Uniwersytet Rzeszowski
Kolegium Nauk Przyrodniczych
Instytut Informatyki

Sztuczna Inteligencja

Projekt

Model przewidujący długość życia w kraju

Prowadzący:
Dr inż. Jacek Bartman

Autor:
Jakub Flis
117794

Rzeszów 2024

Spis treści

1. Temat i cel projektu

Model przewidujący średnią długość życia w oparciu o wskaźniki dla kraju.

Celem projektu jest zaprojektowanie i stworzenie sieci neuronowej, która wykorzysta przygotowane dane, by przewidzieć długość życia w kraju. W oparciu o model zostaną wyciągnięte wnioski dotyczące parametrów modelu, jego skuteczności oraz jego predykcji.

2. Zmienna decyzyjna

- Lifeexpectancy (przewidywana długość życia)

Zmienna decyzyjna liczbowa, przedział 36.3 – 89, 10 brakujących wartości

Brakujące wartości zostały usunięte.

Zmienna została zakodowana do 6 przedziałów (35,45,55,…) przy użyciu logiki rozmytej i liniowej funkcji przynależności, gdzie X5 jest środkiem każdego przedziału z wartością przynależności 1.

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0.5100	0.5100	0.5500	0.5800	0.6200	0.6400
4	1	0.4900	0.4900	0.4500	0.4200	0.3800	0.3600
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0

Podstawowym problemem takiego podejścia jest to, że po wprowadzeniu klas baza danych stała się niezbalansowana. Oto sumy przynależności do każdego przedziału.

```
0.0036    0.1314    0.4311    0.7635    1.2656    0.3221
```

Pomimo słabości pierwszej klasy, nie zostały podjęte żadne kroki. Rozważano połączenie dwóch pierwszych przedziałów, ale to z kolei uniemożliwiłoby poprawne zakodowanie i zdekodowanie wartości regresji. Możliwe że zastosowanie oversamplingu byłoby skuteczne, ale zostało wstępnie odrzucone, ze względu na ryzyko overfittingu modelu.

Zastosowanie logiki rozmytej sprawia, że nadal mamy do czynienia z problemem regresji.

Główną zaletą tego podejścia jest łatwość w imputacji danych: można precyzyjniej wyznaczyć średnią wartość dla każdej klasy i obiektów do nich należących,

W ostatecznym rozrachunku takie podejście nie przynosi znaczących zalet – problem nadal pozostaje problemem regresji, a końcowe przedstawienie go jako klasyfikacja do przedziałów wciąż jest bezcelowe.

```
function coded = fuzzify(col)

    minm = min(col)-10;
    maxm = max(col);

    nrOfRanges = floor((maxm - minm)/10);
    coded = zeros(nrOfRanges, length(col));

    minm = floor(minm/10) * 10 + 5;
    for nr = 1:nrOfRanges
        minm = minm+10;
        for i = 1:size(col)
            coded(nr, i) = max(1 - abs(0.1*(col(i)-minm)), 0);
        end
    end
end
```

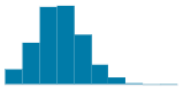
Powyżej przedstawiono funkcję kodującą kolumnę decyzyjną.

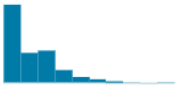
3. Opis bazy danych

Źródło: https://www.kaggle.com/datasets/augustus0498/life-expectancy-who

Country Country	Year Year	Status Developed or Developing status	Lifeexpectancy Life Expectancy in age	AdultMortality Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
193 unique values	2000 2015	Developing 83% Developed 17%	36.3 89	1 723
Afghanistan	2015	Developing	65	263
Afghanistan	2014	Developing	59.9	271
Afghanistan	2013	Developing	59.9	268
Afghanistan	2012	Developing	59.5	272
Afghanistan	2011	Developing	59.2	275

infantdeaths Number of Infant Deaths per 1000 population	Alcohol Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)	percentageexpe... Expenditure on health as a percentage of Gross Domestic Product per capita(%)	HepatitisB Hepatitis B (HepB) immunization coverage among 1-year-olds (%)	Measles Measles - number of reported cases per 1000 population
0 1800	0.01 17.9	0 19.5k	1 99	0 212k
62	0.01	71.27962362	65	1154
64	0.01	73.52358168	62	492
66	0.01	73.21924272	64	430
69	0.01	78.1842153	67	2787
71	0.01	7.097108703	68	3013

# BMI Average Body Mass Index of entire population	# under-fivedeaths Number of under-five deaths per 1000 population	# Polio Polio (Pol3) immunization coverage among 1-year-olds (%)	# Totalexpenditure General government expenditure on health as a percentage of total government expenditure (%)	# Diphtheria Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
 1 87.3	 0 2500	 3 99	 0.37 17.6	 2 99
19.1	83	6	8.16	65
18.6	86	58	8.18	62
18.1	89	62	8.13	64
17.6	93	67	8.52	67
17.2	97	68	7.87	68

# HIV/AIDS Deaths per 1 000 live births HIV/AIDS (0-4 years)	# GDP Gross Domestic Product per capita (in USD)	# Population Population of the country	# thinness1-19years Prevalence of thinness among children and adolescents for Age 10 to 19 (%)	# thinness5-9years Prevalence of thinness among children for Age 5 to 9(%)
 0.1 50.6	 1.68 119k	 34 1.29b	 0.1 27.7	 0.1 28.6
0.1	584.25921	33736494	17.2	17.3
0.1	612.696514	327582	17.5	17.5
0.1	631.744976	31731688	17.7	17.7
0.1	669.959	3696958	17.9	18

