Wykorzystanie biblioteki DEAP w problemie optymalizacji parametrów klasyfikatorów oraz selekcji cech

Obliczenia ewolucyjne

Projekt nr 4

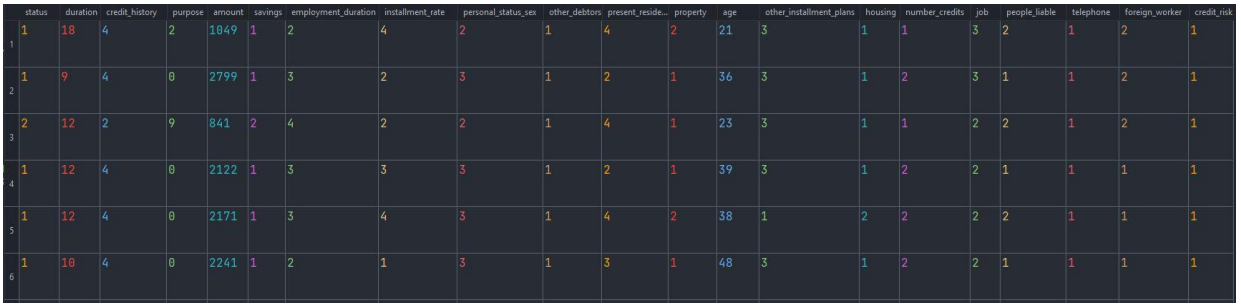
Błażej Zieliński 130605

Jakub Zygmunt 130606

grupa DS2

1. Przedstawienie zbioru danych

Wybrany zbiór danych opisuje ryzyko kredytów zaciąganych przez wybranych obywateli południowych niemiec w latach 1973 - 1975. Baza danych zawiera 1000 rekordów, z których każdy podzielony jest na 20 kolumn, natomiast informacje w niej zawarte to między innymi: oszczędności kredytobiorcy, przeznaczenie kredytu, historia kredytowa, wysokość kredytu, długość zatrudnienia oraz binarny wskaźnik ryzyka kredytowego.



	status	duration	credit_history	purpose	amount	savings	employment_duration	installment_rate	personal_status_sex	other_debtors	present_reside...	property	age	other_installment_plans	housing	number_credits	job	people_liable	telephone	foreign_worker	credit_risk
1	1	18	4	2	1049	1	2	4	2	1	4	2	21	3	1	1	3	2	1	2	1
2	1	9	4	0	2799	1	3	2	3	1	2	1	36	3	1	2	3	1	1	2	1
3	2	12	2	9	841	2	4	2	2	1	4	1	23	3	1	1	2	2	1	2	1
4	1	12	4	0	2122	1	3	3	3	1	2	1	39	3	1	2	2	1	1	1	1
5	1	12	4	0	2171	1	3	4	3	1	4	2	38	1	2	2	2	2	1	1	1
6	1	10	4	0	2241	1	2	1	3	1	3	1	48	3	1	2	2	1	1	1	1

[Link do repozytorium github aplikacji]

2. Wykorzystywane technologie

Zdecydowano się na wykorzystanie języka Python oraz przeznaczonych dla niego bibliotek, między innymi: matplotlib, statistics numpy, sklearn, pandas i deap

3. Wymagania środowiska do uruchomienia aplikacji

a. Python w wersji 3.10

b. Biblioteki języka python takie jak:

   i. numpy w wersji 1.22.2

   ii. pandas w wersji 1.4.2

   iii. sklearn w wersji 1.1.1

   iv. matplotlib w wersji 3.5.1

   v. deap w wersji 1.3.1

4. Przedstawienie wyników dla domyślnych parametrów klasyfikatora

   a. **SVC - maszyna wektorów pomocniczych**

```
Wynik klasyfikatora SVC : 0.749
```

Osiągnięty rezultat: 74,9%

   b. **LR - regresja logistyczna**

```
Wynik klasyfikatora LogisticRegression : 0.742
```

Osiągnięty rezultat: 74,2%

   c. **DT - drzewo decyzyjne**

```
Wynik klasyfikatora DecisionTreeClassifier : 0.658
```

Osiągnięty rezultat: 65,8%

   d. **KNN - K najbliższych sąsiadów**

```
Wynik klasyfikatora KNeighborsClassifier : 0.7070000000000001
```

Osiągnięty rezultat: 70,7%

   e. **RF - random forest**

```
Wynik klasyfikatora RandomForestClassifier : 0.733
```

Osiągnięty rezultat: 73,3%

Dyskusja

➔ W przypadku drzewa decyzyjnego oraz random forest wyniki są bardzo mało powtarzalne w porównaniu z resztą klasyfikatorów.

