

Diagnozowanie raka piersi
Obrazowanie biomedyczne - projekt

Klaudia Gora, Jan Machowski, Mikołaj Mazurek

14 stycznia 2020

Spis treści

1	Wstęp	4
1.1	Cel projektu	4
2	Zbiór danych	4
3	Implementacja	7
3.1	Rozkład danych	8
3.2	Rozkład danych po standaryzacji	10
4	Opracowanie wyników i wnioski	12
	Bibliografia	14

Spis rysunków

1	Przykładowe dane	6
2	Obserwacja rozkładu danych w zbiorze danych (1/2)	8
3	Obserwacja rozkładu danych w zbiorze danych (2/2)	9
4	Obserwacja rozkładu danych w zbiorze danych po standaryzacji (1/2)	10
5	Obserwacja rozkładu danych w zbiorze danych po standaryzacji (2/2)	11
6	Logistic Regression Classifier	12
7	3-Nearest Neighbors Classifier	12
8	Support Vector Classifier Liner Kernel	12
9	Support Vector Classification RBF Kernel	13
10	Gaussian Naive Bayes Classification	13
11	Decision Tree Classifier	13
12	Random Forest Classifier	13
13	MLP Classifier	13
14	Obliczone p-wartości	14
15	Statystyczne porównanie danych	14

1 Wstęp

Rak piersi jest jednym z najczęstszych nowotworów występujących u kobiet na całym świecie. Zostało to udokumentowane przez *World Cancer Research Fund* [4] w 2018 r. Liczba przypadków raka piersi zdiagnozowanych w 2018 r. Wyniosła 2088 849 i stanowi 25,4% ogólnej liczby zdiagnozowanych przypadków. Rak piersi występuje również u mężczyzn, jednak z częstotliwością 100 razy mniejszą. Wczesna diagnoza może znacznie wpłynąć na zwiększenie szans na zachowanie piersi dotkniętej nowotworem, a także na przeżycie pacjenta. Obecnie najczęstszą metodą diagnozowania tego schorzenia jest wykonanie biopsji cienkoigłowej, której skuteczność waha się od 65 do 98%. Różnorakie algorytmy mogą być wykorzystane w celu poprawienia tej skuteczności.

1.1 Cel projektu

Celem projektu jest zaimplementowanie wybranych klasyfikatorów oraz porównanie ich skuteczności na wybranych danych. Zostaną one przetestowane w zakresie diagnozowania raka piersi. Do ich zaimplementowania wykorzystany zostanie język programowania Python oraz biblioteka *sklearn* [2].

2 Zbiór danych

W projekcie wykorzystano istniejące dane ze strony UCI Machine Learning Repository, o nazwie *Breast Cancer Wisconsin (Diagnostic) Data Set* [1]. Zbiór danych zawiera 569 instancji, z których 357 dla klasy raka łagodnego, a 212 dla raka złośliwego. Zaimplementowany algorytm ma na celu zdiagnozowanie czy dana osoba posiada raka piersi w stadium złośliwym czy łagodnym. W pliku znajdują się dane, których atrybuty przedstawiono poniżej:

- id,
- diagnosis,
- radius_mean,
- texture_mean,
- perimeter_mean,
- area_mean,
- smoothness_mean,
- compactness_mean,

- concavity_mean,
- concavepoints_mean,
- symmetry_mean,
- fractal_dimension_mean,
- radius_se,
- texture_se,
- perimeter_se,
- area_se,
- smoothness_se,
- compactness_se,
- concavity_se,
- concavepoints_se,
- symmetry_se,
- fractal_dimension_se,
- radius_worst,
- texture_worst,
- perimeter_worst,
- area_worst,
- smoothness_worst,
- compactness_worst,
- concavity_worst,
- concavepoints_worst,
- symmetry_worst,
- fractal_dimension_worst.

Przykładowe dane wykorzystywane w problemie diagnozowania raka piersi przedstawiono w Tabeli 1.