# Incomecompositi... Human Development Index in terms of income composition of resources (index ranging from 0 to 1)	# Schooling Number of years of Schooling(years)
 0 0.95	 0 20.7
0.479	10.1
0.476	10
0.47	9.9
0.463	9.8
0.454	9.5

Liczba rekordów: 2938

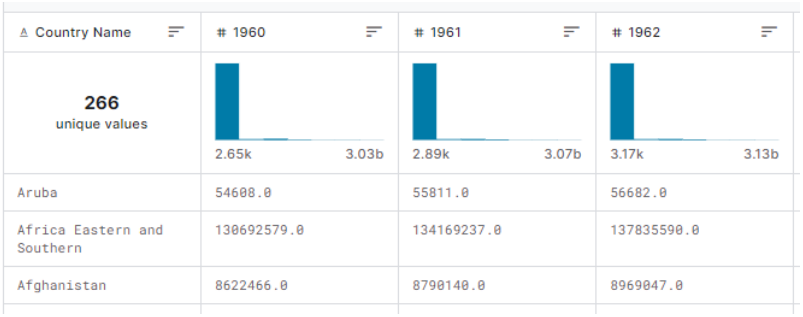
Liczba atrybutów niezależnych: 21

Liczba atrybutów zależnych: 1 (life expectancy – przewidywana dł. życia)

Liczba atrybutów kategorialnych: 1 (status – kraj rozwinięty lub nie)

Powyższa baza danych zawiera bardzo niepoprawne dane w kolumnie Population. Konieczne było ręczne uzupełnienie tej kolumny przy pomocy danych z poniższego źródła.

Źródło danych na temat liczby ludności w krajach:
https://www.kaggle.com/datasets/ayushparwal2026/country-population-from-1960-to-2022



Country Name	# 1960	# 1961	# 1962
266 unique values	2.65k – 3.03b	2.89k – 3.07b	3.17k – 3.13b
Aruba	54608.0	55811.0	56682.0
Africa Eastern and Southern	130692579.0	134169237.0	137835590.0
Afghanistan	8622466.0	8790140.0	8969047.0

4. Opis kolumn i przetworzenie danych

- Country (kraj)

Zmienna kategorialna, 193 unikalne wartości, 0 brakujących wartości.

Kolumna została pominięta, by zwiększyć zdolność modelu do uogólnienia problemu, zamiast przeuczania go do rozpoznawania wzorców dla konkretnych krajów. Ponadto wśród danych występuje wiele innych informacji, które powinny oddawać pełny obraz każdego kraju.

- Year (rok)

Zmienna liczbowa, przedział 2000-2015, 0 brakujących wartości, brak wartości odstających.

Wartości z przedziału 2000-2015 pomniejszono o 2000, by ułatwić dalszą normalizację.

- Status

Zmienna binarna – kraj rozwijający się lub rozwinięty, 0 brakujących wartości

Kolumna została zakodowana binarnie.

- AdultMortality (śmiertelność dorosłych)

Prawdopodobieństwo śmierci między 15 a 60 rokiem życia na 1000 mieszkańców

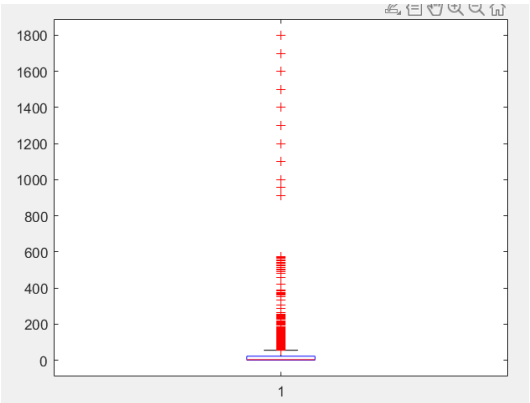
Zmienna liczbowa, przedział 1-723, 10 brakujących wartości (1%), istnieją dane lekko odstające.

10 instancji brakujących usunięto. Wartości odstające nie zostały usunięte.

- infantdeaths (śmierci noworodków)

Liczba śmierci noworodków na 1000 mieszkańców

Zmienna liczbowa, przedział 0 – 1800, 0 brakujących wartości. Występują wartości odstające, ale nie zostały one usunięte.



- Alcohol

Spożycie alkoholu w litrach na osobę 15+.

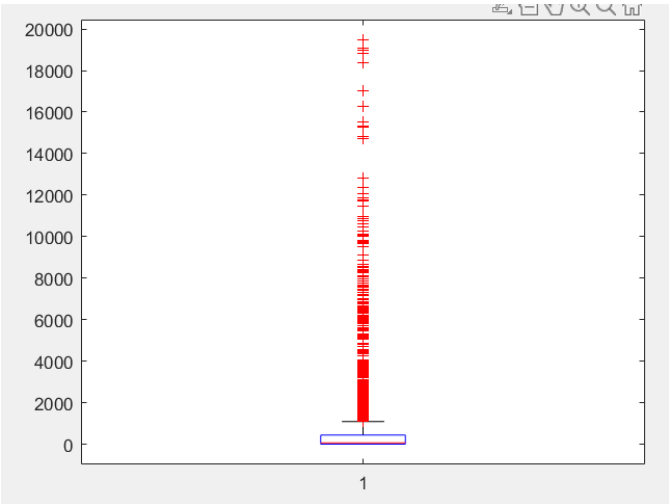
Zmienna liczbowa, przedział 0.01 - 17.9, 179 brakujących wartości (6%), brak odstających wartości.

