SPRAWOZDANIE – LABORATORIUM 5

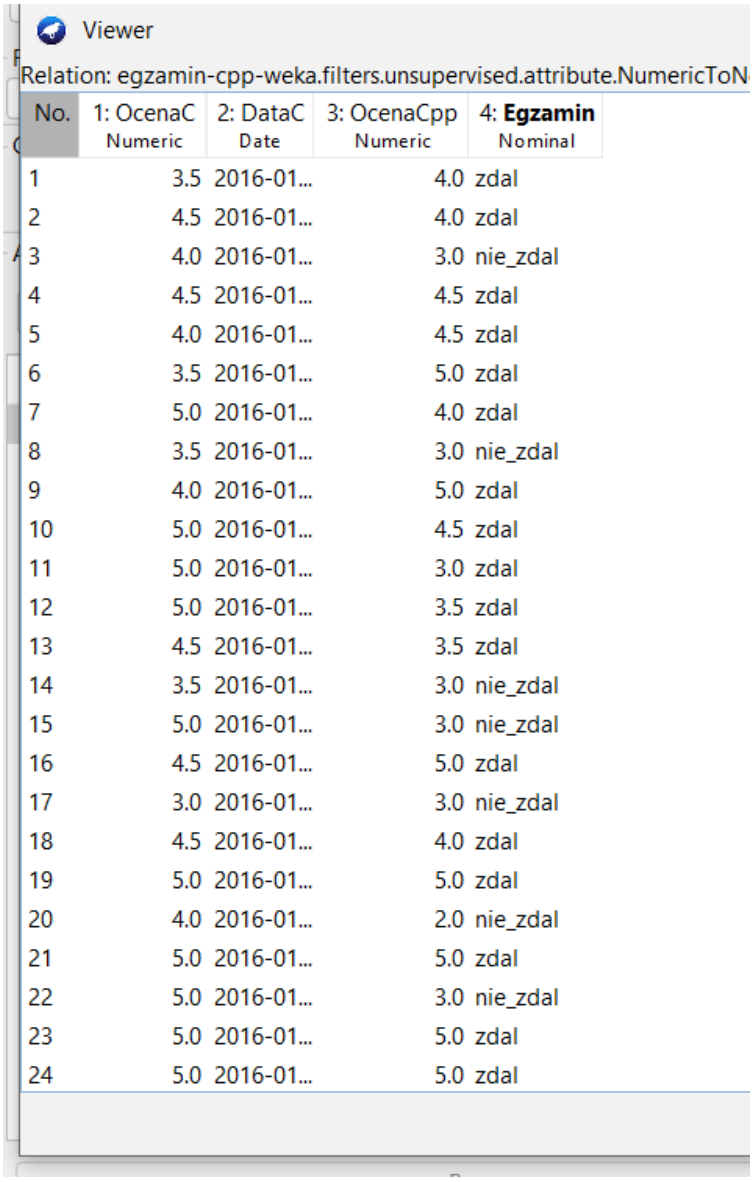
Karolina Kotłowska, 9 kwietnia 2023

5.1

```
  java -cp Weka-3-8/weka.jar weka.core.converters.CSVLoader egzamin-cpp.csv
-S 1 -F ; -D 3 -format yyyy-mm-dd > egzamin-cpp-train.arff

  java  -cp  Weka-3-8/weka.jar  weka.core.converters.CSVLoader  egzamin-cpp-
train.csv -S 1 -F ; -D 3 -format yyyy-mm-dd > egzamin-cpp-train.arff

  java  -cp  Weka-3-8/weka.jar  weka.core.converters.CSVLoader  egzamin-cpp-
test.csv -S 1 -F ; -D 3 -N 5 -format yyyy-mm-dd > egzamin-cpp-train.arff
```

5.2

egzamin-cpp.arff



No.	1: OcenaC Numeric	2: DataC Date	3: OcenaCpp Numeric	4: **Egzamin** Nominal
1	3.5	2016-01...	4.0	zdal
2	4.5	2016-01...	4.0	zdal
3	4.0	2016-01...	3.0	nie_zdal
4	4.5	2016-01...	4.5	zdal
5	4.0	2016-01...	4.5	zdal
6	3.5	2016-01...	5.0	zdal
7	5.0	2016-01...	4.0	zdal
8	3.5	2016-01...	3.0	nie_zdal
9	4.0	2016-01...	5.0	zdal
10	5.0	2016-01...	4.5	zdal
11	5.0	2016-01...	3.0	zdal
12	5.0	2016-01...	3.5	zdal
13	4.5	2016-01...	3.5	zdal
14	3.5	2016-01...	3.0	nie_zdal
15	5.0	2016-01...	3.0	nie_zdal
16	4.5	2016-01...	5.0	zdal
17	3.0	2016-01...	3.0	nie_zdal
18	4.5	2016-01...	4.0	zdal
19	5.0	2016-01...	5.0	zdal
20	4.0	2016-01...	2.0	nie_zdal
21	5.0	2016-01...	5.0	zdal
22	5.0	2016-01...	3.0	nie_zdal
23	5.0	2016-01...	5.0	zdal
24	5.0	2016-01...	5.0	zdal