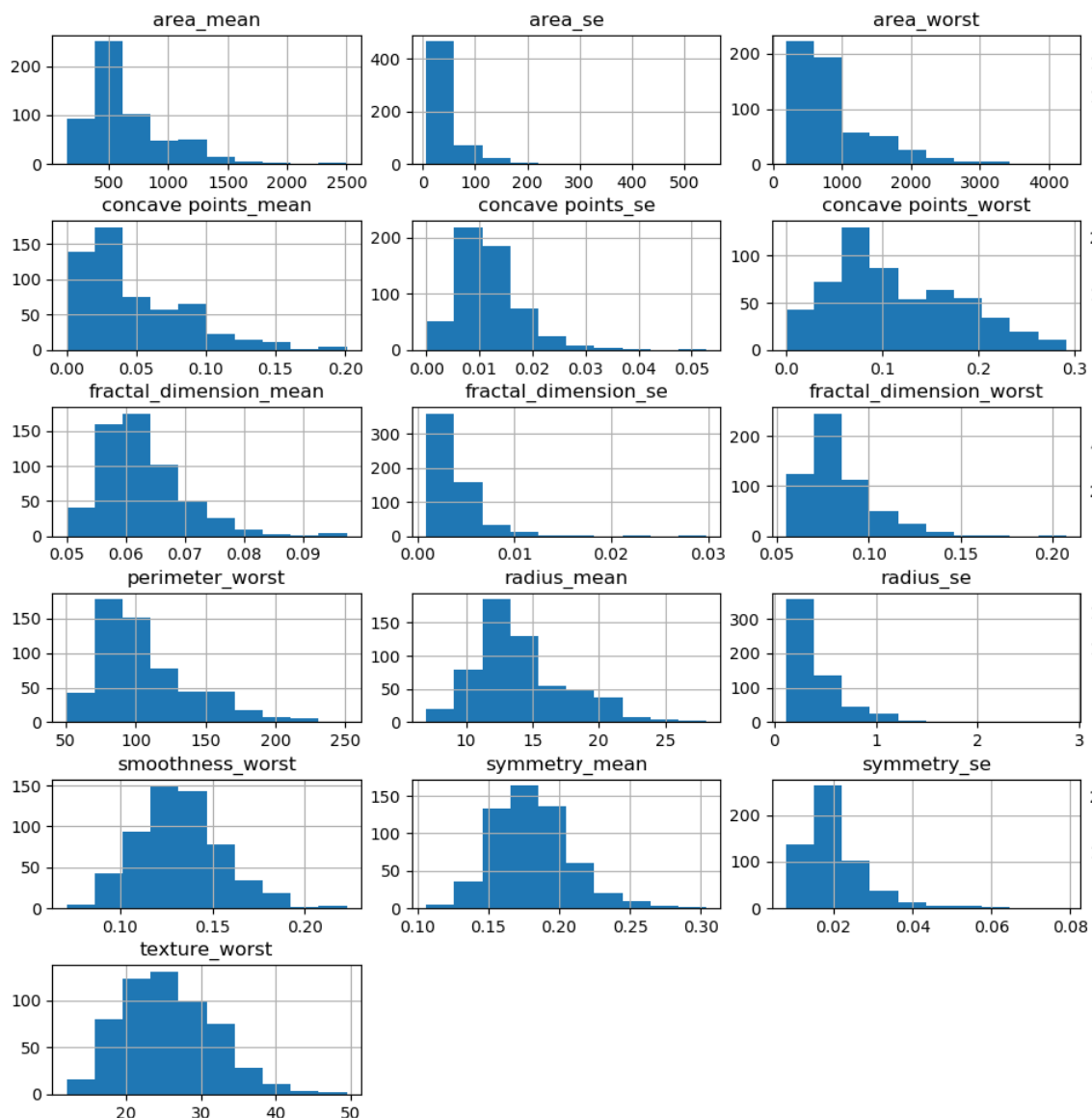
id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323
84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364
84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025
84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259
84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076