6% to umiarkowana liczba braków. Ze względu na niewielkie odchylenie standardowe zmiennej (około 4), możliwe jest uzupełnienie braków poprzez wstawienie średniej z klas przynależnych.

- percentageexpenditure (procent wydatku)

Jaki procent GDP jest przeznaczane na służbę zdrowia.

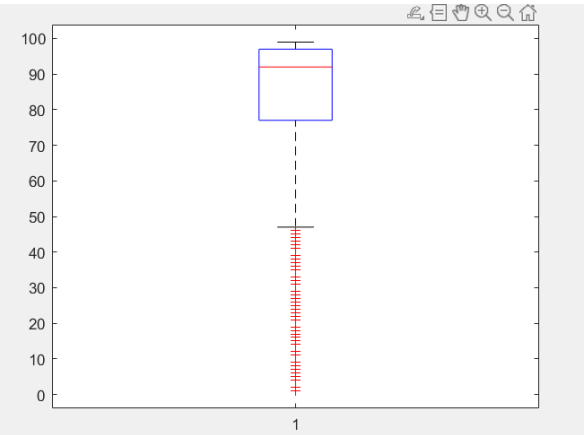
Zmienna liczbowa, przedział 0-19.5k, średnia 738, 0 brakujących wartości, dużo odstających wartości, których nie usunięto.



- HepatitisB (wirusowe zapalenie wątroby typu B)

Procent 1-latków zaszczepionych na WZW B.

Zmienna liczbowa, przedział 1-99, 553 brakujące wartości (19%), istnieją dane lekko odstające.



Przy tak wysokiej liczbie brakujących rekordów potrzebna była skuteczniejsza metoda imputacji danych. W projekcie zastosowano imputację poprzez regresję z najbardziej skorelowaną kolumną (Diphtheria – 0.59).

Dane odstające nie zostały usunięte.

```
missingIndices = ismissing(tab.HepatitisB);


X_train = tab.Diphtheria(~missingIndices);
y_train = tab.HepatitisB(~missingIndices);
X_test = tab.Diphtheria(missingIndices);

model = fitlm(X_train, y_train);
y_pred = predict(model, X_test);
tab.HepatitisB(missingIndices) = y_pred;

reg_test = predict(model, X_train);

err = mape(y_train, reg_test)
```

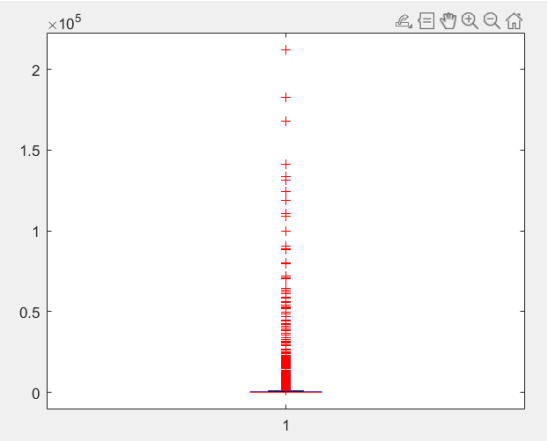
Dla powyżej zrealizowanej regresji wartości brakujących w kolumnie HepatitisB wyznaczono wartość MAPE 17,7. Jako że jest to wartość poniżej 20, jest to stosunkowo dobry model.

```
err =

   17.7094
```

- Measles (świnka)

Liczba chorych na świnkę na 1000 mieszkańców.

Zmienna liczbowa, przedział 0-212k, średnia 2.42k, 0 brakujących wartości, dane bardzo mocno odstające, których nie usuwano.



- BMI

Przeciętne BMI w populacji.

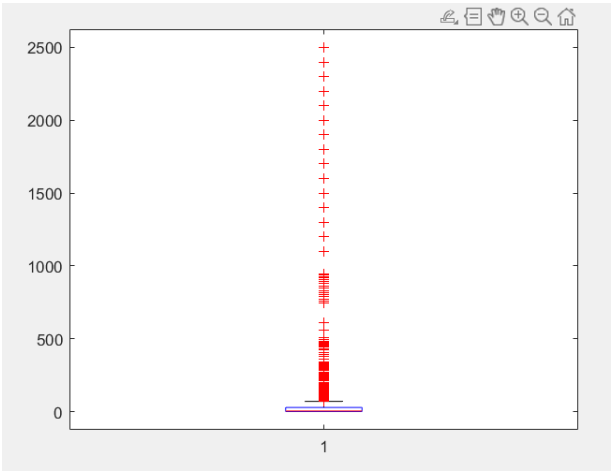
Zmienna liczbowa, przedział 1 – 87.3, 34 brakujące wartości (1%), brak wartości odst.

Brakujące wartości usunięto.

- under-fivedeaths (śmierci poniżej 5 lat)

Liczba śmierci poniżej 5. roku życia na 1000 mieszkańców.

Zmienna liczbowa, przedział 0-2500, 0 brakujących wartości, wartości mocno odstające, których nie usuwano.



- **Polio**

Procent zaszczepionych 1-latków na polio.

