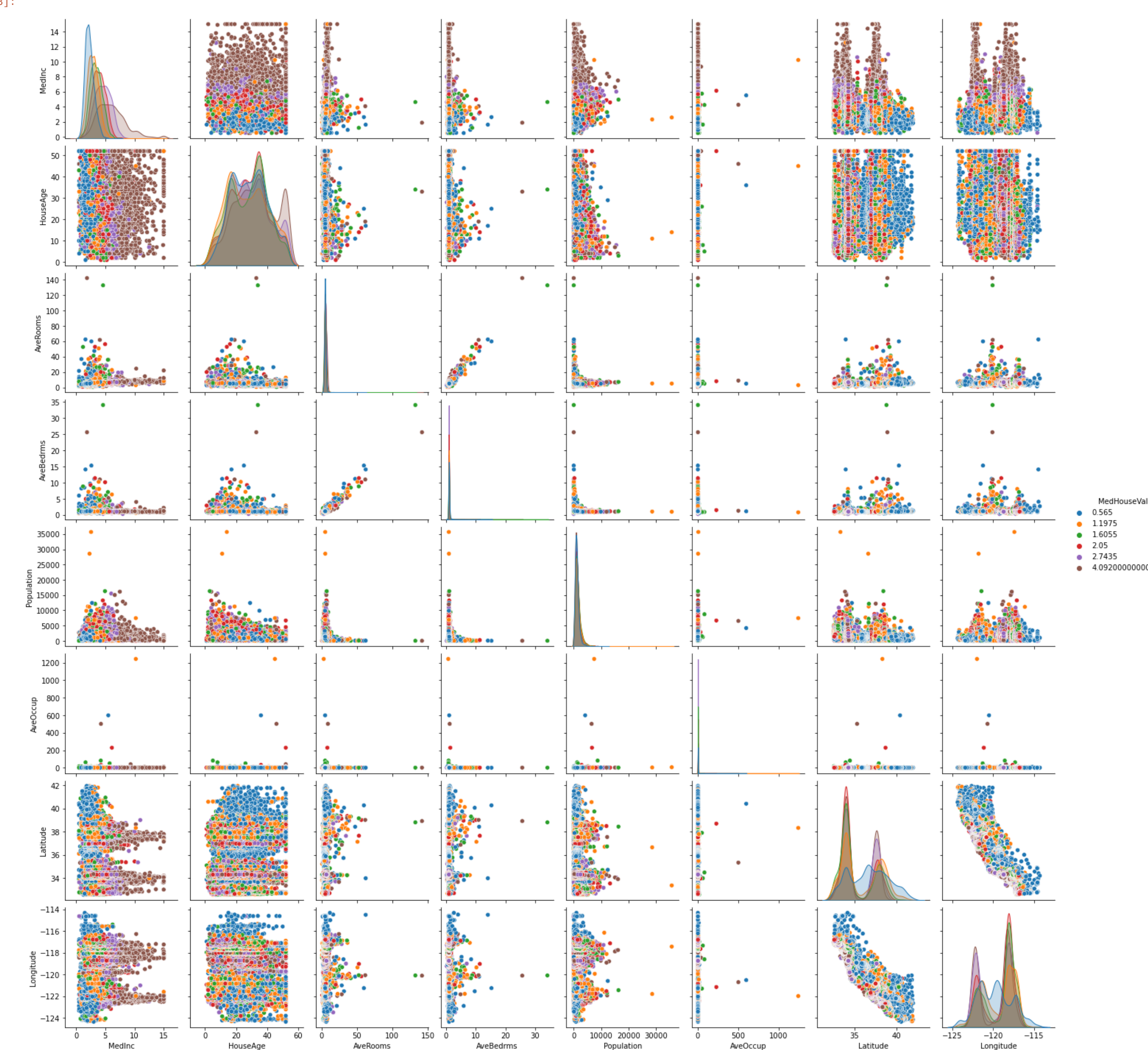
Lab1 - modele liniowe

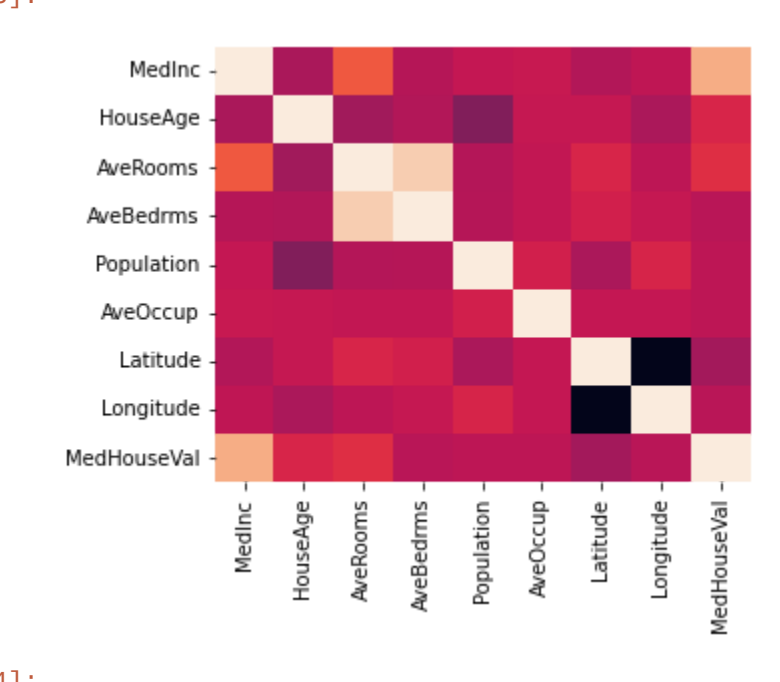
Bartosz Dudek

```
California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block group
        - HouseAge      median house age in block group
        - AveRooms      average number of rooms per household
        - AveBedrms     average number of bedrooms per household
        - Population    block group population
        - AveOccup      average number of household members
        - Latitude      block group latitude
        - Longitude     block group longitude

The target variable is the median house value for California districts,
expressed in hundreds of thousands of dollars ($100,000).

This dataset was derived from the 1990 U.S. census, using one row per census
block group. A block group is the smallest geographical unit for which the U.S.
Census Bureau publishes sample data (a block group typically has a population
of 600 to 3,000 people).

An household is a group of people residing within a home. Since the average
number of rooms and bedrooms in this dataset are provided per household, these
columns may take surpinsingly large values for block groups with few households
