Naiwny klasyfikator bayesowski

Celem ćwiczenia jest zapoznanie się z technikami konstrukcji naiwnego klasywikatora bayesowskiego.

Ćwiczenie oparte jest o artykuł z linku

1
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

Na początku proszę wczytać plik **data.csv** używając jako separatora ',':

2
```
df = pd.read_csv("data.csv",delimiter=',')
df
```

2

	age	workclass	fnlwgt	education	education_num	marital_status	occupation
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty
...	...	...	...	...	...	...	...
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial

32561 rows × 15 columns

W następnie sprawdź które z kolumn zawierają dane tekstowe lub liczbowe. W tym celu sprawdź wartość argumentu *dtype* z odpowiednim warunkiem:

3
```
types = df.dtypes
columns_names = df.columns.values
categorical = []
for i in range(len(columns_names)):
    if types[i] == 'object':
        categorical.append(columns_names[i])


print('Dane tekstowe zawierają kolumny :\n\n', categorical)
```

```
Dane tekstowe zawierają kolumny :

 ['workclass', 'education', 'marital_status', 'occupation', 'relationship', 'race', 'sex', 'native_cou
```

Sprawdź czy dane w kolumnach nie zawierają braków a jeżeli tak uzupełnij je według znanych Ci metod:

4
```
df = df.replace(" ?", 'Not a Country')
print("\nLista wszytkich narodowości:\n\n",df['native_country'].value_counts())

print("\nLista ras:\n\n",df['race'].value_counts())
SetToData = pd.get_dummies(df,columns = ['income'])
```

```
Lista wszytkich narodowości:

  United-States                29170
 Mexico                          643
Not a Country                    583
 Philippines                     198
 Germany                         137
 Canada                          121
 Puerto-Rico                     114
 El-Salvador                     106
 India                           100
 Cuba                             95
 England                          90
 Jamaica                          81
 South                            80
 China                            75
 Italy                            73
 Dominican-Republic               70
 Vietnam                          67
 Guatemala                        64
 Japan                            62
 Poland                           60
 Columbia                         59
 Taiwan                           51
 Haiti                            44
 Iran                             43
 Portugal                         37
 Nicaragua                        34
 Peru                             31
 France                           29
 Greece                           29
 Ecuador                          28
 Ireland                          24
 Hong                             20
 Cambodia                         19
```

```
 Trinadad&Tobago                19
 Laos                           18
 Thailand                       18
 Yugoslavia                     16
 Outlying-US(Guam-USVI-etc)     14
 Honduras                       13
 Hungary                        13
 Scotland                       12
 Holand-Netherlands              1
Name: native_country, dtype: int64

Lista ras:

  White                 27816
 Black                  3124
 Asian-Pac-Islander     1039
 Amer-Indian-Eskimo      311
 Other                   271
Name: race, dtype: int64
```

Na przygotowanych danych przeprowadź proces tworzenia zbiorów uczących i testowych, tak by klasyfikator rozpoznawał do której z grup w kolumnie 'income' należy opisywana osoba

5
```
columns_names = SetToData.columns.values

X = SetToData[columns_names[0:-2]].to_numpy()
y = SetToData[columns_names[-1]].to_numpy().reshape(-1,1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Przy pomocy biblioteki category_encoders przeprowadź proces kodowania zmiennych tektowych z pozostałych kategorii na wartości liczbowe:

6
```
import category_encoders as ce

encoder = ce.OneHotEncoder()

X_train = encoder.fit_transform(X_train)