Zmienna liczbowa, przedział 3-99, 19 brakujących wartości (1%), występują wartości odstające.

Brakujące wartości usunięto. Wartości odstających nie modyfikowano.

- **Totalexpenditure (całkowity wydatek)**

Procent budżetu państwa przeznaczany na służbę zdrowia.

Zmienna liczbowa, przedział 0.37-17.6, 226 brakujących wartości (8%), dane lekko odstające.

Brakujące wartości uzupełniono średnimi z przynależnych klas. Wartości odstających nie zmieniano.

- Diphtheria

Procent wyszczepienia 1-latków na tężec.

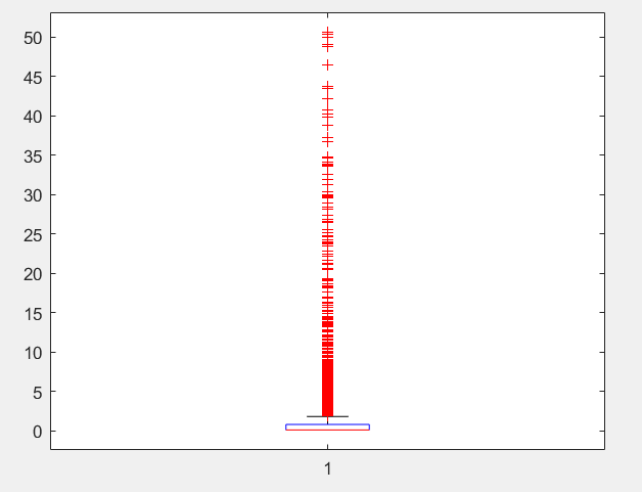
Zmienna liczbowa, przedział 2-99%, 19 brakujących wartości (1%), istnieją dane lekko odst.

Brakujące wartości usunięto. Danych odstających nie modyfikowano.

- HIV/AIDS

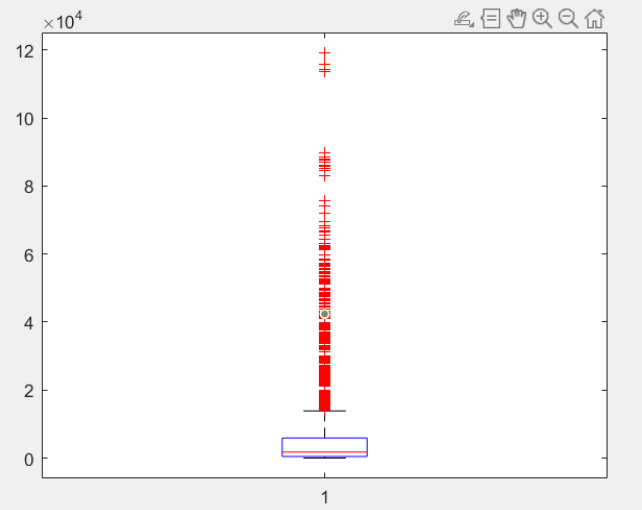
Śmierci 0-4-latków na AIDS na 1000 narodzin.

Zmienna liczbowa, przedział 0.1-50.6, 0 brakujących wartości, dane mocno odstające, ale niezmieniane.



- GDP

Zmienna liczbowa, przedział 1.68-119k, 448 brakujących wartości (15%), średnia 7.48k, istnieją wartości znacznie odstające, ale nie zmieniono ich.

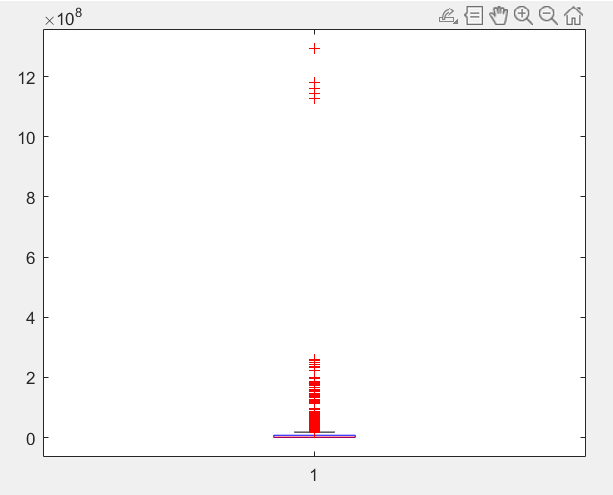


Dane próbowano uzupełnić z użyciem regresji z kolumną totalexpenditure (kor. 0.8). Ze względu na charakter danych, nie powiodło się to. Użyto imputacji średnimi z klas.

- **Population**

Liczba mieszkańców kraju.

Zmienna liczbowa, przedział 34 - 1.29b, średnia 12.8m, 652 brakujących wartości (22%), istnieją wartości bardzo silnie odstające. Wartości odstające są rzeczywiste, więc nie podlegają zmianom.



Kolumna zawierała bardzo uszkodzone dane. Zostały one uzupełnione ręcznie.

- **thinness1-19years**

Procent niedożywienia wśród 1-19-latków.

Zmienna liczbowa, przedział 0.1 – 27.7, 34 brakujące wartości (1%), istnieją wartości lekko odstające.

Wartości brakujące usunięto.

- thinness5-9years

Procent niedożywienia wśród 5-9 latków.

Zmienna liczbowa, przedział 0.1 – 28.6, 34 brakujące wartości (1%), istnieją wartości lekko odstające.

