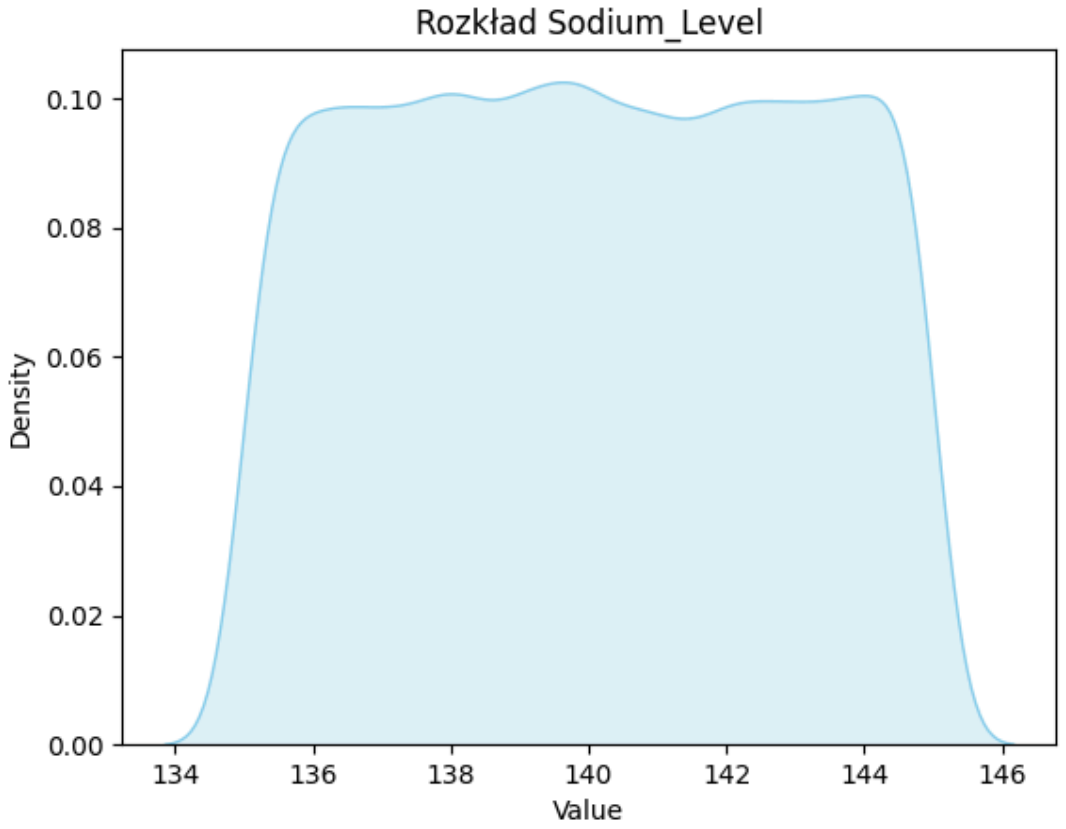
Prototypowanie

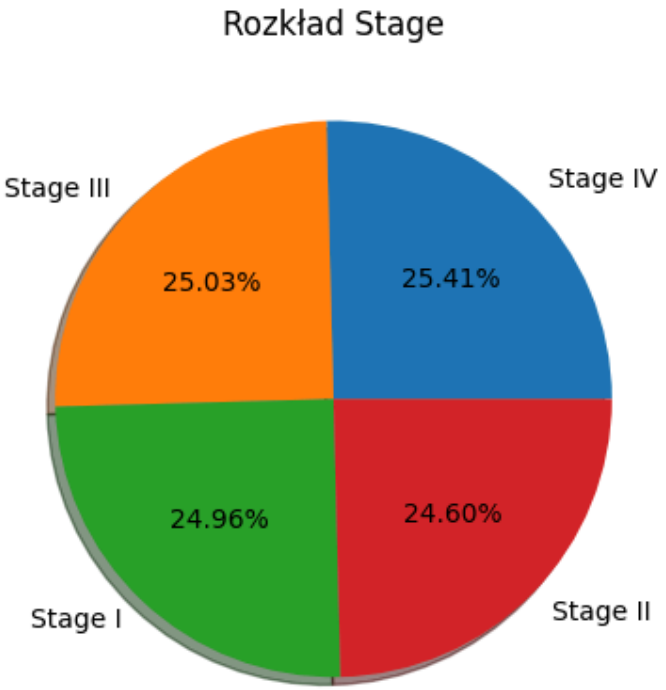
Analiza danych

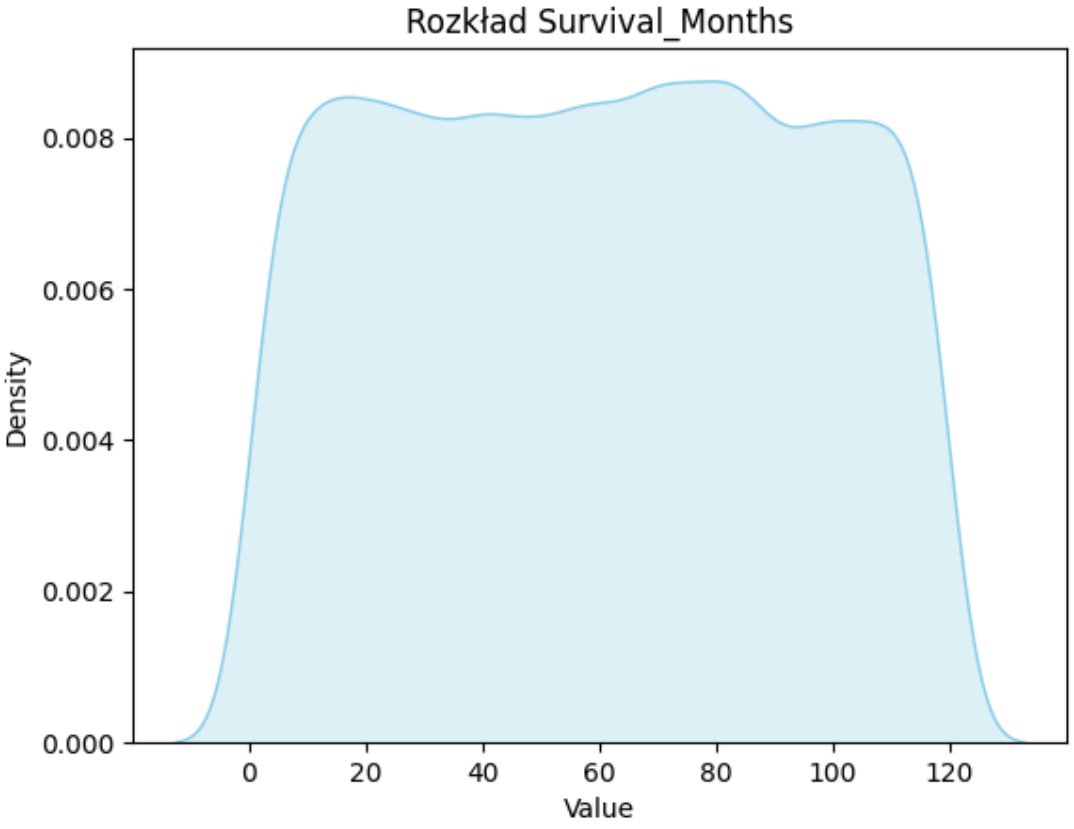
Dokładne efekty działania można zobaczyć w Analizying_data_set.ipynb.

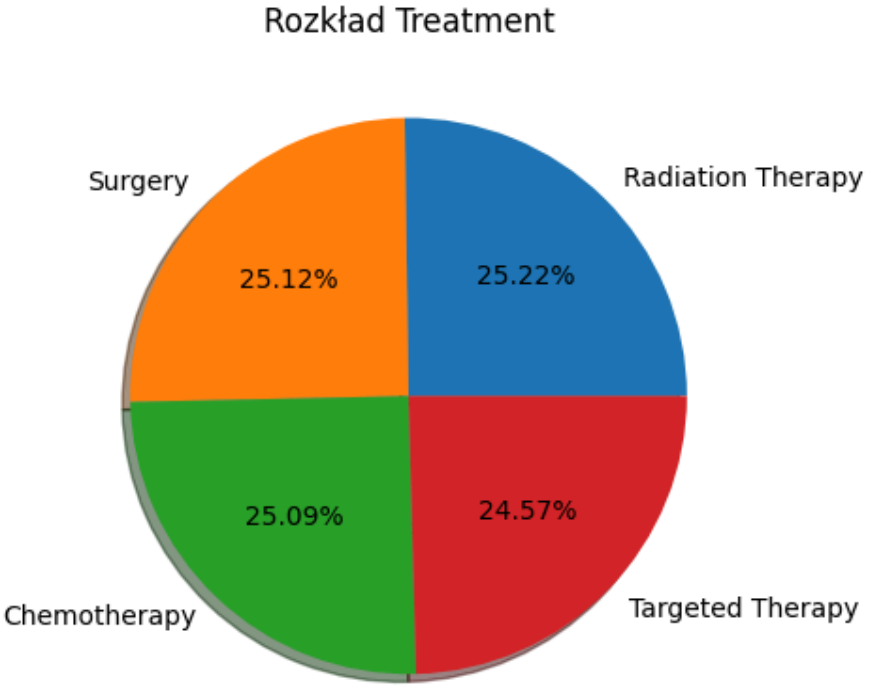
Własna analiza danych

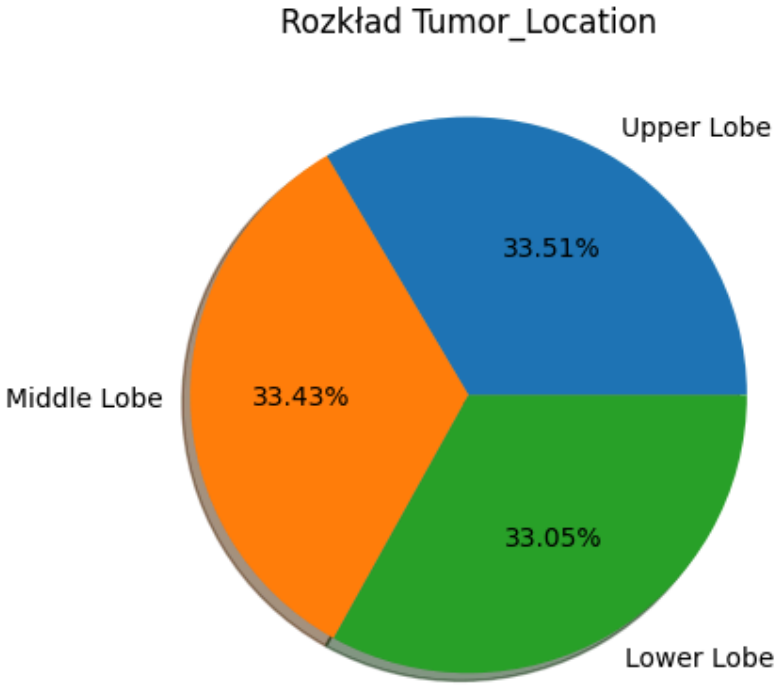
Jak można zobaczyć trzeba było zmienić kolumnę Performance_Status z numerycznej na kategorialną ponieważ ma konkretnie 5 różnych wartości (kategori) z których można wybrać wartość. Dalej można zobaczyć rozkład wartości każdej kolumn które zostały przedstawione w formie wykresów. Można na nich zauważyć, że dane są dość równomiernie rozłożone.

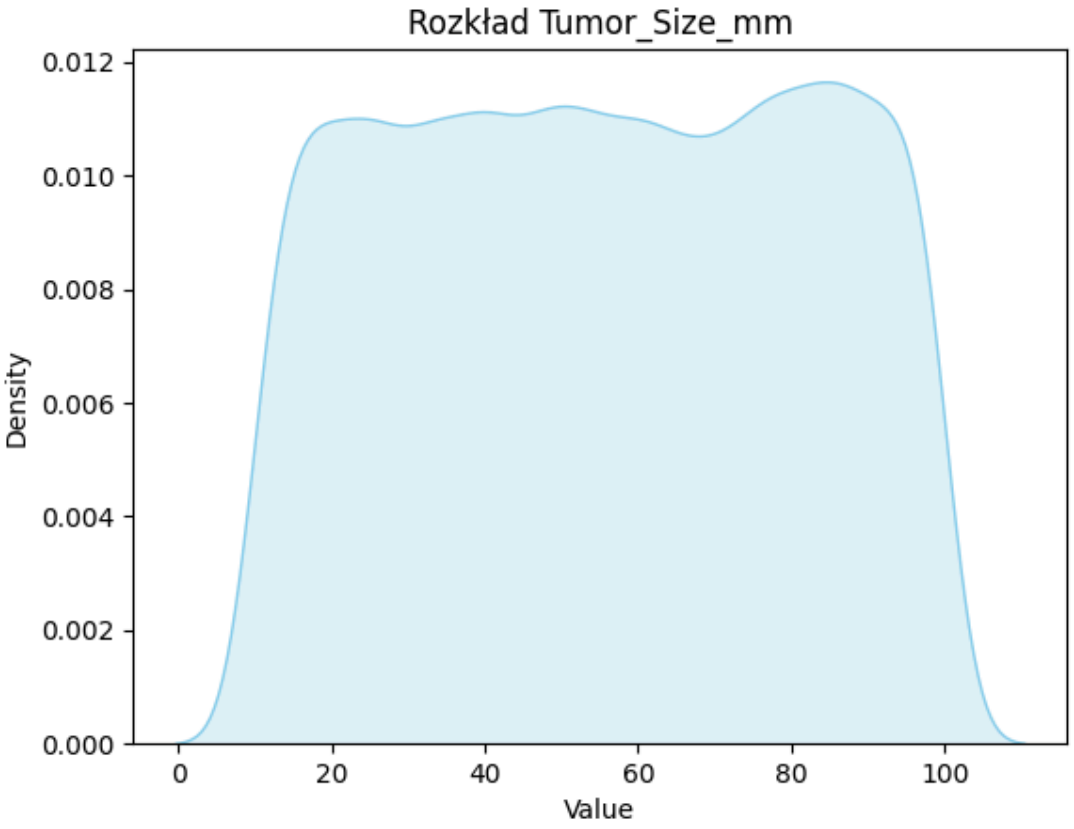


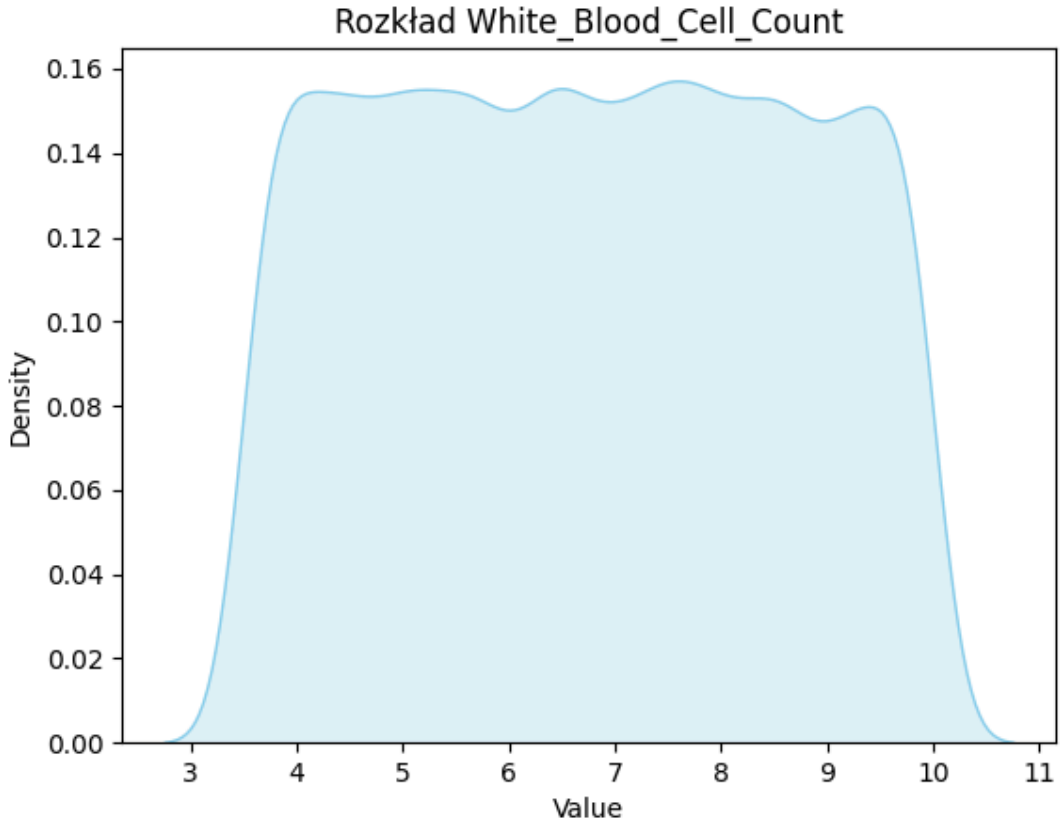

Rozkład Stage
Stage III
Stage IV
25.03%
25.41%
24.96%
24.60%
Stage I
Stage II

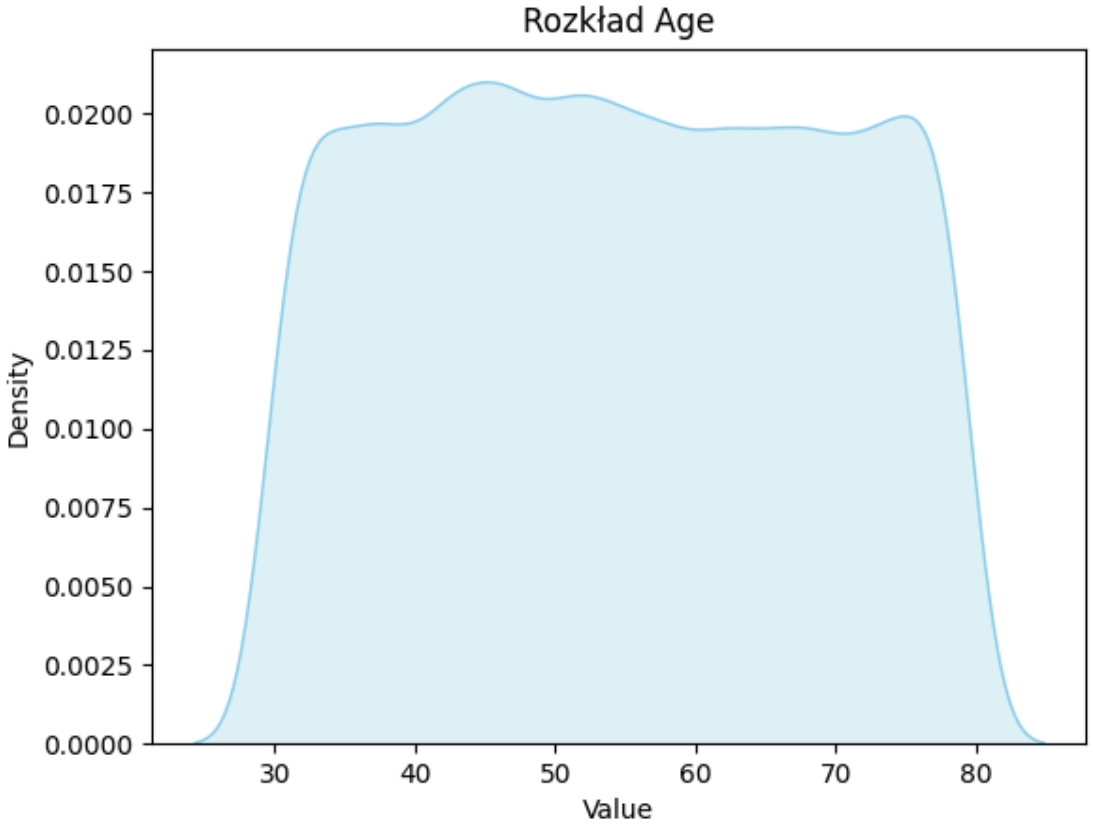

Rozkład Survival_Months
0.008
0.006
0.004
0.002
0.000
Density
0
20
40
60
80
100
120
Value