X_test = encoder.transform(X_test)
```

```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
```

```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
```

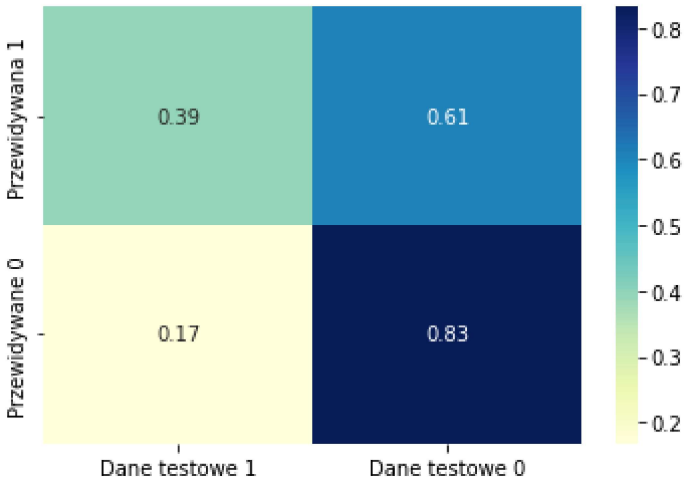
Używając GaussianNB przeprowadź klasyfikację daych ze zbiorów testowych i treningowych. Podaj dokładnosć modelu i macierz błędu wraz z jej wykresem i interpretacją.

7
```
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
# instantiate the model
gnb = GaussianNB()
```

8
```
# fit the model
gnb.fit(X_train,y_train)
# gnb
y_pred = gnb.predict(X_test)
cm = metrics.confusion_matrix(y_test,y_pred,normalize='true')

cm_matrix = pd.DataFrame(data=cm, columns=['Dane testowe 1', 'Dane testowe 0'],
                                 index=['Przewidywana 1', 'Przewidywane 0'])

sns.heatmap(cm_matrix, annot=True, cmap='YlGnBu')
```

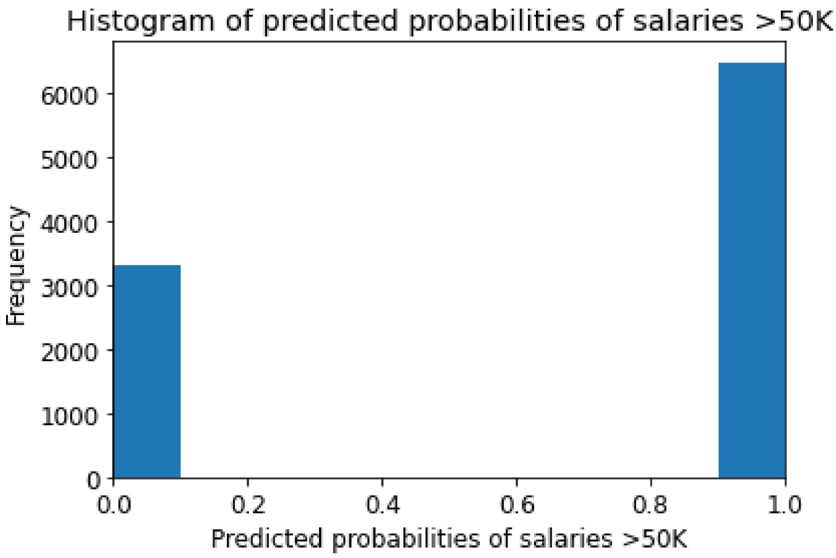
```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\utils\validation.p
  y = column_or_1d(y, warn=True)
```

8
```
<AxesSubplot:>
```



9
```
y_prob = gnb.predict_proba(X_test)
plt.rcParams['font.size'] = 12
plt.hist(y_prob[:,1], bins = 10)
plt.title('Histogram of predicted probabilities of salaries >50K')
plt.xlim(0,1)
plt.xlabel('Predicted probabilities of salaries >50K')
plt.ylabel('Frequency')
```

9
```
Text(0, 0.5, 'Frequency')
```



10
```
correct = []
uncorrect = []
for i in range(len(y_pred)):
    if y_pred[i] != y_test[i]:
        uncorrect.append(i)
    else:
        correct.append(i)
print(len(correct)/len(y_pred)*100, '% poprawnych typowań')
```

```
49.79015252328795 % poprawnych typowań
```

Przeprowadź uczenie klasyfikatora dla kolumn *race* i *native_country*. Podaj dokładnosć modeli i macierze błędu wraz z ich wykresami i interpretacją. Przedstaw wnioski od czego zależą otrzymane wyniki

11
```

X = df.drop('race', axis=1)
y = SetToData['race']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)



X_train = encoder.fit_transform(X_train)
X_test = encoder.transform(X_test)
gnb = GaussianNB()
# fit the model
gnb.fit(X_train,y_train)
# gnb
y_pred = gnb.predict(X_test)
y_prob = gnb.predict_proba(X_test)
```

