

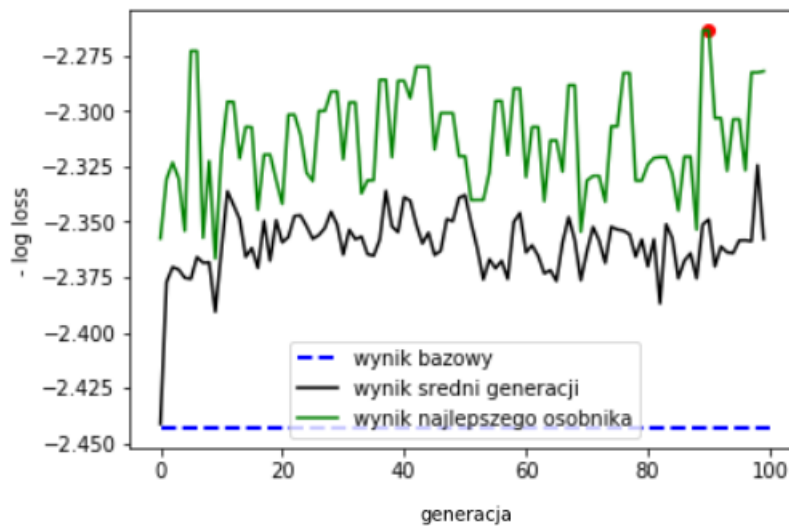
## PB14 - Wykorzystanie algorytmu ewolucyjnego do doboru parametrów SVM do zadania klasyfikacji

### 1 DODATKOWE SZCZEGÓŁY IMPLEMENTACYJNE

Została zmodyfikowana metoda *get\_params*, która, oprócz wyświetlania najlepszych parametrów klasyfikatora SVM (pod względem metryki *textitlogloss*) uzyskanych w procesie optymalizacji, czyli fenotypu najlepszego osobnika, zwraca teraz też ich wartości do zmiennej.

Dodatkowo, na potrzeby eksperymentów związanych z tworzeniem populacji ze zmodyfikowanego rozkładu, przed drugim uruchomieniem algorytmu zmieniono metodę *\_create\_individual*, tak aby losowanie wektorów osobników populacji początkowej następowało z rozkładu jednostajnego z przedziału (0, 5).

Stworzono funkcję pomocniczną *save\_results*, która zapisuje słownik z wynikami przeprowadzonych eksperymentów do pliku w formacie csv. Przykładowy wykres wartości metryki *logloss* w funkcji numeru generacji przedstawia Rysunek 1.1 (czerwoną kropką zaznaczono wynik najlepszego osobnika w całym procesie optymalizacji).



Rysunek 1.1: Wykres wartości metryki *logloss* w funkcji numeru generacji populacji.

## 2 OPIS PROCEDURY EKSPERYMENTALNEJ

### 2.1 IMPLIMENTACJA METODY

Algorytm genetyczny został zaimplementowany samodzielnie w języku Python. Modele maszyny wektorów nośnych były budowane przy użyciu *sklearn.svm.SVC* z biblioteki *scikit-learn*. Do obliczenia wartości funkcji przystosowania osobników użyto funkcji *K*-krotnej walidacji krzyżowej z *sklearn.model\_selection.KFold* razem z metrykami *log\_loss* i *accuracy\_score* z *sklearn.metrics*.

### 2.2 PLAN TESTÓW

W projekcie badaliśmy wpływ rozmiaru populacji i liczby generacji oraz sposobu inicjalizacji populacji początkowej algorytmu ewolucyjnego na wyniki optymalizacji parametrów *C* i *gamma* maszyny wektorów nośnych. Użyte w eksperymentach wartości parametrów przedstawia Tabela 2.1. Dla każdego zbioru danych, dla każdej możliwej kombinacji wartości parametrów (25) został następnie uruchomiony zaprojektowany algorytm ewolucyjny. Po serii tych eksperymentów rozkład jednostajny na przedziale (0, 1), z którego losowana była populacja początkowa (tj. cechy osobników), został zmieniony na rozkład jednostajny na przedziale (0, 10). Po tej zmianie algorytm ewolucyjny został ponownie uruchomiony dla tych samych 25 kombinacji wartości parametrów eksperymentalnych.

