

## Zbiór MNIST

1. Wektor cech polegający na zlokalizowaniu „wierzchołków” cyfry. Przekształcenie oryginalnego wektora polega w pierwszym etapie na zastosowaniu progowania, tzn. z zakresu pikseli [0, 255] przechodzimy na zakres [0, 1] – progiem jest wartość piksela = 50 (mniej niż 50 -> 0, więcej niż 50 -> 1). Następnie sprawdzamy sąsiadów każdego piksela i jeśli ma on mniej niż 4 sąsiadów z wartością 1 to określany jest jako „wierzchołek” i przypisywana jest mu wartość 1, w innym przypadku 0.
2. Wektor składający się z wyciętej z oryginalnego wektora, centralnie położonej macierzy 10x10, tzn. elementy z oryginalnego wektora o pierwszej oraz drugiej z przedziału [10, 20]. W poniższych eksperymentach pozostałe parametry klasyfikatorów były domyślne dla biblioteki scikit-learn.

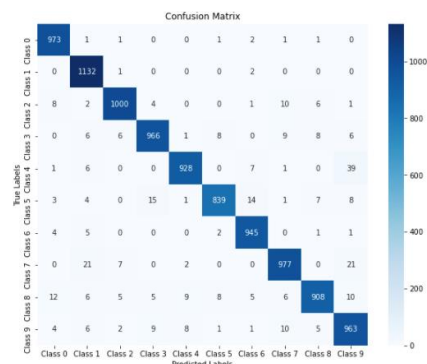
## MNIST Cecha 1 Eksperyment 1 Klasyfikator: kNN

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

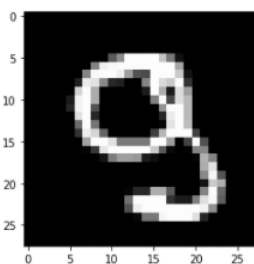
n_neighbors	weights	accuracy	n_neighbors	weights	accuracy
1	uniform	0.9606	25	uniform	0.9514
1	distance	0.9606	25	distance	0.9524
5	uniform	0.9631	50	uniform	0.9431
5	distance	0.9640	50	distance	0.9439

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.9631
Precision(weighted)	0.9635
Precision(macro)	0.9638
Recall(weighted)	0.9631
Recall (macro)	0.9625
F1(weighted)	0.9631
F1(macro)	0.9629

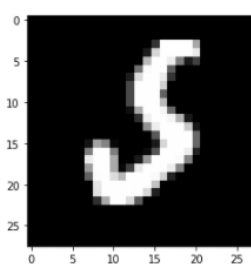


3. Przykładowe obrazy źle sklasyfikowane:



Prawdziwa klasa = 9

Predykcja = 8



Prawdziwa klasa = 5

Predykcja = 6

4. Wnioski:

Optymalną wartością parametru  $k$  ( $n\_neighbors$ ) jest 5. Widać tendencję, że przy zwiększaniu parametru  $k$  ( $n\_neighbors$ ) wydajność algorytmu maleje. Parametr  $weights$  nie wykazuje istotnego wpływu na wyniki, minimalnie lepszy 'distance'. Na podstawie macierzy pomyłek można stwierdzić, że najczęstszym błędem jest przewidzenie 9 zamiast 4.

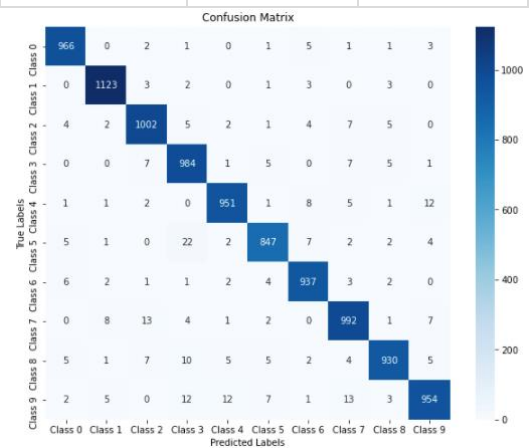
## MNIST Cecha 1 Eksperyment 2 Klasyfikator: MLP