Wartości brakujące usunięto.

- Incomecompositionofresources

Human Development Index pod kątem udziału surowców w dochodach.

Zmienna liczbowa, przedział 0 – 0.95, 167 brakujących wartości (6%), brak wartości odst.

Wartości uzupełniono średnimi z przynależnych klas.

- Schooling

Przeciętna długość nauki.

Zmienna liczbowa, przedział 0 – 20.7, 163 brakujące wartości (6%), brak wartości odst.

Wartości uzupełniono średnimi z przynależnych klas.

Zmienne liczbowe poddano normalizacji do przedziału 0-1, w związku z występowaniem wielu zmiennych z zakresem dodatnim, lub dodatnim procentowym. Kod normalizacji:

```
means = zeros(1,size(data,2));

for i=1:size(data,2)
    means(i) = mean(data(:,i));
end

stds = zeros(1,size(data,2));
for i=1:size(data,2)
    stds(i) = std(data(:,i));
end

% normalize
for i=[1 3:13]
    data(:,i) = (data(:,i) - means(i)) / stds(i);
end
```

Kod wyznaczający wartość średnią (dla rozmytej przynależności do klas decyzyjnych) do imputacji brakujących danych:

```
function column = fillAvg(X, decisions)

    nrOfClasses = size(decisions, 1);

    avg = 0;
    nr = 0;
    for i = 1:size(X)
        if ~isnan(X(i))
            for cl = 1:nrOfClasses
                avg = avg + decisions(cl, i)*X(i);
                nr = nr+1;
            end
        end
    end
    avg = avg/nr;

    for i = 1:size(X)
        if isnan(X(i))
            X(i) = avg;
        end
    end
    column = X;
end
```

5. Usunięcie danych skorelowanych

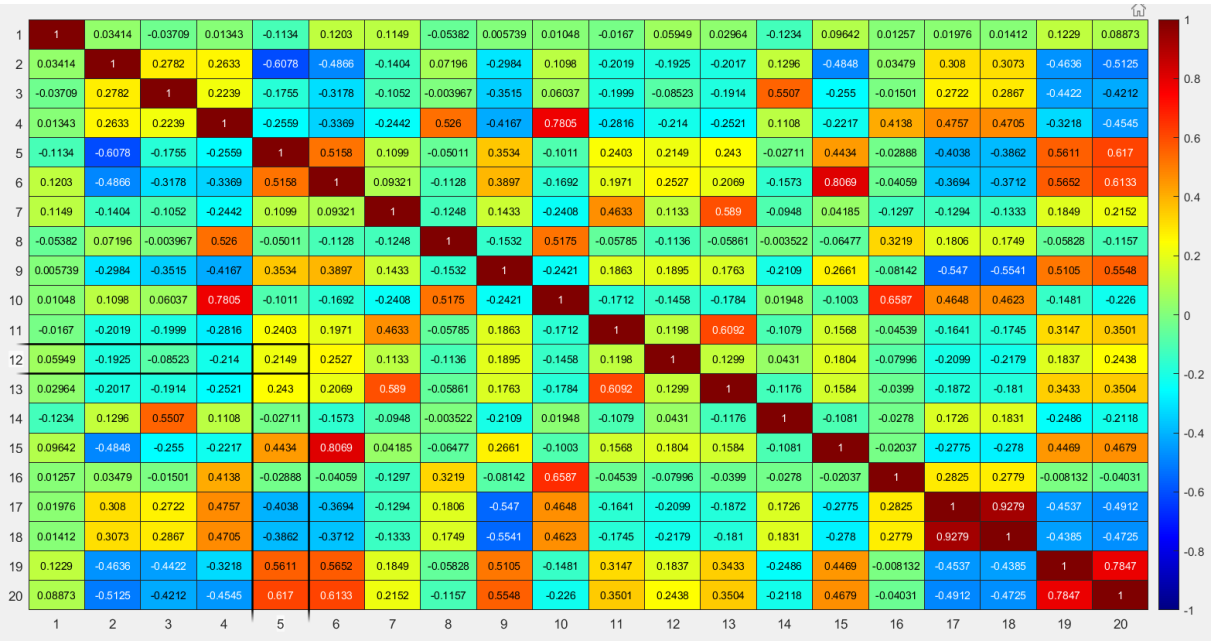


Tabela korelacji Pearsona sugeruje usunięcie kolumn 6 (percentageexpenditure), 10 (under-fivedeaths), 18 (thinness5-9years) i 20 (Schooling).

Wykorzystano VIF (Variance Inflation Factor). Kolumny mające jego wartość powyżej 5 to under-fivedeaths oraz thinness5-9years. Kolumna Schooling posiada wartość 4.2, którą można zachować.

Zdecydowano się na ostrożne usuwanie kolumn, po przeprowadzeniu paru prób usuwania różnych kombinacji kolumn skorelowanych. W tym zbiorze danych lepiej zachować jak najwięcej kolumn.

Kod wyznaczający wskaźnik VIF:

```
function vif = computeVIF(X)

    [n, p] = size(X);
    vif = zeros(1, p);
    for i = 1:p
        Xi = X(:, i);
        X_rest = X(:, [1:i-1, i+1:p]);
        lm = fitlm(X_rest, Xi);
        R2 = lm.Rsquared.Ordinary;
        vif(i) = 1 / (1 - R2);
    end
end
```

Zamiast usuwania kolumn, warto rozważyć zastosowanie bardziej zaawansowanych metod przetwarzania danych, np. metodę analizy PCA do tworzenia liniowych kombinacji cech.