and many empty houses, such as vacation resorts.

   MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
0  8.3252      41.0  6.984127   1.023810       322.0  2.555556     37.88
1  8.3014      21.0  6.238137   0.971880      2401.0  2.109842     37.86
2  7.2574      52.0  8.288136   1.073446       496.0  2.802260     37.85
3  5.6431      52.0  5.817352   1.073059       558.0  2.547945     37.85
4  3.8462      52.0  6.281853   1.081081       565.0  2.181467     37.85

   Longitude  MedHouseVal
0    -122.23        4.526
1    -122.22        3.585
2    -122.24        3.521
3    -122.25        3.413
4    -122.25        3.422

RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   MedInc       20640 non-null  float64
 1   HouseAge     20640 non-null  float64
 2   AveRooms     20640 non-null  float64
 3   AveBedrms    20640 non-null  float64
 4   Population   20640 non-null  float64
 5   AveOccup     20640 non-null  float64
 6   Latitude     20640 non-null  float64
 7   Longitude    20640 non-null  float64
 8   MedHouseVal  20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB
```

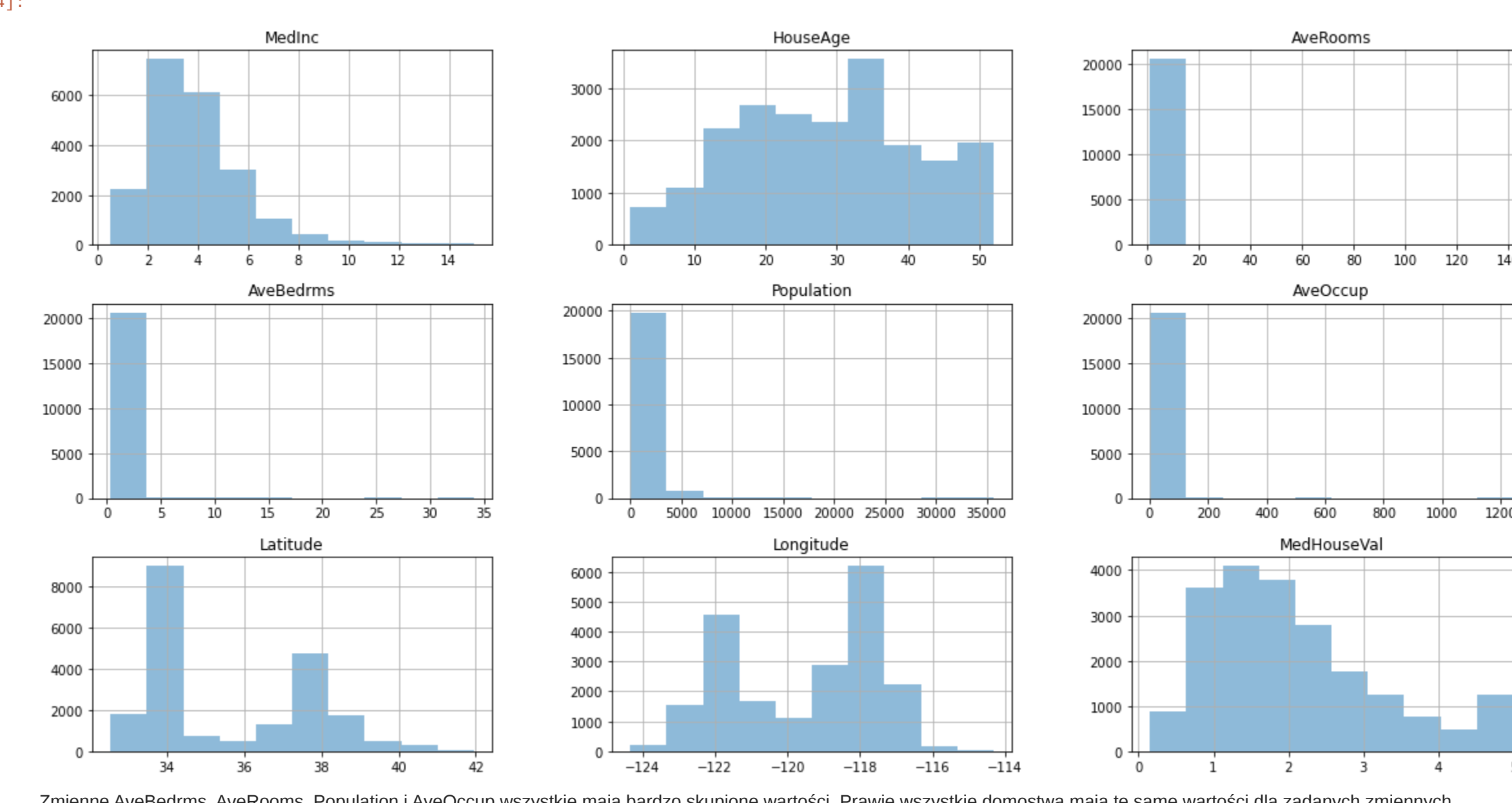
Out[18]:



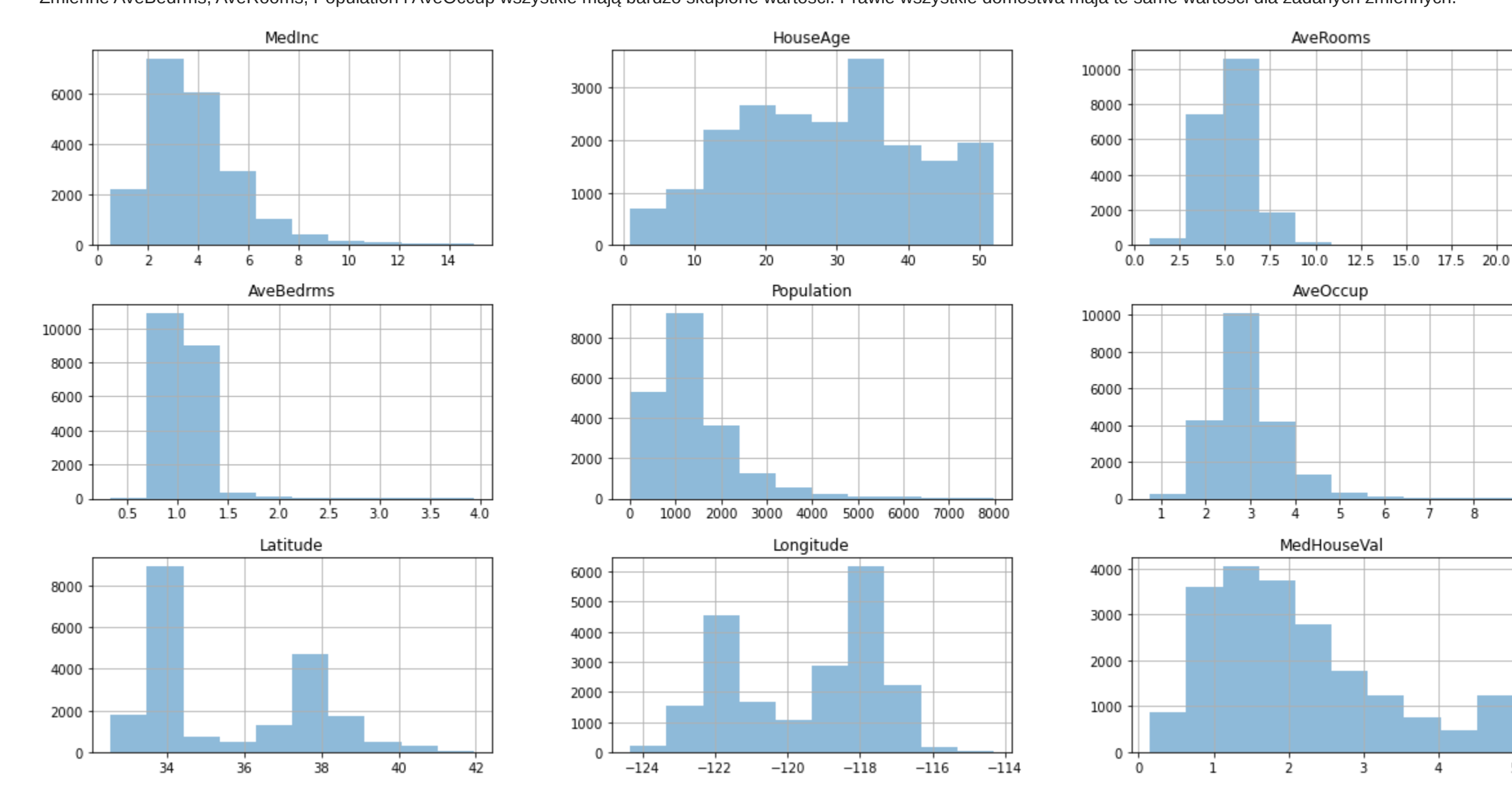
Out[3]:



Out[4]:



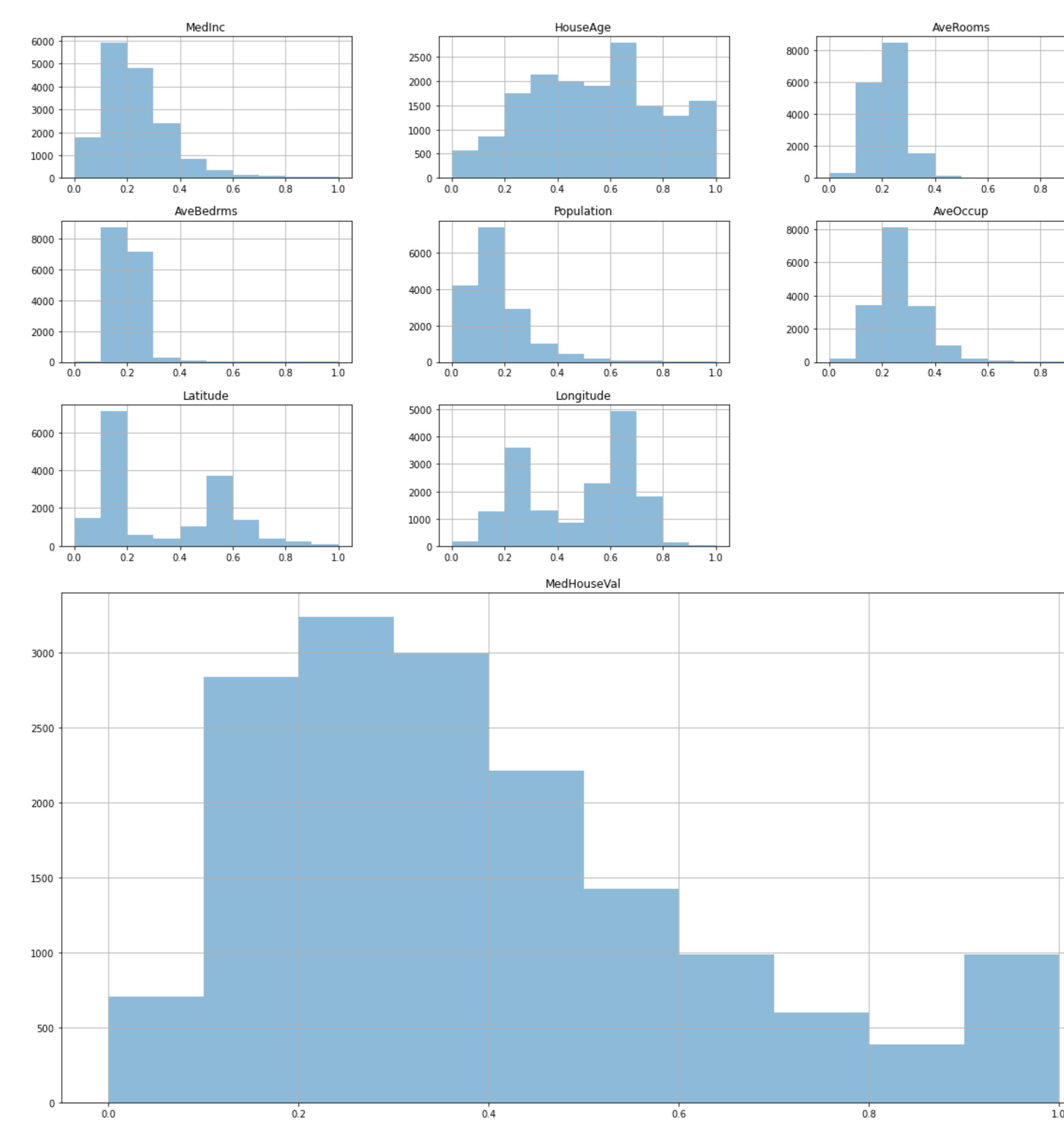
Zmienne AveBedrms, AveRooms, Population i AveOccup wszystkie mają bardzo skupione wartości. Prawie wszystkie domostwa maja te same wartości dla zadanych zmiennych.



Zmienne które wcześniej były skupione są dokładniejsze na wykresach co pozwala odczytać konkretniejsze wartości. Podczas regresji najważniejsze będzie MedInc

```
Int64Index: 20463 entries, 0 to 20639
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   MedInc      20463 non-null  float64
 1   HouseAge    20463 non-null  float64
 2   AveRooms    20463 non-null  float64
 3   AveBedrms   20463 non-null  float64
 4   Population  20463 non-null  float64
 5   AveOccup    20463 non-null  float64
 6   Latitude    20463 non-null  float64
 7   Longitude   20463 non-null  float64
dtypes: float64(8)
memory usage: 1.4 MB
```

Po przeskalowaniu



Wartości błędu modelu

```
0.10369863579416368
0.39360645615487266 MedHouseVal    0.394217
dtype: float64
Score:  0.646675667509965
```

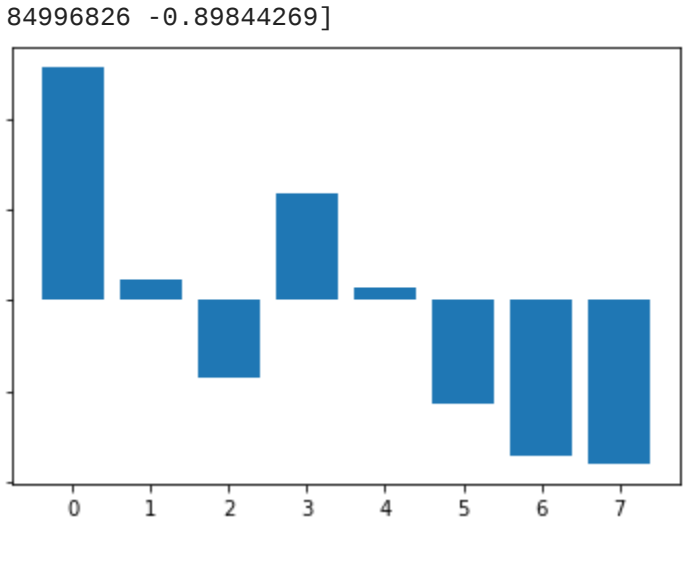
Średnie wartości są bardzo zbliżone, w stosunku do średnich wartości jest dość spory.

```
MAE - 0.1010422473738371
```