egzamin-cpp-train.arff

Viewer

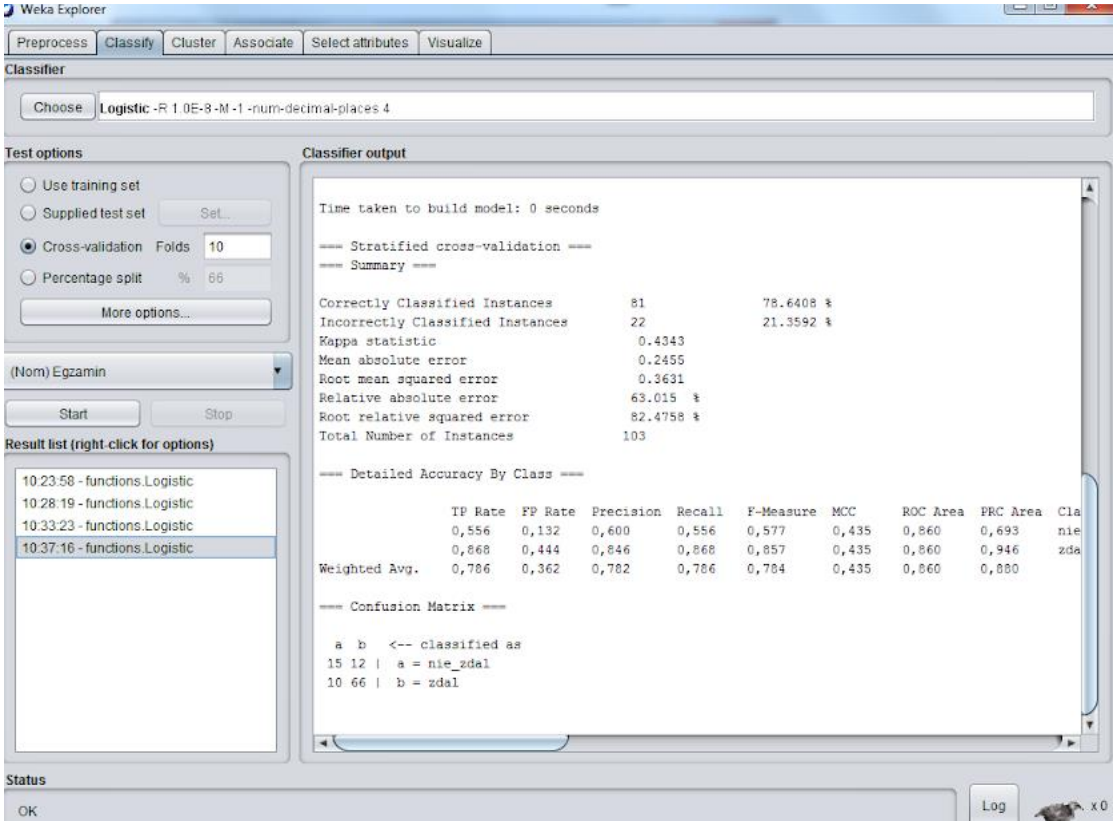
Relation: egzamin-cpp-train-weka.filters.unsupervised.attribute.N

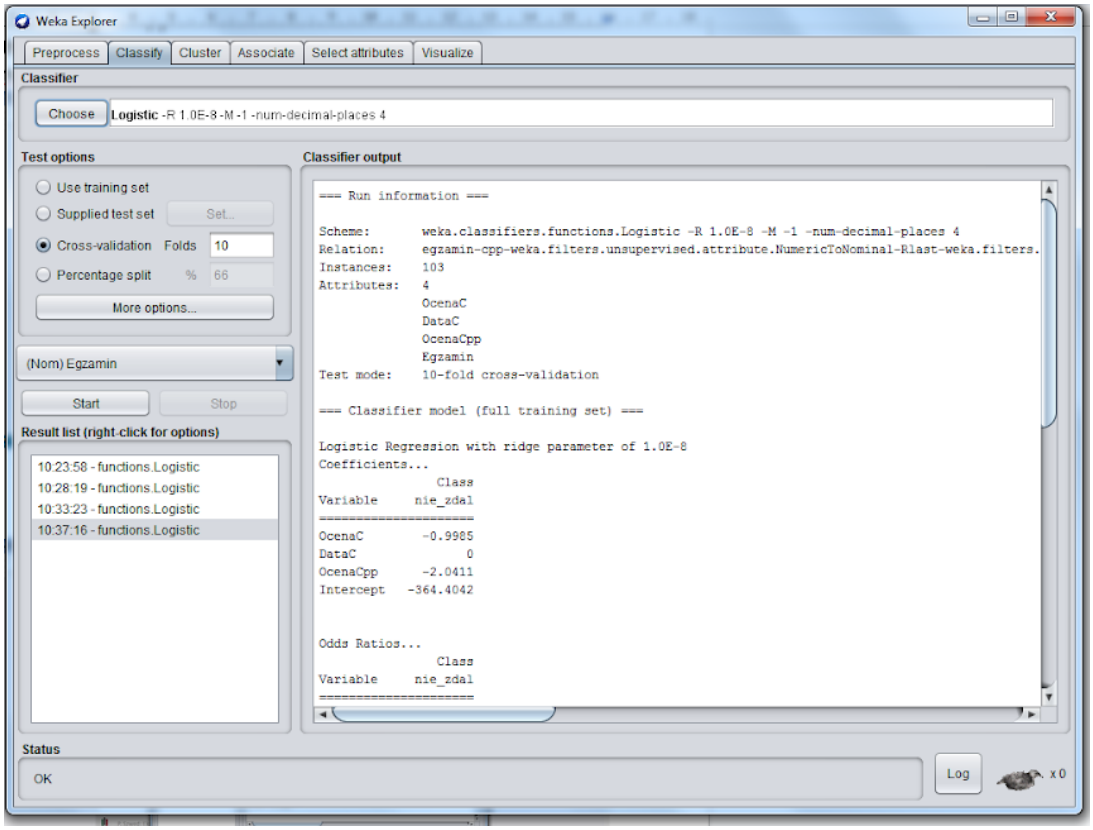
No.	1: OcenaC Numeric	2: DataC Date	3: OcenaCpp Numeric	4: **Egzamin** Nominal
1	3.5	2016-01...	4.0	zdal
2	4.5	2016-01...	4.0	zdal
3	4.0	2016-01...	3.0	nie_zdal
4	4.5	2016-01...	4.5	zdal
5	4.0	2016-01...	4.5	zdal
6	3.5	2016-01...	5.0	zdal
7	5.0	2016-01...	4.0	zdal
8	3.5	2016-01...	3.0	nie_zdal
9	4.0	2016-01...	5.0	zdal
10	5.0	2016-01...	4.5	zdal
11	5.0	2016-01...	3.0	zdal
12	5.0	2016-01...	3.5	zdal
13	4.5	2016-01...	3.5	zdal
14	3.5	2016-01...	3.0	nie_zdal
15	5.0	2016-01...	3.0	nie_zdal
16	4.5	2016-01...	5.0	zdal
17	3.0	2016-01...	3.0	nie_zdal
18	4.5	2016-01...	4.0	zdal
19	5.0	2016-01...	5.0	zdal
20	4.0	2016-01...	2.0	nie_zdal
21	5.0	2016-01...	5.0	zdal
22	5.0	2016-01...	3.0	nie_zdal
23	5.0	2016-01...	5.0	zdal
24	5.0	2016-01...	5.0	zdal