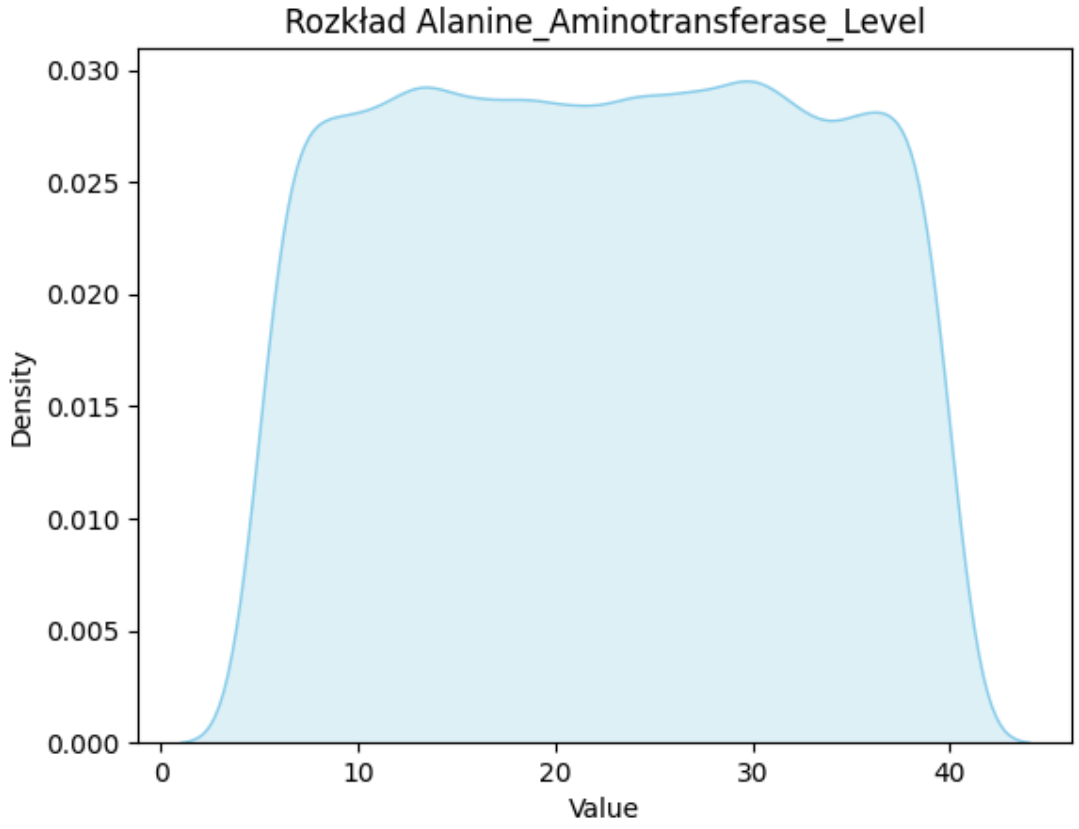

Rozkład Treatment
Surgery
Radiation Therapy
25.12%
25.22%
25.09%
24.57%
Chemotherapy
Targeted Therapy

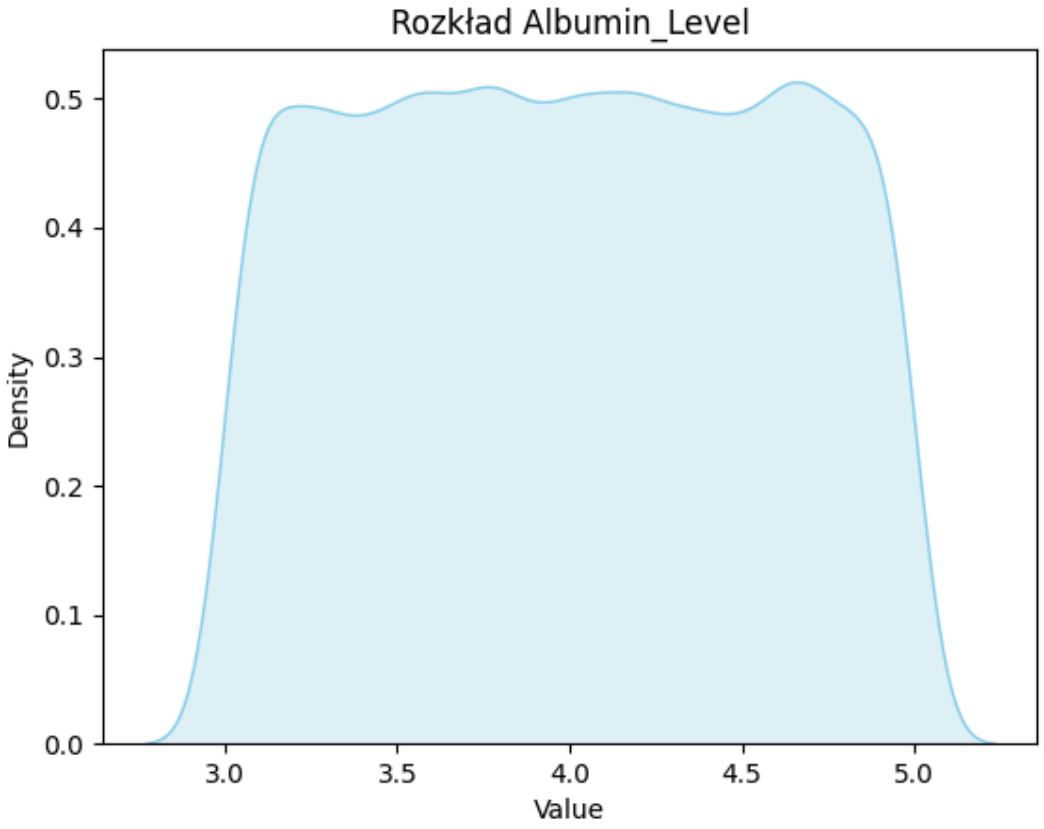

Rozkład Tumor_Location
Upper Lobe
33.51%
Middle Lobe
33.43%
33.05%
Lower Lobe

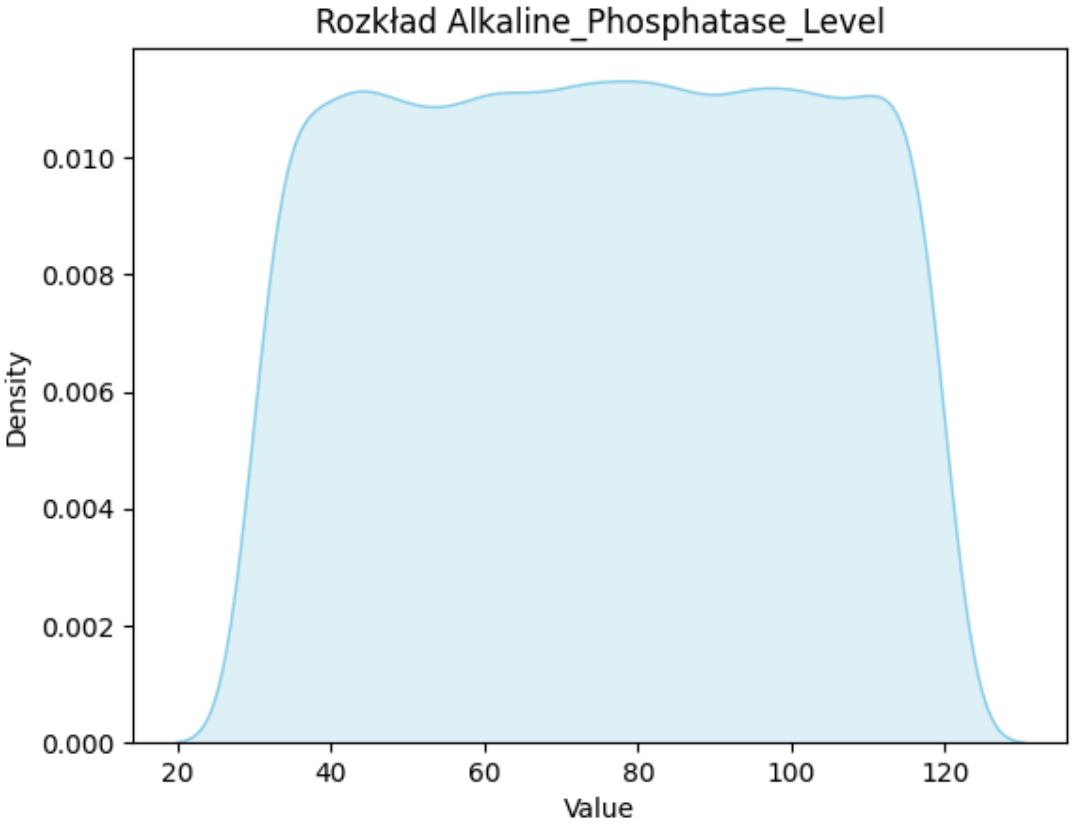

Rozkład Tumor_Size_mm
Density
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
20
40
60
80
100
Value

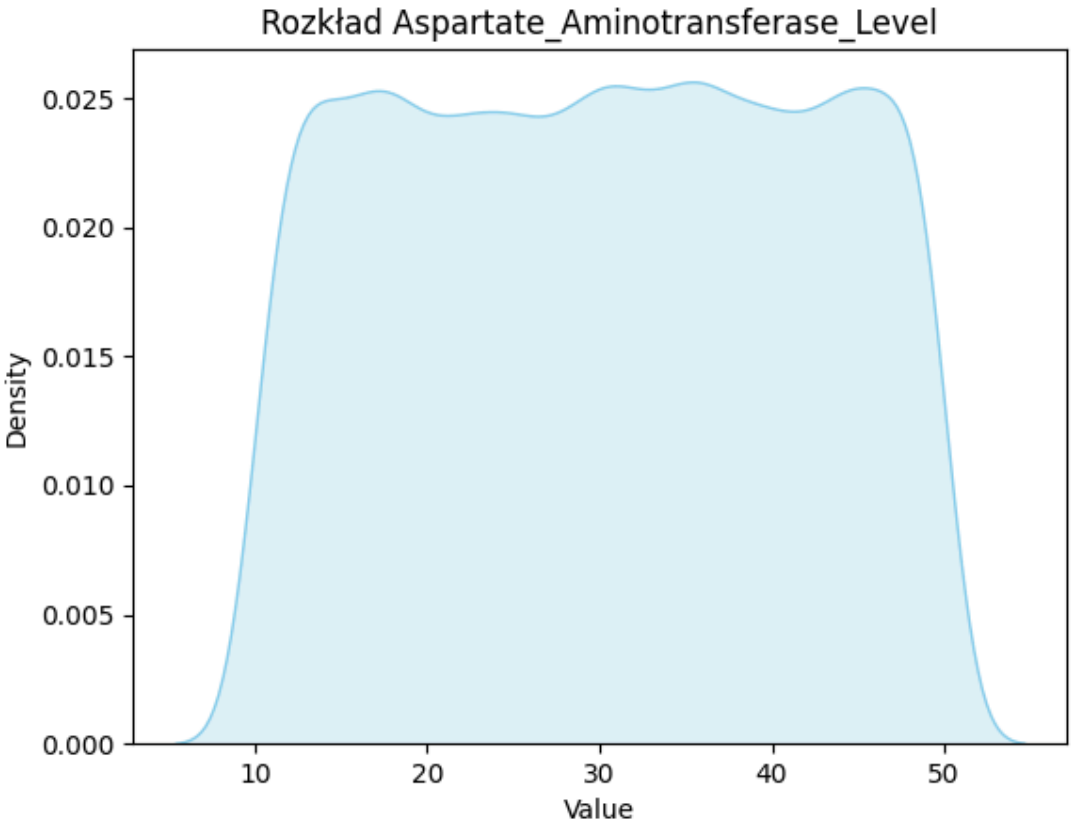

Rozkład White_Blood_Cell_Count
Density
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
3
4
5
6
7
8
9
10
11
Value

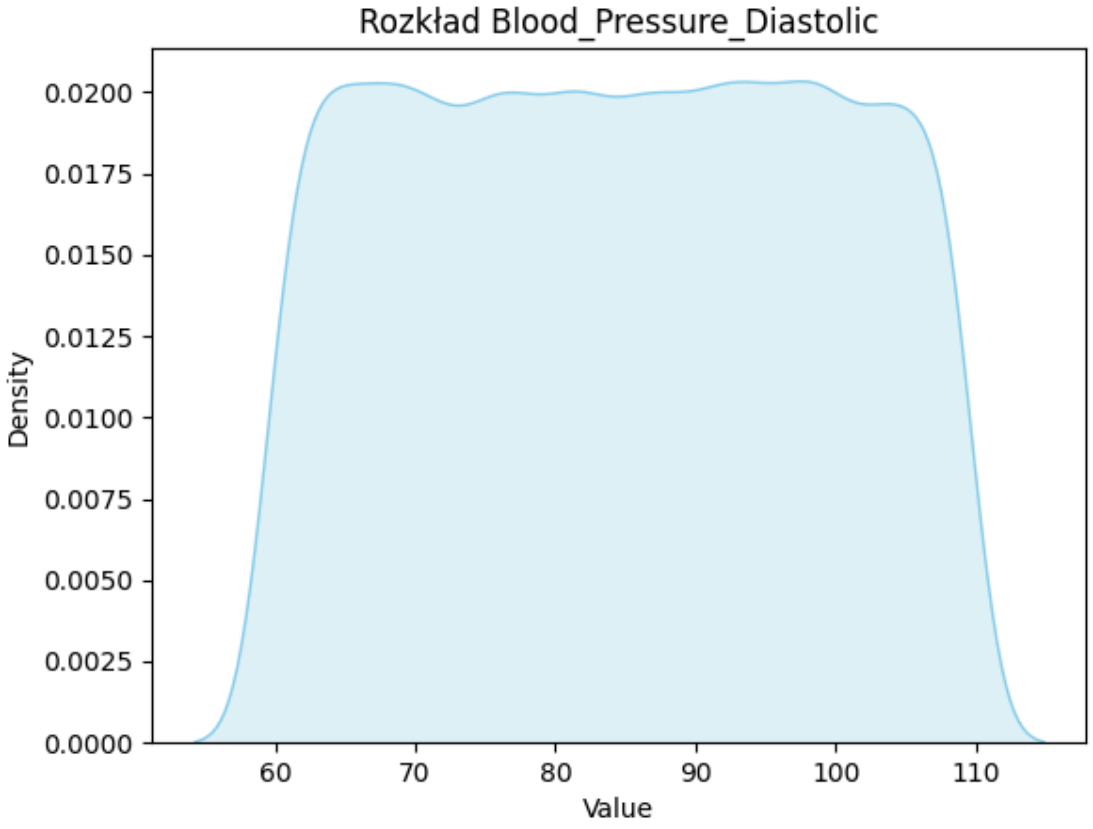

Rozkład Age
Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
30
40
50
60
70
80
Value

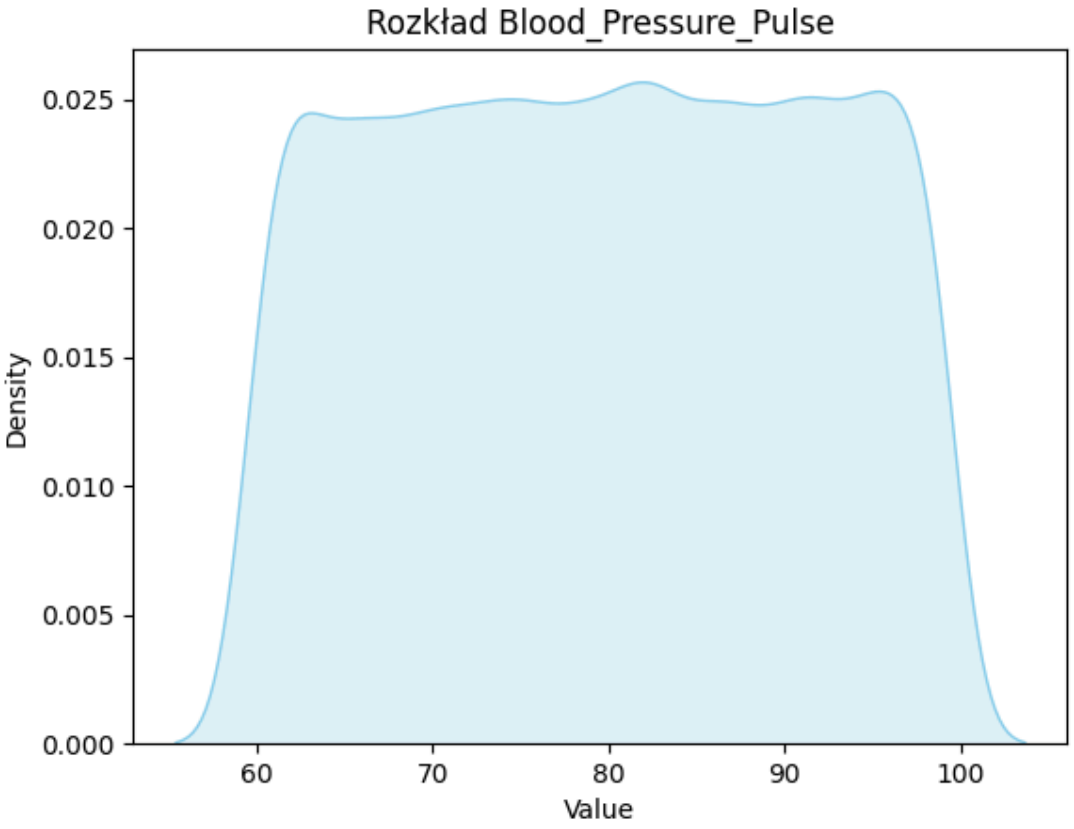

Rozkład Alanine_Aminotransferase_Level
Density
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
10
20
30
40
Value