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

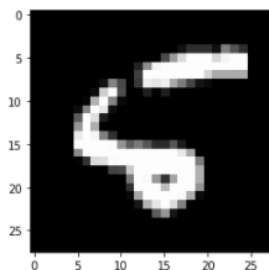
hls	activation	solver	accuracy	hls	activation	solver	accuracy
10	relu	sgd	0.8969	50	logistic	adam	0.9532
10	relu	adam	0.8656	100	logistic	sgd	0.9572
50	relu	sgd	0.9550	100	logistic	adam	0.9635
50	relu	adam	0.9450	10	tanh	sgd	0.9168
100	relu	sgd	0.9614	10	tanh	adam	0.9127
100	relu	adam	0.9573	50	tanh	sgd	0.9580
10	logistic	sgd	0.9182	50	tanh	adam	0.9492
10	logistic	adam	0.9164	100	tanh	sgd	0.9644
50	logistic	sgd	0.9535	100	tanh	adam	0.9604

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.9686
Precision(weighted)	0.9687
Precision(macro)	0.9686
Recall(weighted)	0.9686
Recall (macro)	0.9682
F1(weighted)	0.9686
F1(macro)	0.9683

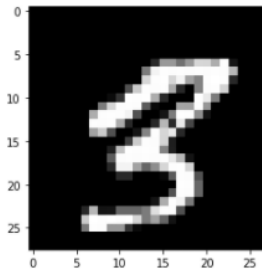


3. Przykładowe obrazy źle sklasyfikowane:



Prawdziwa klasa = 5

Predykcja = 6



Prawdziwa klasa = 3

Predykcja = 8

4. Wnioski:

Można stwierdzić, że im więcej neuronów w warstwie ukrytej tym dokładniejszy model, np. dla 10 neuronów najgorsze wyniki, dla 100 neuronów najlepsze. Solver 'sgd' daje minimalnie lepsze wyniki niż solver 'adam'. Funkcje aktywacji nie wykazywały większego wpływu na wyniki, z wyłączeniem 'relu', która dawała wyraźnie słabsze wyniki. Najczęstszym błędem modelu jest przewidzenie 3 zamiast 5.

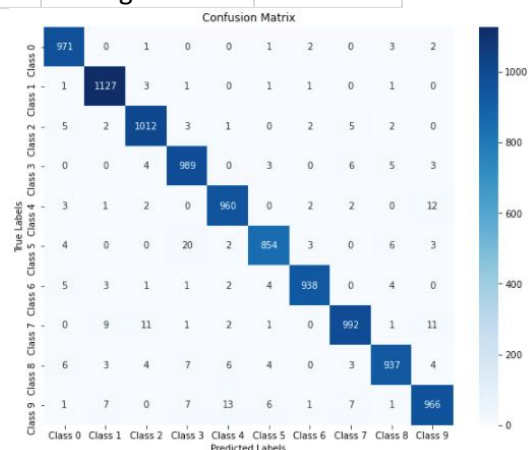
## MNIST Cecha 1 Eksperyment 3 Klasyfikator: SVM

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

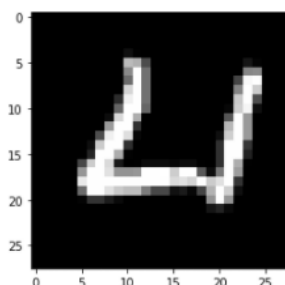
C	kernel	accuracy	C	kernel	accuracy
1	linear	0.9151	50	linear	0.9015
1	poly	0.9652	50	poly	0.9633
1	rbf	0.9644	50	rbf	0.9711
1	sigmoid	0.6101	50	sigmoid	0.5260
10	linear	0.9043	100	linear	0.9004
10	poly	0.9633	100	poly	0.9633
10	rbf	0.9718	100	rbf	0.9711
10	sigmoid	0.4832	100	sigmoid	0.5590