5.3

egzamin-cpp.arff

WALIDACJA KRZYŻOWA

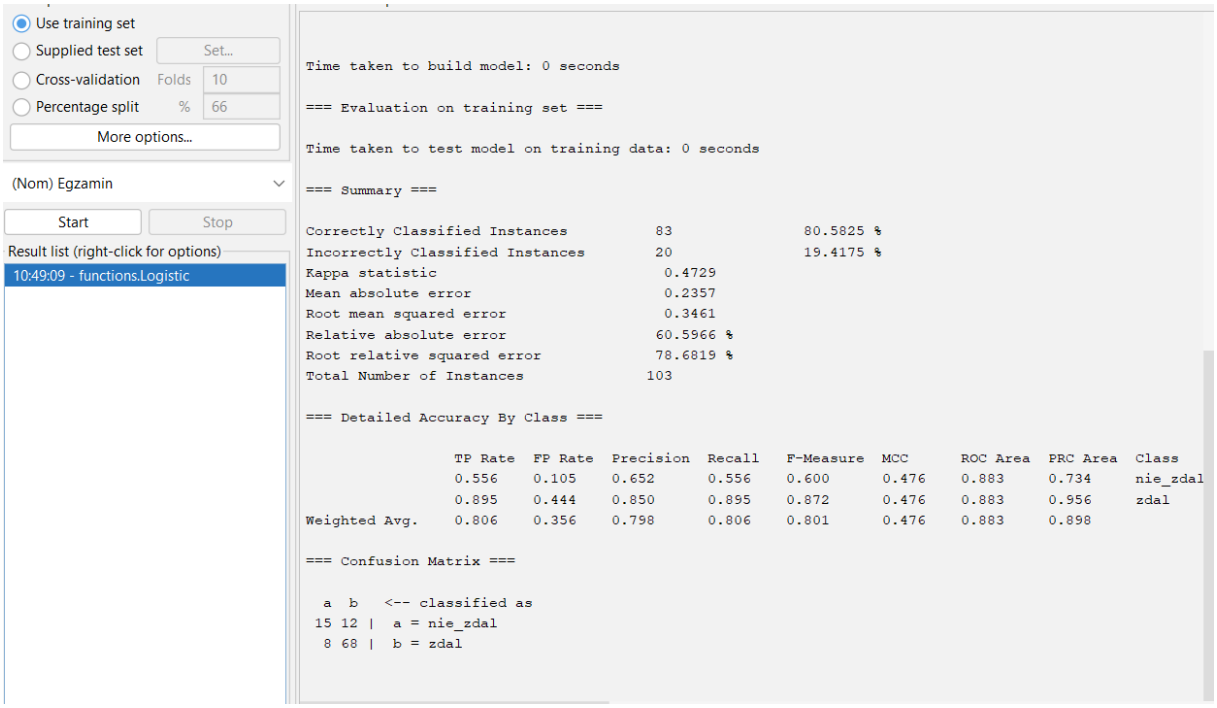




Dla walidacji krzyżowej $e^{-0.99} = 0.371$ -> 37.1% wzrost oceny o 1, zwiększa szanse zdania egzaminu o 37.1%

ZBIÓR UCZĄCY:

```
=== Run Information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     egzamin-cpp-weka.filters.unsupervised.attribute.NumericToNominal-Rlast-weka.filters.unsup
Instances:    103
Attributes:   4
              OcenaC
              DataC
              OcenaCpp
              Egzamin
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                Class
Variable     nie_zdal
====================
OcenaC        -0.9985
DataC               0
OcenaCpp      -2.0411
Intercept   -364.4042


Odds Ratios...
                Class
Variable     nie_zdal
====================
OcenaC          0.3684
DataC                1
OcenaCpp        0.1299
```




```
Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances          83               80.5825 %
Incorrectly Classified Instances        20               19.4175 %
Kappa statistic                          0.4729
Mean absolute error                      0.2357
Root mean squared error                  0.3461
Relative absolute error                 60.5966 %
Root relative squared error             78.6819 %
Total Number of Instances              103

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.556    0.105    0.652      0.556    0.600      0.476    0.883     0.734     nie_zdal
                 0.895    0.444    0.850      0.895    0.872      0.476    0.883     0.956     zdal
Weighted Avg.    0.806    0.356    0.798      0.806    0.801      0.476    0.883     0.898

=== Confusion Matrix ===

  a  b   <-- classified as
 15 12 |  a = nie_zdal
  8 68 |  b = zdal
```

Analiza dla zbioru uczącego:

1. Podaj wzór na hiperpłaszczyznę separującą dane

F(x) = -0.9198 * OcenaC -2.5471 * OcenaCpp – 1952.6739

2. Podaj o ile wzrost/spadek ocen wpływa na szanse zdania/niezdania egzaminu

$e^{-0.91} = 0.402$ -> 40,2% wzrost oceny o 1, zwiększa szanse zdania egzaminu o 40,2%

3. Jak wpływa na egzamin zmiana daty wpisu zaliczenia?

Data nie wpływa na egzamin.

4. Zinterpretuj wyniki klasyfikacji.

Dokładność zdefiniowana jako $precision = \frac{TP}{TP+FP}$ wynosi 0.65.

Trafność zdefiniowana jako $accuracy = \frac{TP+TN}{TP+FP+FN+TN}$ wynosi 0.80.

Czułość zdefiniowana jako $recall = \frac{TP}{TP+FN}$ wynosi 0.75.