<b>Rozmiar populacji</b>	10	20	50	100	200
<b>Liczba generacji</b>	5	10	20	50	100

Tabela 2.1: Zestawienie wartości parametrów wykorzystanych w eksperymentach.

## 3 WYNIKI BADAŃ EKSPERYMENTALNYCH

Poniżej znajdują się wyniki eksperymentów na trzech zbiorach danych ("tanim", "średnim" i "drogim") przy dwóch różnych sposobach inicjalizowania populacji początkowej. Nazwy poszczególnych kolumn oznaczają:

- pop - liczebność populacji,
- iter - liczba generacji działania algorytmu,
- gen num - numer generacji, z której pochodzi najlepszy osobnik,
- C - uzyskana wartość optymalizowanego parametru *C*,
- gamma - uzyskana wartość optymalizowanego parametru *gamma*,

- time - całkowity czas optymalizacji parametrów (działania algorytmu przy zadanych parametrach) w sekundach,
- cv score - średnia wartość metryki *logloss* uzyskana podczas 2-krotnej walidacji krzyżowej na zbiorze uczącym,
- logloss - wartość metryki *logloss* uzyskana na zbiorze testowym,
- accuracy - wartość metryki *accuracy* uzyskana na zbiorze testowym.

Wartości metryk zostały uzyskane w wyniku klasyfikacji przy użyciu modelu SVM o optymalnych parametrach *C* i *gamma* w danym procesie, wyuczonego na zbiorze uczącym.

### 3.1 INICJALIZACJA POPULACJI ROZKŁADEM JEDNOSTAJNYM NA PRZEDZIALE (0, 1)

#### 3.1.1 ZBIÓR "TANI"

*Wyniki eksperymentów na zbiorze "tanim" przy inicjowaniu populacji rozkładem jednostajnym na przedziale (0, 1).*

pop	iter	gen num	C	gamma	time	cv score	logloss	accuracy
10	5	3	0.775885	0.741334	1.04	2.371989	1.639387	0.735294
10	10	4	0.862755	0.61262	2.65	2.375861	1.300349	0.735294
10	20	3	0.854879	0.650106	8.34	2.372864	1.517469	0.735294
10	50	18	0.828379	0.838094	45.73	2.351378	1.300563	0.735294
10	100	98	0.990735	0.642568	195.91	2.346512	1.361659	0.72549
20	5	0	0.991201	0.676812	1.44	2.364096	1.242285	0.72549
20	10	8	0.882012	0.521156	4.11	2.370742	1.440953	0.715686
20	20	3	0.810099	0.792571	12.14	2.3432	1.436476	0.735294
20	50	45	0.965561	0.883801	59.95	2.336284	1.289719	0.72549
20	100	33	0.858748	0.900308	224.52	2.338063	1.2993	0.72549
50	5	0	0.954712	0.960456	3.31	2.326466	1.254817	0.715686
50	10	1	0.559279	0.921179	9.54	2.349331	1.413862	0.705882
50	20	15	0.910123	0.756723	28.68	2.352239	1.29743	0.735294
50	50	43	0.906265	0.769172	140.99	2.34305	1.335147	0.735294
50	100	55	0.859524	0.832056	536.92	2.339384	1.253516	0.735294
100	5	3	0.822393	0.539237	5.92	2.351654	1.383943	0.715686
100	10	1	0.826309	0.779784	16.24	2.341964	1.335005	0.735294
100	20	8	0.95189	0.809941	67.41	2.325684	1.337457	0.72549
100	50	7	0.873937	0.837696	322.54	2.332779	1.375047	0.735294
100	100	97	0.941286	0.895356	1014.99	2.322137	1.335966	0.72549
200	5	0	0.964847	0.939523	11.84	2.341327	1.402856	0.715686
200	10	0	0.924643	0.871136	31.17	2.339811	1.29886	0.72549
200	20	5	0.886385	0.87004	106.72	2.334496	1.304115	0.72549
200	50	41	0.874116	0.769028	529.85	2.335952	1.361472	0.745098
200	100	79	0.890947	0.897066	1911.65	2.325546	1.325503	0.72549