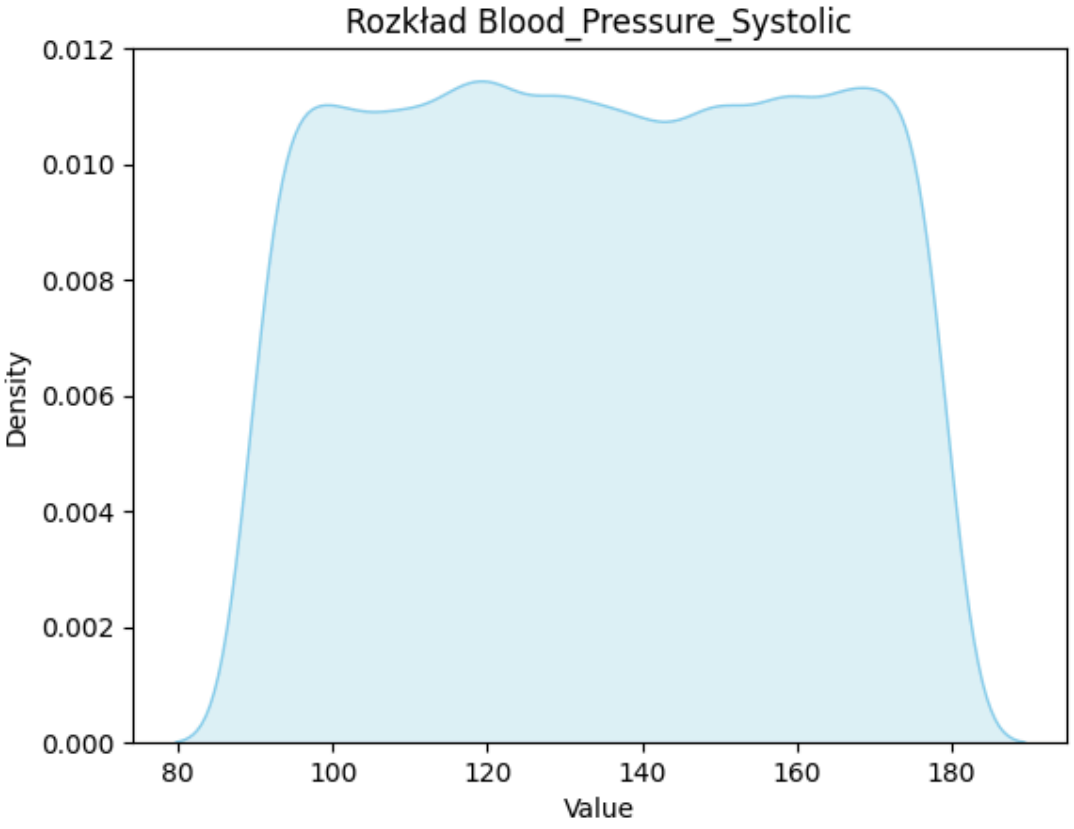

Rozkład Albumin_Level
Density
0.0
0.1
0.2
0.3
0.4
0.5
3.0
3.5
4.0
4.5
5.0
Value

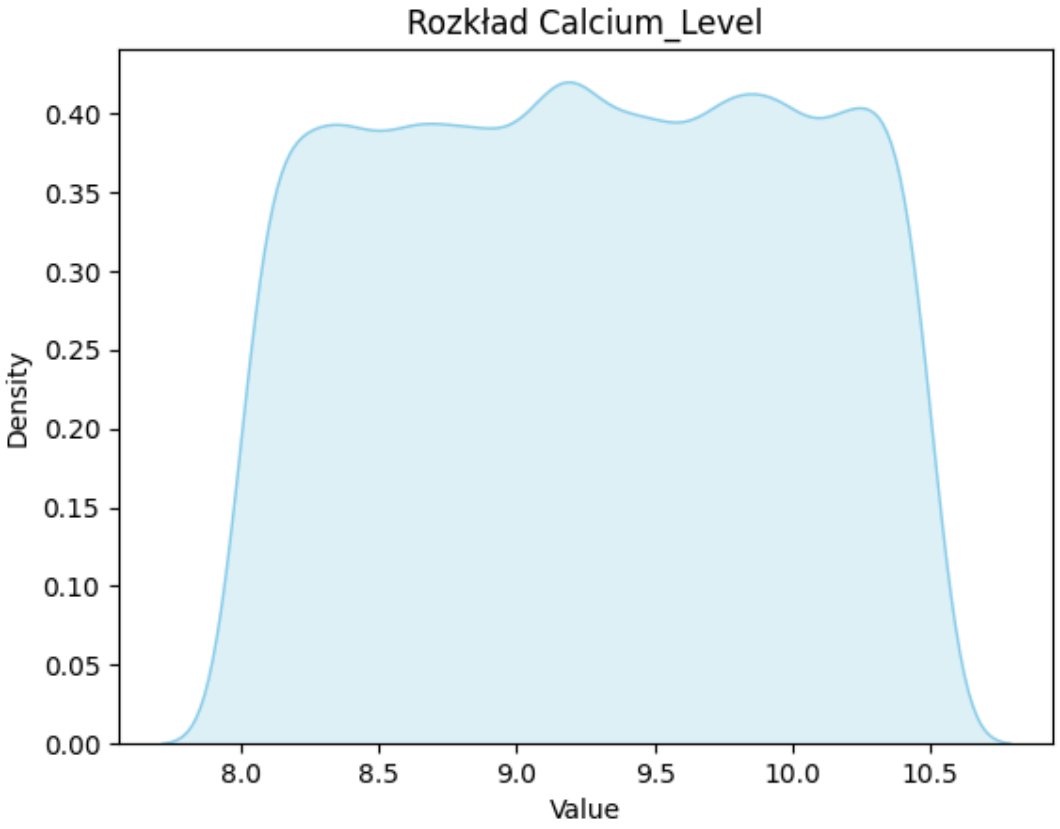

Rozkład Alkaline_Phosphatase_Level
Density
0.000
0.002
0.004
0.006
0.008
0.010
20
40
60
80
100
120
Value

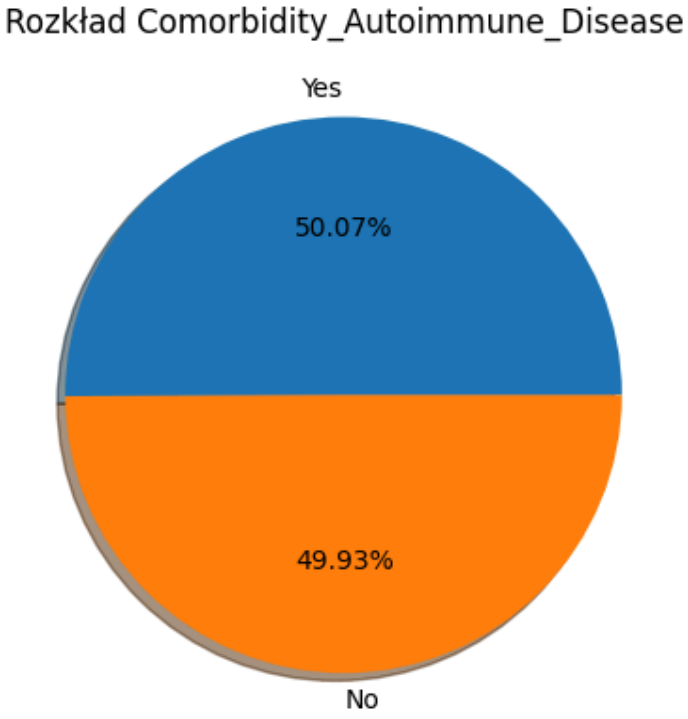

Rozkład Aspartate_Aminotransferase_Level
Density
0.025
0.020
0.015
0.010
0.005
0.000
10
20
30
40
50
Value

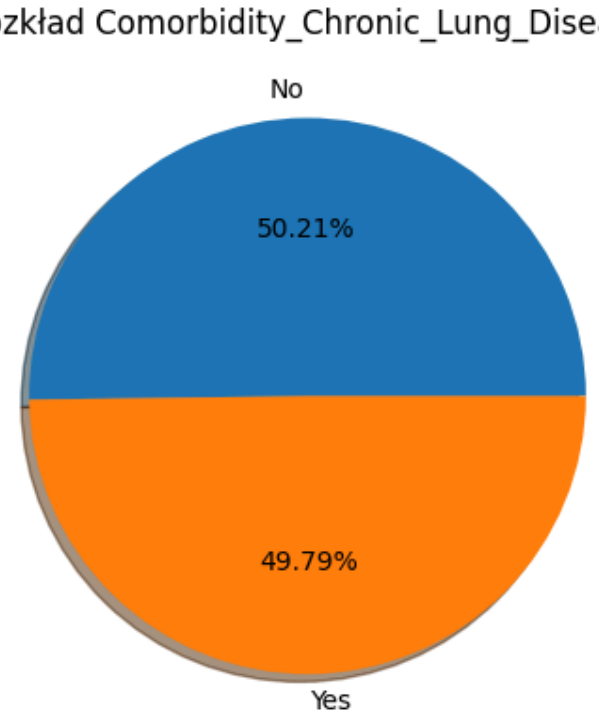

Rozkład Blood_Pressure_Diastolic
Density
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
60
70
80
90
100
110
Value

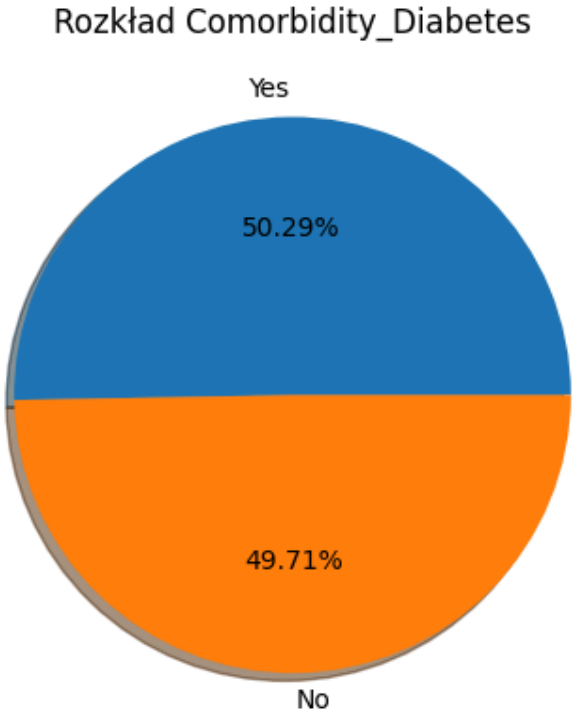

Rozkład Blood_Pressure_Pulse
Density
0.025
0.020
0.015
0.010
0.005
0.000
60
70
80
90
100
Value

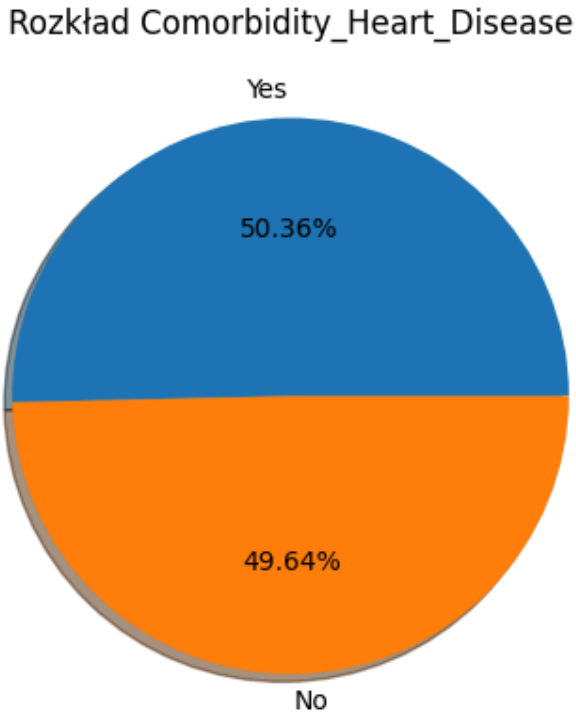

Rozkład Blood_Pressure_Systolic
Density
0.012
0.010
0.008
0.006
0.004
0.002
0.000
80
100
120
140
160
180
Value

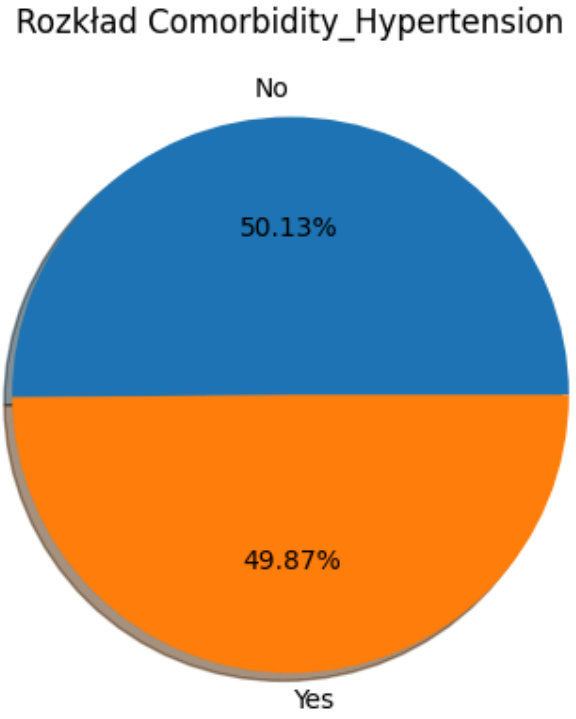

Rozkład Calcium_Level
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
8.0
8.5
9.0
9.5
10.0
10.5
Value