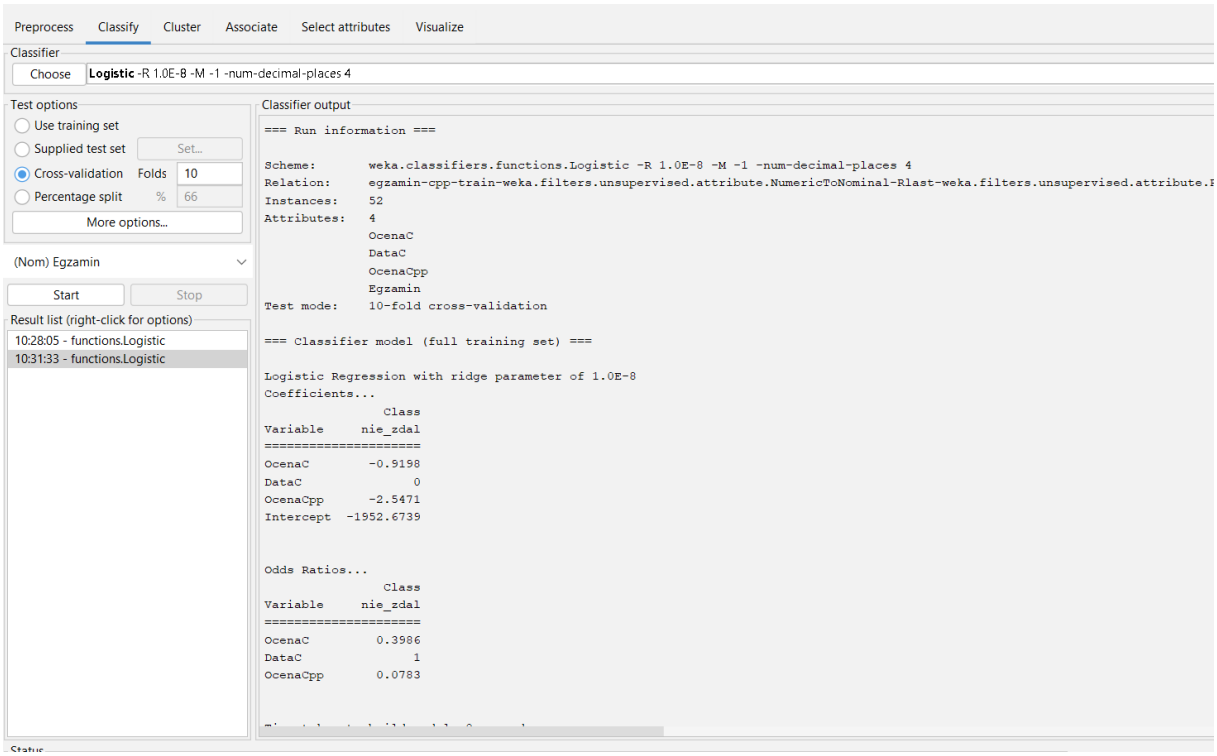
Miara F1 zdefiniowana jako $F1 = 2 * \frac{precission*recall}{precission+recall}$ wynosi 0.36.

5. Porównaj wyniki testów z użyciem zbioru uczącego i walidacji krzyżowej

Porównanie w formie screenshotów wyżej.

egzamin-cpp-train.arff

WALIDACJA KRZYŻOWA

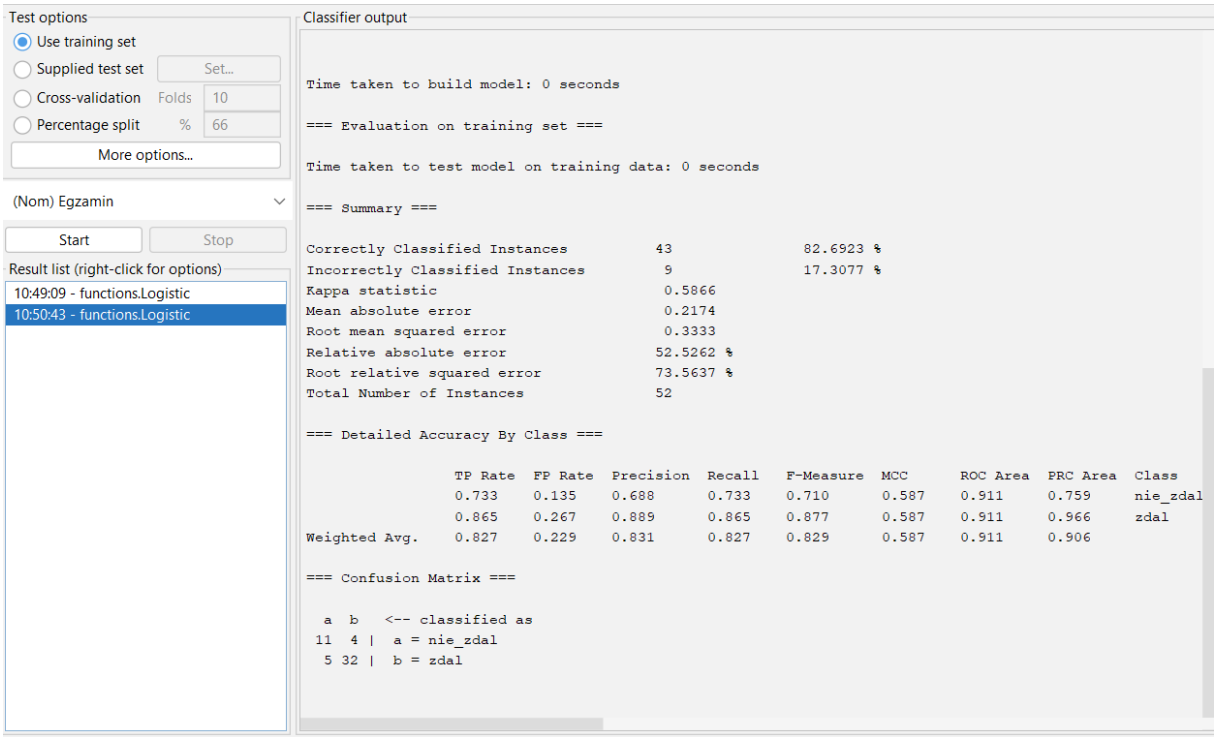



```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          40               76.9231 %
Incorrectly Classified Instances        12               23.0769 %
Kappa statistic                          0.4378
Mean absolute error                      0.2374
Root mean squared error                  0.3723
Relative absolute error                 57.1204 %
Root relative squared error             81.8155 %
Total Number of Instances               52

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.600    0.162    0.600      0.600    0.600      0.438    0.883     0.661     nie_zdal
                 0.838    0.400    0.838      0.838    0.838      0.438    0.883     0.958     zdal
Weighted Avg.    0.769    0.331    0.769      0.769    0.769      0.438    0.883     0.872

=== Confusion Matrix ===

  a  b   <-- classified as
  9  6 |  a = nie_zdal
  6 31 |  b = zdal
```