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.9746
Precision(weighted)	0.9746
Precision(macro)	0.9747
Recall(weighted)	0.9746
Recall (macro)	0.9742
F1(weighted)	0.9745
F1(macro)	0.9744

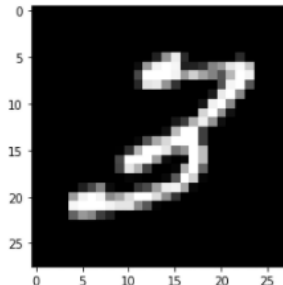


3. Przykładowe obrazy źle sklasyfikowane:



Prawdziwa klasa = 4

Predykcja = 0



Prawdziwa klasa = 3

Predykcja = 8

4. Wnioski:

Widać wyraźnie, że najlepsze wyniki daje model z jądrem 'rbf', a najgorsze z jądrem 'sigmoid'. Parametr C daje minimalnie lepsze wyniki, gdy jest większy niż 1, jednak jego zmiana nie wpływa istotnie na wyniki modelu. Na podstawie macierzy pomyłek można stwierdzić, że najczęstszym błędem jest przewidzenie 3 zamiast 5.

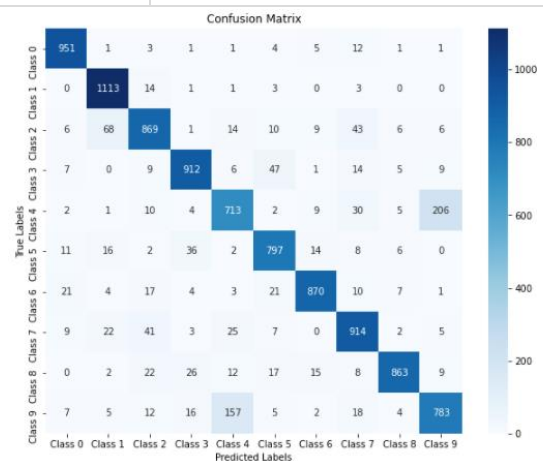
## MNIST Cecha 2 Eksperyment 4 Klasyfikator: kNN

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

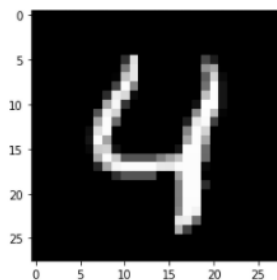
n_neighbors	weights	accuracy
1	uniform	0.8663
1	distance	0.8663
5	uniform	0.8761
5	distance	0.8783
25	uniform	0.8619
25	distance	0.8648
50	uniform	0.8482
50	distance	0.8518

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.8785
Precision(weighted)	0.8785
Precision(macro)	0.8786
Recall(weighted)	0.8785
Recall (macro)	0.8775
F1(weighted)	0.8780
F1(macro)	0.8776

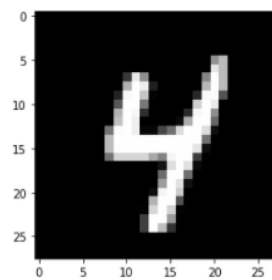


3. Przykładowe obrazy źle sklasyfikowane:



Prawdziwa klasa = 4

Predykcja = 9



Prawdziwa klasa = 4

Predykcja = 9

4. Wnioski:

Optymalną wartością parametru  $k(n\_neighbors)$  jest 5. Widać tendencję, że przy zwiększaniu parametru  $k(n\_neighbors)$  wydajność algorytmu maleje. Parametr  $weights$  nie wykazuje istotnego wpływu na wyniki, minimalnie lepszy 'distance'. Na podstawie macierzy pomyłek można stwierdzić, że najczęstszym błędem jest pomylenie 4 z 9.