5. Przedstawienie wyników dla genetycznej optymalizacji parametrów

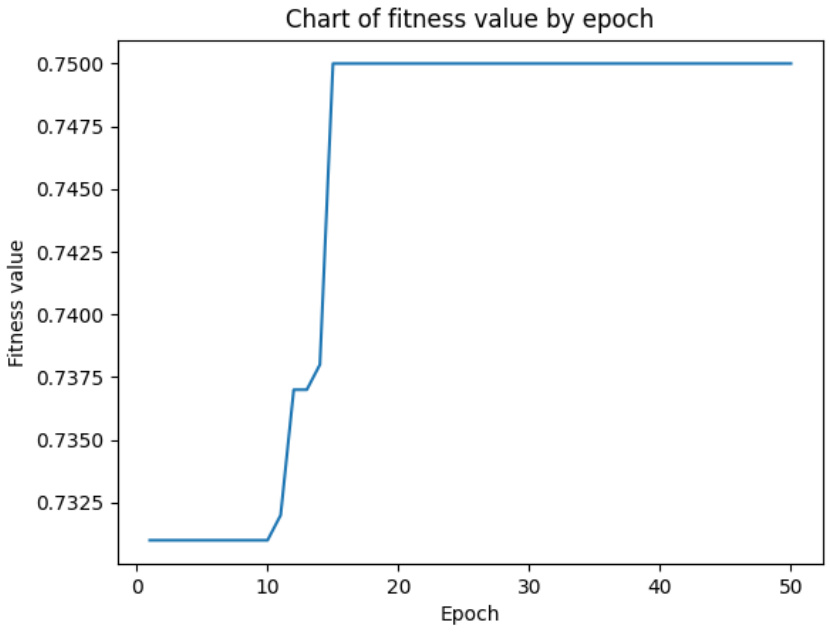
a. SVC - maszyna wektorów pomocniczych

Optymalizowane parametry:

- kernel: Określa typ jądra, który ma być użyty w algorytmie
  - wartości: ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’
- C: Parametr regularyzacji
  - wartości: 0.1-5
- degree: Stopień wielomianu, używany dla typu jądra “poly”
  - wartości: 0.1-5
- gamma: Współczynnik kernela dla ‘rbf’, ‘poly’ i ‘sigmoid’.
  - wartości: 0.001-2
- coefficient: Współczynnik niezależny dla różnych funkcji jądra. Ma znaczenie tylko w „poly” i „sigmoid”.
  - wartości: 0.01-1

Wielkość populacji - 10

Ilość epok - 50



Czas wykonania: 82.13

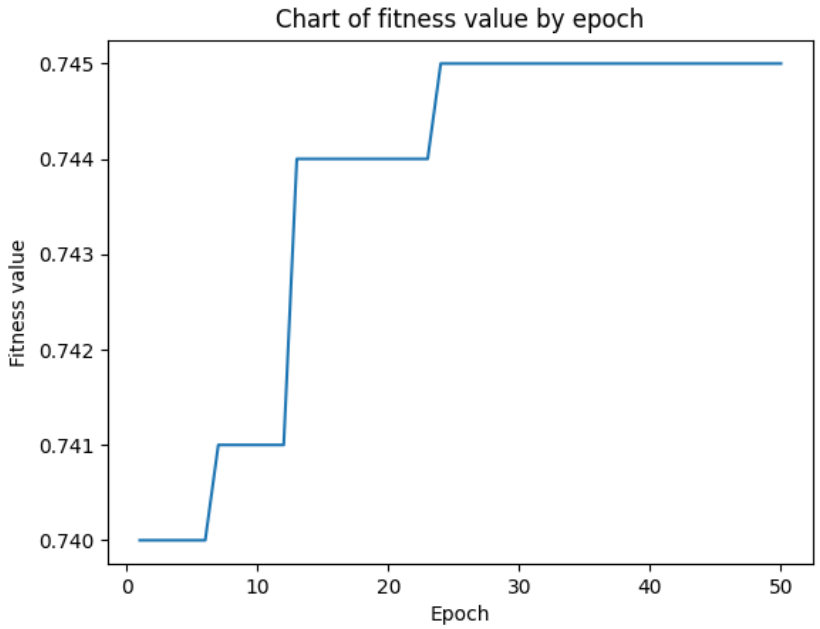
Najlepszy osobnik, rezultat 75%
"['rbf', 4.42521291961968, 3.6830481801353385, 0.08950300517554714, 0.00634221919196]"