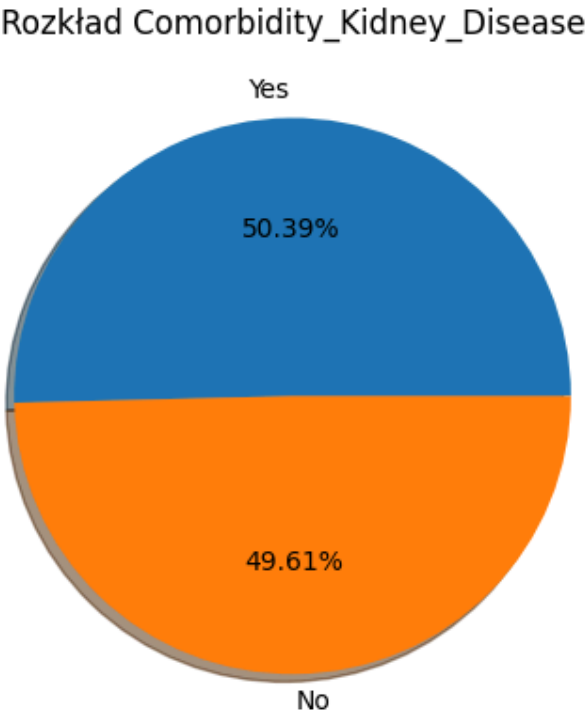

Rozkład Comorbidity_Autoimmune_Disease
Yes
50.07%
49.93%
No

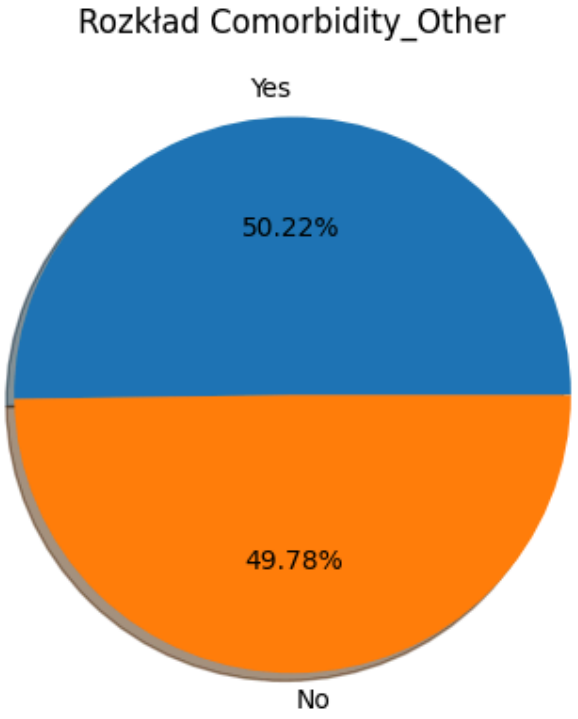

Rozkład Comorbidity_Chronic_Lung_Disease
No
50.21%
49.79%
Yes

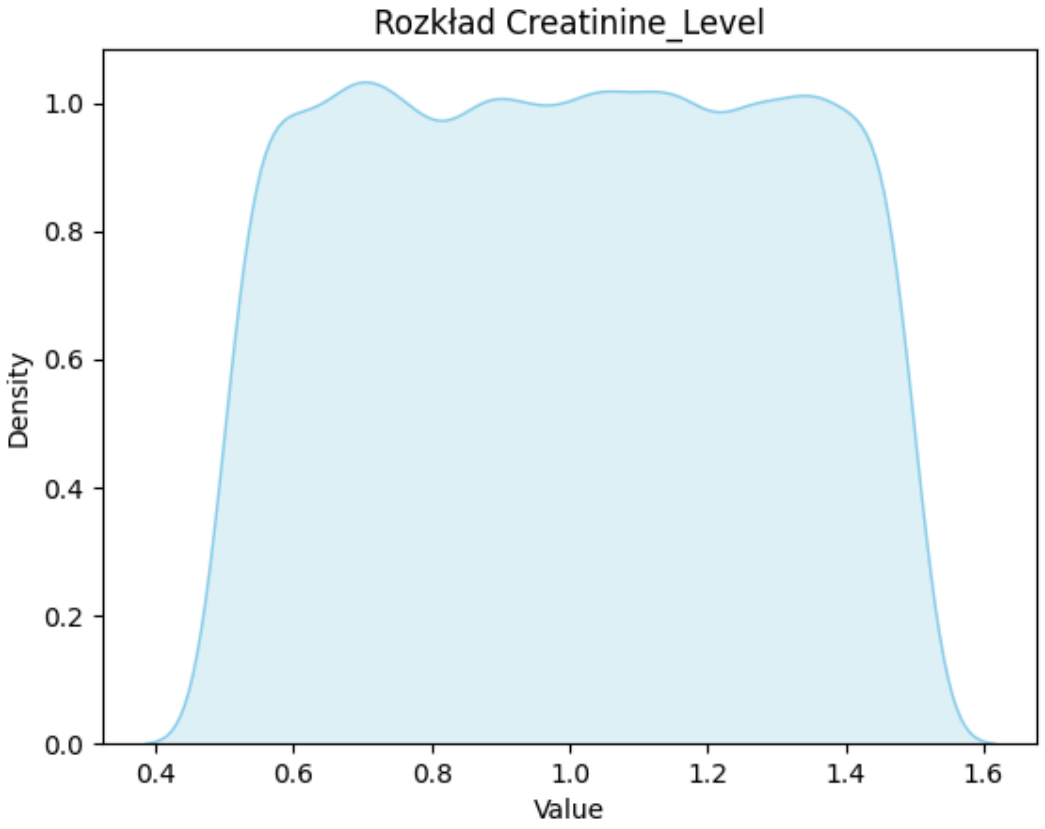

Rozkład Comorbidity_Diabetes
Yes
50.29%
49.71%
No

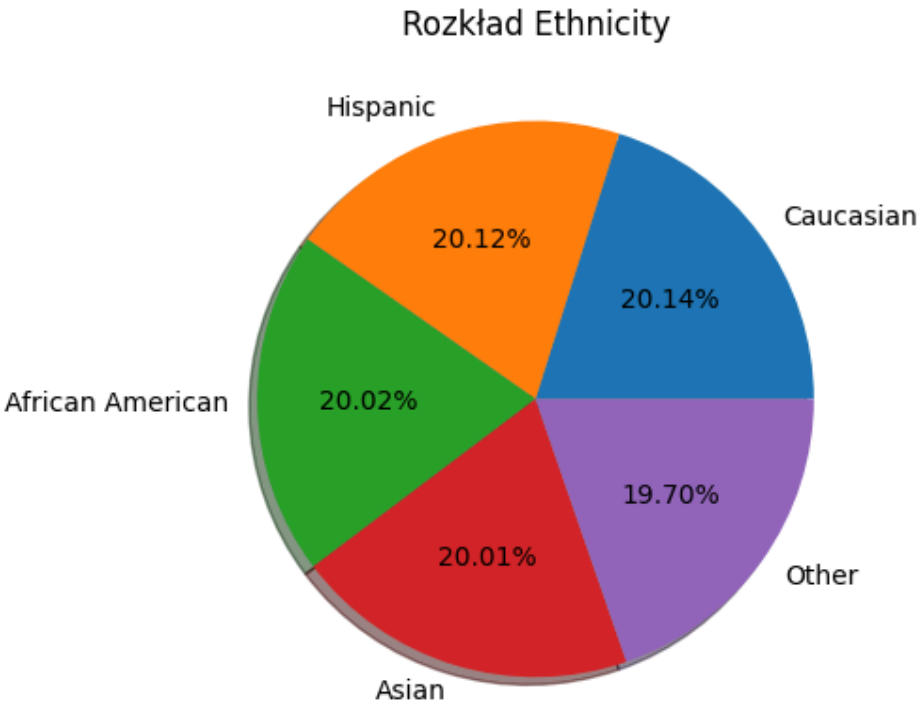

Rozkład Comorbidity_Heart_Disease
Yes
50.36%
49.64%
No

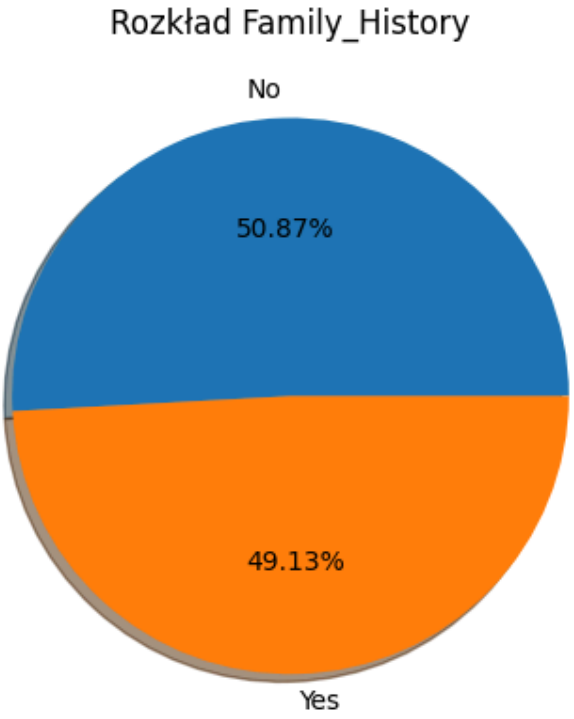

Rozkład Comorbidity_Hypertension
No
50.13%
49.87%
Yes

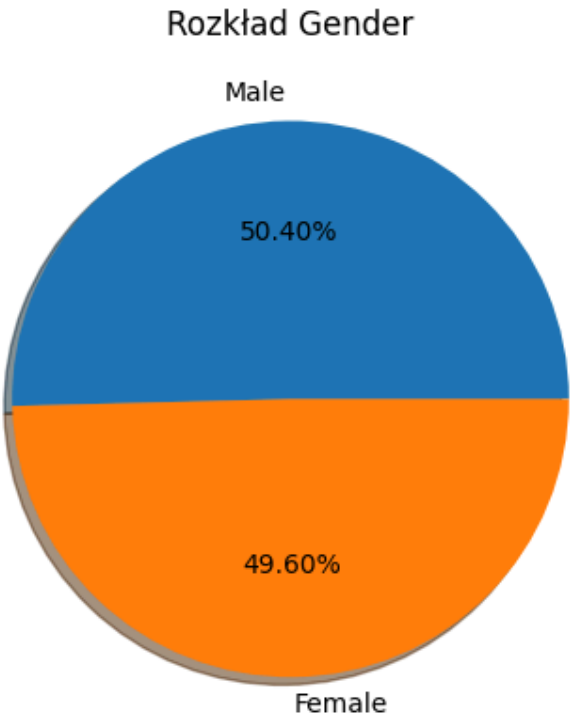

Rozkład Comorbidity_Kidney_Disease
Yes
50.39%
49.61%
No

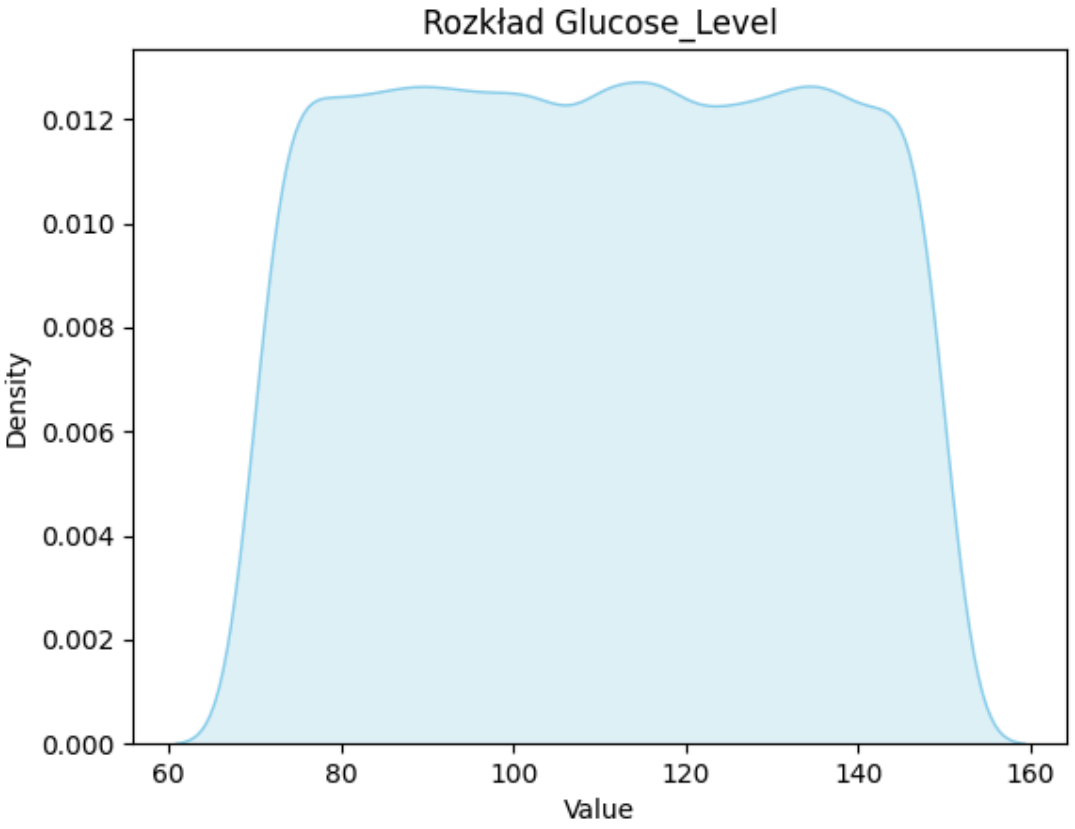

Rozkład Comorbidity_Other
Yes
50.22%
49.78%
No

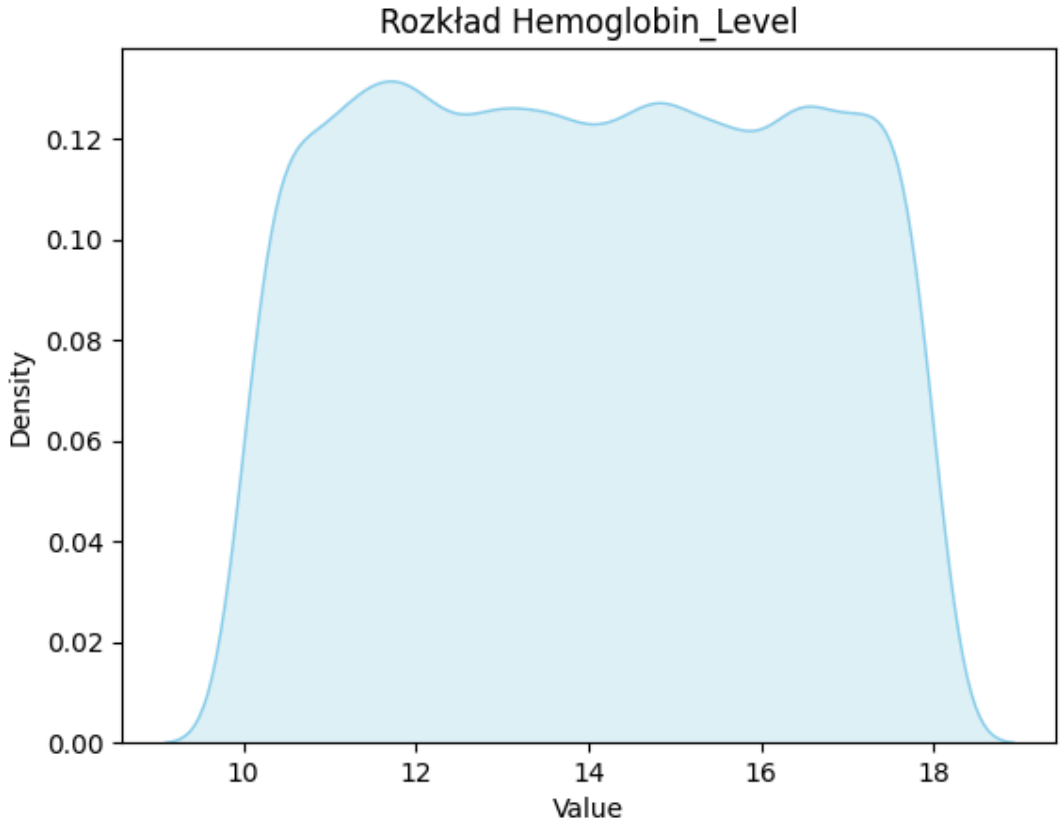

Rozkład Creatinine_Level
Density
Value
0.0
0.2
0.4
0.6
0.8
1.0
0.4
0.6
0.8
1.0
1.2
1.4
1.6

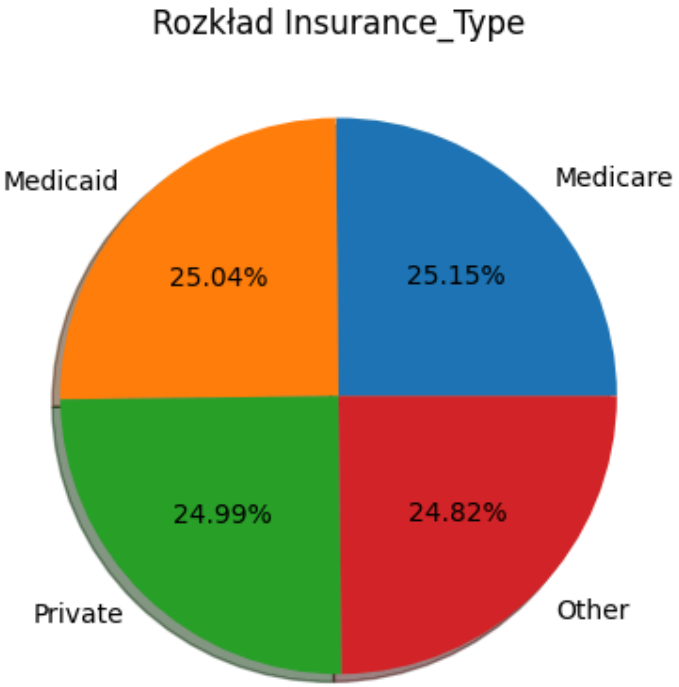

Rozkład Ethnicity
Hispanic
20.12%
Caucasian
20.14%
African American
20.02%
19.70%
20.01%
Other
Asian

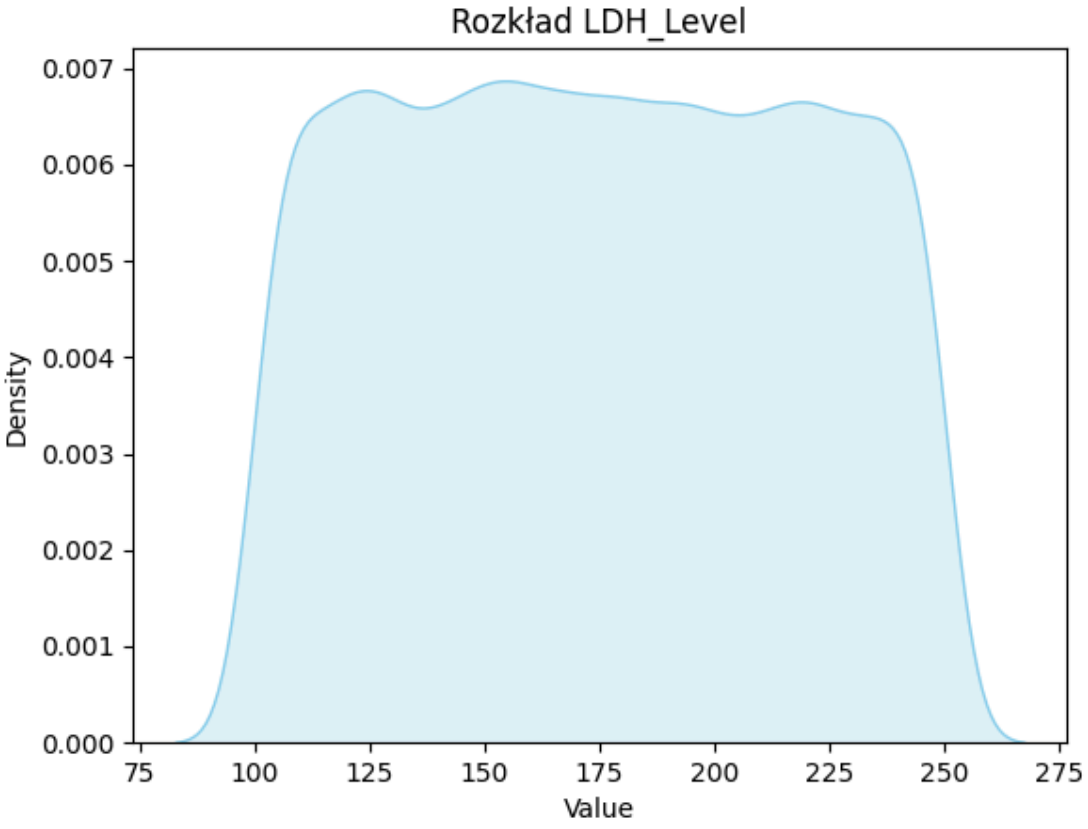

Rozkład Family_History
No
50.87%
49.13%
Yes

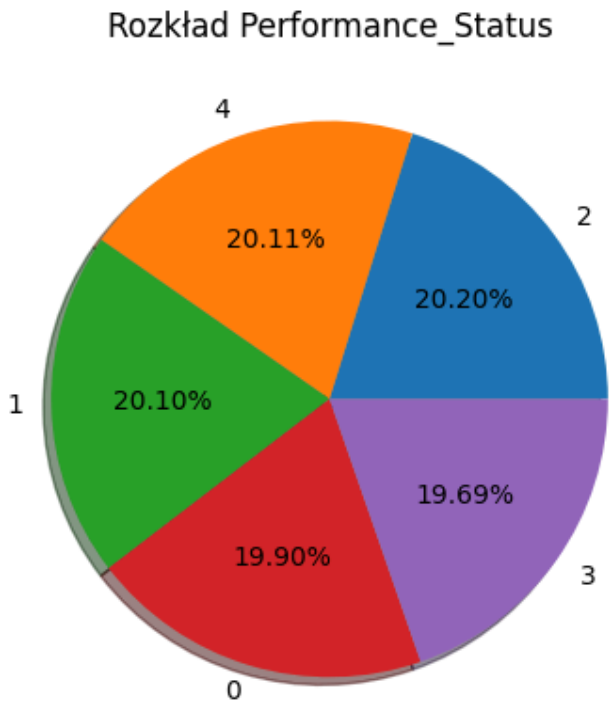

Rozkład Gender
Male
50.40%
49.60%
Female

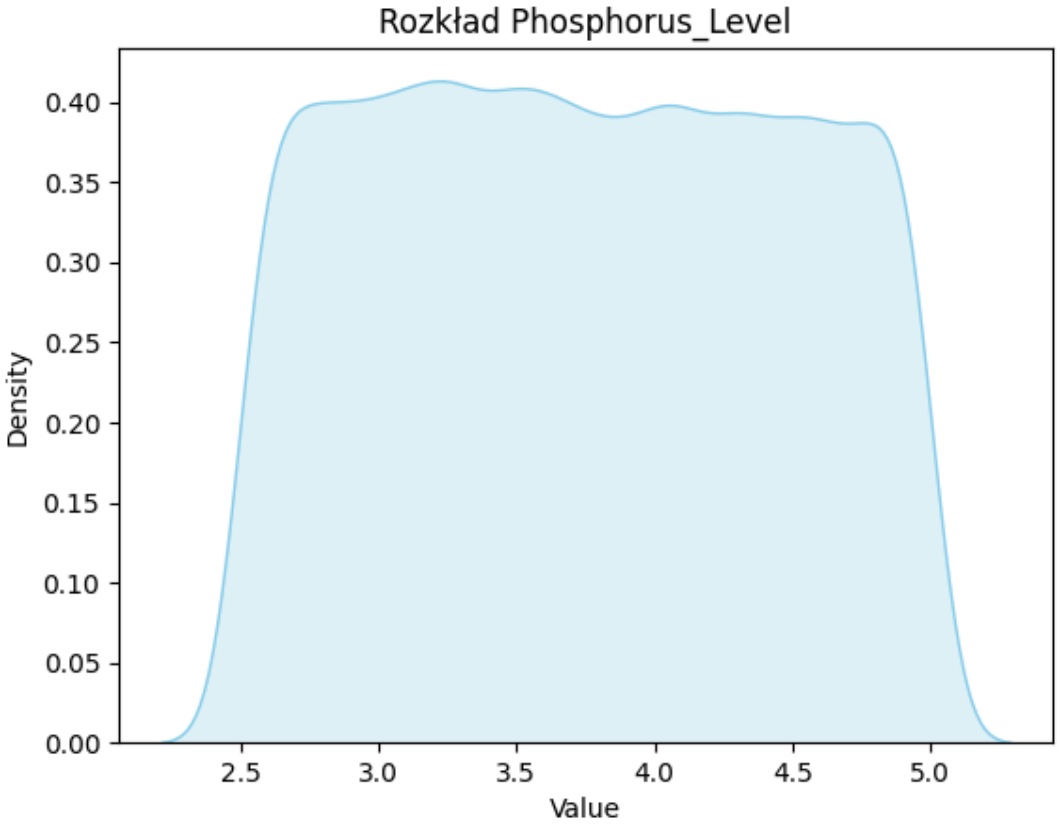

Rozkład Glucose_Level
Density
0.012
0.010
0.008
0.006
0.004
0.002
0.000
60
80
100
120
140
160
Value