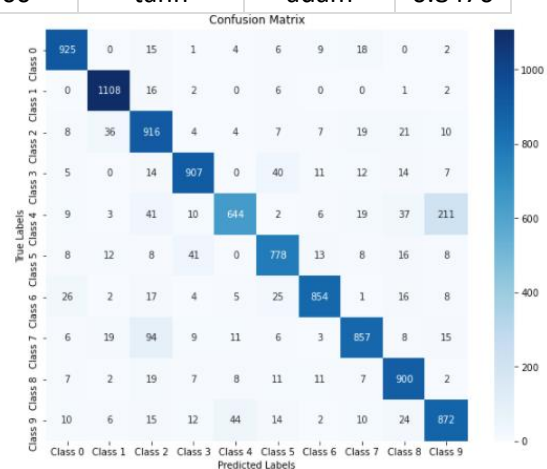
## MNIST Cecha 2 Eksperyment 5 Klasyfikator: MLP

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

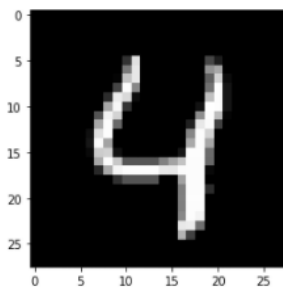
hls	activation	solver	accuracy	hls	activation	solver	accuracy
10	relu	sgd	0.4105	50	logistic	adam	0.8493
10	relu	adam	0.7614	100	logistic	sgd	0.8700
50	relu	sgd	0.7555	100	logistic	adam	0.8710
50	relu	adam	0.8609	10	tanh	sgd	0.7221
100	relu	sgd	0.8298	10	tanh	adam	0.7377
100	relu	adam	0.8744	50	tanh	sgd	0.8284
10	logistic	sgd	0.7320	50	tanh	adam	0.8333
10	logistic	adam	0.7565	100	tanh	sgd	0.8576
50	logistic	sgd	0.8484	100	tanh	adam	0.8476

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.8761
Precision(weighted)	0.8791
Precision(macro)	0.8789
Recall(weighted)	0.8761
Recall (macro)	0.8747
F1(weighted)	0.8749
F1(macro)	0.8741

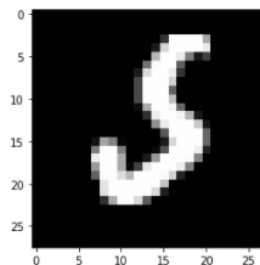


3. Przykładowe obrazy źle sklasyfikowane:



Prawdziwa klasa = 4

Predykcja = 9



Prawdziwa klasa = 5

Predykcja = 2

4. Wnioski:

Można stwierdzić, że im więcej neuronów w warstwie ukrytej tym dokładniejszy model, np. dla 10 neuronów najgorsze wyniki, dla 100 neuronów najlepsze. Solver 'adam' daje minimalnie lepsze wyniki niż solver 'sgd'. Funkcje aktywacji nie wykazywały większego wpływu na wyniki, z wyłączeniem 'relu', która dawała wyraźnie słabsze wyniki. Najczęstszym błędem modelu jest przewidzenie 9 zamiast 4.

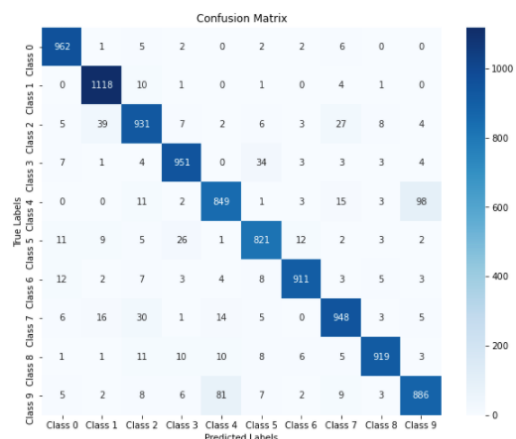
## MNIST Cecha 2 Eksperyment 6 Klasyfikator: SVM