### 3.1.2 ZBIÓR "ŚREDNI"

*Wyniki eksperymentów na zbiorze "średnim" przy inicjowaniu populacji rozkładem jednostajnym na przedziale (0, 1).*

pop	iter	gen num	C	gamma	time	cv score	logloss	accuracy
10	5	3	0.583313	0.876033	0.9	2.376071	1.586244	0.696078
10	10	5	0.617034	0.555163	2.52	2.377227	1.448491	0.676471
10	20	8	0.507644	0.930632	8.43	2.365265	1.567837	0.676471
10	50	13	0.502328	0.908584	45.35	2.348504	1.423047	0.676471
10	100	93	0.705663	0.697588	173.7	2.328258	1.361726	0.696078
20	5	4	0.813933	0.801347	1.36	2.344713	1.624254	0.72549
20	10	4	0.67745	0.643677	3.77	2.351094	1.515612	0.705882
20	20	3	0.444417	0.859861	11.45	2.349657	1.817341	0.656863
20	50	11	0.829294	0.66087	64.18	2.352808	1.376056	0.72549
20	100	30	0.735298	0.649867	243.72	2.327358	1.390432	0.705882
50	5	4	0.757886	0.571231	3.09	2.340524	1.413652	0.696078
50	10	6	0.63553	0.42136	8.91	2.368091	1.748562	0.676471
50	20	12	0.750386	0.592301	32.54	2.353272	1.398061	0.705882
50	50	14	0.511788	0.71066	179.28	2.356804	1.647214	0.686275
50	100	78	0.720232	0.631464	571.78	2.295972	1.490703	0.705882
100	5	0	0.997353	0.755967	7.54	2.34796	1.49959	0.735294
100	10	2	0.502407	0.702069	18.35	2.362611	1.60099	0.686275
100	20	8	0.731289	0.759372	52.28	2.322591	1.422814	0.715686
100	50	38	0.693835	0.579956	303.4	2.318138	1.503484	0.686275
100	100	35	0.657951	0.540167	977.43	2.305616	1.389641	0.676471
200	5	1	0.864375	0.662769	11.24	2.349409	1.327915	0.72549
200	10	1	0.72737	0.725455	29.46	2.32262	1.386085	0.705882
200	20	9	0.728142	0.651474	89.09	2.299347	1.466367	0.705882
200	50	8	0.731327	0.633242	553.39	2.298736	1.479212	0.705882
200	100	46	0.694776	0.591155	2141.81	2.300198	1.473277	0.696078

### 3.1.3 ZBIÓR "DROGI"

*Wyniki eksperymentów na zbiorze "drogim" przy inicjowaniu populacji rozkładem jednostajnym na przedziale (0, 1).*

pop	iter	gen num	C	gamma	time	cv score	logloss	accuracy
10	5	0	0.498917	0.238895	1.17	2.318614	1.797332	0.588235
10	10	2	0.951934	0.772284	3.39	2.248885	1.316599	0.715686
10	20	1	0.789687	0.865127	8.47	2.284584	1.362445	0.715686
10	50	23	0.522148	0.897803	45.36	2.258524	1.598832	0.696078
10	100	18	0.778453	0.903627	215.56	2.24919	1.597466	0.715686
20	5	3	0.734826	0.792028	1.8	2.267316	1.530664	0.705882
20	10	8	0.773557	0.940156	4.15	2.249166	1.421809	0.72549
20	20	9	0.809572	0.948005	13.49	2.266422	1.644196	0.72549
20	50	23	0.743888	0.929618	72.13	2.254207	1.460628	0.715686
20	100	69	0.912512	0.830734	254.93	2.2361	1.411296	0.715686
50	5	1	0.83546	0.765408	3.06	2.272703	1.409115	0.72549
50	10	8	0.661562	0.801659	8.8	2.284497	1.476152	0.705882
50	20	10	0.678544	0.862862	32.01	2.259467	1.402371	0.705882
50	50	33	0.77309	0.897324	168.93	2.245521	1.697309	0.715686
50	100	7	0.735145	0.950181	647.44	2.245206	1.535622	0.715686
100	5	4	0.874356	0.666245	6.58	2.286599	1.443904	0.696078
100	10	4	0.865063	0.883355	17.77	2.228063	1.339122	0.715686
100	20	14	0.758646	0.753038	55.48	2.235956	1.871298	0.715686
100	50	15	0.720993	0.853719	267.03	2.234466	1.386049	0.715686
100	100	33	0.677532	0.736065	953.56	2.234557	1.45305	0.705882
200	5	2	0.838728	0.844323	11.79	2.257453	1.396679	0.715686
200	10	7	0.746672	0.794966	35.85	2.245369	1.403357	0.705882
200	20	1	0.782966	0.882653	104.8	2.244267	1.433705	0.715686
200	50	43	0.724877	0.833313	463.23	2.233841	1.42213	0.715686
200	100	64	0.710254	0.814443	1813.25	2.234269	1.668056	0.705882