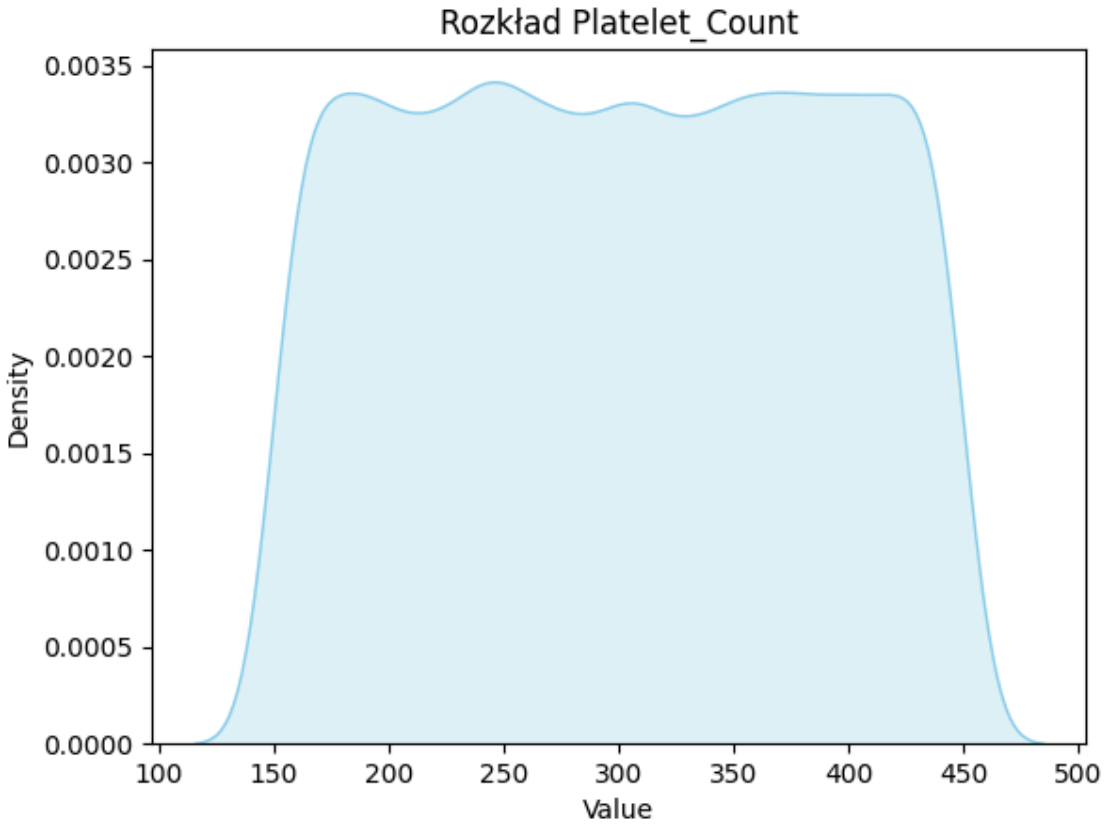

Rozkład Hemoglobin_Level
Density
0.12
0.10
0.08
0.06
0.04
0.02
0.00
10
12
14
16
18
Value

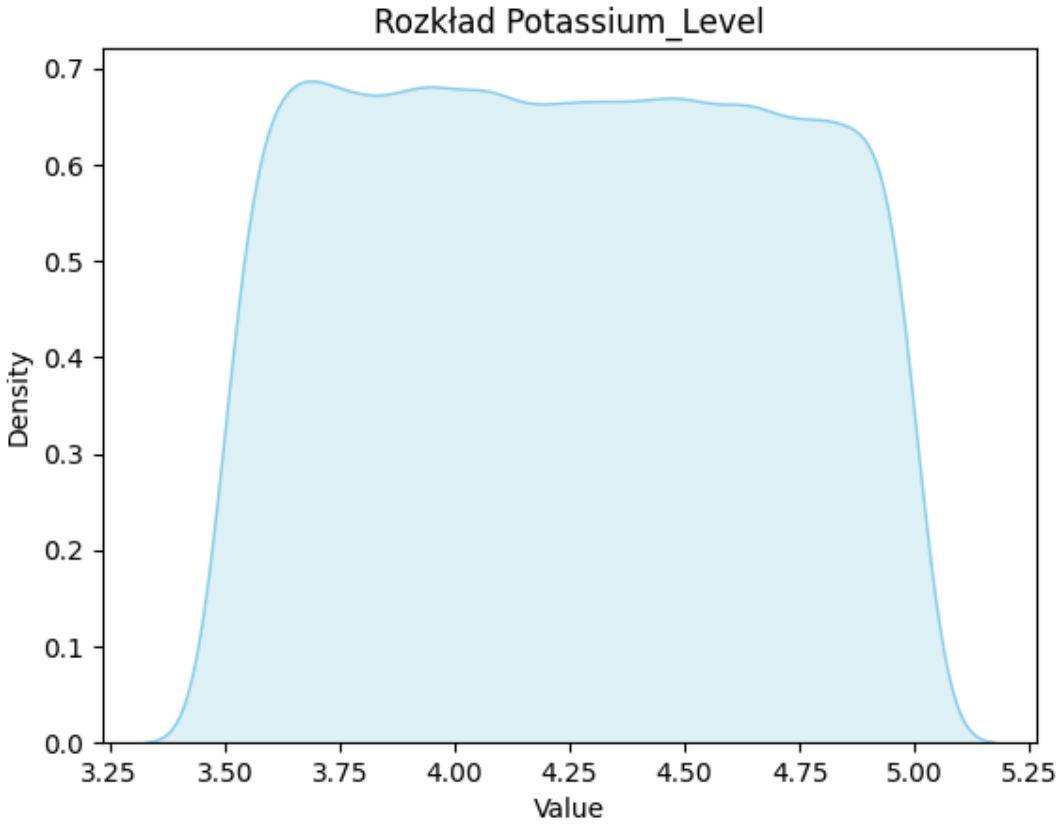

Rozkład Insurance_Type
Medicaid
Medicare
25.04%
25.15%
24.99%
24.82%
Private
Other

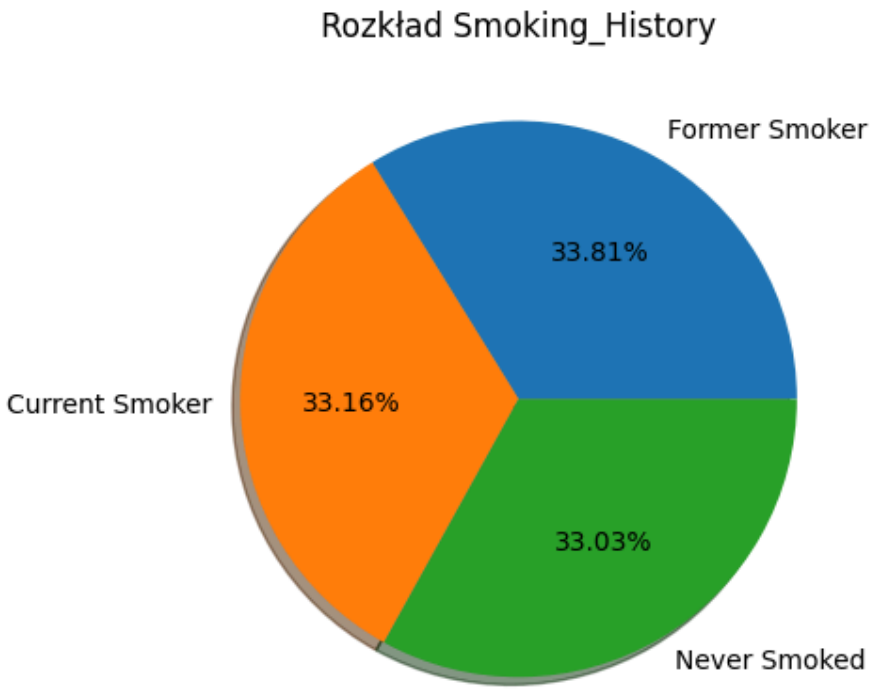

Rozkład LDH_Level
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
Density
75
100
125
150
175
200
225
250
275
Value

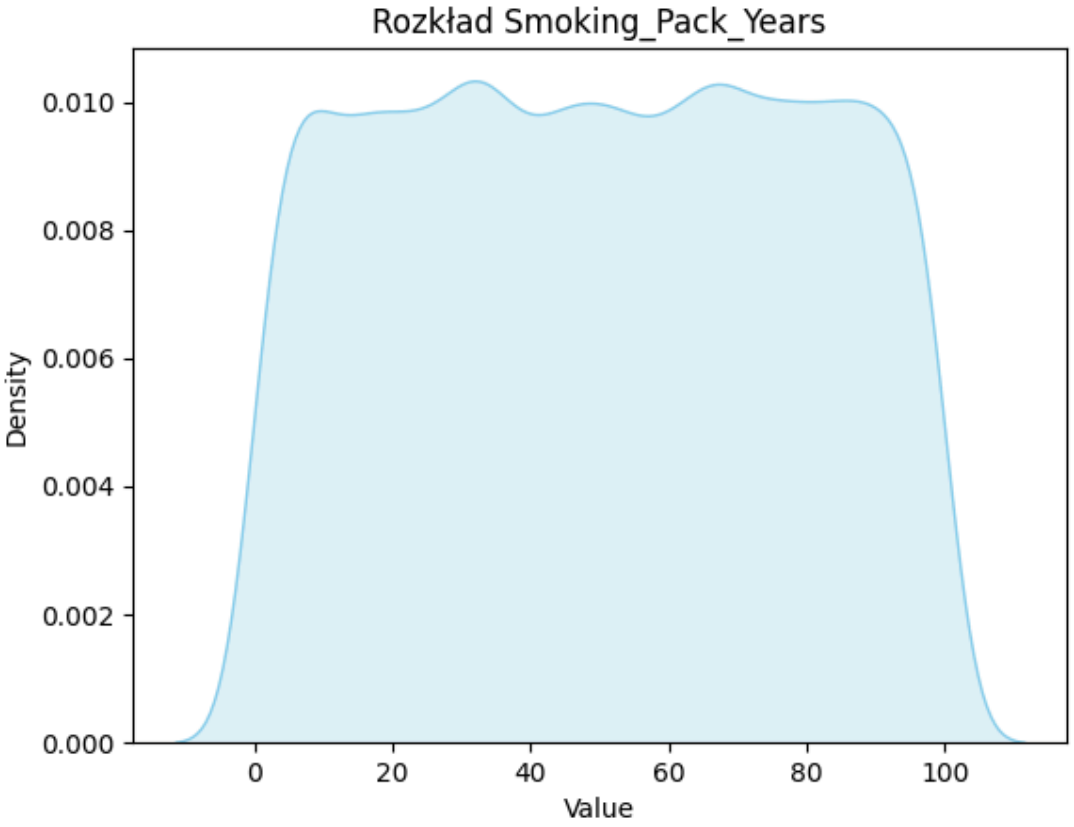

Rozkład Performance_Status
4
2
1
3
0
20.11%
20.20%
20.10%
19.69%
19.90%


Rozkład Phosphorus_Level
Density
Value
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
2.5
3.0
3.5
4.0
4.5
5.0

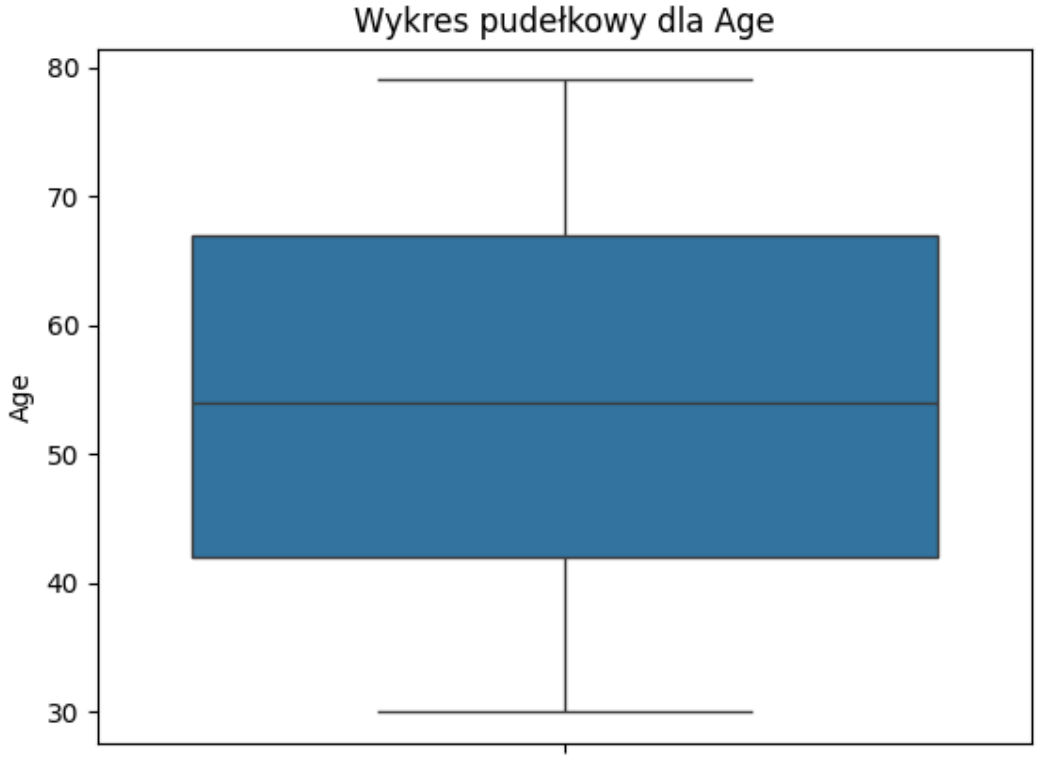

Rozkład Platelet_Count
Density
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
100
150
200
250
300
350
400
450
500
Value

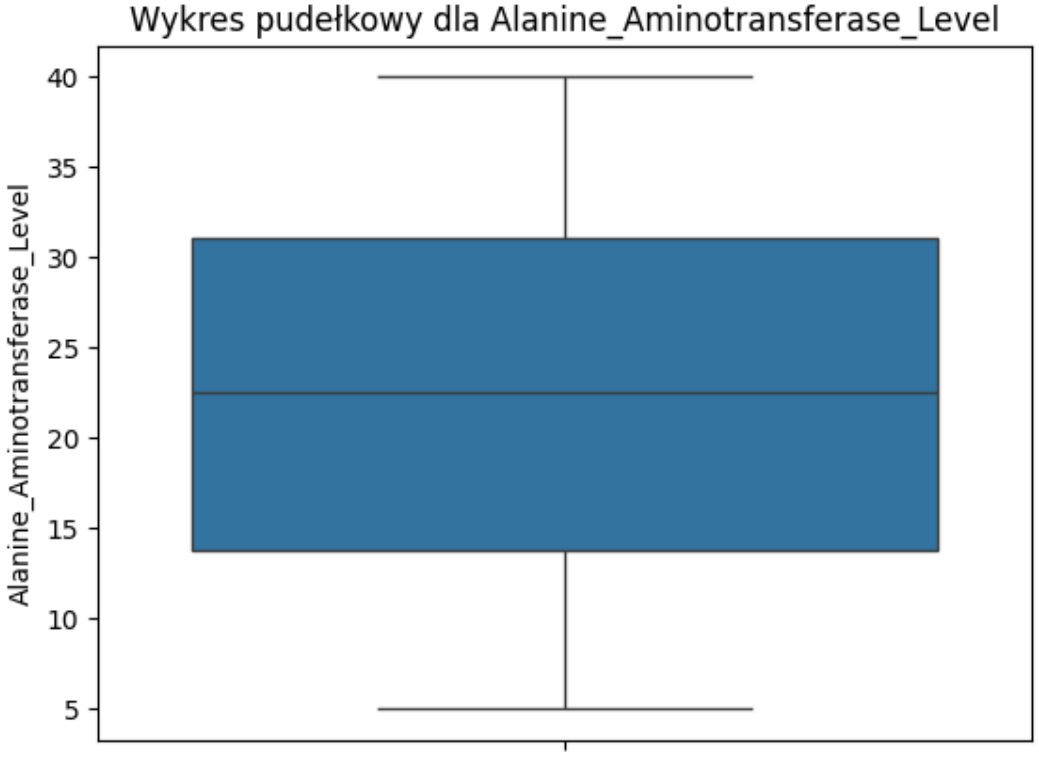

Rozkład Potassium_Level
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
3.25
3.50
3.75
4.00
4.25
4.50
4.75
5.00
5.25
Value

