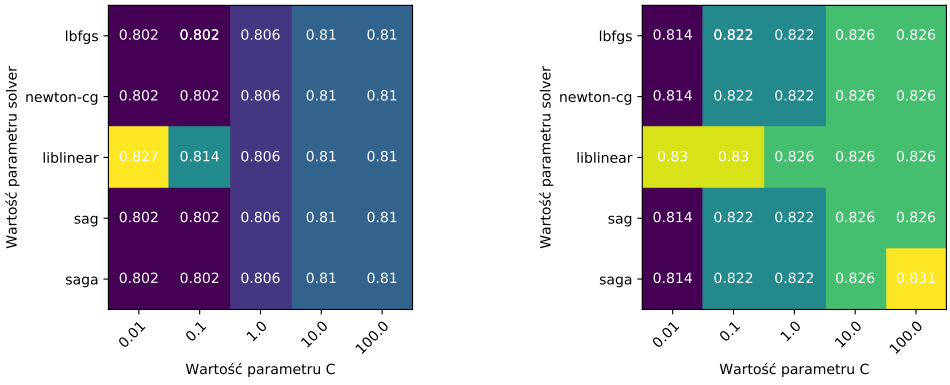
3.4 Regresja logistyczna - LogisticRegression

Wcześniej użyte modele naprowadzały, iż liniowe klasifikatory radzą sobie całkiem nieźle dlatego postanowiono spróbować klasyfikatora `sklearn.linear_model.LogisticRegression`[7]. Strojonymi parametrami były:

- `C` - parametry regularyzacyjny, im mniejsza wartość tym większa regularyzacja, a co za tym idzie mniejsza szansa przeuczenia,
- `solver` - algorytm używany do optymalizacji.

Wyniki przedstawiono w postaci heatmap na Rys. 23.

Rysunek 23: Wyniki klasyfikatora `LogisticRegression` w postaci heatmapy na oryginalnych (lewo) i zmienionych danych (prawo).



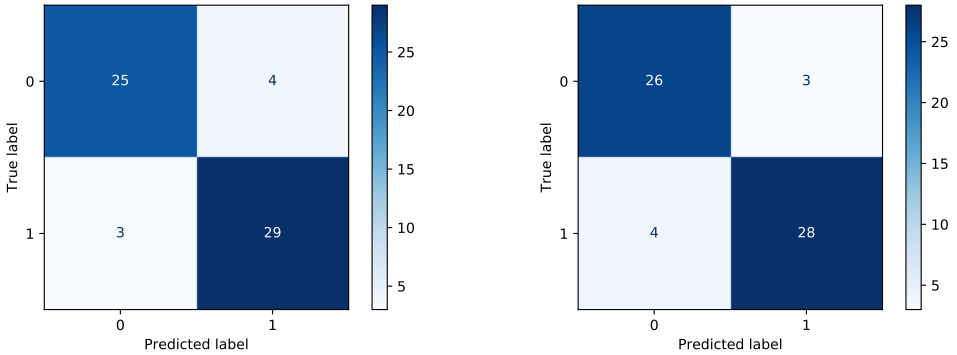
Rzeczywiście regresja logistyczna poradziła sobie całkiem dobrze. Na Rys. 23 widać, że radzi sobie bardzo dobrze niezależnie od parametrów. Ich wpływ na wyniki jest marginalny.

Tabela 14: Wyniki trafności dla najlepszego uzyskanego modelu `LogisticRegression`.

Oryginalne dane		Zmienione dane	
Uczący	Testowy	Uczący	Testowy
0.827	0.885	0.831	0.885

W Tabeli 14 widać, że klasyfikator poradził sobie bardzo dobrze dla obu przypadków danych.

Rysunek 24: Macierz pomyłek dla najlepszego modelu klasyfikatora `LogisticRegression` na oryginalnym (lewo) i zmodyfikowanym (prawo) zbiorze.



4 Wnioski

Tak jak zakładano od samego początku problem ten jest dosyć dobrze separowalny liniowo, co potwierdzają wyniki klasyfikatorów liniowych, które osiągały najwyższe wyniki. Niestety nie udało się osiągnąć trafności na poziomie 90%, co było powodem nachodzenia na siebie części przypadków z różnych klas decyzyjnych. Trzeba jednak brać pod uwagę, iż posiadany zbiór danych był niewielki i błąd rzędu 10% to tak naprawdę było mniej niż 10 przypadków.

Próbowano także skorzystać z klasyfikatorów złożonych w celu połączenia wszystkich nauczonych modeli. Wypróbowano `AdaBoostClassifier`[2], `VotingClassifier`[20] oraz `StackingClassfier`[16], `RandomForestClassifier`[15] jednak żaden z nich nie poprawił znacząco uzyskanych wcześniej wyników. Warto jednak wspomnieć, iż `RandomForestClassifier` uzyskiwał trafności na poziomie 0.885, a czasami nawet 0.9 co jest najwyższym uzyskanym wynikiem, gdzie pojedyncze drzewo `DecisionTreeClassifier` radziło sobie jednak gorzej.

Przeprowadzone eksperymenty udowodniły także, że dla tego zbioru danych używanie techniki *one-hot-encoding* nie miało zbyt dużego sensu. Wyniki poprawiały się jednak był to raczej niski zysk.

Literatura

[1] A Gentle Introduction to k-fold Cross-Validation. URL: https://machinelearningmastery.com/k-fold-cross-validation/.

[2] AdaBoostClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html.

[3] DecisionTreeClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

[4] Fluoroscopy. URL: https://en.wikipedia.org/wiki/Fluoroscopy.

[5] get_dummies. URL: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html.

[6] GridSearchCV. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[7] LogisticRegression. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[8] MLPCLassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[9] Odcinek ST. URL: https://pl.wikipedia.org/wiki/Odcinek_ST.

[10] One Hot Encoder. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

[11] PCA. URL: https://scikit-learn.org/stable/modules/decomposition.html#pca.

[12] Pearson Product-Moment Correlation. URL: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php.

[13] Pipeline. URL: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html.

[14] plot_tree. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html.

[15] RandomForestClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[16] StackingClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html.

[17] StandardScaler. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[18] SVC. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.

[19] train_test_split. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[20] VotingClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html.

[21] Jason Brownlee. Why One-Hot Encode Data in Machine Learning? URL: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.