Model nie dopasował się idealnie do danych treningowych więc napewno nie będzie overfittingu, występuje dość znaczny błąd co może oznaczać underfitting ale nie jest jeszcze na tyle duży

Wagi cech

```
[ 1.2828417   0.11619419 -0.41949517  0.59526072  0.0712893  -0.57141647
 -0.84996826 -0.89844269]
```



Wartości błędu dla degree = 3

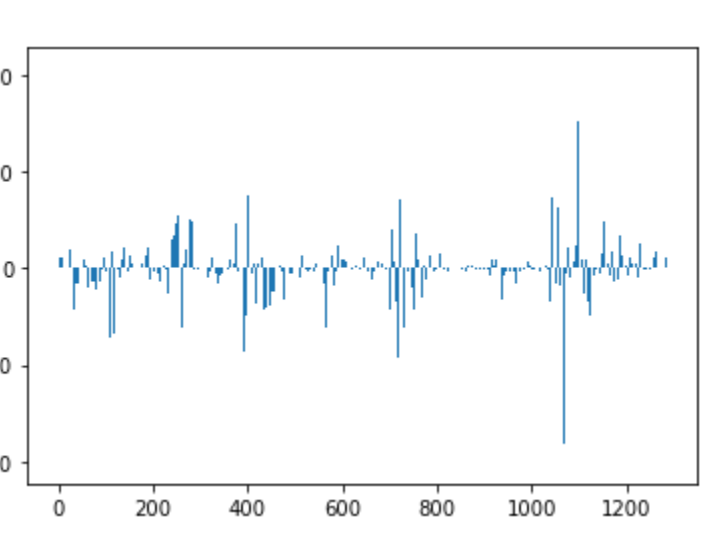
```
0.08523310640440387
0.3937983576473915 MedHouseVal    0.394217
dtype: float64
Score:  0.739512918144162
```

Dla degree = 5 gorsze wyniki

```
0.09301928821245872
0.3896716047226991 MedHouseVal    0.394217
dtype: float64
Score:  0.06539251184511363
```

Wartości poprawiły się dla degree = 3. W sytuacji degree = 5 znacznie się pogarszają.

Wagi cech degree = 5

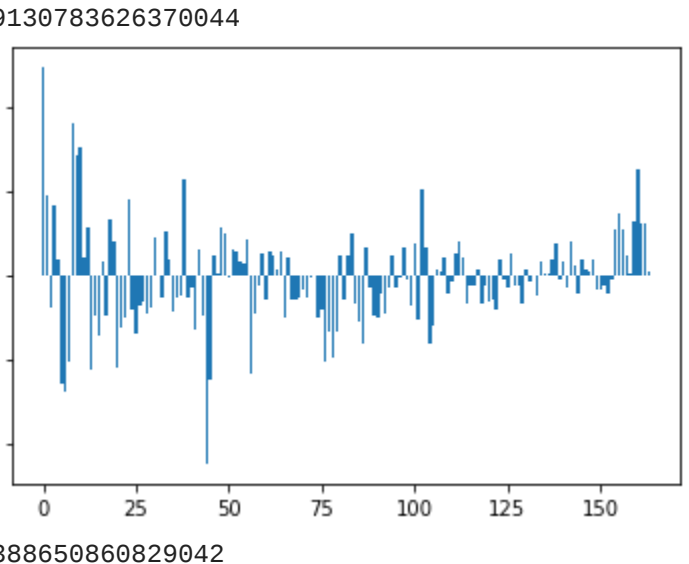


```
wartość max - 20787.062346901275
wartość min - -20258.9878730695
liczba zerowych wag - 0
```

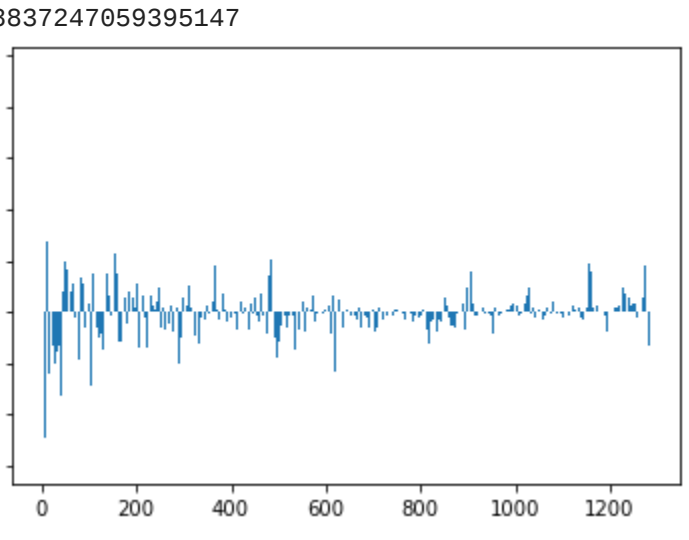
Brak wag zero

Wartości dla kolejno degree 3 potem 5 modelu Ridge

```
0.09130783626370044
```



```
1.2388650860829042
-1.1147369944464605
0
0.08837247059395147
```