Dla walidacji krzyżowej $e^{-0.91} = 0.402$ -> 40.2% wzrost oceny o 1, zwiększa szanse zdania egzaminu o 40.2%

ZBIÓR UCZĄCY

```
=== Run information ===

Scheme:       weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Relation:     egzamin-cpp-train-weka.filters.unsupervised.attribute.NumericToNominal-Rlast-weka.filters
Instances:    52
Attributes:   4
              OcenaC
              DataC
              OcenaCpp
              Egzamin
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                Class
Variable     nie_zdal
====================
OcenaC        -0.9198
DataC               0
OcenaCpp      -2.5471
Intercept  -1952.6739


Odds Ratios...
                Class
Variable     nie_zdal
====================
OcenaC          0.3986
DataC                1
OcenaCpp        0.0783
```



Analiza dla zbioru uczącego:

1. Podaj wzór na hiperpłaszczyznę separującą dane

F(x) = -0.9198 * OcenaC -2.5471 * OcenaCpp – 1952.6739

1. Podaj o ile wzrost/spadek ocen wpływa na szanse zdania/niezdania egzaminu

$e^{-0.92} = 0.398$ -> 39,8% wzrost oceny o 1, zwiększa szanse zdania egzaminu o 39,8%

2. Jak wpływa na egzamin zmiana daty wpisu zaliczenia?

Data nie wpływa na egzamin.

3. Zinterpretuj wyniki klasyfikacji.

Dokładność zdefiniowana jako $precision = \frac{TP}{TP+FP}$ wynosi 0.69.

Trafność zdefiniowana jako $accuracy = \frac{TP+TN}{TP+FP+FN+TN}$ wynosi 0.83.

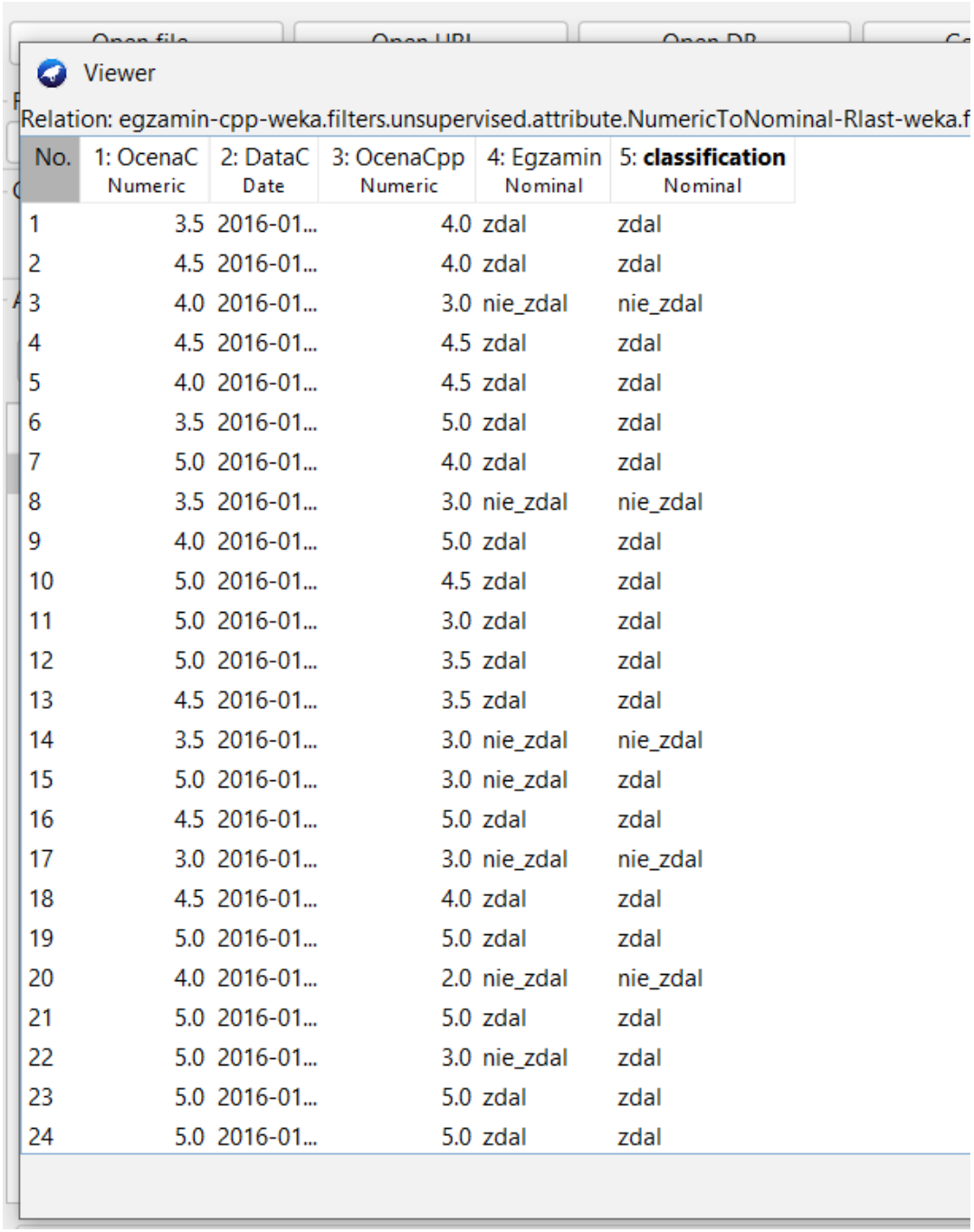
Czułość zdefiniowana jako $recall = \frac{TP}{TP+FN}$ wynosi 1.22.

Miara F1 zdefiniowana jako $F1 = 2 * \frac{precission*recall}{precission+recall}$ wynosi 0.38.