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

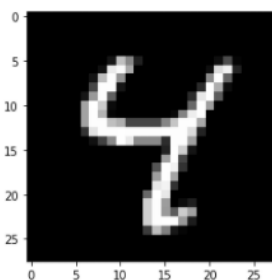
C	kernel	accuracy	C	kernel	accuracy
1	poly	0.9139	50	poly	0.9078
1	rbf	0.919	50	rbf	0.9207
1	sigmoid	0.4385	50	sigmoid	0.4342
10	poly	0.9126	100	poly	0.9067
10	rbf	0.9242	100	rbf	0.9194
10	sigmoid	0.4348	100	sigmoid	0.4342

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.9296
Precision(weighted)	0.9295
Precision(macro)	0.9296
Recall(weighted)	0.9296
Recall (macro)	0.9290
F1(weighted)	0.9294
F1(macro)	0.9292

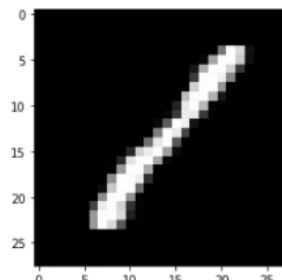


3. Przykładowe obrazy źle sklasyfikowane:



Prawdziwa klasa = 4

Predykcja = 9



Prawdziwa klasa = 1

Predykcja = 2

4. Wnioski:

Widać wyraźnie, że najlepsze wyniki daje model z jądrem 'rbf', a najgorsze z jądrem 'sigmoid'. Parametr C nie ma znacznego wpływu na wyniki. Na podstawie macierzy pomyłek można stwierdzić, że najczęstszym błędem jest pomylenie 4 z 9.

## Cora

1. Wektor cech wbudowany: 1433 elementowymi wektorami kodującymi wystąpienie słów w artykule lub nie.
2. Wektor cech z ekstrakcji pewnych cech: Samodzielnie wyznaczono 8 elementowy wektor cech, który składa się z 2 informacji: Indeks '0' mówi o liczbie cytacji dla danego artykułu, np. artykuł cytuje 3 dokumenty wartość dla indeksu '0' jest równa 3. Indeksy od '1' do '7', wielkość tej informacji jest zależna od liczby klas artykułów w zbiorze Cora. Zbiór Cora posiada 7 klas artykułów. Algorytm zlicza liczbę cytowań poszczególnych klas artykułów dla każdego artykułu, wartości są zamieszczane w liście 7 elementowej np. [2., 0., 2., 2., 0., 0., 0.]. Następnie 2 informacje są łączone w jeden wektor cech [3., 2., 0., 2., 2., 0., 0.].

W eksperymentach pozostałe parametry klasyfikatorów były domyślne dla biblioteki scikit-learn.

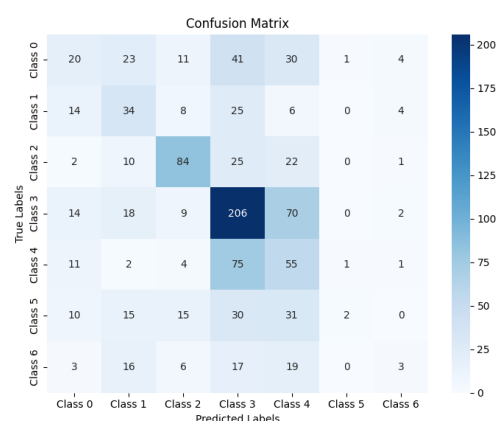
### Cora Cecha 1 Eksperyment 7 Klasyfikator: kNN

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

n_neighbors	weights	accuracy
1	uniform	0.3438
1	distance	0.3438
5	uniform	0.3734
5	distance	0.3703
25	uniform	0.3500
25	distance	0.3578
50	uniform	0.2922
50	distance	0.2969
100	uniform	0.3078
100	distance	0.3047

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.4040
Precision(weighted)	0.4060
Precision(macro)	0.3713
Recall(weighted)	0.4040
Recall (macro)	0.3131
F1(weighted)	0.3709
F1(macro)	0.2970



### 3. Wnioski

Im mniejsza liczba sąsiadów branych pod uwagę w klasyfikatorze knn tym dokładność jest wyższa. Wagi uniform i distance dają podobne wyniki. Najwięcej poprawnych wyników uzyskano dla klasy 3, ponieważ jest ona najliczniejszą z klas.