b. LR - regresja logistyczna

Optymalizowane parametry:

- solver: algorytm używany do rozwiązania problemu
  - wartości: ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’
- C: Parametr regularyzacji
  - wartości: 0.1-5
- fit_intercept: Określa, czy do funkcji decyzyjnej należy dodać stałą (tzw. stronniczość lub przecięcie).
  - wartości: 0-1
- max_iter: maksymalna ilość iteracji algorytmu
  - wartości: 100-1000

Wielkość populacji - 50
Ilość epok - 50



Czas wykonania: 54.87

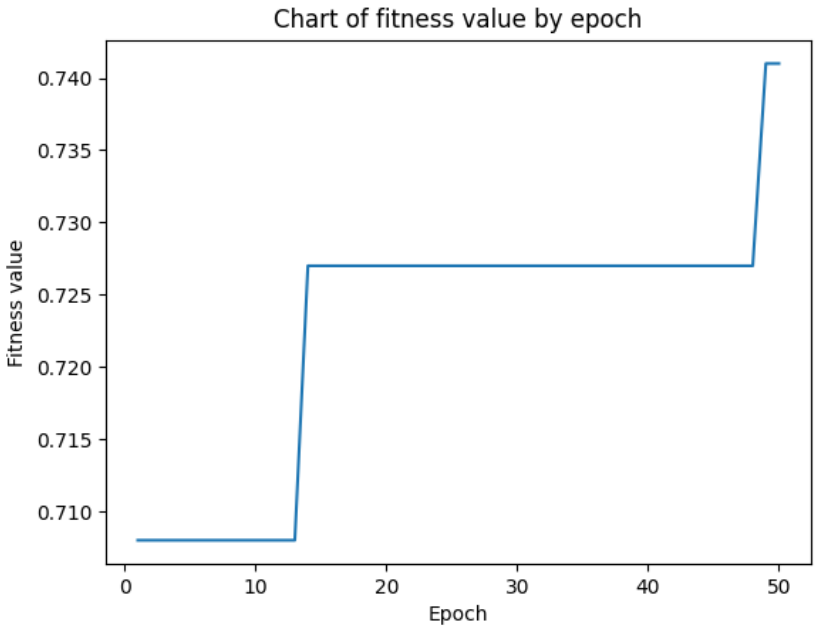
Najlepszy osobnik, rezultat 74,5%
"['liblinear', 0.5675191825122563, 0, 614]"

c. DT - drzewo decyzyjne

Optymalizowane parametry:

- criterion: Funkcja pomiaru jakości podziału
  - wartości: “gini”, “entropy”, “log_loss”
- splitter: Strategia używana przy wyborze podziału każdego węzła
  - wartości: “best”, “random”
- max_depth: Maksymalna głębokość drzewa
  - wartości: 2-8
- min_samples_split: Minimalna liczba próbek wymagana do podziału węzła wewnętrznego:
  - wartości: 0.01-1
- min_samples_leaf: Minimalna liczba próbek, które muszą znajdować się w węźle liścia
  - wartości: 0.01-5

Wielkość populacji - 50
Ilość epok - 50



Czas wykonania: 31.34

Najlepszy osobnik, rezultat 74,1%

"['log_loss', 'random', 5, 0.14448771756080656, 0.03612751616188254]"

