# Sprawozdanie

Indukcyjne Metody Analizy Danych

Ćwiczenie 1. Klasyfikator oparty na twierdzeniu Bayesa przy naiwnym założeniu o wzajemnej niezależności atrybutów

Autor: Paweł Mielniczuk

# Spis treści:



## 1. Opis klasyfikatora

W implementacji i analizie został użyty naiwny klasyfikator Bayes'a. W odróżnieniu od rozwiązania nienaiwnego, bazuje on na założeniu że wszystkie zdarzenia, w przypadku użytych danych są to cechy (features), są od siebie niezależne i nie mają na siebie wpływu i nie występują pomiędzy nimi żadne korelacje. W zadaniu rozwiązywany będzie problem klasyfikacji. W przypadku sieci bayesowskich należałoby wszystkie takie relacje zamodelować.

## 2. Opis zbiorów

Podczas analizy i implementacji użyte zostały cztery zbiory danych. Zbiory podzielone są na dwie części. Pierwszą z nich są cechy, dokładnie wektor, cech oraz etykiety mówiące o przynależności wektora cech do konkretnej klasy.

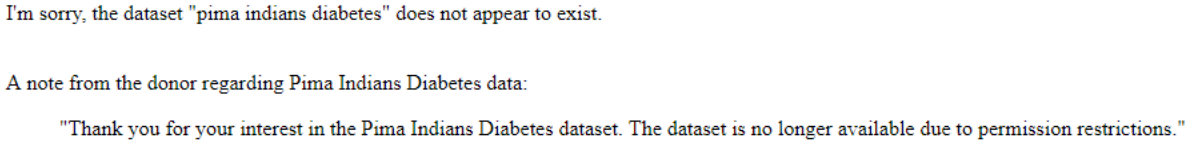
Wszystkie zbiory dostępne są do pobrania ze strony
*https://archive.ics.uci.edu/ml/datasets.html*

Zbiory danych zostały ściągnięte i załadowane przy użyciu biblioteki *pandas* lub bezpośrednio załadowane za pomocą biblioteki *scikit-learn*.

Zbiory danych:

- Iris data set
- Wine data set
- Glass identification data set
- Pima diabetes data set

Ciekawostką jest, że w trakcie badania klasyfikatora i tworzenia sprawozdania ostatni ze zbiorów *Pima diabetes* został usunięty ze strony UCI ze przez ograniczenie uprawnień do udostępniania danego zbioru.




*Rysunek 1 Wiadomość ze strony https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes mówiąca o braku dalszego dostępu do danego zbioru.*

Poniżej zaprezentowano opis zbiorów. Opis ten pomoże w zrozumieniu danych, które będą analizowane. Dobre zrozumienie danych z którymi się pracuje jest niezbędną częścią do poprawnego przeprowadzenia badań.

## Zbiór Iris

Jest to prawdopodobnie jeden z najbardziej znanych i podstawowych zbiorów danych przy problemach klasyfikacji i rozpoznawania wzorców.

Zbiór składa się ze 150 instancji, podzielonych na 3 równe zbiory po 50 klas każda.

Definicje atrybutów:

- Sepal – zielony płatek u dołu kielicha służący do ochrony kwiatu w trakcie kwitnięcia,
- Petal – płatek kwiatu, służący do przyciągania uwagi ptaków i insektów

Cechy zbioru zawierają cztery informacje:

1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm

Ostatnią, piątą kolumną jest klasa mówiąca o typie irysa. Możliwe są trzy klasy:

1. Iris Setosa
2. Iris Versicolour
3. Iris Virginica

## Zbiór Wine

Zbiór ten został skonstruowany w wyniku analizy składu chemicznego win stworzonych w tym samym rejonie Włoch lecz przy użyciu trzech różnych odmian uprawnych.

Zbiór składa się ze 178 instancji.

Definicje atrybutów oraz cechy zbioru:

1. Alcohol – alkohol
2. Malic acid – kwas jabłkowy
3. Ash – popiół
4. Alkalinity of ash – alkaliczność popiołu

5. Magnesium – magnez
6. Total phenols – całkowita zawartość fenoli
7. Flavonoids – flawonoidy
8. Nonflavanoid phenols – fenole nieflawonowe
9. Proanthocyanidins – proantocyjanidyny
10. Color intensity, intensywność koloru
11. Hue – odcień
12. OD280/OD315 of diluted wines - OD280 / OD315 rozcieńczonych win
13. Proline – Proline

Pierwszy atrybut w pliku zawierającym dane jest identyfikatorem klasy od 1 do 3.

Rozłożenie instancji klas jest następujące:

- Klasa 1 – 59 instancji,
- Klasa 2 – 71 instancji,
- Klasa 3 – 48 instancji.