### 3.2 INICJALIZACJA POPULACJI ROZKŁADEM JEDNOSTAJNYM NA PRZEDZIALE (0, 10)

#### 3.2.1 ZBIÓR "TANI"

*Wyniki eksperymentów na zbiorze "tanim" przy inicjowaniu populacji rozkładem jednostajnym na przedziale (0, 10).*

pop	iter	gen num	C	gamma	time	cv score	logloss	accuracy
10	5	3	6.200002	1.141829	1.83	2.283397	1.265425	0.754902
10	10	0	8.131339	2.485682	4.39	2.314621	1.335839	0.735294
10	20	12	4.257617	1.852351	13.51	2.295609	1.215829	0.754902
10	50	2	4.885378	0.711335	56.23	2.260047	1.287287	0.754902
10	100	5	3.871355	1.03575	231.34	2.261426	1.170609	0.745098
20	5	2	6.749784	0.890763	1.82	2.264712	1.223727	0.754902
20	10	7	8.468546	1.754818	4.91	2.278581	1.164444	0.754902
20	20	6	6.925979	1.031326	17.36	2.266686	1.172379	0.754902
20	50	44	6.012299	0.732016	86.99	2.241381	1.190057	0.754902
20	100	60	9.53045	0.734791	306	2.235877	1.220786	0.754902
50	5	4	3.545346	1.278675	4.69	2.276611	1.179453	0.745098
50	10	1	4.645771	0.071521	10.08	2.253624	1.409153	0.676471
50	20	17	5.013222	1.321122	30.34	2.267474	1.246558	0.754902
50	50	9	8.357896	0.67645	182.27	2.229672	1.205878	0.754902
50	100	89	3.364823	2.219496	554.27	2.280693	1.199325	0.72549
100	5	3	5.760027	0.649505	7.61	2.255572	1.234954	0.754902
100	10	5	7.371377	0.839293	19.44	2.249488	1.245772	0.754902
100	20	6	7.736571	0.712183	53.77	2.248061	1.172785	0.754902
100	50	3	8.312286	0.040928	278.65	2.182017	1.3489	0.705882
100	100	50	6.138155	0.922795	996.02	2.250202	1.220114	0.754902
200	5	4	8.526678	0.645125	13.15	2.239398	1.164818	0.754902
200	10	1	6.626876	0.769886	33.56	2.250435	1.185254	0.754902
200	20	19	6.149831	1.06778	92.78	2.26004	1.219916	0.754902
200	50	3	8.685977	0.556309	466.95	2.223739	1.222229	0.754902
200	100	37	6.180297	1.005777	2257.44	2.25525	1.214828	0.745098

### 3.2.2 ZBIÓR "ŚREDNI"

*Wyniki eksperymentów na zbiorze "średnim" przy inicjowaniu populacji rozkładem jednostajnym na przedziale (0, 10).*