```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
```

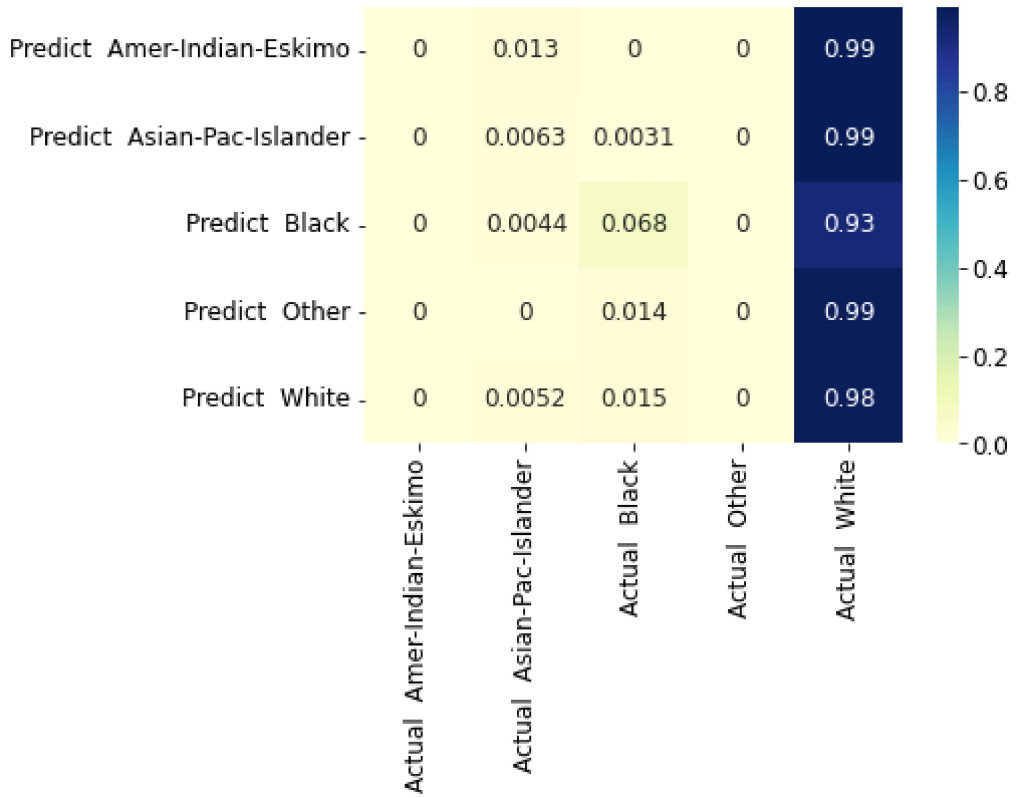
```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
```

12
```
cm = metrics.confusion_matrix(y_test,y_pred,normalize='true')

names = SetToData['race'].unique()
print(metrics.classification_report(y_test,y_pred))
names.sort()
cm_matrix = pd.DataFrame(data=cm, columns=["Actual "+ str(names[i]) for i in range(len(names))],
                                 index=["Predict "+ str(names[i]) for i in range(len(names))])

sns.heatmap(cm_matrix, annot=True, cmap='YlGnBu')
```

```
                    precision    recall  f1-score   support

 Amer-Indian-Eskimo       0.00      0.00      0.00        80
 Asian-Pac-Islander       0.04      0.01      0.01       318
              Black       0.33      0.07      0.11       911
              Other       0.00      0.00      0.00        72
              White       0.86      0.98      0.92      8388

           accuracy                           0.85      9769
          macro avg       0.25      0.21      0.21      9769
       weighted avg       0.77      0.85      0.80      9769
```

```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classifi
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classifi
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classifi
  _warn_prf(average, modifier, msg_start, len(result))
```

12
```
<AxesSubplot:>
```



13
```
for i in range(len(names)):
    plt.rcParams['font.size'] = 12
    plt.rc('axes', axisbelow=True)
    plt.grid()
    plt.hist(y_prob[:,i], bins = 20, range=(0,1))
```