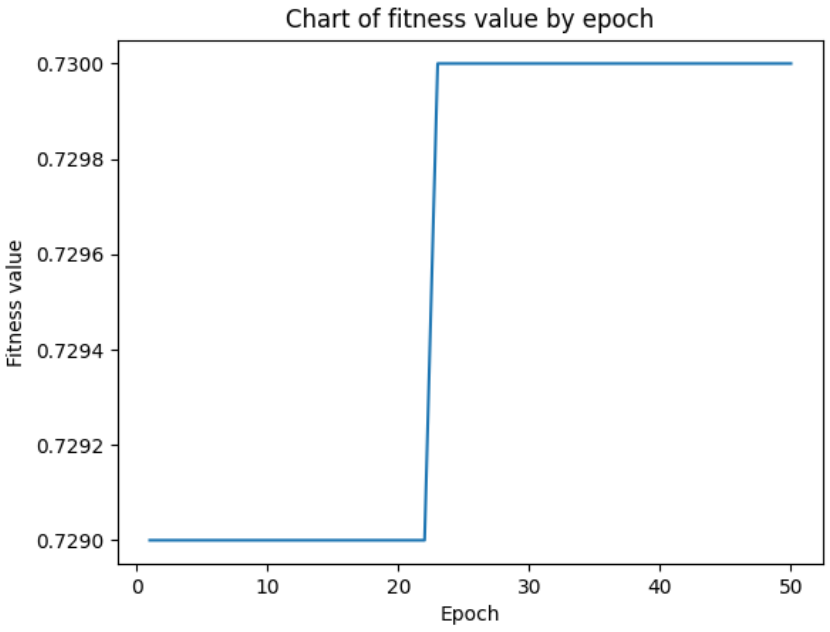
d. KNN - K najbliższych sąsiadów

Optymalizowane parametry:

- n_neighbors: Liczba sąsiadów używanych domyślnie w zapytaniach sąsiadów
    - wartości: 1-10
- weights: Funkcja wagi używana w prognozie
    - wartości: ‘uniform’, ‘distance’
- algorithm: Algorytm używany do znajdywania najbliższego sąsiada
    - wartości: ‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’
- leaf_size: Rozmiar liścia przekazany do ‘ball_tree’ i ‘kd_tree’
    - wartości: 20-40
- p: Parametr mocy dla metryki Minkowskiego
    - wartości: 2-5

Wielkość populacji - 10

Ilość epok - 50



Czas wykonania: 46.77

Najlepszy osobnik, rezultat 73%

"[10, 'distance', 'brute', 35, 5]”

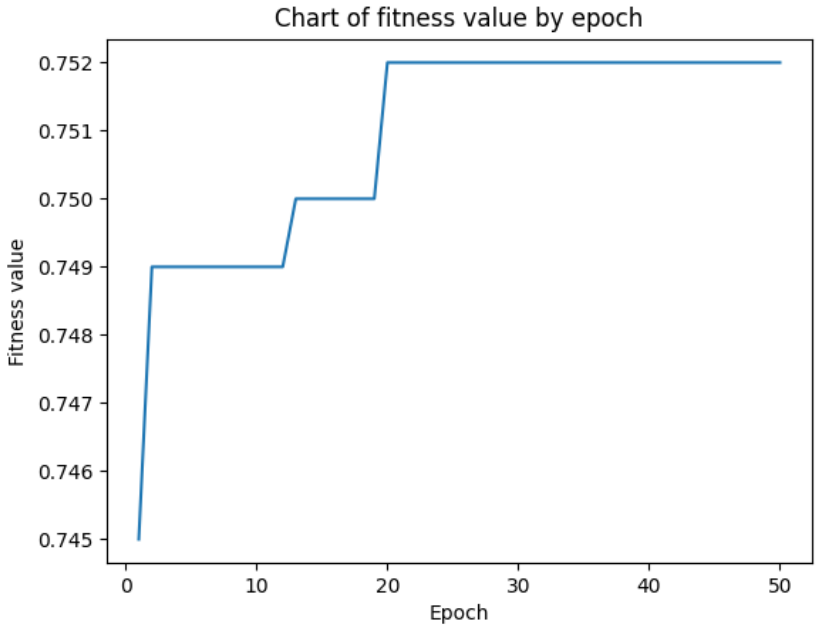
e. RF - random forest

Optymalizowane parametry:

- criterion: Funkcja pomiaru jakości podziału
  - wartości: “gini”, “entropy”, “log_loss”
- n_estimators: Liczba drzew
  - wartości: 10-50
- max_depth: Maksymalna głębokość drzewa
  - wartości: 2-8
- min_samples_split: Minimalna liczba próbek wymagana do podziału węzła wewnętrznego:
  - wartości: 2-4
- min_samples_leaf: Minimalna liczba próbek, które muszą znajdować się w węźle liścia
  - wartości: 1-4

Wielkość populacji - 10

Ilość epok - 50



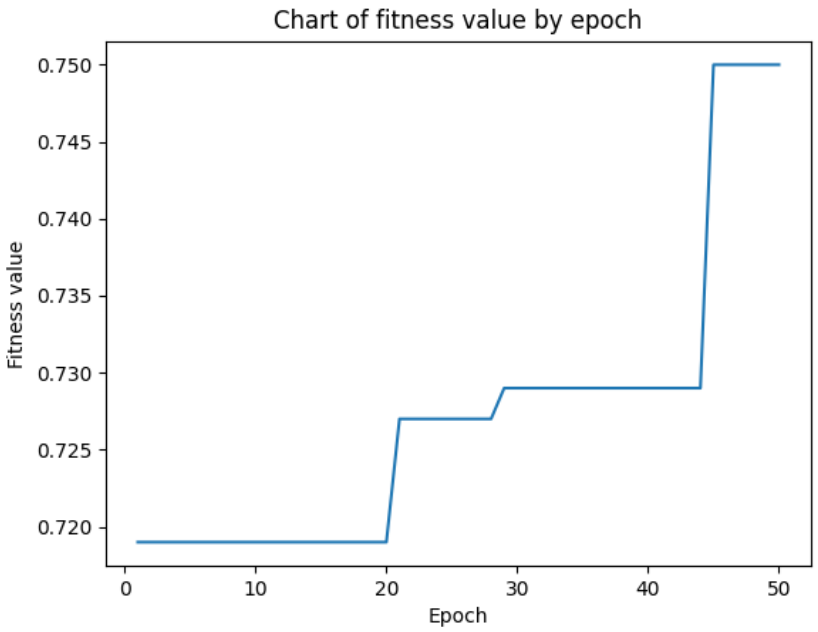
Czas wykonania: 41.17

Najlepszy osobnik, rezultat 75,2%
"[45, 'log_loss', 7, 3, 1]"

6. Przedstawienie wyników dla genetycznej optymalizacji parametrów i selekcji cech

a. SVC - maszyna wektorów pomocniczych

Wielkość populacji - 10
Ilość epok - 50



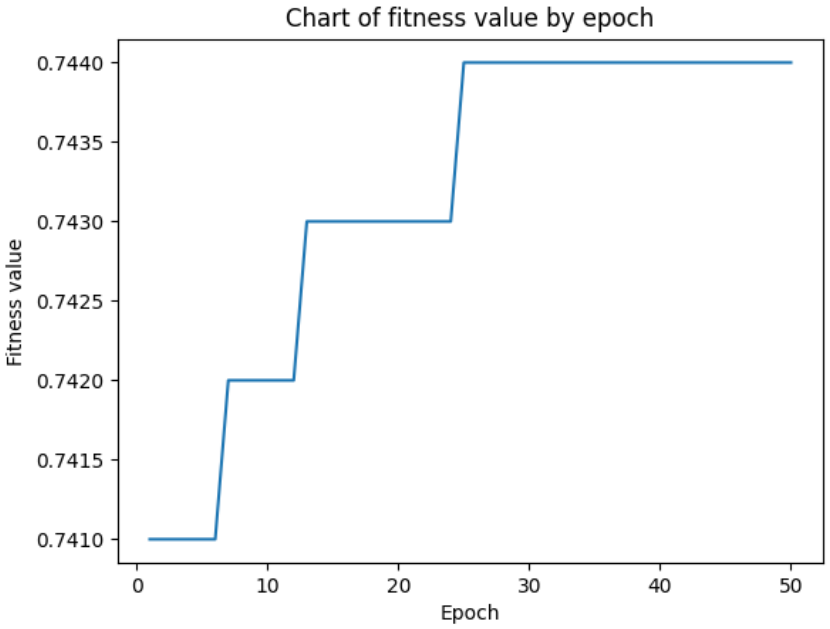
Czas wykonania: 79.12

Najlepszy osobnik, rezultat 75%
"['rbf', 0.7579598444849946, 4.2259208400124235, 1.0127524587837429, 0.6210313713651605, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1]"