Rysunek 1: Przykładowe dane

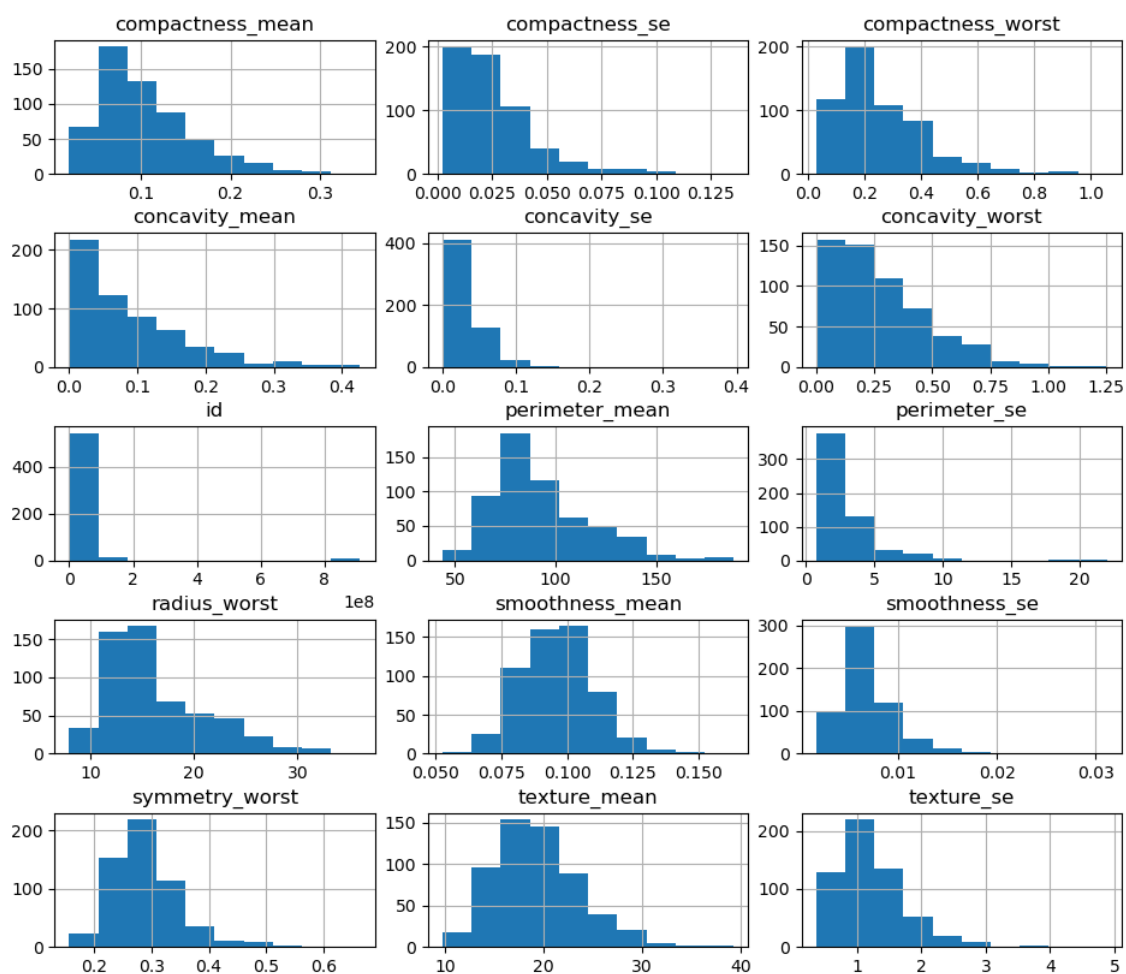
3 Implementacja

Implementacja została wykonana w języku Python 3.7 w środowisku PyCharm. Klasyfikatory zaimportowano z pakietu *sklearn*. Przed przystąpieniem do badań dane zostały ustandaryzowane. W celu uniknięcia zjawiska zwanego *Overfittingiem* zastosowano 5-krotną walidację krzyżową. Do parametryzacji klasyfikatorów wykorzystano funkcję *GridSearchCV()*, która spośród podanych parametrów wybrała te, które skutkowały najwyższą jakością klasyfikacji.