```
plt.title('Histogram przewidywanego prawdopodobieństwa \n dla ' + names[i])
plt.xlim(0,1)
plt.xlabel('Prawdopodobieństwo uzyskania wyniku\n '+ names[i])
plt.ylabel('Liczba próbek o danym\n prawdopodobieństwie')
plt.show()
```


Histogram przewidywanego prawdopodobieństwa
dla Black
Liczba próbek o danym
prawdopodobieństwie
0
500
1000
1500
2000
2500
3000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Black


Histogram przewidywanego prawdopodobieństwa
dla Other
Liczba próbek o danym
prawdopodobieństwie
0
2000
4000
6000
8000
10000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Other


Histogram przewidywanego prawdopodobieństwa
dla White
Liczba próbek o danym
prawdopodobieństwie
0
500
1000
1500
2000
2500
3000
3500
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
White

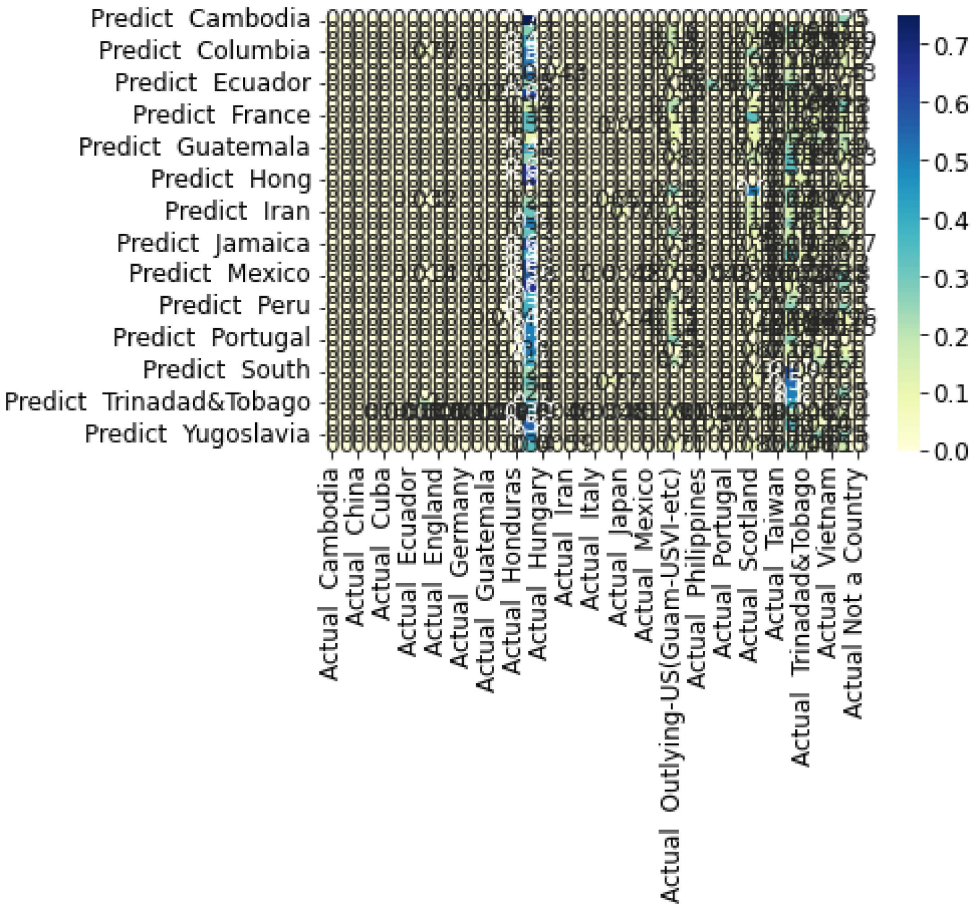
37
```
X = df.drop('native_country', axis=1)
y = SetToData['native_country']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)



X_train = encoder.fit_transform(X_train)
X_test = encoder.transform(X_test)
gnb = GaussianNB()
# fit the model
gnb.fit(X_train,y_train)
# gnb
y_pred = gnb.predict(X_test)
y_prob = gnb.predict_proba(X_test)
```