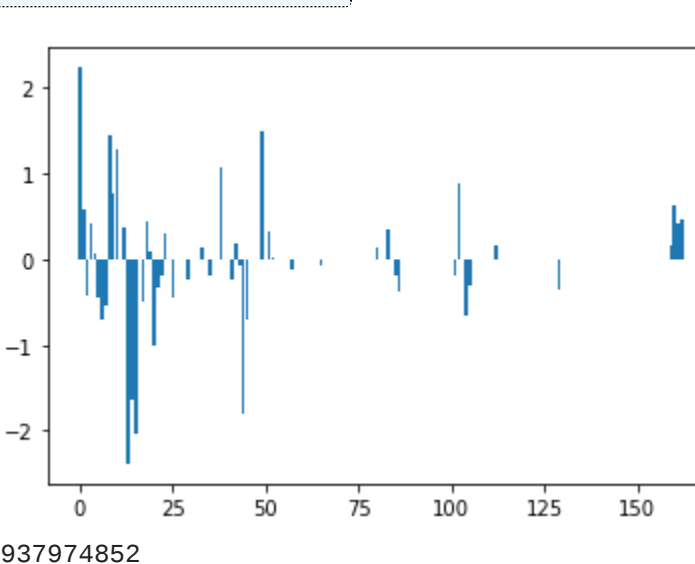
```
0.9536951824915761
-0.5917674013117712
0
0.18849694800937775
0.18849694800937775
```

Nie pojawiają sie zerowe wagi, model Ridge radzi sobie znacznie lepiej.

Out[115]:

```
LassoCV
LassoCV(cv=5, random_state=0)
```

Model Lasso



```
2.2394773937974852
-2.3857762180290014
-50
0.08999053316141939
```

Wyeliminowano 30% cech

Najlepsze wyniki otrzymały: model LassoCV przy zbiorze rozszerzonym (degree=3), Ridge przy zbiorze rozszerzonym (degree=5)

Zadanie 2

Out[150]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007

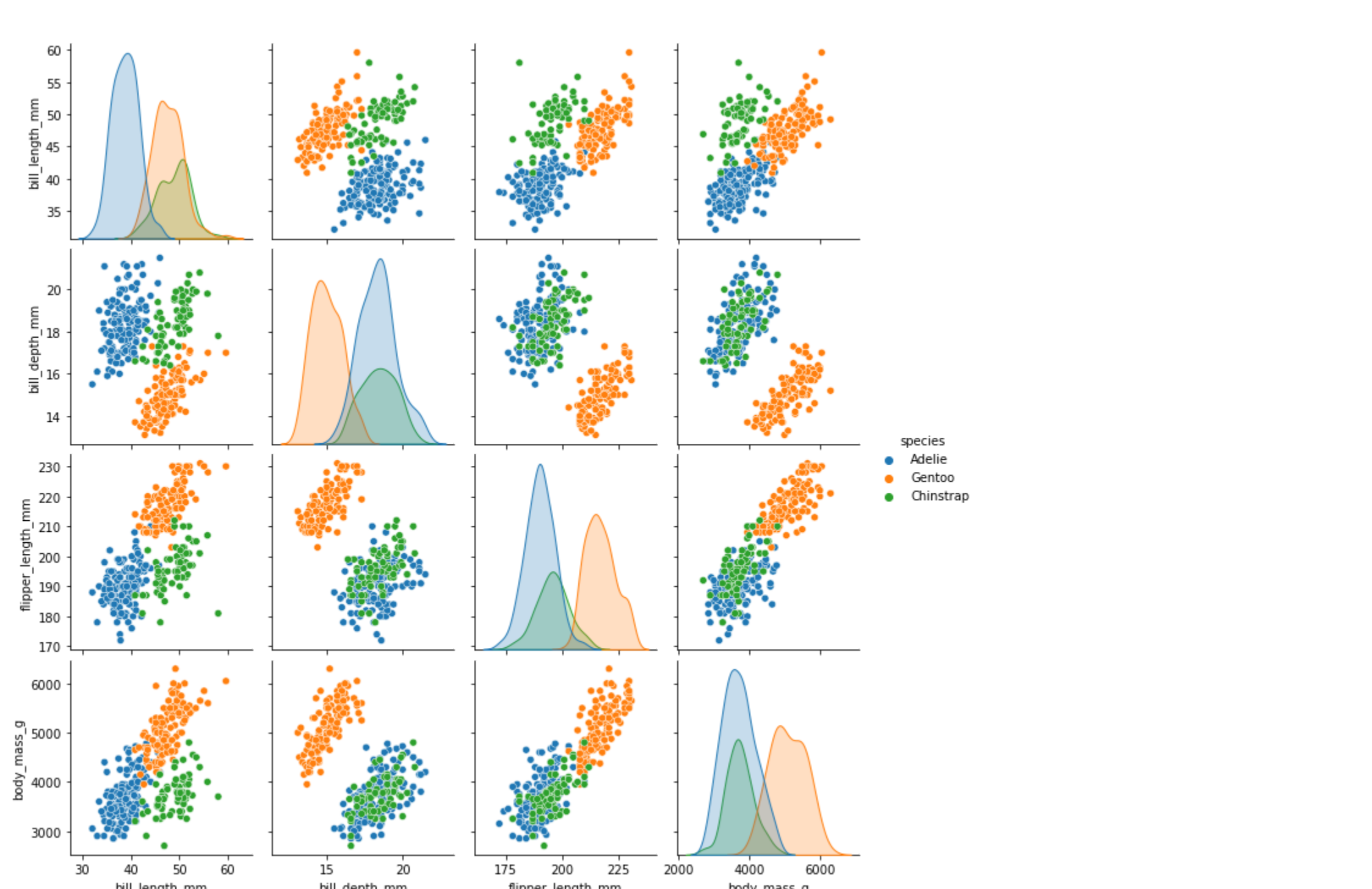
```
Int64Index: 333 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   species            333 non-null    object
 1   island             333 non-null    object
 2   bill_length_mm     333 non-null    float64
 3   bill_depth_mm      333 non-null    float64
 4   flipper_length_mm  333 non-null    float64
 5   body_mass_g        333 non-null    float64
 6   sex                333 non-null    object
dtypes: float64(4), object(3)
memory usage: 20.8+ KB
```