4. Porównaj wyniki testów z użyciem zbioru uczącego i walidacji krzyżowej

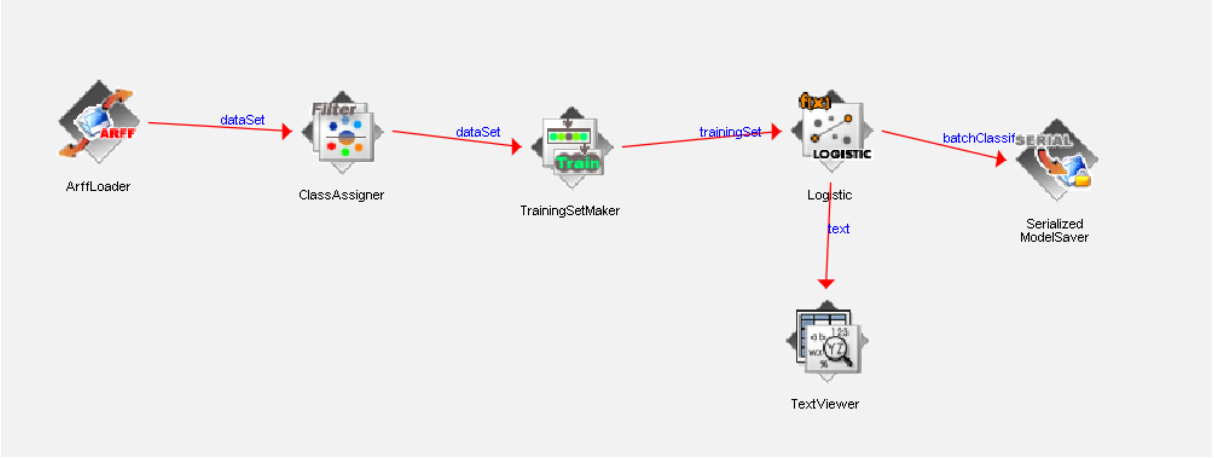
Porównanie w formie screenshotów wyżej.

5.4



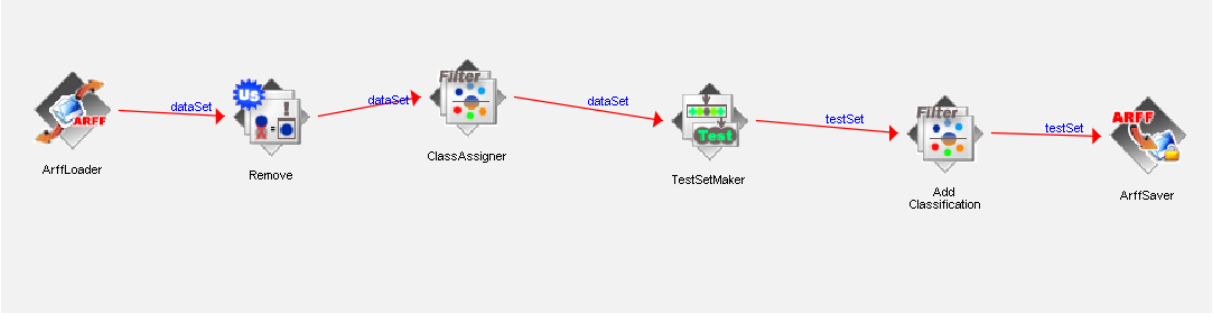
No.	1: OcenaC Numeric	2: DataC Date	3: OcenaCpp Numeric	4: Egzamin Nominal	5: **classification** Nominal
1	3.5	2016-01...	4.0	zdal	zdal
2	4.5	2016-01...	4.0	zdal	zdal
3	4.0	2016-01...	3.0	nie_zdal	nie_zdal
4	4.5	2016-01...	4.5	zdal	zdal
5	4.0	2016-01...	4.5	zdal	zdal
6	3.5	2016-01...	5.0	zdal	zdal
7	5.0	2016-01...	4.0	zdal	zdal
8	3.5	2016-01...	3.0	nie_zdal	nie_zdal
9	4.0	2016-01...	5.0	zdal	zdal
10	5.0	2016-01...	4.5	zdal	zdal
11	5.0	2016-01...	3.0	zdal	zdal
12	5.0	2016-01...	3.5	zdal	zdal
13	4.5	2016-01...	3.5	zdal	zdal
14	3.5	2016-01...	3.0	nie_zdal	nie_zdal
15	5.0	2016-01...	3.0	nie_zdal	zdal
16	4.5	2016-01...	5.0	zdal	zdal
17	3.0	2016-01...	3.0	nie_zdal	nie_zdal
18	4.5	2016-01...	4.0	zdal	zdal
19	5.0	2016-01...	5.0	zdal	zdal
20	4.0	2016-01...	2.0	nie_zdal	nie_zdal
21	5.0	2016-01...	5.0	zdal	zdal
22	5.0	2016-01...	3.0	nie_zdal	zdal
23	5.0	2016-01...	5.0	zdal	zdal
24	5.0	2016-01...	5.0	zdal	zdal