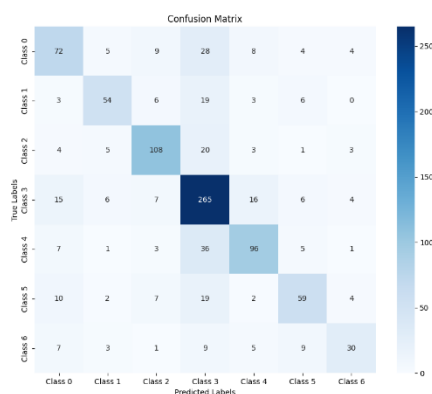
## Cora Cecha 1 Eksperyment 8 Klasyfikator: MLP

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

hls	activation	solver	accuracy	hls	activation	solver	accuracy
5	identity	adam	0.6406	(5, 5)	identity	adam	0.5891
5	identity	sgd	0.4469	(5, 5)	identity	sgd	0.4250
5	identity	lbfgs	0.6156	(5, 5)	identity	lbfgs	0.5953
25	identity	adam	0.6547	(5, 5, 5)	identity	adam	0.5656
25	identity	sgd	0.5578	(5, 5, 5)	identity	sgd	0.4281
25	identity	lbfgs	0.6422	(5, 5, 5)	identity	lbfgs	0.5563
50	identity	adam	0.6609	5	relu	adam	0.6047
50	identity	sgd	0.5781	5	relu	sgd	0.3688
50	identity	lbfgs	0.6609	5	relu	lbfgs	0.5734
25	relu	adam	0.6781	(5, 5)	relu	adam	0.5250
25	relu	sgd	0.4000	(5, 5)	relu	sgd	0.3156
25	relu	lbfgs	0.6469	(5, 5)	relu	lbfgs	0.5203
50	relu	adam	0.6625	(5, 5, 5)	relu	adam	0.4562
50	relu	sgd	0.4156	(5, 5, 5)	relu	sgd	0.2828
50	relu	lbfgs	0.6500	(5, 5, 5)	relu	lbfgs	0.4359
5	tanh	adam	0.6312	(5, 5)	tanh	adam	0.5359
5	tanh	sgd	0.4328	(5, 5)	tanh	sgd	0.3531
5	tanh	lbfgs	0.5672	(5, 5)	tanh	lbfgs	0.5359
25	tanh	adam	0.6672	(5, 5, 5)	tanh	adam	0.5250
25	tanh	sgd	0.5219	(5, 5, 5)	tanh	sgd	0.3531
25	tanh	lbfgs	0.6656	(5, 5, 5)	tanh	lbfgs	0.5172
50	tanh	adam	0.6594				
50	tanh	sgd	0.5484				
50	tanh	lbfgs	0.6687				

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.6840
Precision(weighted)	0.6846
Precision(macro)	0.6836
Recall(weighted)	0.6840
Recall (macro)	0.6305
F1(weighted)	0.6793
F1(macro)	0.6520



3. Wnioski:

Dla badanych scenariuszy testowych solvery adam i lbfgs radzą sobie zdecydowanie lepiej niż sgd. Można zauważyć tendencję do zwiększania się dokładności wraz z większą liczbą neuronów w warstwie ukrytej. Natomiast zwiększona liczba warstw ukrytych wpływa niekorzystnie na uzyskiwane wyniki.