b. LR - regresja logistyczna

Wielkość populacji - 50

Ilość epok - 50



Czas wykonania: 57.77

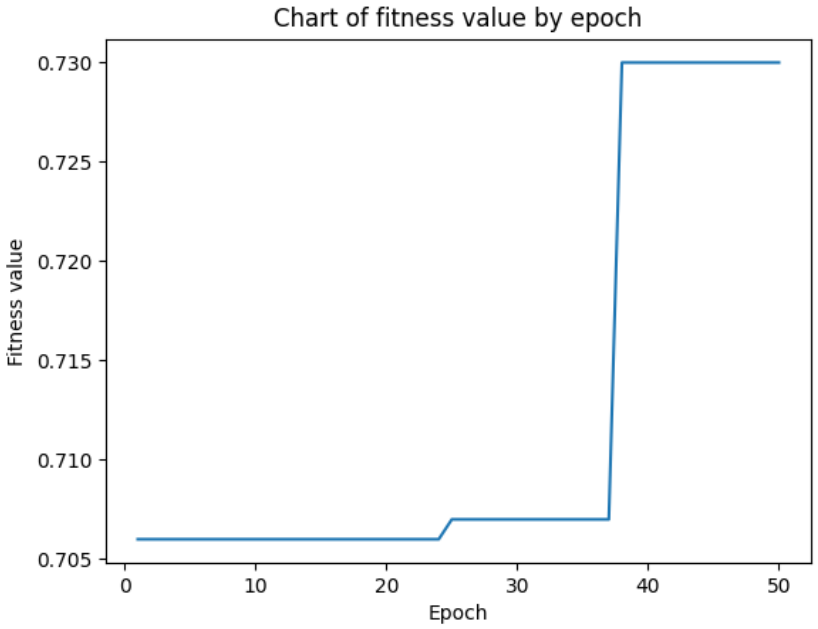
Najlepszy osobnik, rezultat 74.4%

"['lbfgs', 1.0611177611481304, 0, 806, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]"

c. DT - drzewo decyzyjne

Wielkość populacji - 50

Ilość epok - 50



Czas wykonania: 32.38

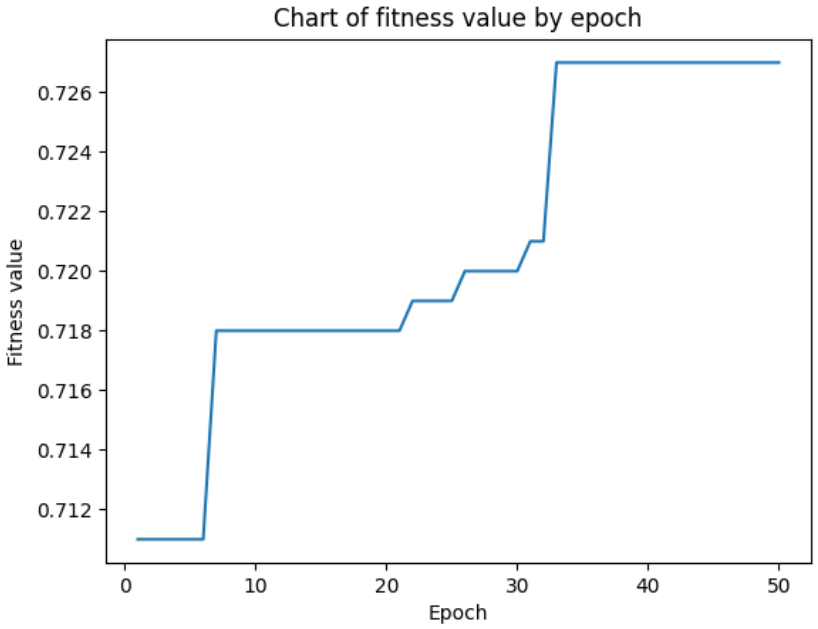
Najlepszy osobnik, rezultat 73%

"['log_loss', 'best', 4, 0.11469683353479082, 0.054588107434576344, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1]"