pop	iter	gen num	C	gamma	time	cv score	logloss	accuracy
10	5	2	4.210969	2.346232	1.56	2.295243	1.330491	0.794118
10	10	2	1.984242	2.91749	4.01	2.283937	1.275355	0.77451
10	20	19	4.208015	4.065623	12.57	2.293146	1.308217	0.784314
10	50	36	7.86615	1.737911	65.61	2.282742	1.359633	0.803922
10	100	3	5.304857	2.090131	247.65	2.280183	1.335532	0.784314
20	5	0	4.429249	3.738299	2.02	2.295799	1.321826	0.784314
20	10	2	7.804582	0.99723	6.55	2.271879	1.289294	0.784314
20	20	4	5.368216	2.191709	25.52	2.281566	1.311587	0.784314
20	50	46	2.82151	4.00809	129.87	2.282159	1.338717	0.784314
20	100	65	4.50327	2.230916	398.37	2.273145	1.305933	0.794118
50	5	1	3.876886	1.944297	7.02	2.287347	1.314148	0.784314
50	10	6	5.421475	2.191377	17.71	2.27483	1.32955	0.784314
50	20	19	3.877565	2.186777	45.62	2.278481	1.295872	0.803922
50	50	6	4.430782	1.905586	194.49	2.276811	1.328644	0.794118
50	100	68	7.080306	2.316395	614.58	2.274722	1.287643	0.813725
100	5	4	4.799429	1.711978	7.44	2.279487	1.294737	0.784314
100	10	5	3.862868	3.35176	19.63	2.279929	1.361676	0.794118
100	20	12	6.053327	2.118969	57.95	2.273388	1.342638	0.803922
100	50	33	5.850822	2.329582	313.4	2.271917	1.40191	0.813725
100	100	9	3.82968	2.877185	1368.35	2.270971	1.290899	0.784314
200	5	3	5.201249	1.92281	19.52	2.266819	1.308112	0.794118
200	10	1	4.909842	1.294052	39.96	2.265926	1.291878	0.764706
200	20	19	5.434341	1.97408	149.04	2.269738	1.309931	0.794118
200	50	8	4.3629	2.371978	718.88	2.269267	1.287321	0.794118
200	100	7	4.180534	2.662845	2143.99	2.271437	1.347307	0.794118

### 3.2.3 ZBIÓR "DROGI"

*Wyniki eksperymentów na zbiorze "drogim" przy inicjowaniu populacji rozkładem jednostajnym na przedziale (0, 10).*

pop	iter	gen num	C	gamma	time	cv score	logloss	accuracy
10	5	4	5.342617	4.435369	0.96	2.231863	1.327162	0.745098
10	10	2	1.831292	3.637094	2.63	2.234246	1.34815	0.72549
10	20	4	3.180908	4.327073	8.72	2.233035	1.290034	0.735294
10	50	8	4.504743	5.39172	46.6	2.230554	1.344548	0.705882
10	100	91	4.230199	4.456992	195.35	2.223204	1.408079	0.735294
20	5	3	3.554042	3.491802	2.24	2.234471	1.340022	0.735294
20	10	2	3.642608	3.52862	5.77	2.234601	1.271516	0.735294
20	20	6	3.328605	4.754344	16.68	2.225926	1.408666	0.715686
20	50	10	2.594965	3.599979	69.54	2.219404	1.279914	0.735294
20	100	41	3.776987	3.575837	248.45	2.22102	1.337346	0.735294
50	5	2	1.277379	4.07283	5.45	2.234366	1.376824	0.715686
50	10	7	4.315767	3.594954	16.35	2.221961	1.376849	0.735294
50	20	12	4.073571	3.972275	43.9	2.220088	1.320316	0.735294
50	50	32	3.411611	3.261005	167.21	2.22418	1.306324	0.735294
50	100	16	5.350751	4.360709	640.09	2.22456	1.405044	0.745098
100	5	1	9.494216	0.059406	6.99	2.206845	1.351693	0.696078
100	10	3	3.478495	3.956318	18.66	2.222903	1.338909	0.735294
100	20	9	3.278624	3.794142	55.02	2.22284	1.344396	0.735294
100	50	0	8.997209	0.059623	310.55	2.210296	1.4097	0.686275
100	100	47	3.980218	4.076469	1211.22	2.221863	1.39554	0.735294
200	5	4	4.108456	4.182859	15.01	2.229417	1.378256	0.735294
200	10	8	4.104505	3.729021	35.52	2.226191	1.278617	0.72549
200	20	9	3.507914	3.732463	104.63	2.220195	1.388671	0.72549
200	50	3	3.466952	3.910759	571.33	2.219035	1.376684	0.735294
200	100	85	3.748505	3.889421	2048.93	2.214077	1.294672	0.735294

### 3.3 ZESTAWIENIE WYNIKÓW OPTIMALIZACJI

Poniżej zostały zestawione uzyskane wartości metryk *logloss* i *accuracy* na zbiorze testowym w zależności od liczby generacji. Na końcu tej sekcji znajduje się porównanie wartości metryki *logloss* dla najlepszych modeli pod jej względem spośród kandydatów o różnej liczby generacji przy tej samej liczebności populacji oraz zestawienie numerów generacji, w której znaleziono parametry *C* i *gamma* danego modelu. W tabelach lepsze wyniki (czyli mniejszą wartość *loglossa* lub większą wartość *accuracy*) pogrubiono.