6. Dobór parametrów sieci

```
net = patternnet([24, 12, 6], 'trainlm');

net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'tansig';
net.layers{3}.transferFcn = 'tansig';
net.layers{4}.transferFcn = 'softmax';

net.trainParam.max_fail = 10;
net.trainParam.mu_dec = 0.015;
net.trainParam.mu_inc = 10;
```

Zastosowano sieć z trzema warstwami ukrytymi o liczebności neuronów kolejno: 24, 12, 6. Liczba warstw oraz liczby neuronów zostały wyznaczone eksperymentalnie, jak poniżej:

dla lm								logsig	
12,12,12	12,9,6	6,6,6	6,6,6,6	6,6	24,24	12,12	12,6	24,12,6	tansig
0,87699	0,82232	0,86333	0,84738	0,83827	0,85649	0,83144	0,87016	0,8861	86105
0,84738	0,85649	0,85877	0,84966	0,82005	0,89066	0,86105	0,87472	0,8656	87016
0,87472	0,88155	0,83599	0,85649	0,82916	0,88155	0,85194	0,8451	0,88155	86105
0,88383	0,82688	0,84738	0,84738	0,85877	0,84966	0,85649	0,86333	0,84738	87699
0,852	0,85194	0,84966	0,82916	0,84738	0,85421	0,83827	0,84055	0,86333	84966
0,84738	0,84738	0,84738	0,82916	0,85194	0,83827	0,82916	0,84738	0,86105	87472
0,84055	0,85421	0,85421	0,83144	0,86788	0,85649	0,83371	0,86333	0,86788	85421
0,87244	0,85877	0,84282	0,84966	0,82688	0,8451	0,8656	0,86105	0,89977	84738
0,81549	0,84282	0,84966	0,87016	0,85194	0,87244	0,86333	0,86788	0,88838	86105
0,85877	0,86333	0,87244	0,81549	0,8451	0,90433	0,86333	0,86333	0,87016	87244
0,86788	0,87927	0,85649	0,86788	0,82688	0,86105	0,84282	0,83827	0,83599	84738
0,81321	0,87244	0,88383	0,83371	0,83371	0,85421	0,86788	0,86333	0,86974	82916
0,85422	0,85478333	0,85516	0,84396	0,8415	0,86371	0,85042	0,8582	0,86974	85877,1

28,14,14	14,14,14	14,14,7	7,7,7	10,7,6	14,10,6	7,7,6
0,87016	0,85649	0,87016	0,89977	0,86788	0,83827	0,82688
0,87244	0,8656	0,84282	0,87472	0,8451	0,87699	0,81093
0,86333	0,89294	0,86105	0,86105	0,85649	0,84282	0,83827
0,87016	0,88383	0,87699	0,85649	0,8861	0,85421	0,87016
0,87016	0,87472	0,87472	0,86333	0,85877	0,85877	0,87472
0,84055	0,85194	0,86105	0,87016	0,85877	0,87927	0,89522
0,85194	0,83599	0,82916	0,85877	0,83371	0,86788	0,84966
0,8861	0,83599	0,87016	0,88155	0,88155	0,87016	0,84738
0,89522	0,85421	0,86105	0,87016	0,84966	0,8656	0,85421
0,8451	0,8656	0,84738	0,87016	0,87699	0,87244	0,86333
0,86652	0,861731	0,85945	0,85877	0,8615	0,86264	0,85308

Rodzaj zastosowanej sieci to patternnet. Jest to bardziej współczesna sieć, popularna w problemach klasyfikacji, zwłaszcza z wyjściową funkcją aktywacji softmax.

Podczas poszukiwania optymalnych ustawień sieci przetestowano 4 metody z optymalnymi parametrami uczenia: lm, gdm, scg oraz rp.

Podstawowym kryterium oceny była wartość accuracy.

Proces dobierania parametrów opierał się na wykonywaniu 10 prób dla kilku różnych wartości. Najpierw modyfikowano tylko pierwszy parametr, a następnie tylko drugi. Poniżej przedstawiono pomiary dla scg:

K	L	M	N	O	P	Q	R	S	T	U	V
sigma = 5e-3		sigma = 5e-1		sigma = 5e-5		sigma = 5e-4		sigma = 5e-3		sigma = 5e-3	
lambda = 5e-5		lambda = 5e-5		lambda = 5e-5		lambda = 5e-5		lambda = 5e-4		lambda = 5e-6	
0,87244		0,87472		0,85421		0,87927		0,89066		0,84966	
0,86788		0,87244		0,83144		0,86105		0,8246		0,83371	
0,87244		0,86788		0,87244		0,8656		0,8656		0,86105	
0,85421		0,85877		0,84282		0,8451		0,83827		0,8656	
0,8861		0,83371		0,89522		0,8451		0,8656		0,87016	
0,84966		0,85421		0,88383		0,86788		0,8451		0,88383	
0,86333		0,8656		0,87016		0,87699		0,8451		0,82232	
0,86105		0,85877		0,85877		0,83144		0,8451		0,85194	
0,8451		0,84966		0,86105		0,81321		0,84738		0,80866	
0,85877		0,84738		0,85421		0,85649		0,83371		0,87472	
0,863098		0,858314		0,862415		0,854213		0,850112		0,852165	