d. KNN - K najbliższych sąsiadów

Wielkość populacji - 10

Ilość epok - 50



Czas wykonania: 41.02

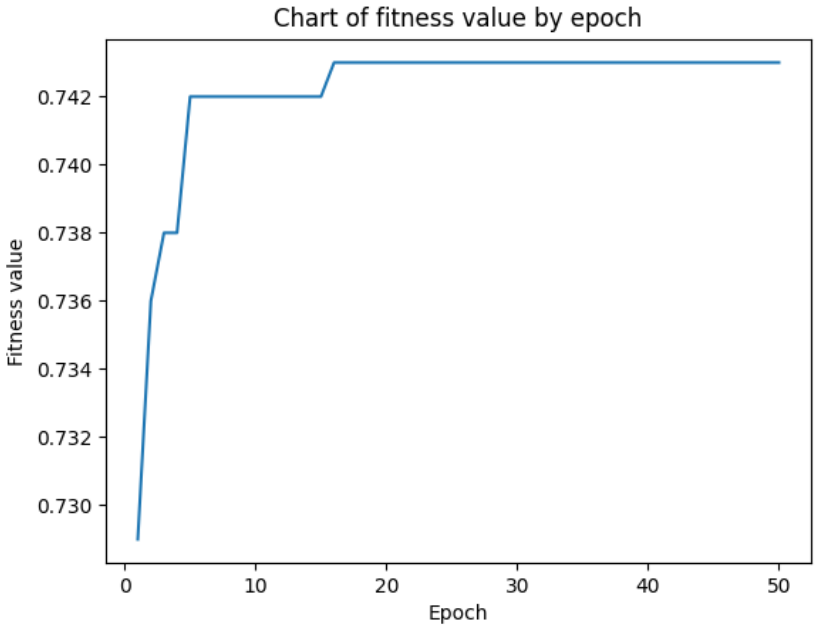
Najlepszy osobnik, rezultat 72,7%

"[9, 'distance', 'ball_tree', 30, 5, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1]"

e. RF - random forest

Wielkość populacji - 10

Ilość epok - 50



Czas wykonania: 45.81

Najlepszy osobnik, rezultat 74,3%

"[15, 'gini', 6, 3, 2, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1]"

7. Podsumowanie

		Dokładność początkowa	Czas algorytmu	Wielkość populacji	Dokładność po algorytmie genetycznym
SVC	Bez Selekcji	74,9%	82,13s	10	75%
	Selekcja		79,12s		75%
LR	Bez Selekcji	74,2%	54.87s	50	74,5%
	Selekcja		57.77s		74.4%
DT	Bez Selekcji	65,8%	31.34s	50	74,1%
	Selekcja		32.38s		73%
KNN	Bez Selekcji	70,7%	46.77s	10	73%
	Selekcja		41.02s		72,7%
RF	Bez Selekcji	73,3%	41.17s	10	75,2%
	Selekcja		45.81s		74,3%

Wnioski

- W przypadku analizowanego zbioru selekcja cech nie miała większego wpływu na otrzymywane wyniki.
- Klasyfikatory takie jak SVC, Random forest oraz K-nearest neighbors charakteryzują się znacznie dłuższym (około 5 razy) czasem wykonania klasyfikacji aniżeli w przypadku Logistic regression oraz Decision tree.
- Najlepszy wynik po optymalizacji parametrów uzyskał Random forest, bardzo dobrze poradził sobie również SVC oraz Logistic regression.
- Optymalizacja parametrów w omawianym zbiorze przynosi znaczące rezultaty dla klasyfikatorów Decision tree, K-nearest neighbors oraz Random forest. w przypadku SVC oraz Logistic regression wyniki poprawiły się o zaledwie 0.1-0.