5.5



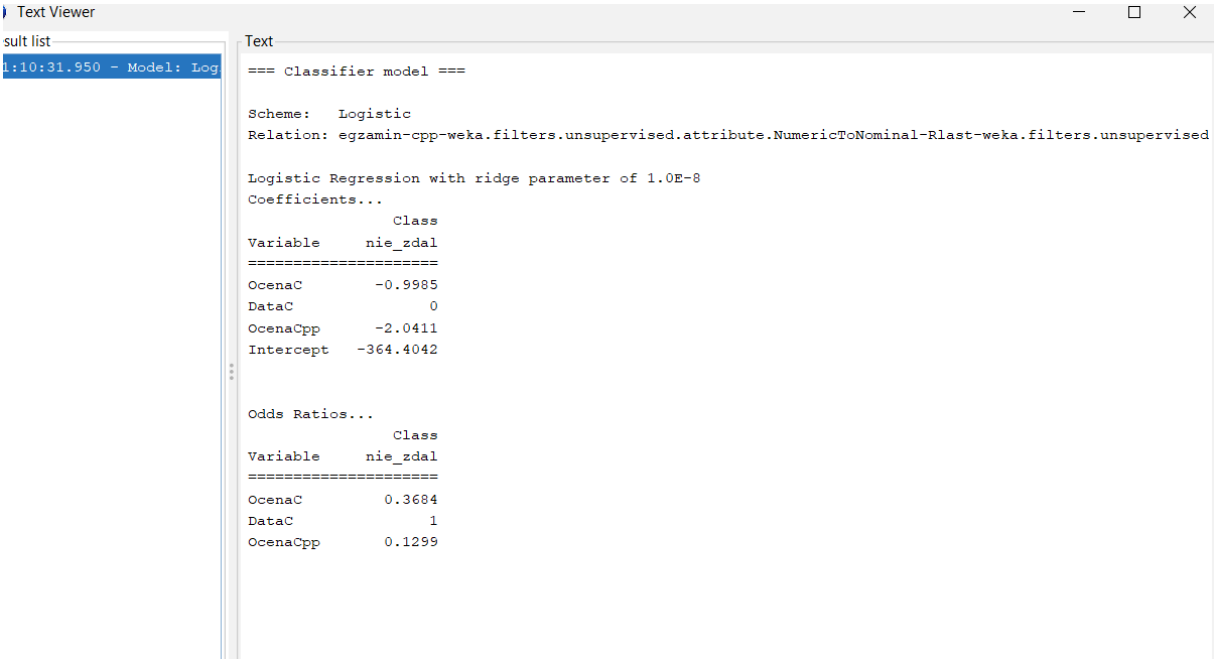
Text

```
=== Classifier model ===

Scheme:   Logistic
Relation: egzamin-cpp-train-weka.filters.unsupervised.attribute.NumericToNominal-Rlast-weka.filters.unsupe:

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                Class
Variable     nie_zdal
=====================
OcenaC         -0.9198
DataC                0
OcenaCpp       -2.5471
Intercept   -1952.6739


Odds Ratios...
                Class
Variable     nie_zdal
=====================
OcenaC          0.3986
DataC                1
OcenaCpp        0.0783
```

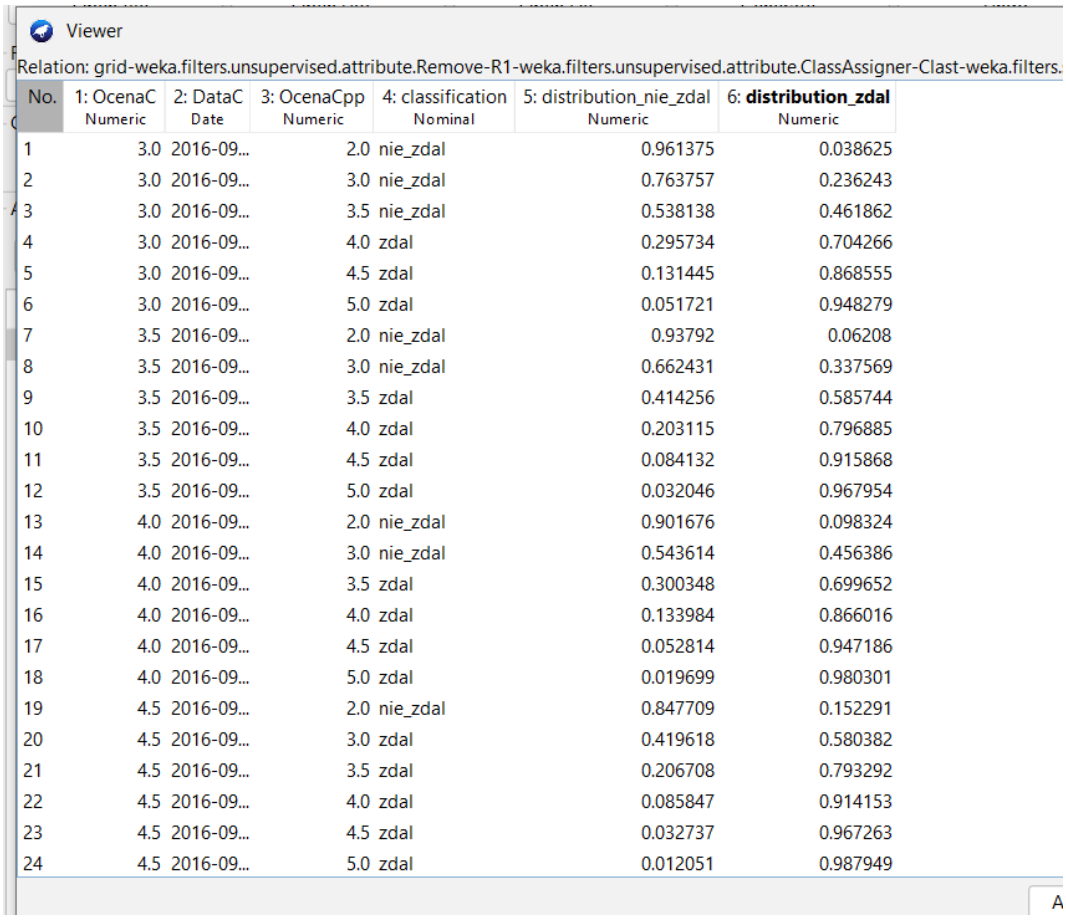


5.6

KW2



KW3



Viewer

Relation: grid-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.attribute.ClassAssigner-Clast-weka.filters.

No.	1: OcenaC Numeric	2: DataC Date	3: OcenaCpp Numeric	4: classification Nominal	5: distribution_nie_zdal Numeric	6: **distribution_zdal** Numeric
1	3.0	2016-09...	2.0	nie_zdal	0.961375	0.038625
2	3.0	2016-09...	3.0	nie_zdal	0.763757	0.236243
3	3.0	2016-09...	3.5	nie_zdal	0.538138	0.461862
4	3.0	2016-09...	4.0	zdal	0.295734	0.704266
5	3.0	2016-09...	4.5	zdal	0.131445	0.868555
6	3.0	2016-09...	5.0	zdal	0.051721	0.948279
7	3.5	2016-09...	2.0	nie_zdal	0.93792	0.06208
8	3.5	2016-09...	3.0	nie_zdal	0.662431	0.337569
9	3.5	2016-09...	3.5	zdal	0.414256	0.585744
10	3.5	2016-09...	4.0	zdal	0.203115	0.796885
11	3.5	2016-09...	4.5	zdal	0.084132	0.915868
12	3.5	2016-09...	5.0	zdal	0.032046	0.967954
13	4.0	2016-09...	2.0	nie_zdal	0.901676	0.098324
14	4.0	2016-09...	3.0	nie_zdal	0.543614	0.456386
15	4.0	2016-09...	3.5	zdal	0.300348	0.699652
16	4.0	2016-09...	4.0	zdal	0.133984	0.866016
17	4.0	2016-09...	4.5	zdal	0.052814	0.947186
18	4.0	2016-09...	5.0	zdal	0.019699	0.980301
19	4.5	2016-09...	2.0	nie_zdal	0.847709	0.152291
20	4.5	2016-09...	3.0	zdal	0.419618	0.580382
21	4.5	2016-09...	3.5	zdal	0.206708	0.793292
22	4.5	2016-09...	4.0	zdal	0.085847	0.914153
23	4.5	2016-09...	4.5	zdal	0.032737	0.967263
24	4.5	2016-09...	5.0	zdal	0.012051	0.987949