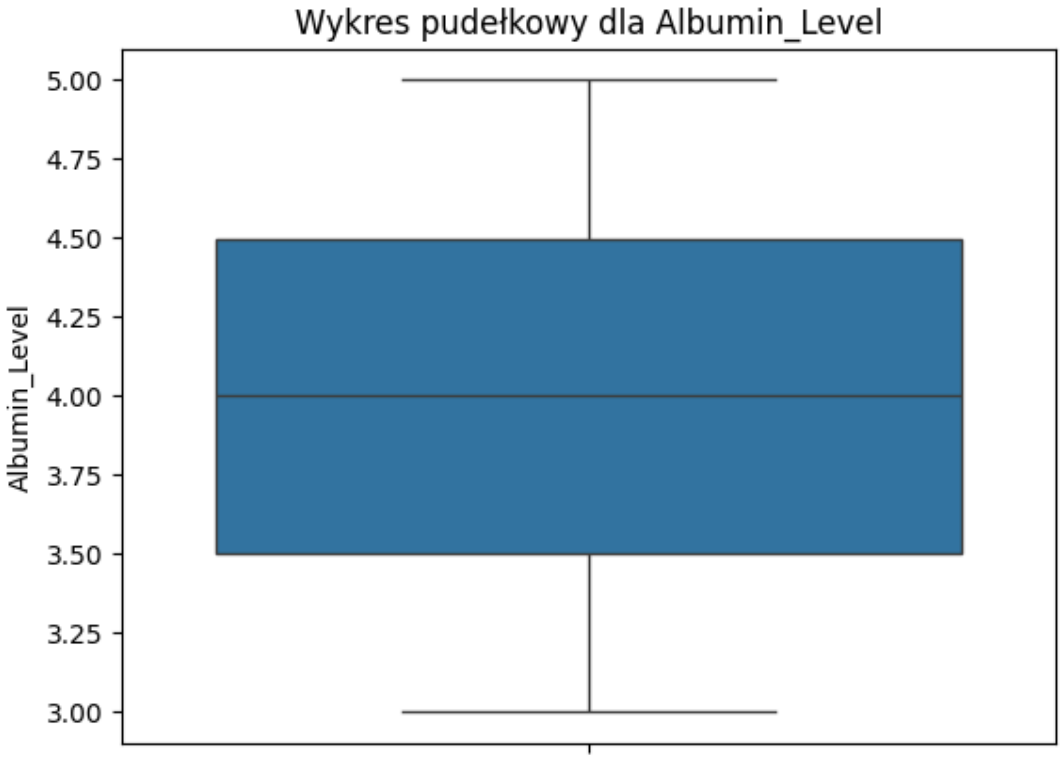

Rozkład Smoking_History
Former Smoker
33.81%
Current Smoker
33.16%
33.03%
Never Smoked

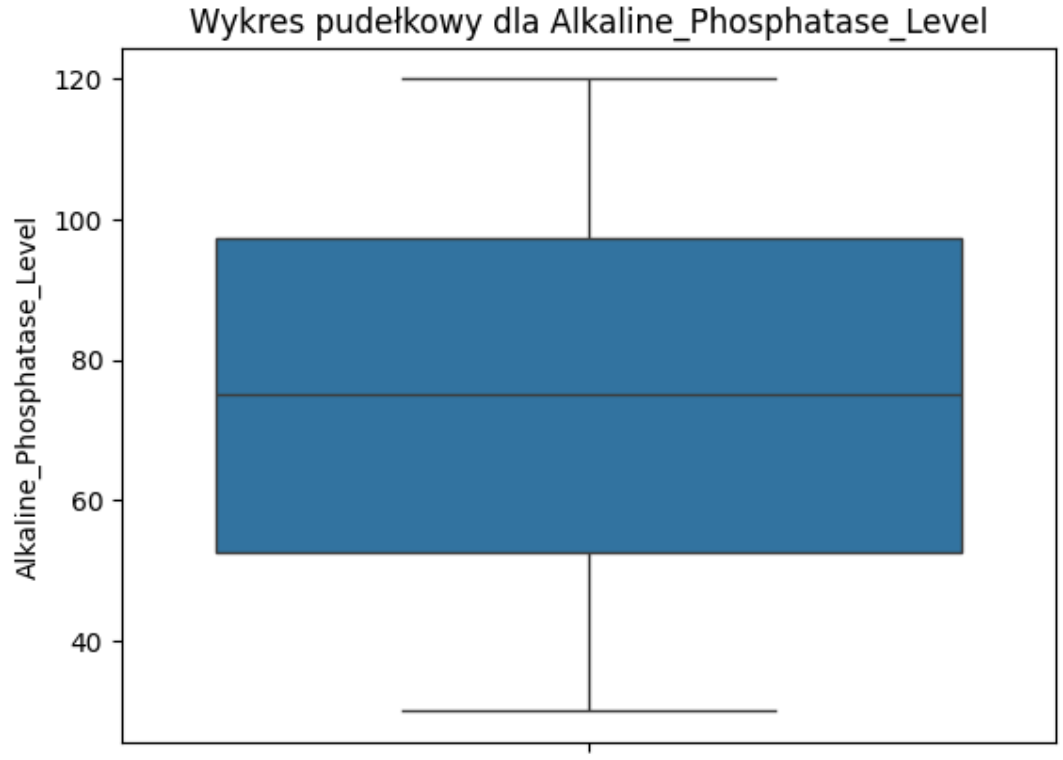

Rozkład Smoking_Pack_Years
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
20
40
60
80
100
Value

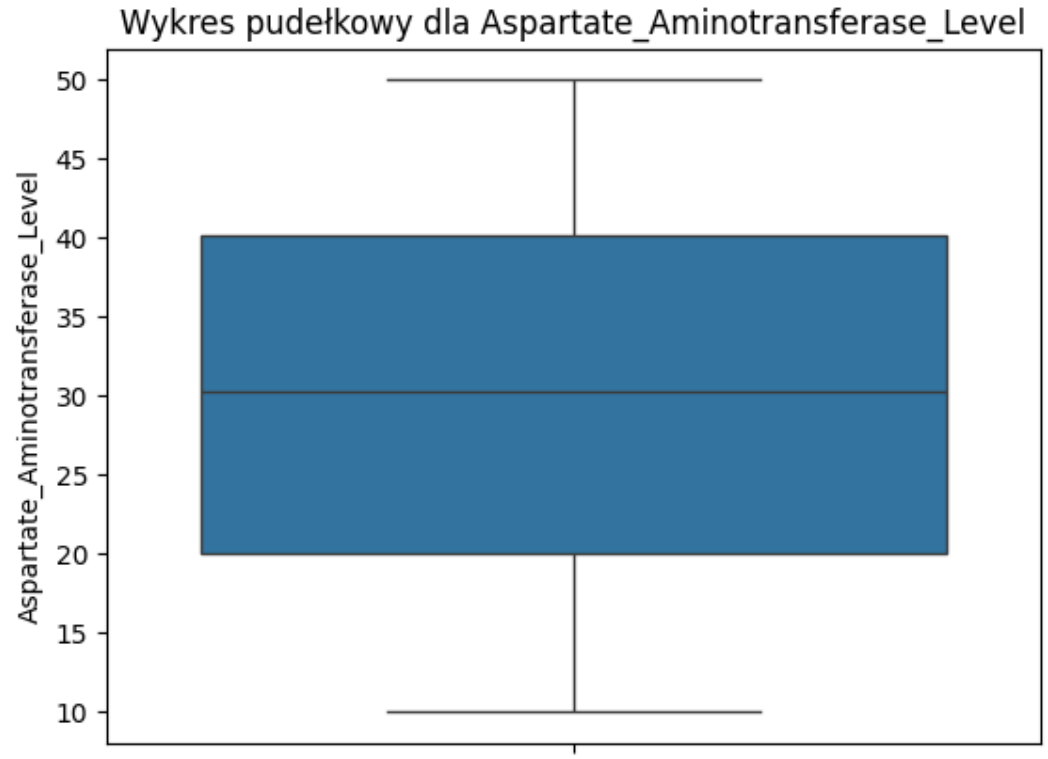
W dalszej części można też zauważyć wykresy pudełkowe dla wartości numerycznych które wydają się podobne do siebie. Można na nich zobaczyć, że minimalna wartość, wartość w 1/4, wartość w połowie, wartość w 3/4 i wartość maksymalna są odpowiednio w podobnych odległościach.

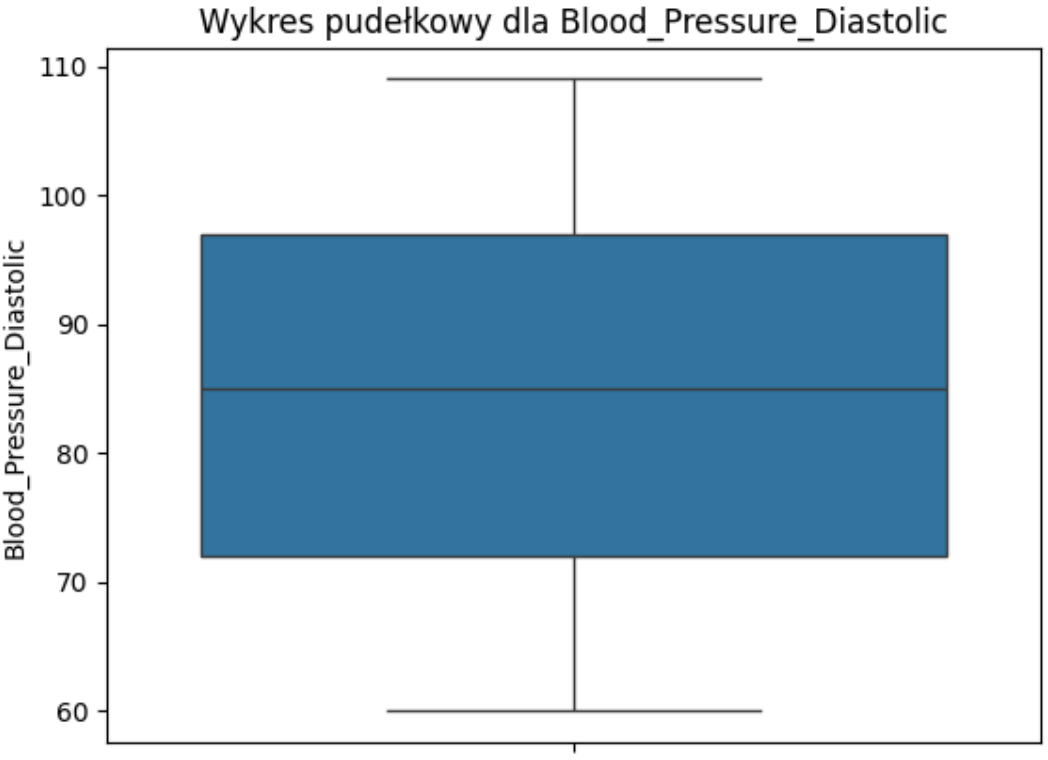

Wykres pudełkowy dla Age
Age
80
70
60
50
40
30

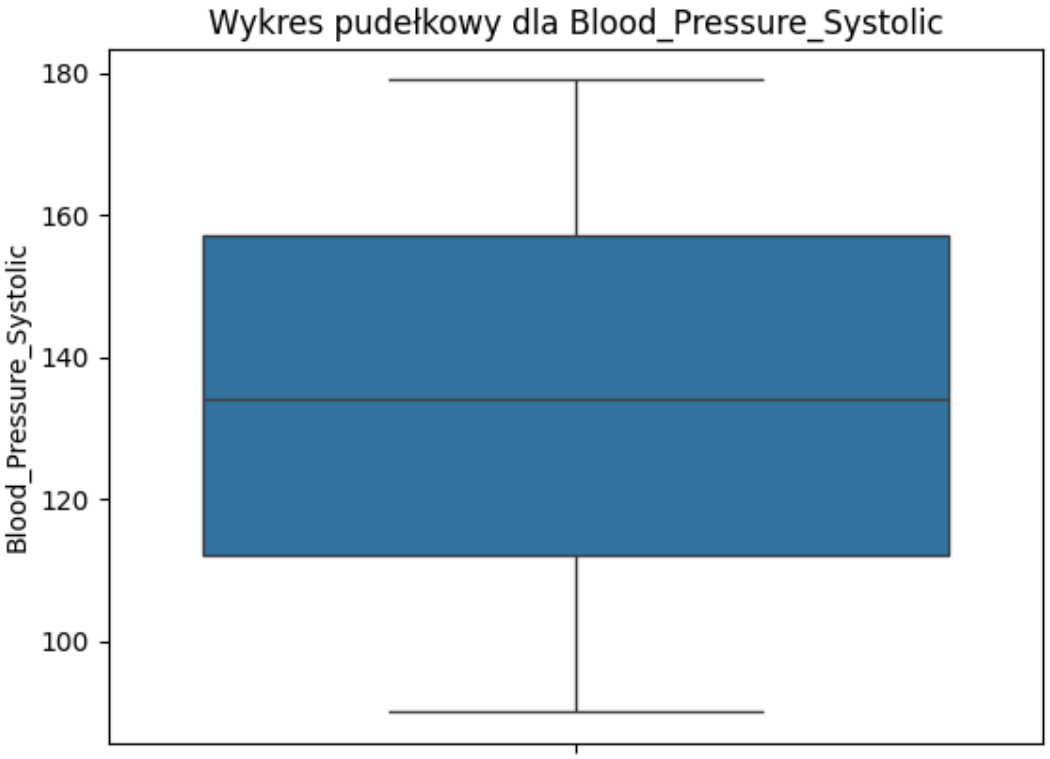

Wykres pudełkowy dla Alanine_Aminotransferase_Level
Alanine_Aminotransferase_Level
40
35
30
25
20
15
10
5

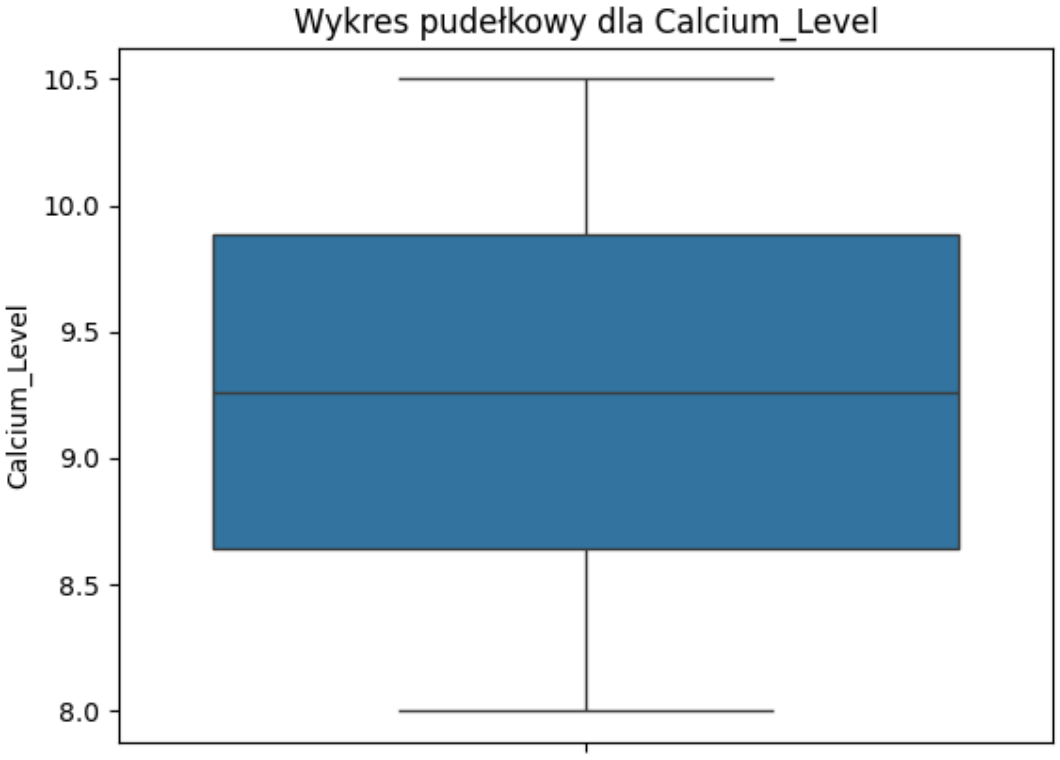

Wykres pudełkowy dla Albumin_Level
Albumin_Level
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00

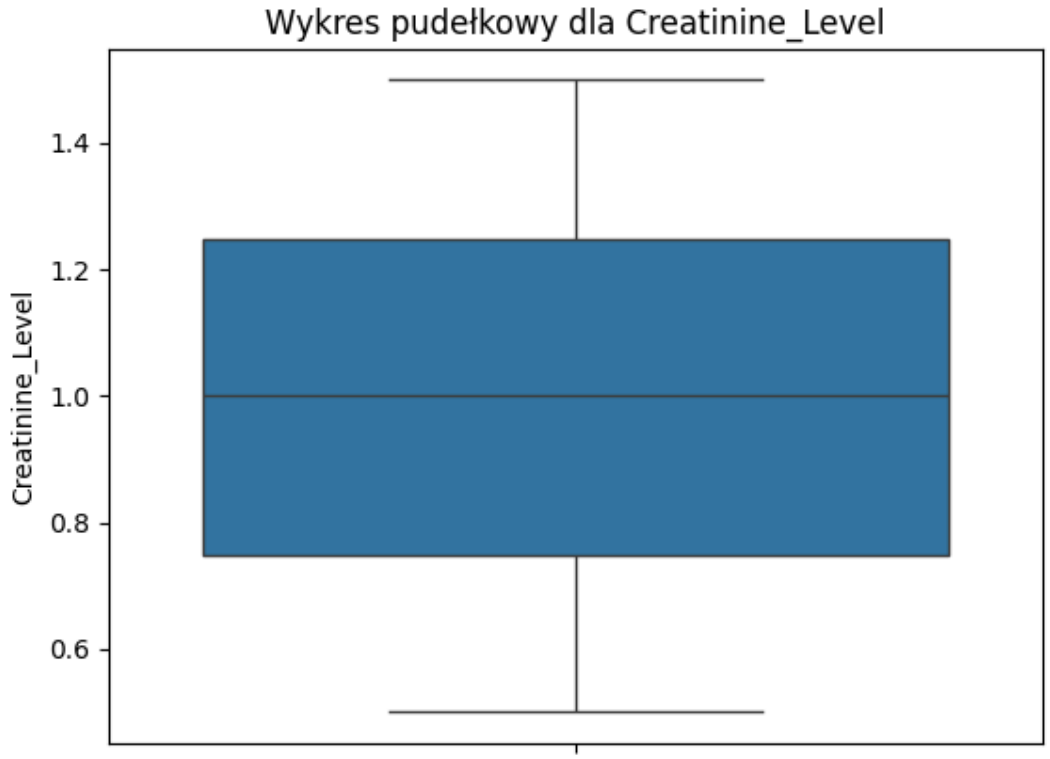

Wykres pudełkowy dla Alkaline_Phosphatase_Level
Alkaline_Phosphatase_Level
120
100
80
60
40