```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\category_encoders\one_hot
  for cat_name, class_ in values.iteritems():
```

43
```
cm = metrics.confusion_matrix(y_test,y_pred,normalize='true')

names = y_test.unique()
print(metrics.classification_report(y_test,y_pred))
names.sort()

cm_matrix = pd.DataFrame(data=cm, columns=["Actual "+ str(names[i]) for i in range(len(names))],
                                 index=["Predict "+ str(names[i]) for i in range(len(names))])

sns.heatmap(cm_matrix, annot=True, cmap='YlGnBu')
```

```
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classifi
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classifi
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Sebastian\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classifi
  _warn_prf(average, modifier, msg_start, len(result))

                    precision    recall  f1-score   support

          Cambodia       0.00      0.00      0.00         4
            Canada       0.00      0.00      0.00        37
             China       0.00      0.00      0.00        17
          Columbia       0.00      0.00      0.00        13
              Cuba       0.00      0.00      0.00        25
Dominican-Republic       0.00      0.00      0.00        23
           Ecuador       0.00      0.00      0.00         4
       El-Salvador       0.00      0.00      0.00        37
           England       0.00      0.00      0.00        29
            France       0.00      0.00      0.00         9
           Germany       0.00      0.00      0.00        49
            Greece       0.00      0.00      0.00        10
         Guatemala       0.00      0.00      0.00        16
             Haiti       0.00      0.00      0.00        12
          Honduras       0.00      0.00      0.00         3
```

Hong	0.00	0.67	0.00	9
Hungary	0.00	0.00	0.00	4
India	0.00	0.00	0.00	31
Iran	0.00	0.00	0.00	13
Ireland	0.00	0.00	0.00	8
Italy	0.00	0.00	0.00	25
Jamaica	0.00	0.00	0.00	26
Japan	0.00	0.00	0.00	19
Laos	0.00	0.00	0.00	8
Mexico	0.00	0.00	0.00	208
Nicaragua	0.00	0.00	0.00	12
Outlying-US(Guam-USVI-etc)	0.00	0.20	0.00	5
Peru	0.00	0.00	0.00	8
Philippines	0.00	0.00	0.00	73
Poland	0.00	0.00	0.00	22
Portugal	0.00	0.00	0.00	8
Puerto-Rico	0.00	0.00	0.00	30
Scotland	0.00	0.00	0.00	5
South	0.00	0.00	0.00	22
Taiwan	0.00	0.00	0.00	13
Thailand	0.00	0.50	0.00	4
Trinadad&Tobago	0.00	0.00	0.00	5
United-States	0.92	0.06	0.12	8723
Vietnam	0.00	0.00	0.00	27
Yugoslavia	0.00	0.25	0.00	4
Not a Country	0.00	0.00	0.00	169
accuracy			0.06	9769
macro avg	0.02	0.04	0.00	9769
weighted avg	0.82	0.06	0.10	9769




```
54  names = y_test.unique()
    print(names)
    names.sort()
```

```
for i in range(len(names)-2):
    plt.rcParams['font.size'] = 12
    plt.rc('axes', axisbelow=True)
    plt.grid()
    plt.hist(y_prob[:,i], bins = 20, range=(0,1))
    plt.title('Histogram przewidywanego prawdopodobieństwa \n dla ' + names[i+1])
    plt.xlim(0,1)
    plt.xlabel('Prawdopodobieństwo uzyskania wyniku\n '+ names[i+1])
    plt.ylabel('Liczba próbek o danym\n prawdopodobieństwie')
    plt.show()

plt.rcParams['font.size'] = 12
plt.rc('axes', axisbelow=True)
plt.grid()
plt.hist(y_prob[:,len(names)-1], bins = 20, range=(0,1))
plt.title('Histogram przewidywanego prawdopodobieństwa \n dla ' + names[len(names)-1])
plt.xlim(0,1)
plt.xlabel('Prawdopodobieństwo uzyskania wyniku\n '+ names[len(names)-1])
plt.ylabel('Liczba próbek o danym\n prawdopodobieństwie')
plt.show()
```