3.1 Rozkład danych

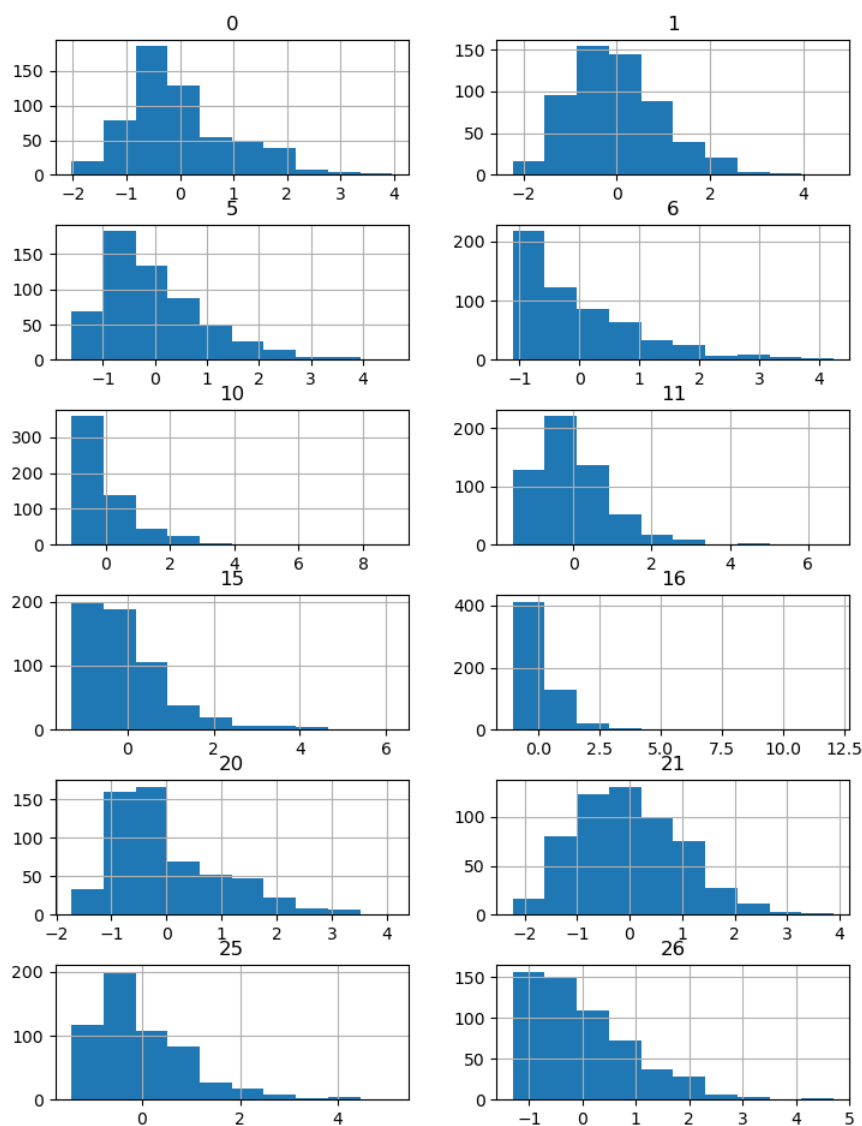


Rysunek 2: Obserwacja rozkładu danych w zbiorze danych (1/2)