Tabela 3.1: Wartości metryki *logloss* na zbiorze testowym dla liczby generacji = 5.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	1.639387	<b>1.265425</b>	1.586244	<b>1.330491</b>	1.797332	<b>1.327162</b>
20	1.242285	<b>1.223727</b>	1.624254	<b>1.321826</b>	1.530664	<b>1.340022</b>
50	1.254817	<b>1.179453</b>	1.413652	<b>1.314148</b>	1.409115	<b>1.376824</b>
100	1.383943	<b>1.234954</b>	1.49959	<b>1.294737</b>	1.443904	<b>1.351693</b>
200	1.402856	<b>1.164818</b>	1.327915	<b>1.308112</b>	1.396679	<b>1.378256</b>

Tabela 3.2: Wartości metryki *accuracy* na zbiorze testowym dla liczby generacji = 5.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	0.735294	<b>0.754902</b>	0.696078	<b>0.794118</b>	0.588235	<b>0.745098</b>
20	0.72549	<b>0.754902</b>	0.72549	<b>0.784314</b>	0.705882	<b>0.735294</b>
50	0.715686	<b>0.745098</b>	0.696078	<b>0.784314</b>	<b>0.72549</b>	0.715686
100	0.715686	<b>0.754902</b>	0.735294	<b>0.784314</b>	0.696078	0.696078
200	0.715686	<b>0.754902</b>	0.72549	<b>0.794118</b>	0.715686	<b>0.735294</b>

Tabela 3.3: Wartości metryki *logloss* na zbiorze testowym dla liczby generacji = 10.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	<b>1.300349</b>	1.335839	1.448491	<b>1.275355</b>	<b>1.316599</b>	1.34815
20	1.440953	<b>1.164444</b>	1.515612	<b>1.289294</b>	1.421809	<b>1.271516</b>
50	1.413862	<b>1.409153</b>	1.748562	<b>1.32955</b>	1.476152	<b>1.376849</b>
100	1.335005	<b>1.245772</b>	1.60099	<b>1.361676</b>	1.339122	<b>1.338909</b>
200	1.29886	<b>1.185254</b>	1.386085	<b>1.291878</b>	1.403357	<b>1.278617</b>

Tabela 3.4: Wartości metryki *accuracy* na zbiorze testowym dla liczby generacji = 10.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	0.735294	0.735294	0.676471	<b>0.77451</b>	0.715686	<b>0.72549</b>
20	0.715686	<b>0.754902</b>	0.705882	<b>0.784314</b>	0.72549	<b>0.735294</b>
50	<b>0.705882</b>	0.676471	0.676471	<b>0.784314</b>	0.705882	<b>0.735294</b>
100	0.735294	<b>0.754902</b>	0.686275	<b>0.794118</b>	0.715686	<b>0.735294</b>
200	0.72549	<b>0.754902</b>	0.705882	<b>0.764706</b>	0.705882	<b>0.72549</b>

Tabela 3.5: Wartości metryki *logloss* na zbiorze testowym dla liczby generacji = 20.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	1.517469	<b>1.215829</b>	1.567837	<b>1.308217</b>	1.362445	<b>1.290034</b>
20	1.436476	<b>1.172379</b>	1.817341	<b>1.311587</b>	1.644196	<b>1.408666</b>
50	1.29743	<b>1.246558</b>	1.398061	<b>1.295872</b>	1.402371	<b>1.320316</b>
100	1.337457	<b>1.172785</b>	1.422814	<b>1.342638</b>	1.871298	<b>1.344396</b>
200	1.304115	<b>1.219916</b>	1.466367	<b>1.309931</b>	1.433705	<b>1.388671</b>

Tabela 3.6: Wartości metryki *accuracy* na zbiorze testowym dla liczby generacji = 20.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	0.735294	<b>0.754902</b>	0.676471	<b>0.784314</b>	0.715686	<b>0.735294</b>
20	0.735294	<b>0.754902</b>	0.656863	<b>0.784314</b>	<b>0.72549</b>	0.715686
50	0.735294	<b>0.754902</b>	0.705882	<b>0.803922</b>	0.705882	<b>0.735294</b>
100	0.72549	<b>0.754902</b>	0.715686	<b>0.803922</b>	0.715686	<b>0.735294</b>
200	0.72549	<b>0.754902</b>	0.705882	<b>0.794118</b>	0.715686	<b>0.72549</b>