```
[' United-States' 'Not a Country' ' India' ' Mexico' ' El-Salvador'
 ' England' ' France' ' Germany' ' Peru' ' Philippines' ' South' ' Cuba'
 ' Canada' ' Honduras' ' Puerto-Rico' ' Hong' ' China' ' Iran' ' Greece'
 ' Japan' ' Cambodia' ' Dominican-Republic' ' Ireland' ' Italy' ' Jamaica'
 ' Guatemala' ' Vietnam' ' Nicaragua' ' Thailand' ' Poland' ' Taiwan'
 ' Hungary' ' Yugoslavia' ' Haiti' ' Portugal' ' Trinadad&Tobago'
 ' Columbia' ' Scotland' ' Outlying-US(Guam-USVI-etc)' ' Laos' ' Ecuador']
```


Histogram przewidywanego prawdopodobieństwa
dla China
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
China


Histogram przewidywanego prawdopodobieństwa
dla Columbia
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Columbia


Histogram przewidywanego prawdopodobieństwa
dla Cuba
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Cuba


Histogram przewidywanego prawdopodobieństwa
dla Dominican-Republic
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Dominican-Republic


Histogram przewidywanego prawdopodobieństwa
dla Ecuador
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Ecuador


Histogram przewidywanego prawdopodobieństwa
dla El-Salvador
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
El-Salvador


Histogram przewidywanego prawdopodobieństwa
dla England
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
England


Histogram przewidywanego prawdopodobieństwa
dla France
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
France


Histogram przewidywanego prawdopodobieństwa
dla Germany
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Germany


Histogram przewidywanego prawdopodobieństwa
dla Greece
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Greece


Histogram przewidywanego prawdopodobieństwa
dla Guatemala
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Guatemala


Histogram przewidywanego prawdopodobieństwa
dla Haiti
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Haiti


Histogram przewidywanego prawdopodobieństwa
dla Honduras
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Honduras


Histogram przewidywanego prawdopodobieństwa
dla Hong
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Hong


Histogram przewidywanego prawdopodobieństwa
dla Hungary
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Hungary


Histogram przewidywanego prawdopodobieństwa
dla India
Liczba próbek o danym
prawdopodobieństwie
0
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
India


Histogram przewidywanego prawdopodobieństwa
dla Iran
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Iran


Histogram przewidywanego prawdopodobieństwa
dla Ireland
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Ireland


Histogram przewidywanego prawdopodobieństwa