Gatunki

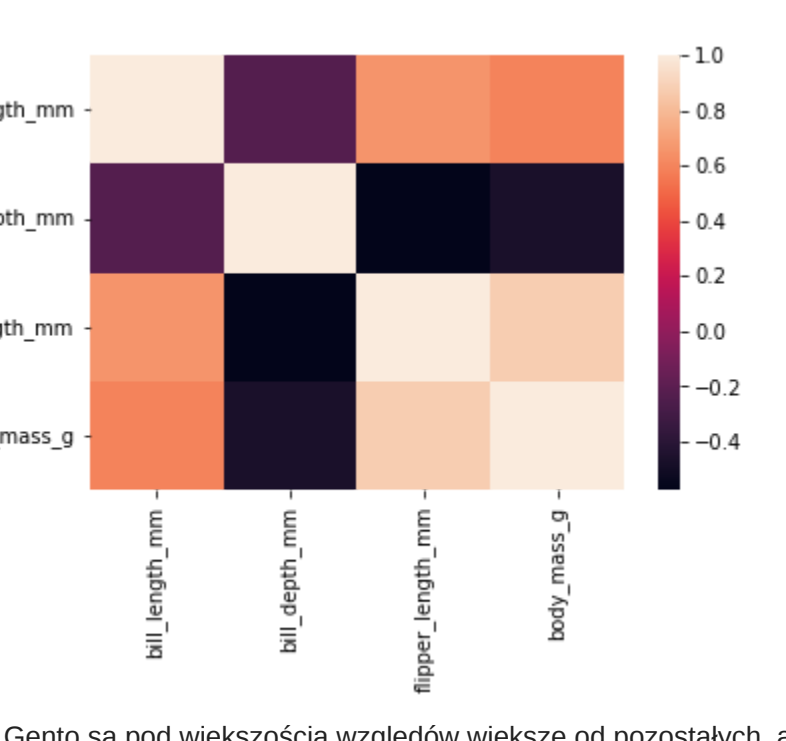
```
Adelie       146
Gentoo       119
Chinstrap     68
Name: species, dtype: int64
```

Dane są niezbalansowane

Out[136]:



Out[137]:



Pingwiny Gento są pod większością względów większe od pozostałych, a Adelie mniejsze. Silna korelacja pomiędzy długością dzioba i długością płetw.

Występowanie gatunków na wyspach:

```
['Torgersen' 'Biscoe' 'Dream']
```

Chinstrap na wyspie:

```
['Dream']
```

Out[184]:

	species	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	39.1	18.7	181.0	3750.0	1
1	Adelie	39.5	17.4	186.0	3800.0	0
2	Adelie	40.3	18.0	195.0	3250.0	0
4	Adelie	36.7	19.3	193.0	3450.0	0
5	Adelie	39.3	20.6	190.0	3650.0	1

Out[272]:

```
LogisticRegression
LogisticRegression(class_weight='balanced', random_state=0)
```

```
Liczba źle sklasyfikowanych: 1 , procent: 2 %
```

W zależności od tego jak podzialiły się zbiory testowy i treningowy wyniki wachają się od dobrych 1 błąd do bardzo złych 14 błędów. (Raczej dostajemy dobre wyniki ale przy pierwszej próbie z nieznanego powodu było 14 błędów - wszystko zaklasyfikował jako Adelie)

```
             feature      coef
0     bill_length_mm  3.985857
1      bill_depth_mm -0.666142
2  flipper_length_mm  0.228105
3        body_mass_g -0.566453
4                sex -1.201405
```

Zdecydowanie najwazniejsza jest długość dzioba co zgadza się z przewidywaniami.

```
Liczba źle sklasyfikowanych: 14 , procent: 33 %
```

Dużo większy błąd, bo najważniejsza cecha zniknęła.

```
Liczba źle sklasyfikowanych: 17 , procent: 40 %
```

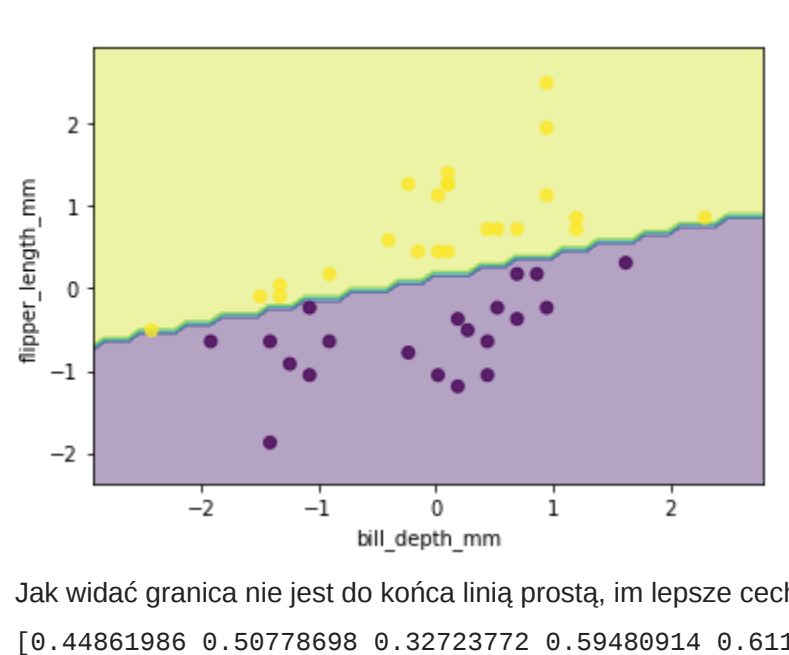
Przy wyborze tylko 2 cech nie będących najważniejszymi

```
             feature      coef
0      bill_depth_mm -0.279318
1  flipper_length_mm  0.993229
```