Tabela 3.7: Wartości metryki *logloss* na zbiorze testowym dla liczby generacji = 50.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	1.300563	<b>1.287287</b>	1.423047	<b>1.359633</b>	1.598832	<b>1.344548</b>
20	1.289719	<b>1.190057</b>	1.376056	<b>1.338717</b>	1.460628	<b>1.279914</b>
50	1.335147	<b>1.205878</b>	1.647214	<b>1.328644</b>	1.697309	<b>1.306324</b>
100	1.375047	<b>1.3489</b>	1.503484	<b>1.40191</b>	<b>1.386049</b>	1.4097
200	1.361472	<b>1.222229</b>	1.479212	<b>1.287321</b>	1.42213	<b>1.376684</b>

Tabela 3.8: Wartości metryki *accuracy* na zbiorze testowym dla liczby generacji = 50.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	0.735294	<b>0.754902</b>	0.676471	<b>0.803922</b>	0.696078	<b>0.705882</b>
20	0.72549	<b>0.754902</b>	0.72549	<b>0.784314</b>	0.715686	<b>0.735294</b>
50	0.735294	<b>0.754902</b>	0.686275	<b>0.794118</b>	0.715686	<b>0.735294</b>
100	<b>0.735294</b>	0.705882	0.686275	<b>0.813725</b>	<b>0.715686</b>	0.686275
200	0.745098	<b>0.754902</b>	0.705882	<b>0.794118</b>	0.715686	<b>0.735294</b>

Tabela 3.9: Wartości metryki *logloss* na zbiorze testowym dla liczby generacji = 100.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	1.361659	<b>1.170609</b>	1.361726	<b>1.335532</b>	1.597466	<b>1.408079</b>
20	1.2993	<b>1.220786</b>	1.390432	<b>1.305933</b>	1.411296	<b>1.337346</b>
50	1.253516	<b>1.199325</b>	1.490703	<b>1.287643</b>	1.535622	<b>1.405044</b>
100	1.335966	<b>1.220114</b>	1.389641	<b>1.290899</b>	1.45305	<b>1.39554</b>
200	1.325503	<b>1.214828</b>	1.473277	<b>1.347307</b>	1.668056	<b>1.294672</b>

Tabela 3.10: Wartości metryki *accuracy* na zbiorze testowym dla liczby generacji = 100.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	0.72549	<b>0.745098</b>	0.696078	<b>0.784314</b>	0.715686	<b>0.735294</b>
20	0.72549	<b>0.754902</b>	0.705882	<b>0.794118</b>	0.715686	<b>0.735294</b>
50	<b>0.735294</b>	0.72549	0.705882	<b>0.813725</b>	0.715686	<b>0.745098</b>
100	0.72549	<b>0.754902</b>	0.676471	<b>0.784314</b>	0.705882	<b>0.735294</b>
200	0.72549	<b>0.745098</b>	0.696078	<b>0.794118</b>	0.705882	<b>0.735294</b>

Tabela 3.11: Najlepsze wartości metryki *logloss* na zbiorze testowym.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	1.300349	<b>1.170609</b>	1.361726	<b>1.275355</b>	1.316599	<b>1.290034</b>
20	1.242285	<b>1.164444</b>	1.376056	<b>1.289294</b>	1.411296	<b>1.271516</b>
50	1.253516	<b>1.179453</b>	1.398061	<b>1.287643</b>	1.402371	<b>1.306324</b>
100	1.335005	<b>1.172785</b>	1.389641	<b>1.290899</b>	1.339122	<b>1.338909</b>
200	1.29886	<b>1.164818</b>	1.327915	<b>1.287321</b>	1.396679	<b>1.278617</b>

Tabela 3.12: Numer generacji, w której osiągnięto najlepszą wartość metryki *logloss*.

pop	tani		średni		drogi	
	(0, 1)	(0, 10)	(0, 1)	(0, 10)	(0, 1)	(0, 10)
10	4	5	93	2	2	4
20	0	7	11	2	69	2
50	55	4	12	68	10	32
100	1	6	35	9	4	3
200	0	4	1	8	2	8