Zestawienie wyników dla optymalnych parametrów różnych metod uczenia. Jak widać, najlepszą metodą okazała się metoda Levenberga-Marquadta.

mu_dec=0.25	lr=10	sgm=5e-3	delt_inc=1.6
mu_inc=15	mc=0.55	lmbd=5e-5	delt_dec=0.1

lm	gdm	scg	rp
0,8861	0,87244	0,863098	0,86
0,8656	0,84282	0,87244	0,84
0,88155	0,87927	0,86788	0,86
0,84738	0,83827	0,87244	0,84
0,86333	0,81777	0,85421	0,82
0,86105	0,8451	0,8861	0,82
0,86788	0,86333	0,84966	0,86
0,89977	0,86788	0,86333	0,87
0,88838	0,84738	0,86105	0,84
0,87016	0,8656	0,8451	0,89
0,83599	0,85877	0,85877	0,85
0,869745	0,854421	0,863098	0,850518

7. Trening modelu

Podział danych dokonano w standardowej proporcji: 70-15-15, jak w kodzie poniżej:

```
[trainInd, valInd, testInd] = dividerand(size(data, 1), 0.70, 0.15, 0.15);


trainInputs = data(trainInd, :)';
trainTargets = decisions(:, trainInd);
valInputs = data(valInd, :)';
valTargets = decisions(:, valInd);
testInputs = data(testInd, :)';
testTargets = decisions(:, testInd);

[net, tr] = train(net, trainInputs, trainTargets);

outputs = net(testInputs);
performance = perform(net, testTargets, outputs);
```

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	28	1000
Elapsed Time	-	00:00:06	-
Performance	0.657	0.00359	0
Gradient	1.7	0.0101	1e-07
Mu	0.001	8.52e-08	1e+10
Validation Checks	0	10	10

Training Algorithms

Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Odczytanie błędów podczas przewidywania danych testowych:

```
ranges = [35 45 55 65 75 85]';

decodedOut = sum(outputs .* ranges);
decodedTargets = sum(testTargets .* ranges);

errors = decodedTargets - decodedOut;
rel_err = errors ./ decodedTargets;
```

Jak widać wyżej, rozmyte wartości decyzji zostały wyostrzone, by odczytać wartości predykcji.

8. Analiza modelu

Najlepszy wytrenowany model osiągnął accuracy 0.898, co jest bardzo dobrym wynikiem. Prawdopodobnie przy przetestowaniu większej liczby modeli, możliwe byłoby uzyskanie wyniku 91-92%.

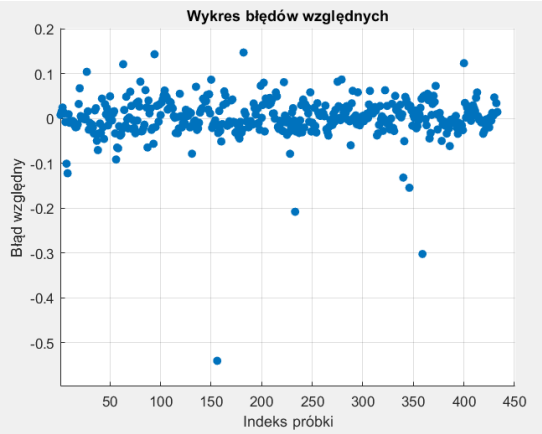
```
Accuracy: 0.89838
Recall: 0.9218
Precision: 0.9725
```

Baza danych nie została stworzona z myślą o klasyfikacji, co z pewnością prowadzi do pogorszenia wyniku. Wartości leżące na granicach przedziałów będą znacząco pogarszać wskaźniki jakości klasyfikacji.

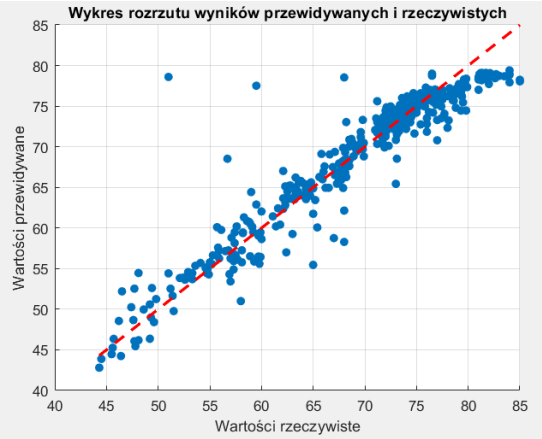
W dalszej części rozdziału zostanie przedstawiona analiza zarówno od strony klasyfikacji, jak i regresji.

Sporządzenie wykresów jakości modelu regresji:

```
figure;
scatter(1:length(decodedTargets), rel_err, 'filled');
xlabel('Indeks próbki');
ylabel('Błąd względny');
title('Wykres błędów względnych');
grid on;

figure;
scatter(1:length(decodedTargets), errors, 'filled');
xlabel('Indeks próbki');
ylabel('Błąd bezwzględny');
title('Wykres błędów bezwzględnych');
grid on;

figure;
scatter(decodedTargets, decodedOut, 'filled');
hold on;

plot([min(decodedTargets), max(decodedTargets)], [min(decodedTargets),
max(decodedTargets)], 'r--', 'LineWidth', 2);

xlabel('Wartości rzeczywiste');
ylabel('Wartości przewidywane');
title('Wykres rozrzutu wyników przewidywanych i rzeczywistych');
grid on;
hold off;
```



Jak widać na wykresie prawie wszystkie wyniki mieszczą się w zakresie -0.1 do 0.1 błędu względnego. Jedynie kilka predykcji wykracza poza ten przedział, co jest bardzo dobrym wynikiem. Błędy rozrzucone są bardzo losowo, co dobrze świadczy o jakości modelu.