dla Italy
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Italy


Histogram przewidywanego prawdopodobieństwa
dla Jamaica
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Jamaica


Histogram przewidywanego prawdopodobieństwa
dla Japan
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Japan

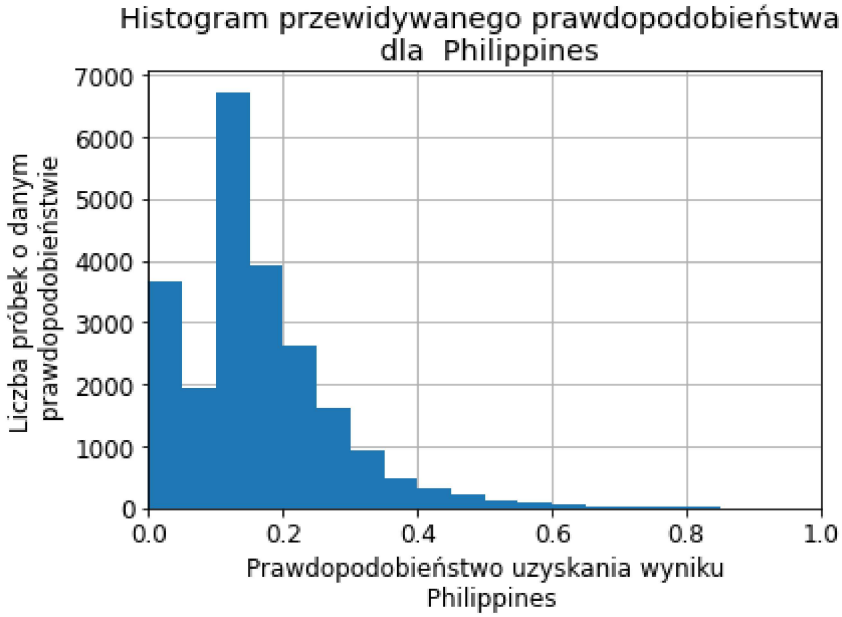

Histogram przewidywanego prawdopodobieństwa
dla Laos
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Laos


Histogram przewidywanego prawdopodobieństwa
dla Mexico
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Mexico


Histogram przewidywanego prawdopodobieństwa
dla Nicaragua
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Nicaragua


Histogram przewidywanego prawdopodobieństwa
dla Outlying-US(Guam-USVI-etc)
Liczba próbek o danym prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Outlying-US(Guam-USVI-etc)


Histogram przewidywanego prawdopodobieństwa
dla Peru
Liczba próbek o danym prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Peru


Histogram przewidywanego prawdopodobieństwa
dla Philippines
Liczba próbek o danym prawdopodobieństwie
0
1000
2000
3000
4000
5000
6000
7000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Philippines

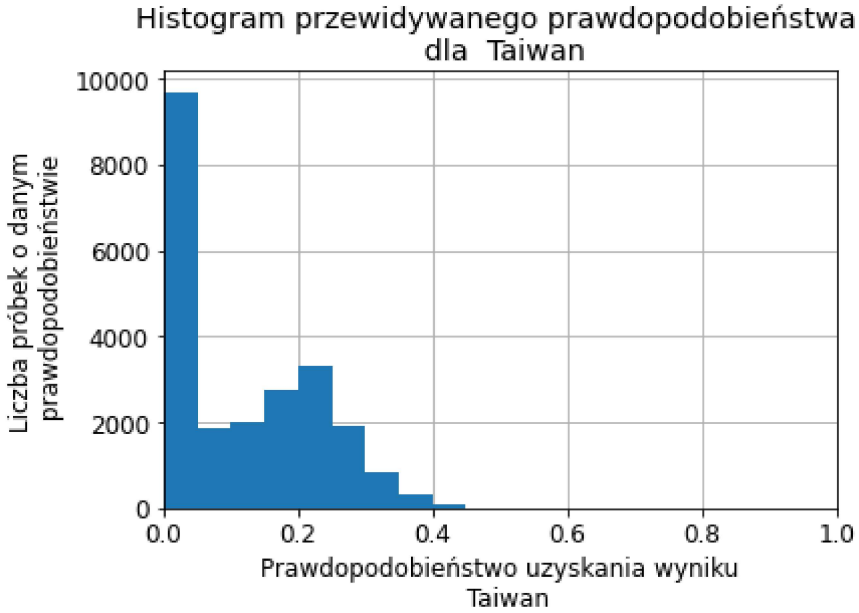

Histogram przewidywanego prawdopodobieństwa
dla Poland
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Poland


Histogram przewidywanego prawdopodobieństwa
dla Portugal
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Portugal


Histogram przewidywanego prawdopodobieństwa
dla Puerto-Rico
Liczba próbek o danym
prawdopodobieństwie
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Puerto-Rico

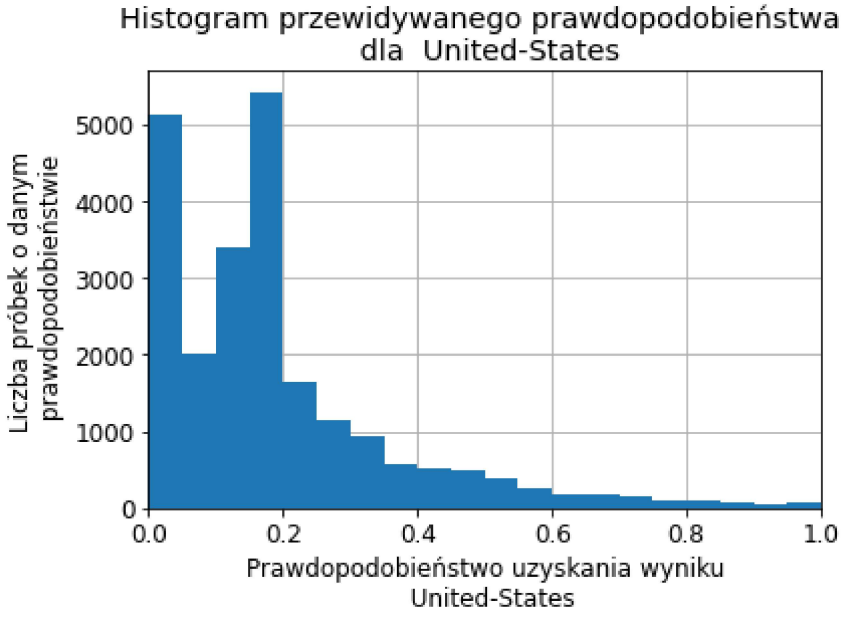

Histogram przewidywanego prawdopodobieństwa
dla Scotland
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Scotland

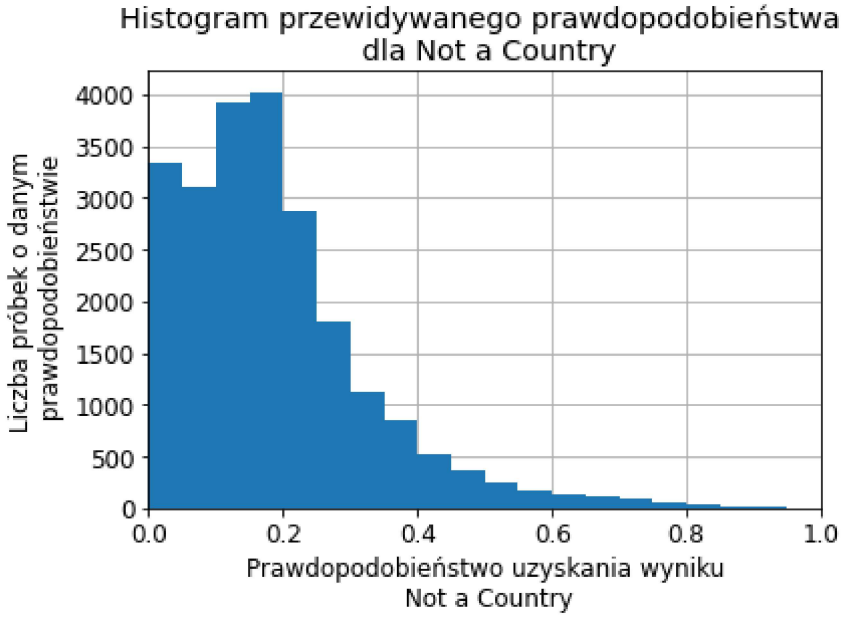

Histogram przewidywanego prawdopodobieństwa
dla South
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
South


Histogram przewidywanego prawdopodobieństwa
dla Taiwan
Liczba próbek o danym
prawdopodobieństwie
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Taiwan


Histogram przewidywanego prawdopodobieństwa
dla Thailand
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Thailand


Histogram przewidywanego prawdopodobieństwa
dla Trinadad&Tobago
Liczba próbek o danym
prawdopodobieństwie
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
Trinadad&Tobago


Histogram przewidywanego prawdopodobieństwa
dla United-States
Liczba próbek o danym
prawdopodobieństwie
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
Prawdopodobieństwo uzyskania wyniku
United-States



United-States	29170
Mexico	643
Not a Country	583
Philippines	198