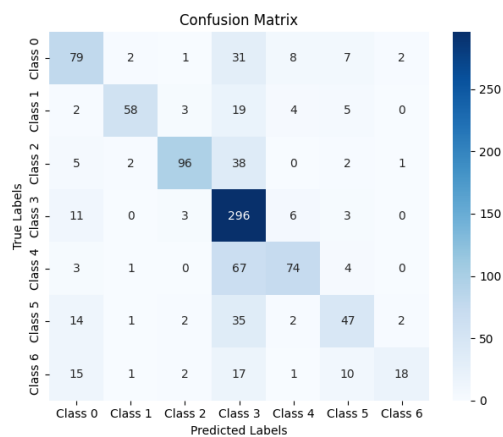
## Cora Cecha 1 Eksperyment 9 Klasyfikator: SVM

- Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

C	kernel	accuracy	C	kernel	accuracy
0.1	linear	0.6438	0.1	rbf	0.2781
0.1	poly	0.2781	0.1	sigmoid	0.2781
1	linear	0.6156	50	linear	0.6156
1	poly	0.2812	50	poly	0.3219
1	rbf	0.5703	50	rbf	0.6547
1	sigmoid	0.6594	50	sigmoid	0.5953
10	linear	0.6156	100	linear	0.6156
10	poly	0.2828	100	poly	0.3422
10	rbf	0.6547	100	rbf	0.6547
10	sigmoid	0.5984	100	sigmoid	0.5953

- Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.6680
Precision(weighted)	0.7059
Precision(macro)	0.7364
Recall(weighted)	0.6680
Recall (macro)	0.5820
F1(weighted)	0.6572
F1(macro)	0.6254



- Wnioski:

Na podstawie wyników uzyskanych z eksperymentu można wywnioskować, że dla jądra miary wartość C powinna być jak najmniejsza, przeciwnie jest natomiast w przypadku wykorzystania pozostałych jąder: poly, rbf i sigmoid.

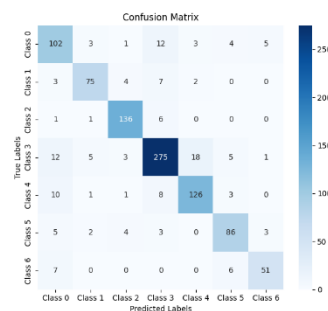
## Cora Cecha 2 Eksperyment 10 Klasyfikator: kNN

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

n_neighbors	weights	accuracy
1	uniform	0.8203
1	distance	0.8203
5	uniform	0.8797
5	distance	0.8656
25	uniform	0.8547
25	distance	0.8609
50	uniform	0.8375
50	distance	0.8531
100	uniform	0.7391
100	distance	0.8515

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.8510
Precision(weighted)	0.8522
Precision(macro)	0.8443
Recall(weighted)	0.851
Recall (macro)	0.8418
F1(weighted)	0.8513
F1(macro)	0.8427



3. Wnioski:

Na podstawie badanych scenariuszy testowych dla zbioru Cora wykorzystując wagę uniform i distance warto ustawiać wartość k niską w granicy 5 a 10, wtedy uzyskamy najlepszą dokładność.