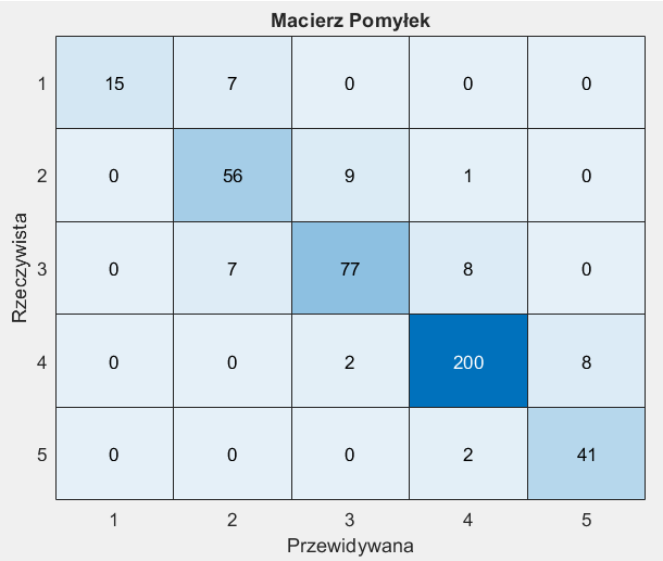


Na wykresie rozrzutu widzimy, że prawie wszystkie predykcje leżą blisko wartości oczekiwanych. Jedynie kilkanaście predykcji znacząco odbiega od oczekiwanych.

```
mse1 =

    8.3892


mae1 =

    1.8635


mape1 =

    2.7399
```

Błąd średni kwadratowy jest umiarkowany, co sugeruje, że model mógłby zostać ulepszony. Wskaźniki MAE i MAPE świadczą o niskim błędzie średnim, więc większość predykcji będzie trafna.

Jako że model może być interpretowany zarówno jako regresja jak i klasyfikacja, możliwe jest przygotowanie macierzy pomyłek jak poniżej:



Widzimy, że najgorzej przewidywaną klasą jest klasa 1. Wynika to z jej niewielkiej liczebności. Model ma skłonność przyporządkowywać obiekty z klasy 1, do znacznie bardziej licznej klasy 2. Jest to prawdopodobnie niemożliwe do naprawienia dla tej bazy danych. Możliwym rozwiązaniem byłoby zastosować przekształcenie na całej kolumnie decyzyjnej, które tak rozrzuciłoby wartości, by dało się je równomiernie podzielić na równe przedziały. Poniżej przedstawiono oceny wyników klasyfikacji dla każdej z klas decyzyjnych:

```
Klasa 1
Accuracy: 0.98383
Recall: 0.68182
Precision: 1
Specificity: 1
Klasa 2
Accuracy: 0.94457
Recall: 0.84848
Precision: 0.8
Specificity: 0.96185
Klasa 3
Accuracy: 0.93995
Recall: 0.83696
Precision: 0.875
Specificity: 0.96774
Klasa 4
Accuracy: 0.9515
Recall: 0.95238
Precision: 0.94787
Specificity: 0.95067
Klasa 5
Accuracy: 0.97691
Recall: 0.95349
Precision: 0.83673
Specificity: 0.97949
```

Kod analizy klasyfikacji modelu:

```
C = confusionmat(vec2ind(testTargets), vec2ind(outputs));


figure;
heatmap(C, 'ColorBarVisible', 'off', 'XLabel', 'Przewidywana', 'YLabel',
'Rzeczywista', 'Title', 'Macierz Pomyłek');

num_classes = size(C, 1);

accuracy = zeros(num_classes, 1);
precision = zeros(num_classes, 1);
recall = zeros(num_classes, 1);
specificity = zeros(num_classes, 1);

for i = 1:num_classes
    TP = C(i, i);
    FP = sum(C(:, i)) - TP;
    FN = sum(C(i, :)) - TP;
    TN = sum(C(:)) - TP - FP - FN;

    accuracy(i) = (TP + TN) / sum(C(:));
    precision(i) = TP / (TP + FP);
    recall(i) = TP / (TP + FN);
    specificity(i) = TN / (TN + FP);
end

for i = 1:num_classes
    disp(['Klasa ', num2str(i)]);
    disp(['Accuracy: ', num2str(accuracy(i))]);
    disp(['Recall: ', num2str(recall(i))]);
    disp(['Precision: ', num2str(precision(i))]);
    disp(['Specificity: ', num2str(specificity(i))]);
end

TP = sum(diag(C))
FN = sum(sum(triu(C, 1)));
FP = sum(sum(tril(C, -1)));
TN = sum(C(:)) - TP - FN - FP

accuracy = (TP + TN) / sum(C(:));
recall = TP / (TP + FN);
precision = TP / (TP + FP);

disp(['Accuracy: ', num2str(accuracy)]);
disp(['Recall: ', num2str(recall)]);
disp(['Precision: ', num2str(precision)]);
```