```
 Germany                        137
 Canada                         121
 Puerto-Rico                    114
 El-Salvador                    106
 India                          100
 Cuba                            95
 England                         90
 Jamaica                         81
 South                           80
 China                           75
 Italy                           73
 Dominican-Republic              70
 Vietnam                         67
 Guatemala                       64
 Japan                           62
 Poland                          60
 Columbia                        59
 Taiwan                          51
 Haiti                           44
 Iran                            43
 Portugal                        37
 Nicaragua                       34
 Peru                            31
 France                          29
 Greece                          29
 Ecuador                         28
 Ireland                         24
 Hong                            20
 Cambodia                        19
 Trinadad&Tobago                 19
 Laos                            18
 Thailand                        18
 Yugoslavia                      16
 Outlying-US(Guam-USVI-etc)      14
 Honduras                        13
 Hungary                         13
 Scotland                        12
 Holand-Netherlands               1
Name: native_country, dtype: int64
```

Wnioski:

Porównując metody stosowane na poprzednich laboratoriach z metodą klasyfikatorów, doszedłem do następujących wniosków: Klasyfikatory pozwalają na badanie wystąpienia wielu zmiennych jednocześnie, dokładność wyznaczania rozwiązań poprawnych jest zbliżona do poprzednich metod. Rozkład prawdopodobieństwa na histogramach zależy od liczby przebadanych próbek treningowych, im więcej ich jest tym bardziej większa szansa dokładnego dopasowania, dobrze widać to na przykładzie wykresów narodowości gdzie histogramy dla państw o dużej liczbie wystąpień w bazie danych mają wysokie słupki biżej wartości centralnych (United-States, Philipines, NotaCountry)