## Cora Cecha 2 Eksperyment 11 Klasyfikator: MLP

1. Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

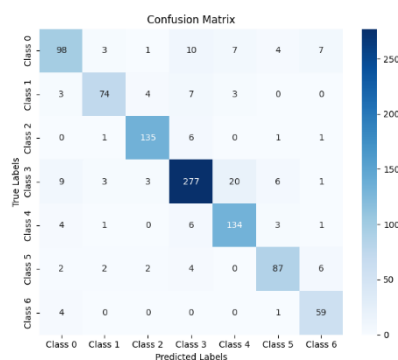
hls	activation	solver	accuracy	hls	activation	solver	accuracy
5	identity	adam	0.8281	(5, 5)	identity	adam	0.8141
5	identity	sgd	0.8281	(5, 5)	identity	sgd	0.8359
5	identity	lbfgs	0.8656	(5, 5)	identity	lbfgs	0.8641
25	identity	adam	0.8719	(5, 5, 5)	identity	adam	0.8172
25	identity	sgd	0.8672	(5, 5, 5)	identity	sgd	0.8219
25	identity	lbfgs	0.8672	(5, 5, 5)	identity	lbfgs	0.8672
50	identity	adam	0.8719	5	relu	adam	0.7922
50	identity	sgd	0.8688	5	relu	sgd	0.6828
50	identity	lbfgs	0.8672	5	relu	lbfgs	0.8469
25	relu	adam	0.8672	(5, 5)	relu	adam	0.6906
25	relu	sgd	0.8562	(5, 5)	relu	sgd	0.3703
25	relu	lbfgs	0.8500	(5, 5)	relu	lbfgs	0.8328
50	relu	adam	0.8750	(5, 5, 5)	relu	adam	0.7063
50	relu	sgd	0.8609	(5, 5, 5)	relu	sgd	0.4750
50	relu	lbfgs	0.8438	(5, 5, 5)	relu	lbfgs	0.7406
5	tanh	adam	0.8313	(5, 5)	tanh	adam	0.8406
5	tanh	sgd	0.6109	(5, 5)	tanh	sgd	0.6469
5	tanh	lbfgs	0.8719	(5, 5)	tanh	lbfgs	0.8578
25	tanh	adam	0.8734	(5, 5, 5)	tanh	adam	0.8094
25	tanh	sgd	0.8641	(5, 5, 5)	tanh	sgd	0.5016
25	tanh	lbfgs	0.8406	(5, 5, 5)	tanh	lbfgs	0.8594
50	tanh	adam	0.8738				
50	tanh	sgd	0.8641				
50	tanh	lbfgs	0.8500				

2. Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.8640
Precision(weighted)	0.8654
Precision(macro)	0.8541
Recall(weighted)	0.8640
Recall (macro)	0.8627
F1(weighted)	0.8637
F1(macro)	0.8570

3. Wnioski:

Podobnie jak w przypadku pierwszego wektora zwiększanie liczby warstw ukrytych nie poprawia wyników dokładności. Przy dużo lepiej zdefiniowanym wektorze cech nadal solvery adam i lbfgs radzą sobie zdecydowanie lepiej niż sgd. Można zauważyć tendencję do zwiększania się dokładności wraz większą liczbą neuronów w warstwie ukrytej.



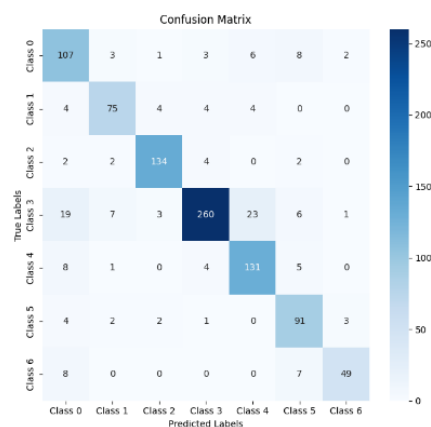
## Cora Cecha 2 Eksperyment 12 Klasyfikator: SVM

- Przedstawienie różnych kombinacji parametrów i wskazanie optymalnej:

C	kernel	accuracy	C	kernel	accuracy
0.1	linear	0.8703	0.1	rbf	0.6688
0.1	poly	0.4063	0.1	sigmoid	0.5922
1	linear	0.8688	50	linear	0.8703
1	poly	0.5469	50	poly	0.7094
1	rbf	0.8594	50	rbf	0.8703
1	sigmoid	0.8172	50	sigmoid	0.7453
10	linear	0.8688	100	linear	0.8672
10	poly	0.6500	100	poly	0.7250
10	rbf	0.8734	100	rbf	0.8719
10	sigmoid	0.7578	100	sigmoid	0.7438

- Szczegółowa analiza optymalnej kombinacji:

miara	wartość
Accuracy	0.8470
Precision(weighted)	0.8567
Precision(macro)	0.8378
Recall(weighted)	0.8470
Recall (macro)	0.8459
F1(weighted)	0.8487
F1(macro)	0.8389



- Wnioski:

Dla badanego scenariusza testowego jądra linear i rbg działają wydajniej na dokładność w porównaniu z poly i sigmoid. Dla jąder rbf i poly wartości C powinny być wyższe niż wykorzystując jądro linear.