Wybrełem głębokośc dzioba i długośćpłetwy przez macierz korelacji. Jak się można było spodziawać wyniki są gorsza ale 60% poprawności nie jast złym wynikiem.

```
['Adelie', 'Chinstrap', 'Adelie', 'Chinstrap', 'Chinstrap', 'Chinstrap', 'Adelie', 'Chinstrap', 'Adelie', 'Adelie', 'Adelie', 'Chinstrap', 'Adelie', 'Adelie', 'Adelie', 'Chinstrap',
'Chinstrap', 'Chinstrap', 'Adelie', 'Chinstrap', 'Adelie', 'Adelie', 'Adelie', 'Chinstrap', 'Chinstrap', 'Chinstrap', 'Adelie', 'Adelie', 'Adelie', 'Adelie', 'Chinstrap', 'Chinstra
p', 'Chinstrap', 'Adelie', 'Chinstrap', 'Chinstrap', 'Chinstrap', 'Chinstrap', 'Chinstrap', 'Adelie', 'Chinstrap', 'Adelie', 'Chinstrap'] ['Adelie' 'Chinstrap' 'Adelie' 'Chinstrap'
'Chinstrap' 'Chinstrap'
 'Adelie' 'Chinstrap' 'Adelie' 'Adelie' 'Adelie' 'Chinstrap' 'Adelie'
 'Adelie' 'Adelie' 'Chinstrap' 'Chinstrap' 'Chinstrap' 'Adelie'
 'Chinstrap' 'Adelie' 'Adelie' 'Adelie' 'Chinstrap' 'Chinstrap'
 'Chinstrap' 'Adelie' 'Adelie' 'Adelie' 'Adelie' 'Chinstrap' 'Chinstrap'
 'Chinstrap' 'Adelie' 'Chinstrap' 'Chinstrap' 'Chinstrap' 'Chinstrap'
 'Chinstrap' 'Adelie' 'Chinstrap' 'Adelie' 'Chinstrap']
```

Wszystkie się pokrywają. Przemnozenie zbioru testowego przez wagi daje ten sam efekt co .predict()



Jak widać granica nie jest do końca linią prostą, im lepsze cechy wybierzemy to klasyfikacja będzie dokonywana z większą dokładnością.

```
[0.44861986 0.50778698 0.32723772 0.59480914 0.61167746 0.56875682
 0.37294071 0.59355818 0.2915398  0.16677676 0.43890494 0.63495621
 0.46026953 0.23265877 0.36321788 0.74866234 0.76593039 0.7269497
 0.21234696 0.56043051 0.3306733  0.28833564 0.34590259 0.58595474
 0.57451005 0.67277235 0.40454076 0.29915975 0.37108771 0.33128086
 0.74866234 0.77334142 0.54489255 0.48170633 0.53321702 0.6060822
 0.82280391 0.88880584 0.57129989 0.42998239 0.51830509 0.2016305
 0.56681034] [[0.55138014 0.44861986]
 [0.49221302 0.50778698]
 [0.67276228 0.32723772]
 [0.40519086 0.59480914]
 [0.38832254 0.61167746]
 [0.43124318 0.56875682]
 [0.62705929 0.37294071]
 [0.40644182 0.59355818]
 [0.7084602  0.2915398 ]
 [0.83322324 0.16677676]
 [0.56109506 0.43890494]
 [0.36504379 0.63495621]
 [0.53973047 0.46026953]
 [0.76734123 0.23265877]
 [0.63678212 0.36321788]
 [0.25133766 0.74866234]
 [0.23406961 0.76593039]
 [0.2730503  0.7269497 ]
 [0.78765304 0.21234696]
 [0.43956949 0.56043051]
 [0.6693267  0.3306733 ]
 [0.71166436 0.28833564]
 [0.65409741 0.34590259]
 [0.41404526 0.58595474]
 [0.42548995 0.57451005]
 [0.32722765 0.67277235]
 [0.59545924 0.40454076]
 [0.70084025 0.29915975]
 [0.62891229 0.37108771]
 [0.66871914 0.33128086]
 [0.25133766 0.74866234]
 [0.22665858 0.77334142]
 [0.45510745 0.54489255]
 [0.51829367 0.48170633]
 [0.46678298 0.53321702]
 [0.3939178  0.6060822 ]
 [0.17719609 0.82280391]
 [0.11119416 0.88880584]
 [0.42870011 0.57129989]
 [0.57001761 0.42998239]
 [0.48169491 0.51830509]
 [0.7983695  0.2016305 ]
 [0.43318966 0.56681034]]
True
```

Otrzymany wynik to True czyli przekonwertowane dane daja ten sam rezultat.