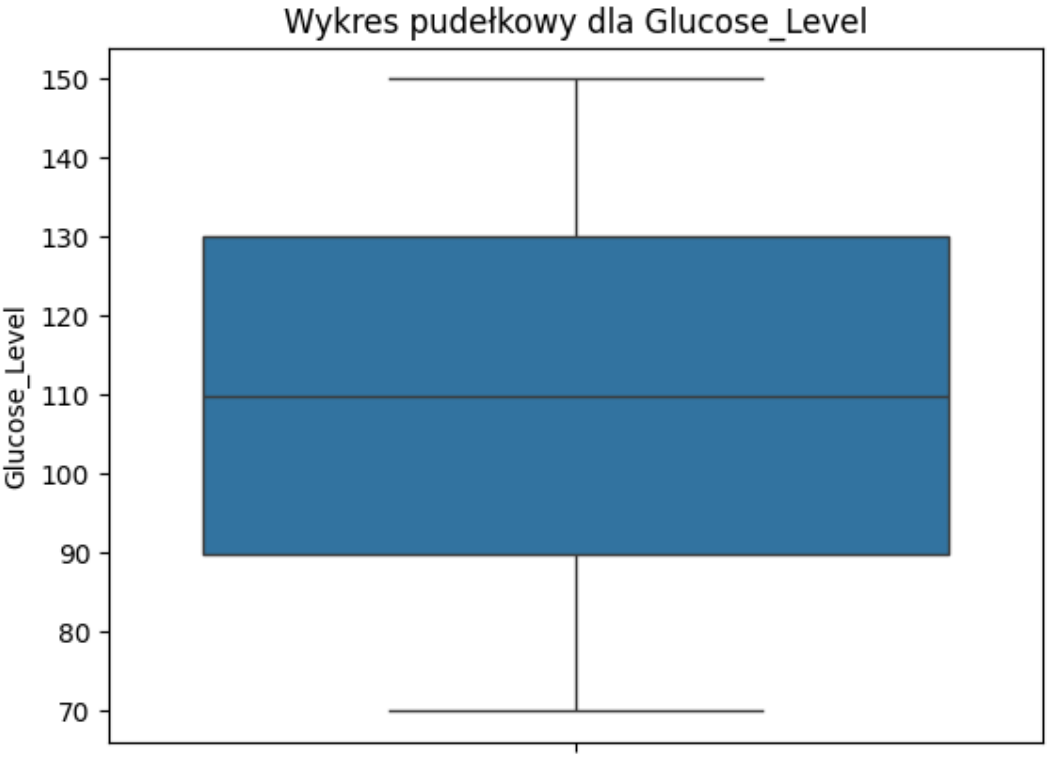

Wykres pudełkowy dla Aspartate_Aminotransferase_Level
Aspartate_Aminotransferase_Level
50
45
40
35
30
25
20
15
10

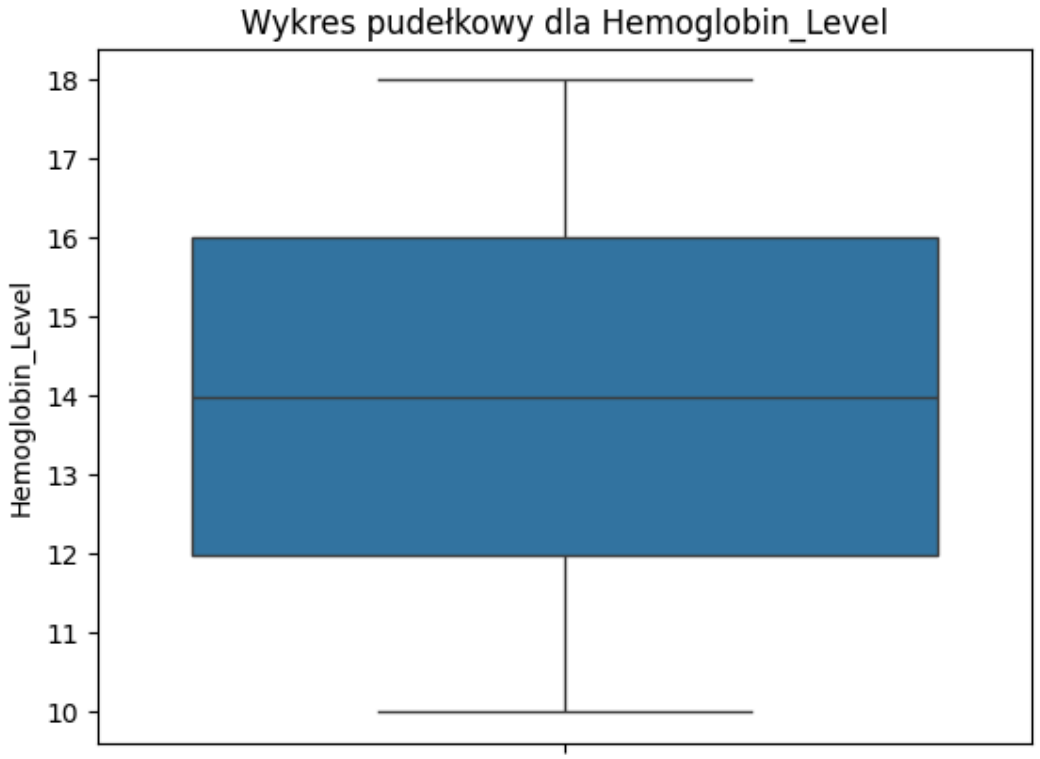

Wykres pudełkowy dla Blood_Pressure_Diastolic
Blood_Pressure_Diastolic
110
100
90
80
70
60

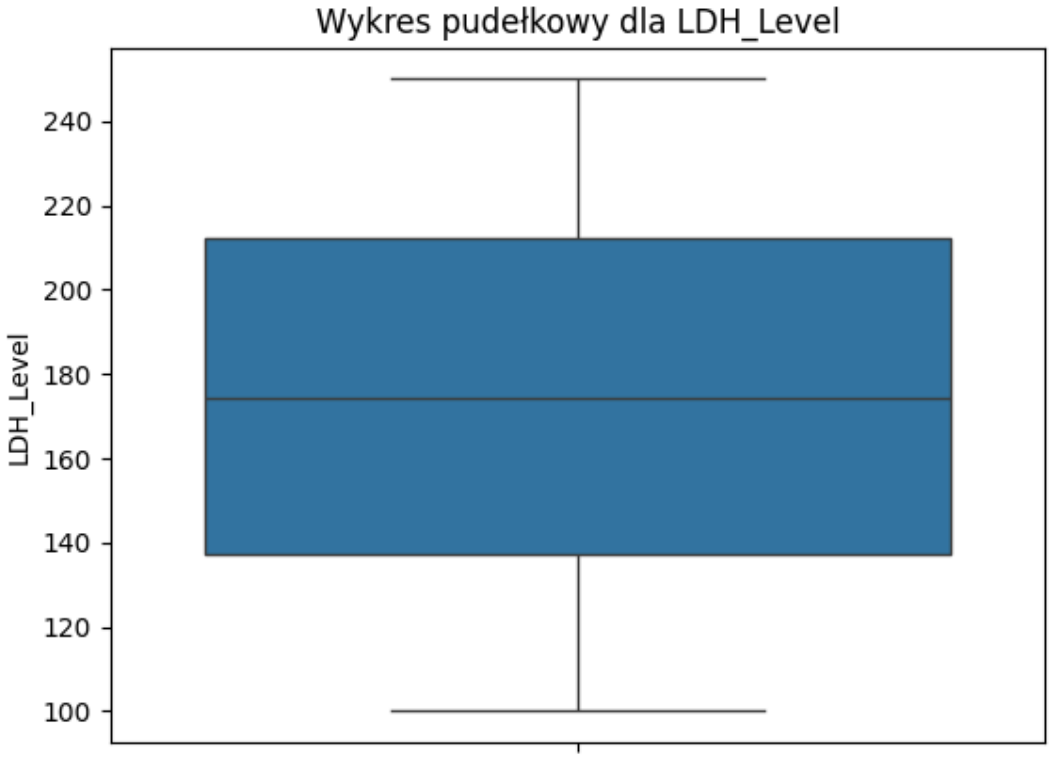

Wykres pudełkowy dla Blood_Pressure_Systolic
Blood_Pressure_Systolic
180
160
140
120
100

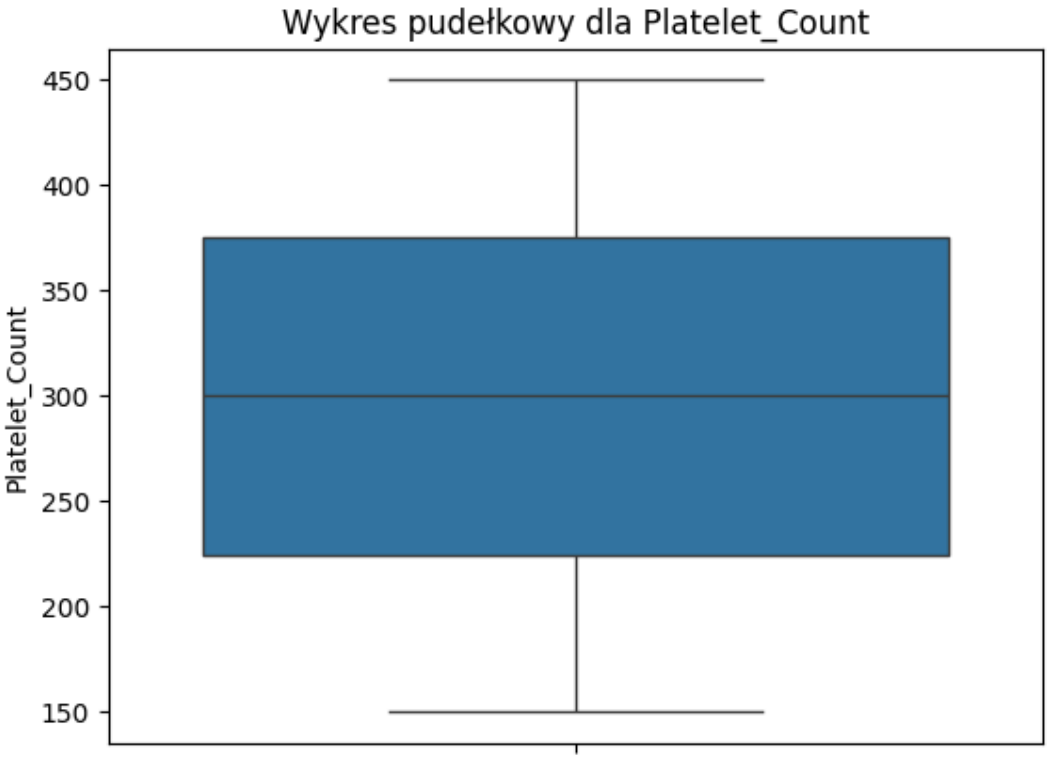

Wykres pudełkowy dla Calcium_Level
Calcium_Level
10.5
10.0
9.5
9.0
8.5
8.0

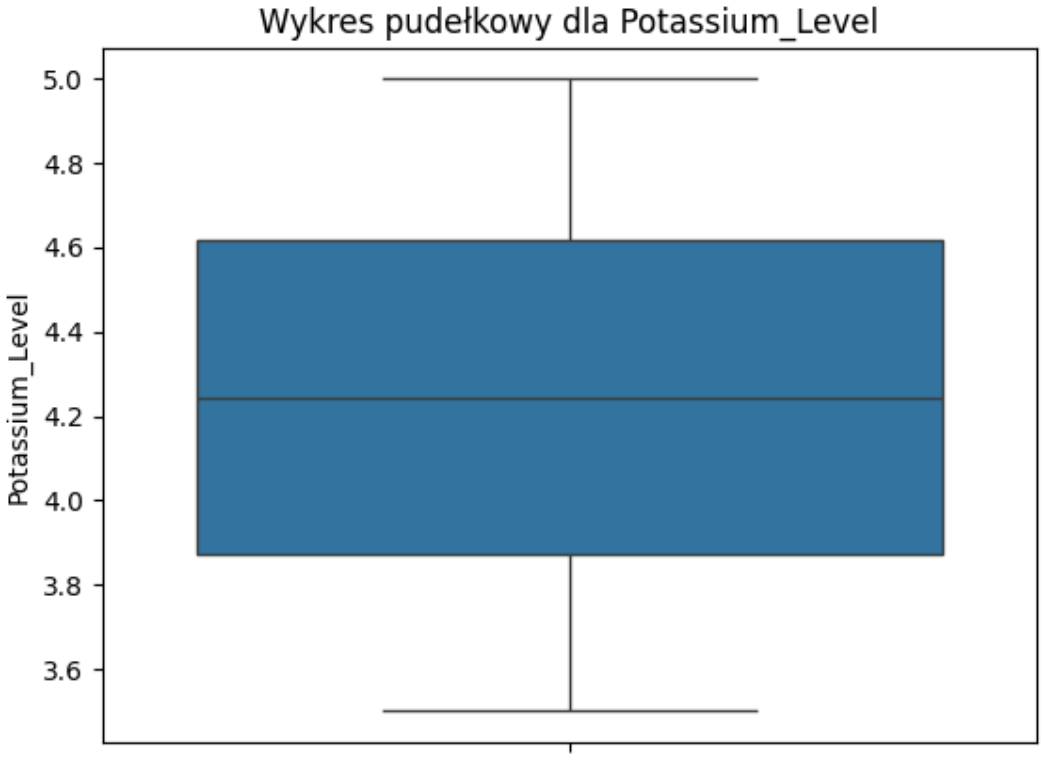

Wykres pudełkowy dla Creatinine_Level
Creatinine_Level
1.4
1.2
1.0
0.8
0.6

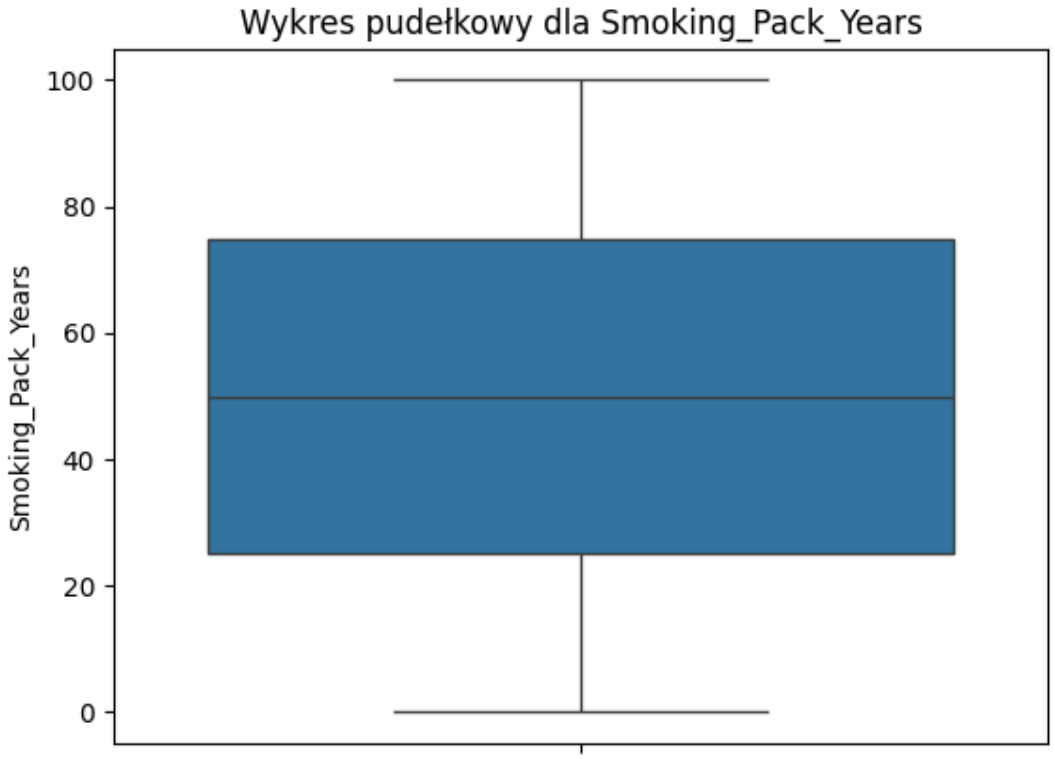

Wykres pudełkowy dla Glucose_Level
Glucose_Level
150
140
130
120
110
100
90
80
70

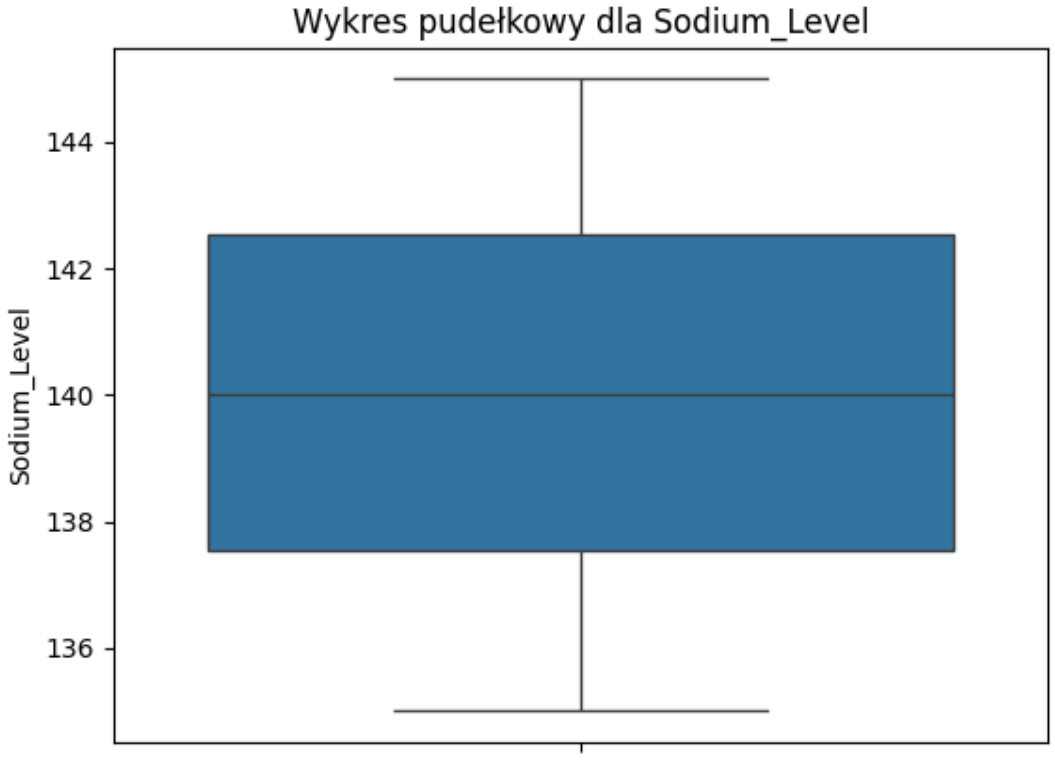

Wykres pudełkowy dla Hemoglobin_Level
Hemoglobin_Level
18
17
16
15
14
13
12
11
10