## 4 WNIOSKI

Realizując projekt, mieliśmy okazję zaobserwować wiele cech działania algorytmu ewolucyjnego w rzeczywistości. Analizując wykres zmian wartości metryki (Rysunek 1.1) w funkcji numeru generacji populacji widać, jak populacja może ewoluować. W naszej implementacji wprowadziliśmy możliwość zgubienia najlepszego osobnika, stąd program nie utożsamia się z jednym optimum lokalnym, a może pogorszyć swój stan, by później znaleźć osobnika lepszego. Na podstawie fluktuacji przebiegu wykresu widać również, że jest to metoda niedeterministyczna. Zastanawiające jest jednak to, że dążenie całej populacji do optimum jest stosunkowo słabe - na początku następuje duża poprawa w wartości metryki, ale osobniki w kolejnych generacjach zdają się być uzyskiwane całkowicie losowo, pomimo zastosowania selekcji turniejowej. Może to wynikać z dobrego początkowego losowania parametrów, istnienia wielu minimów lokalnych o zbliżonych wartościach funkcji celu lub weryfikacji zbyt wąskiego zakresu wartości parametrów  $C$  i  $gamma$ .

Porównując wyniki zgromadzone w tabelach w sekcji 4 dochodzimy do wniosku, że optymalizacja parametrów modelu SVM przy użyciu zaproponowanego algorytmu ewolucyjnego pozwoliła na stworzenie skutecznego klasyfikatora dla każdego zadania o różnej "jakości" danych. Dokładność klasyfikacji, niezależnie od wykorzystywanego zbioru, waha się w zakresie 70-80%, co jest dobrym wynikiem, zwłaszcza, że przypisania odbywały się do 13 klas, a każda z nich była reprezentowana tylko przez kilka przykładów uczących. Różnice w wartości tej metryki są jednak niewielkie przy różnych ustawieniach programu, co pokazuje problem doboru odpowiednich wartości parametrów algorytmu ewolucyjnego. Większe zmiany można zaobserwować dla metryki *logloss*, której wartości, już dla najlepszych osobników w każdym przypadku, zawierają się w zakresie [1.17, 1.41]. Zestawienie najlepszych wartości tej metryki w tabeli 3.11 pozwala zauważyć, że rośnie ona wraz ze wzrostem trudności danych, a co bardziej istotne - przyjmuje mniejsze wartości przy losowaniu populacji początkowej z szerszego zakresu. Prowadzi to do konkluzji, że szeroka eksploracja dziedziny daje lepsze efekty w naszym zadaniu. Z kolei na podstawie uzyskanych rezultatów, ciężko mówić o wyraźnym wpływie liczebności populacji na skuteczność działania algorytmu ewolucyjnego, ponieważ wartości *logloss* dość mocno zmieniają się w poszczególnych przypadkach, lecz w większości uruchomień procesu optymalizacji funkcja celu jest minimalizowana przy populacji liczącej 20 osobników. Zwracając uwagę na numer generacji, z której pochodzi najlepszy osobnik w odniesieniu do liczby generacji działania algorytmu (w tabelach w sekcji 3.2) można dostrzec tendencję do wybierania go z młodszych pokoleń, co prezentuje wolną zbieżność badanej metody. Zachowanie to nie przenosi się jednak na wybór najlepszego osobnika spośród wszystkich kandydatów, ponieważ w tabeli 3.12 brak opisanej zależności.

Podsumowując, algorytm ewolucyjny ze względu na to, że jest metodą losową jest trudny pod względem doboru właściwych parametrów, ale nawet mimo to, przynosi bardzo dobre efekty przy rozwiązywaniu postawionych problemów. Dla wielu typów zadań istnieją inne, dedykowane metody dające lepsze rezultaty, ale zaletą tej metody jest właśnie jej uniwersalność. W naszym projekcie spełnił on swoją rolę doboru optymalnych parametrów modelu SVM jako klasyfikatora na różnych przykładach zbiorów danych. Prawdopodobnie należałoby sprawdzić

jego działanie przy inicjacji populacji początkowej na szerszym zakresie wartości (optymalizacja parametrów SVM jest skomplikowanym problemem i nie ma żadnych dowodów na to jakich wartości należy oczekiwać), ale ze względu na liczbę budowanych modeli w każdej generacji i jeszcze wolniejsze zbieganie algorytmu przy bardziej różnorodnej populacji, nie było to możliwe przy dostępnych zasobach. Czas działania badanej metody rekompensuje w pewnym stopniu łatwość jej implementacji.