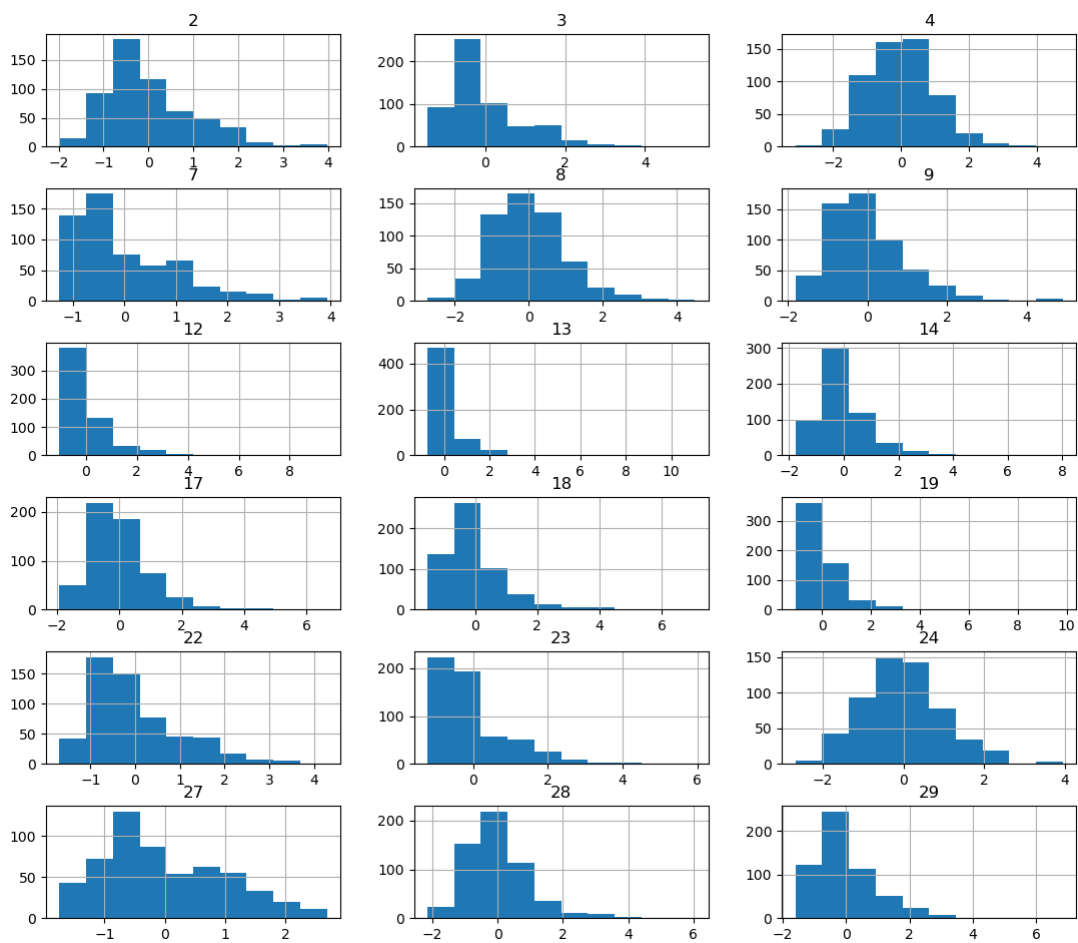


Rysunek 3: Obserwacja rozkładu danych w zbiorze danych (2/2)

3.2 Rozkład danych po standaryzacji



Rysunek 4: Obserwacja rozkładu danych w zbiorze danych po standaryzacji (1/2)

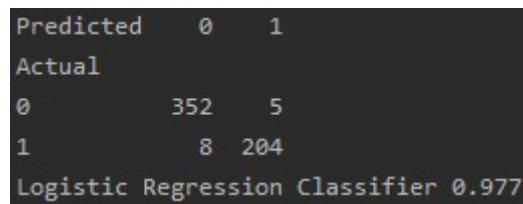


Rysunek 5: Obserwacja rozkładu danych w zbiorze danych po standaryzacji (2/2)

4 Opracowanie wyników i wnioski

Dla każdego klasyfikatora uzyskano dokładność klasyfikacji w każdym z pięciu *foldów*, co umożliwiło przeprowadzenie testu *Wilcoxon* [3]. Test ten wykazał, że klasyfikator *Gaussian Naive Bayes* jest statystycznie różny od *Random Forest* oraz *MLP*, a klasyfikator *Decision Tree* jest statystycznie różny od *Random Forest*.

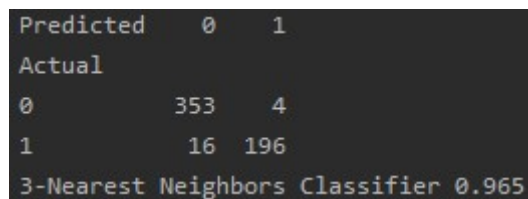
Na rysunkach 6-13 przedstawiono macierz konfuzji obrazującą liczbę poprawnych i niepoprawnych klasyfikacji. Rysunek 14 przedstawia wyznaczone p-wartości do testu Wilcoxon.



```
Predicted    0    1
Actual
0           352    5
1             8   204
Logistic Regression Classifier 0.977
```

This figure shows a confusion matrix for a Logistic Regression Classifier. The matrix is presented in a text-based format with a dark background. It includes the predicted and actual values for two classes (0 and 1). The classifier achieved an accuracy of 0.977.