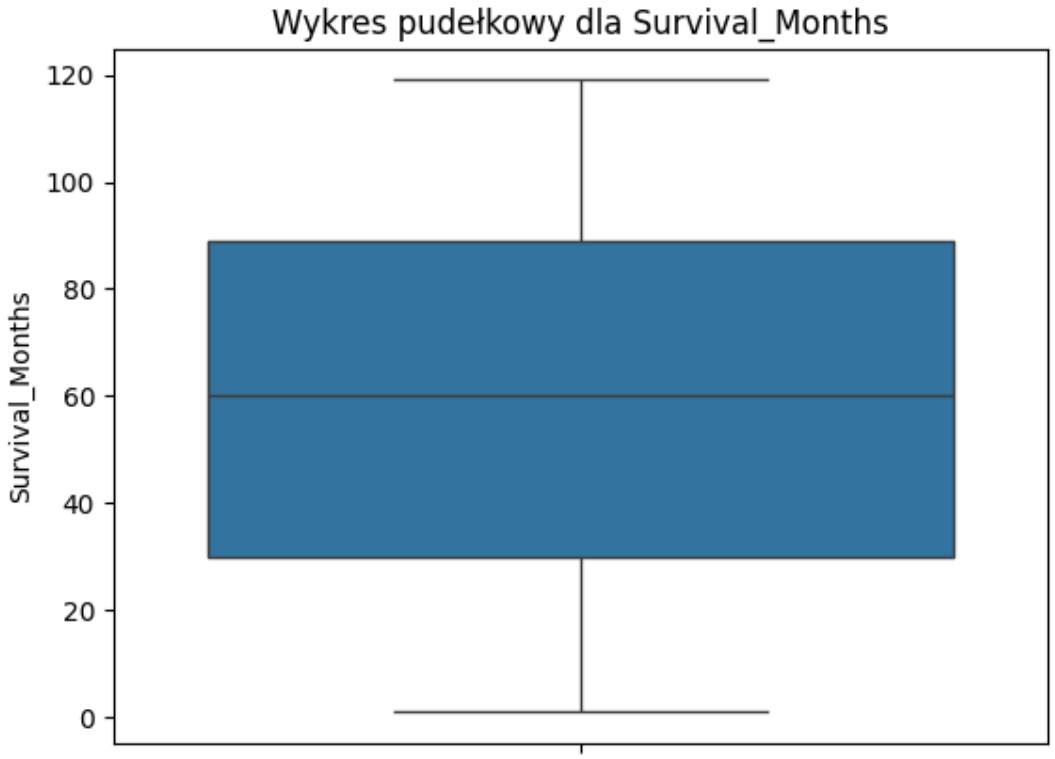

Wykres pudełkowy dla LDH_Level
LDH_Level
240
220
200
180
160
140
120
100

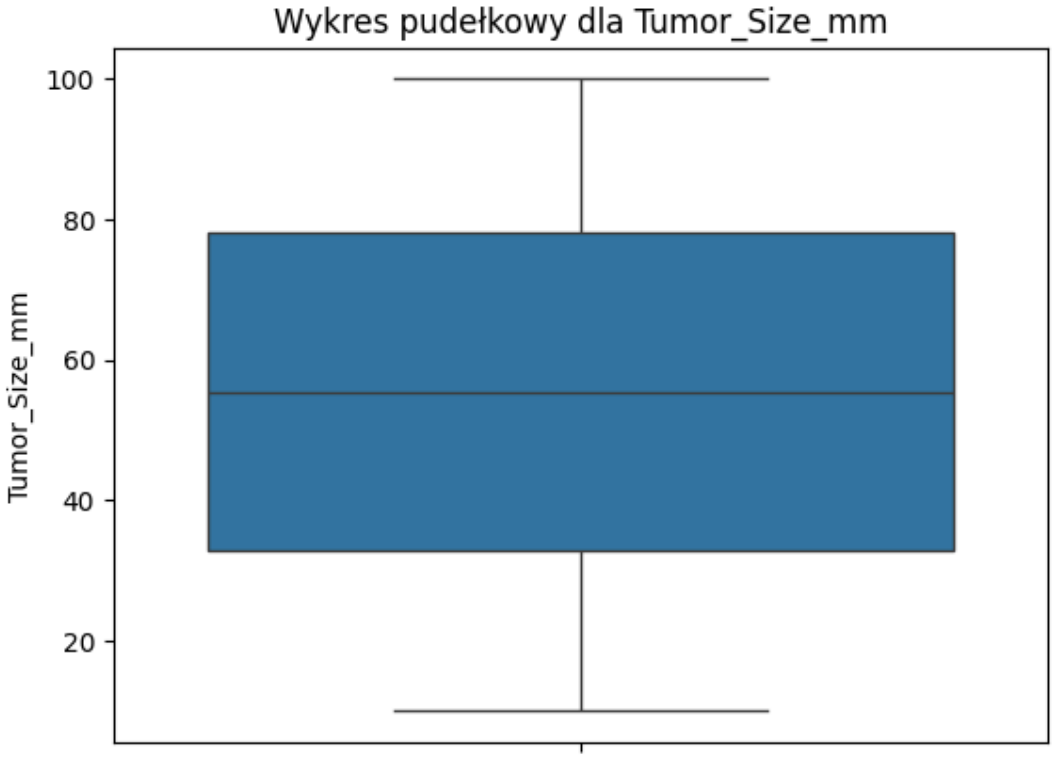

Wykres pudełkowy dla Platelet_Count
Platelet_Count
450
400
350
300
250
200
150


Wykres pudełkowy dla Potassium_Level
Potassium_Level
5.0
4.8
4.6
4.4
4.2
4.0
3.8
3.6

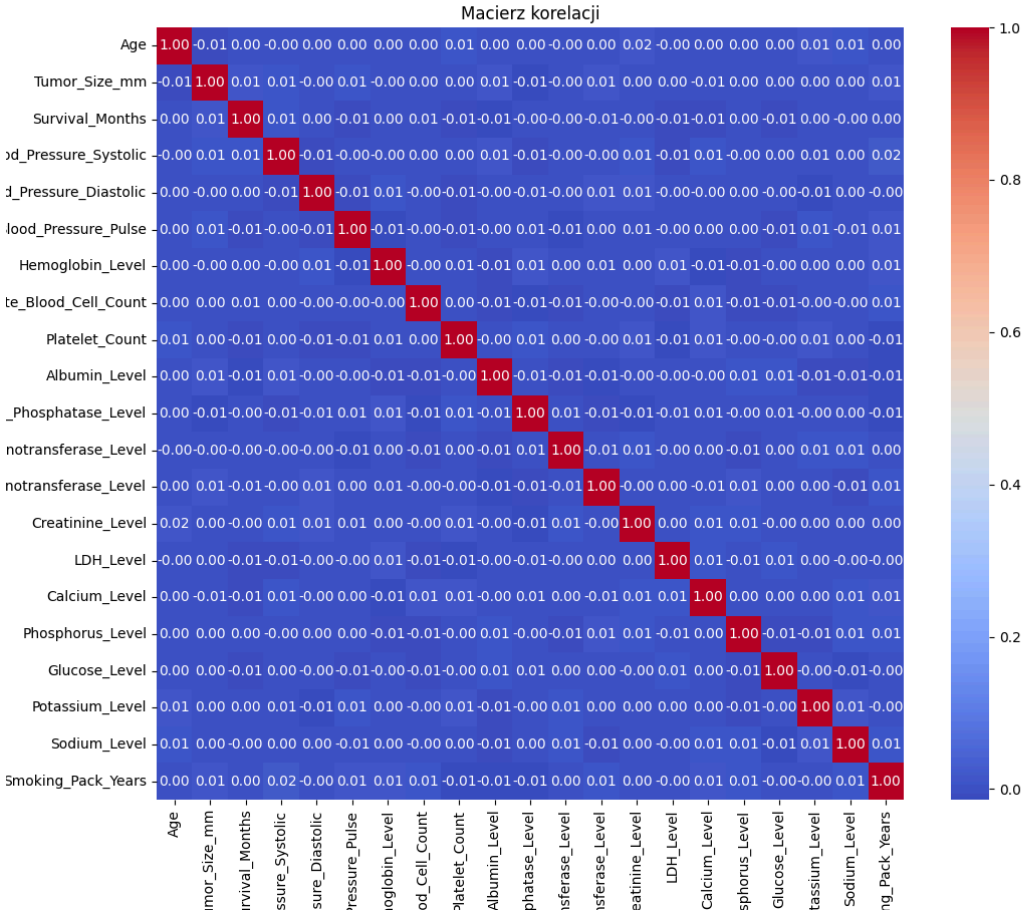

Wykres pudełkowy dla Smoking_Pack_Years
Smoking_Pack_Years
100
80
60
40
20
0


Wykres pudełkowy dla Sodium_Level
Sodium_Level
144
142
140
138
136


Wykres pudełkowy dla Survival_Months
Survival_Months
120
100
80
60
40
20
0


Wykres pudełkowy dla Tumor_Size_mm
Tumor_Size_mm
100
80
60
40
20

Dalej jest przedstawiona tabela korelacji wartości numerycznych na której można zauważyć, że takich korelacji nie ma.



Na koniec zostało sprawdzone czy są jakieś wartości brakujące, ale ze względu na to, że takowych wartości nie ma to nic oprócz tego nie zostało wyświetlone.

Automatyczna

Niestety analiza zautomatyzowanymi narzędziami okazała się niezbyt możliwa, ze względu na niekompatybilność, jak i za dużą ilość danych przez które nie można było wygenerować wykresów.

Efekty które udało się uzyskać to informacje o rodzaju danych w kolumnach, brakujących wartościach i rekomendacje co do tego co można zrobić z danymi kolumnami. Jedynymi zmianami zaproponowanymi w czasie tej analizy było usunięcie kolumny Patient_ID.

Aktualizacja

Udało się uruchomić Sweetviz po zmianie w pliku graph_numeric.py:
z:

warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)

 # ...
warnings.filterwarnings('once', category=np.VisibleDeprecationWarning)

na:

warnings.filterwarnings('ignore', category=DeprecationWarning)
 # ...
warnings.filterwarnings('once', category=DeprecationWarning)

Ogólnie rzecz biorąc wyniki są podobne do wcześniej uzyskanych.

Cały zbiór danych

Można zobaczyć brak korelacji i równomierne rozłożenie danych jak i czasami dużą różnorodność danych.

Porównanie zbioru danych test i train

Można zobaczyć, że procentowe rozłożenie danych jest dość wyrównane co powinno dobrze wpłynąć na trenowanie modelu.

Podsumowanie analizy

Podsumowując należy usunąć Insurance_Type i Patient_ID, pierwsze ze względu różnicy między krajami, a drugie przez to, że jest to id co nie będzie potrzebne przy tworzeniu modelu. Dodatkowo nie widać pomiędzy między kolumnami korelacji, więc trudno będzie wprowadzić jakieś zmiany związane z tym. Jednak można zauważyć wiele wartości które się nie powtarzają przez co można pomyśleć czy nie warto jakoś ich zaokrąglić.

Modele

Bez zaokrąglenia danych

Model	Wynik	Jak został zrobiony	Inne informacje
#1	0.25638457265467657	ExtraTreesClassifier(input_matrix, ExtraTreesClassifier__bootstrap=False, ExtraTreesClassifier__criterion=entropy, ExtraTreesClassifier__max_features=0.4	{'generation': 5, 'mutation_count': 2, 'crossover_count': 0, 'predecessor': ('ExtraTreesClassifier(input_matrix, ExtraTreesClassifier__bootstrap=False,

		, ExtraTreesClassifier__min_samples_leaf=15, ExtraTreesClassifier__min_samples_split=12, ExtraTreesClassifier__n_estimators=100)	ExtraTreesClassifier__criterion=entropy, ExtraTreesClassifier__max_features=0.4, ExtraTreesClassifier__min_samples_leaf=5, ExtraTreesClassifier__min_samples_split=12, ExtraTreesClassifier__n_estimators=100)',), 'operator_count': 1, 'internal_cv_score': np.float64(0.25638457265467657)}
#2	0.25569309463296225	KNeighborsClassifier(input_matrix, KNeighborsClassifier__n_neighbors=82, KNeighborsClassifier__p=2, KNeighborsClassifier__weights=uniform)	{'generation': 3, 'mutation_count': 1, 'crossover_count': 0, 'predecessor': ('KNeighborsClassifier(input_matrix, KNeighborsClassifier__n_neighbors=82, KNeighborsClassifier__p=2, KNeighborsClassifier__weights=distance)',), 'operator_count': 1, 'internal_cv_score': np.float64(0.25569309463296225)}
#3	0.25509031629399037	KNeighborsClassifier(input_matrix, KNeighborsClassifier__n_neighbors=44, KNeighborsClassifier__p=1, KNeighborsClassifier__weights=uniform)	{'generation': 5, 'mutation_count': 1, 'crossover_count': 1, 'predecessor': ('BernoulliNB(input_matrix, BernoulliNB__alpha=100.0, BernoulliNB__fit_prior=True)',), 'operator_count': 1, 'internal_cv_score': np.float64(0.25509031629399037)}

#4	0.255003923398299 14	GradientBoostingClassifier(input_matrix, GradientBoostingClassifier__learning_rate=1.0, GradientBoostingClassifier__max_depth=9, GradientBoostingClassifier__max_features=0.4, GradientBoostingClassifier__min_samples_leaf=19, GradientBoostingClassifier__min_samples_split=3, GradientBoostingClassifier__n_estimators=100, GradientBoostingClassifier__subsample=0.6500000000000001)	{'generation': 0, 'mutation_count': 0, 'crossover_count': 0, 'predecessor': ('ROOT',), 'operator_count': 1, 'internal_cv_score': np.float64(0.255003 92339829914)}
#5	0.255003625748357 1	GradientBoostingClassifier(input_matrix, GradientBoostingClassifier__learning_rate=0.01, GradientBoostingClassifier__max_depth=2, GradientBoostingClassifier__max_features=0.9000000000000001, GradientBoostingClassifier__min_samples_leaf=13, GradientBoostingClassifier__min_samples_split=2, GradientBoostingClassifier__n_estimators=100, GradientBoostingClassifier__subsample=0.7500000000000001)	{'generation': 1, 'mutation_count': 1, 'crossover_count': 0, 'predecessor': ('BernoulliNB(input_matrix, BernoulliNB__alpha=100.0, BernoulliNB__fit_prior=False)',), 'operator_count': 1, 'internal_cv_score': np.float64(0.255003 6257483571)}

Z zaokrągleniem danych do 2 miejsca po przecinku

Model	Wynik	Jak został zrobiony	Inne informacje
#1	0.2585402279477669	SGDClassifier(input_matrix, SGDClassifier__alpha=0.0, SGDClassifier__eta0=0.01, SGDClassifier__fit_intercept=True, SGDClassifier__l1_ratio=0.25, SGDClassifier__learning_rate=constant, SGDClassifier__loss=modified_huber, SGDClassifier__penalty=elasticnet, SGDClassifier__power_t=10.0)	{'generation': 0, 'mutation_count': 0, 'crossover_count': 0, 'predecessor': ('ROOT',), 'operator_count': 1, 'internal_cv_score': np.float64(0.2585402279477669)}
#2	0.2550028072110163	DecisionTreeClassifier(RBFSampler(input_matrix, RBFSampler__gamma=0.2), DecisionTreeClassifier__criterion=entropy, DecisionTreeClassifier__max_depth=6, DecisionTreeClassifier__min_samples_leaf=11, DecisionTreeClassifier__min_samples_split=5)	{'generation': 4, 'mutation_count': 2, 'crossover_count': 0, 'predecessor': ('DecisionTreeClassifier(RBFSampler(input_matrix, RBFSampler__gamma=0.2), DecisionTreeClassifier__criterion=entropy, DecisionTreeClassifier__max_depth=6, DecisionTreeClassifier__min_samples_leaf=19, DecisionTreeClassifier__min_samples_split=5)',), 'operator_count': 2, 'internal_cv_score': np.float64(0.2550028072110163)}
#3	0.2547440005863704	DecisionTreeClassifier(RBFSampler(input_matrix, RBFSampler__gamma=0.2), DecisionTreeClassifier__criterion=gini, DecisionTreeClassifier__max_depth=6, DecisionTreeClassifier__min_	{'generation': 5, 'mutation_count': 3, 'crossover_count': 0, 'predecessor': ('DecisionTreeClassifier(RBFSampler(input_matrix, RBFSampler__gamma

		samples_leaf=11, DecisionTreeClassifier__min_samples_split=5)	=0.2), DecisionTreeClassifier__criterion=entropy, DecisionTreeClassifier__max_depth=6, DecisionTreeClassifier__min_samples_leaf=11, DecisionTreeClassifier__min_samples_split=5)',), 'operator_count': 2, 'internal_cv_score': np.float64(0.2547440005863704)}
#4	0.2545712892074735	DecisionTreeClassifier(RBFSampler(input_matrix, RBFSampler__gamma=0.2), DecisionTreeClassifier__criterion=entropy, DecisionTreeClassifier__max_depth=6, DecisionTreeClassifier__min_samples_leaf=5, DecisionTreeClassifier__min_samples_split=11)	{'generation': 4, 'mutation_count': 3, 'crossover_count': 0, 'predecessor': ('DecisionTreeClassifier(RBFSampler(input_matrix, RBFSampler__gamma=0.2), DecisionTreeClassifier__criterion=entropy, DecisionTreeClassifier__max_depth=6, DecisionTreeClassifier__min_samples_leaf=19, DecisionTreeClassifier__min_samples_split=11)',), 'operator_count': 2, 'internal_cv_score': np.float64(0.2545712892074735)}
#5	0.2543125942015559	KNeighborsClassifier(RobustScaler(input_matrix), KNeighborsClassifier__n_neighbors=15, KNeighborsClassifier__p=1, KNeighborsClassifier__weights=uniform)	{'generation': 4, 'mutation_count': 2, 'crossover_count': 0, 'predecessor': ('BernoulliNB(RobustScaler(input_matrix), BernoulliNB__alpha=1.0, BernoulliNB__fit_prior=True)',), 'operator_count': 2, 'internal_cv_score': np.float64(0.2543125942015559)}

Wybrany model

Wybrałam model #1 z grupy działającej na zaokrąglonych danych, ponieważ daje najlepszy wynik.

Wyniki modelu:

accuracy: 0.2457729468599034
precision: 0.2402012891721489
F1: 0.2190883897895028