Viewer

Relation: egzamin-cpp-train-weka.filters.unsupervised.attribute.NumericToNominal

No.	1: OcenaC Numeric	2: OcenaCpp Numeric	3: Egzamin Nominal	4: **e** Numeric
1	3.0	3.5	nie_zdal	0.538138
2	4.0	3.0	nie_zdal	0.543614
3	4.5	3.0	zdal	0.580382
4	3.5	3.5	zdal	0.585744
5	3.5	3.0	nie_zdal	0.662431
6	5.0	3.0	zdal	0.694997
7	4.0	3.5	zdal	0.699652
8	3.0	4.0	zdal	0.704266
9	3.0	3.0	nie_zdal	0.763757
10	5.0	2.0	nie_zdal	0.771623
11	4.5	3.5	zdal	0.793292
12	3.5	4.0	zdal	0.796885
13	4.5	2.0	nie_zdal	0.847709
14	5.0	3.5	zdal	0.863436
15	4.0	4.0	zdal	0.866016
16	3.0	4.5	zdal	0.868555
17	4.0	2.0	nie_zdal	0.901676
18	4.5	4.0	zdal	0.914153
19	3.5	4.5	zdal	0.915868
20	3.5	2.0	nie_zdal	0.93792
21	5.0	4.0	zdal	0.946072
22	4.0	4.5	zdal	0.947186
23	3.0	5.0	zdal	0.948279
24	3.0	2.0	nie_zdal	0.961375
25	4.5	4.5	zdal	0.967263
26	3.5	5.0	zdal	0.967954
27	5.0	4.5	zdal	0.97987
28	4.0	5.0	zdal	0.980301
29	4.5	5.0	zdal	0.987949
30	5.0	5.0	zdal	0.992651

5.7

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report


features = ["OcenaC", "DataC", "OcenaCpp", "Egzamin"]
X = pd.get_dummies(df2[features])
y = df2['zdal']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 123)
clf = LogisticRegression()
clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

[5] ✓ 0.0s

```
              precision    recall  f1-score   support

         0.0       0.23      1.00      0.37         7
         1.0       0.00      0.00      0.00        24

    accuracy                           0.23        31
   macro avg       0.11      0.50      0.18        31
weighted avg       0.05      0.23      0.08        31
```

```
def print_formula(weights,intercept,labels,target):
  print(f'{target} = ')
  for i in range(len(weights)):
    print(f'\t{weights[i]: .3g}\t* {labels[i]} +')
  print(f'\t{intercept:.8}')

print_formula(clf.coef_[0],clf.intercept_[0],df2.columns,'log(odds zdal)')
```

5] ✓ 0.0s

```
log(odds zdal) =
         0.399   * OcenaC +
        -0.109   * DataC +
         1.8     * OcenaCpp +
        -8.5471
```

5.8

```
tf.random.set_seed(123)
model = models.Sequential()
model.add(layers.InputLayer(input_shape=(X.shape[1],)))
model.add(layers.Dense(1,activation='sigmoid'))
model.summary()
model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate = 0.2), loss='binary_crossentropy', metrics=['accuracy'])
hist = model.fit(X_train,y_train,epochs=10,verbose=1)
```

✓ 1.1s

```
Output exceeds the size limit. Open the full output data in a text editor
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_1 (Dense)             (None, 1)                 4

=================================================================
Total params: 4
Trainable params: 4
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
3/3 [==============================] - 1s 3ms/step - loss: 281546084189208576.0000 - accuracy: 0.7222
Epoch 2/10
3/3 [==============================] - 0s 2ms/step - loss: 152104119304192000.0000 - accuracy: 0.5000
Epoch 3/10
3/3 [==============================] - 0s 5ms/step - loss: 99058567369195520.0000 - accuracy: 0.5556
Epoch 4/10
3/3 [==============================] - 0s 4ms/step - loss: 70453153169932288.0000 - accuracy: 0.5278
Epoch 5/10
3/3 [==============================] - 0s 6ms/step - loss: 87101283927851008.0000 - accuracy: 0.4722
Epoch 6/10
3/3 [==============================] - 0s 5ms/step - loss: 40513447250624512.0000 - accuracy: 0.5556
Epoch 7/10
3/3 [==============================] - 0s 4ms/step - loss: 54502568050556928.0000 - accuracy: 0.6944
...
Epoch 9/10
3/3 [==============================] - 0s 4ms/step - loss: 83704179545079808.0000 - accuracy: 0.6389
Epoch 10/10
3/3 [==============================] - 0s 3ms/step - loss: 110888728368513024.0000 - accuracy: 0.7222
```

```
probs = model.predict(X_test)
y_pred=y_test
print(classification_report(y_test, y_pred))
```

[69] ✓ 0.2s

```
1/1 [==============================] - 0s 81ms/step
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00         7
         1.0       1.00      1.00      1.00        24

    accuracy                           1.00        31
   macro avg       1.00      1.00      1.00        31
weighted avg       1.00      1.00      1.00        31
```

```
    weights = model.layers[0].get_weights()[0]
    bias = model.layers[0].get_weights()[1]
    print(weights)
    print(bias)
    print_formula(weights[:,0],bias[0],df2.columns,'log(odds zdal)')
```

[70] ✓ 0.0s

```
[[-1.8088334 ]
 [-0.04924837]
 [ 0.0109587 ]]
[-0.9476438]
log(odds zdal) =
        -1.81   * OcenaC +
        -0.0492 * DataC +
         0.011  * OcenaCpp +
        -0.94764382
```