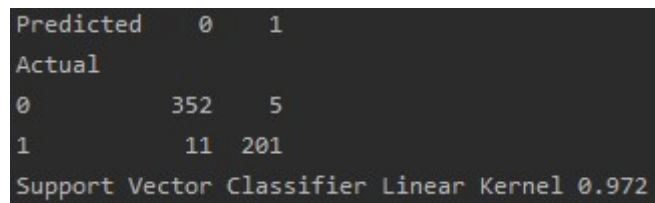
Rysunek 6: Logistic Regression Classifier



```
Predicted    0    1
Actual
0           353    4
1            16   196
3-Nearest Neighbors Classifier 0.965
```

This figure shows a confusion matrix for a 3-Nearest Neighbors Classifier. The matrix is presented in a text-based format with a dark background. It includes the predicted and actual values for two classes (0 and 1). The classifier achieved an accuracy of 0.965.

Rysunek 7: 3-Nearest Neighbors Classifier



```
Predicted    0    1
Actual
0           352    5
1            11   201
Support Vector Classifier Linear Kernel 0.972
```

This figure shows a confusion matrix for a Support Vector Classifier using a Linear Kernel. The matrix is presented in a text-based format with a dark background. It includes the predicted and actual values for two classes (0 and 1). The classifier achieved an accuracy of 0.972.

Rysunek 8: Support Vector Classifier Liner Kernel

```

Predicted    0    1
Actual
0           350    7
1            10  202
Support Vector Classification RBF Kernel 0.970

```

Rysunek 9: Support Vector Classification RBF Kernel

```

Predicted    0    1
Actual
0           340   17
1            23  189
Gaussian Naive Bayes Classification 0.930

```

Rysunek 10: Gaussian Naive Bayes Classification

```

Predicted    0    1
Actual
0           338   19
1            14  198
Decision Tree Classifier 0.942

```

Rysunek 11: Decision Tree Classifier

```

Predicted    0    1
Actual
0           348    9
1            22  190
Random Forest Classifier 0.946

```

Rysunek 12: Random Forest Classifier

```

Predicted    0    1
Actual
0           345   12
1            10  202
MLP Classifier 0.961

```

Rysunek 13: MLP Classifier

```
[[1.    0.216 0.059 0.416 0.043 0.042 0.08  0.068]
 [1.    1.    0.854 0.257 0.043 0.043 0.066 0.066]
 [1.    1.    1.    0.715 0.043 0.042 0.138 0.276]
 [1.    1.    1.    1.    0.043 0.043 0.042 0.066]
 [1.    1.    1.    1.    1.    0.357 0.043 0.042]
 [1.    1.    1.    1.    1.    1.    0.039 0.068]
 [1.    1.    1.    1.    1.    1.    1.    0.496]
 [1.    1.    1.    1.    1.    1.    1.    1.    ]]
```

Rysunek 14: Obliczone p-wartości

```
[[False False False False False False False False]
 [False False False False False False False False]
 [False False False False False False False False]
 [False False False False False False False False]
 [False False False False False False True  True]
 [False False False False False False True  False]
 [False False False False False False False False]
 [False False False False False False False False]]]
```

Rysunek 15: Statystyczne porównanie danych

Literatura

- [1] Breast cancer wisconsin (diagnostic) data set. [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)). Dostęp: 2019-11-25.
- [2] scikit-learn - machine learning in python. <https://scikit-learn.org/stable/>. Dostęp: 2019-11-12.
- [3] Test kolejności par wilcoxon. <https://www.statystyka.az.pl/test-kolejnosci-par-wilcoxona.php>. Dostęp: 2019-11-11.
- [4] Worldwide cancer data - global cancer statistics for the most common cancers. <https://www.wcrf.org/dietandcancer/cancer-trends/worldwide-cancer-data>. Dostęp: 2019-11-25.