## Zbiór Glass identification

Zbiór powstał poprzez analizę składu chemicznego badanego szkła aby określić typ powstałego szkła oraz jego przeznaczenie.

Zbiór składa się z 214 instancji podzielonych na 6 klas.

Rozłożenie instancji klas jest następujące:

- Klasa 1 – 70 instancji,
- Klasa 2 – 76 instancji,
- Klasa 3 – 17 instancji,
- Klasa 4 - 13,
- Klasa 5 - 9,
- Klasa 6 - 29.

Definicje atrybutów oraz cechy zbioru:

1. Id – numer porządkowy
2. Refractive index – współczynnik załamania światła
3. Sodium – sód
4. Magnesium – magnez
5. Aluminium – glin
6. Silicon – krzem
7. Potassium – potas
8. Calcium – wapń
9. Barium – bar
10. Iron – żelazo

### Zbiór Pima diabetes

Celem zbioru jest umożliwienie zdiagnozowania czy dany pacjent ma cukrzycę, bazując na diagnostykach zamieszczonych w cechach zbioru. Wszyscy pacjenci przebadani byli kobietami mającymi przynajmniej 21 lat oraz byli pochodzenia indiańskiego plemienia Pima.

Zbiór składa się z 768 instancji posiadających dwie możliwe klasy 1 – oznaczające że zbadana osoba jest chora na cukrzycę, 0 – oznaczające że dana osoba nie jest chora na cukrzycę.

Definicje atrybutów oraz cechy zbioru:

1. Pregnancies – liczba ciąży
2. Glucose – poziom glukozy
3. Blood ressure – ciśnienie krwi
4. Skin thickness – grubość skóry
5. Insulin – poziom insuliny
6. BMI – body mass index
7. Diabetes pedigree function – funkcja pedigree
8. Age – wiek

## 3. Implementacja

Do implementacji użyty został język programowania Python oraz następujące biblioteki: Numpy, Pandas, Matplotlib, Seaborn, scikit-learn.

W implementacji zostały użyte dwa modele naiwnego Bayes’a z biblioteki scikit-learn.

1. **GaussianNB** – zakłada, że dane są w rozkładzie normalnym. Używany będzie gdy dane nie będą dyskretyzowane
2. **MultinomialNB** – model który zostanie użyty po dyskretyzacji atrybutów. Bazuje na dystrybucji multinomial, która jest pochodną dystrybucji binomialnej.

## Wczytywanie danych

```
dataset = pd.read_csv('./pima-indians-diabetes.csv')
```

```
print(dataset.describe())
```

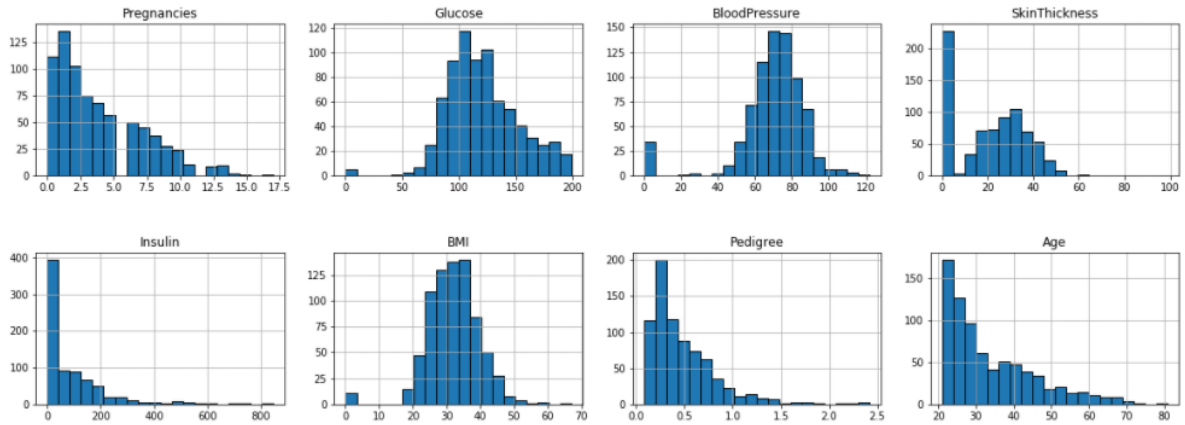
```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI    Pedigree         Age     Outcome
count  768.000000  768.000000  768.000000  768.000000
mean    31.992578    0.471876   33.240885    0.348958
std      7.884160    0.331329   11.760232    0.476951
min      0.000000    0.078000   21.000000    0.000000
25%     27.300000    0.243750   24.000000    0.000000
50%     32.000000    0.372500   29.000000    0.000000
75%     36.600000    0.626250   41.000000    1.000000
max     67.100000    2.420000   81.000000    1.000000
```

*Rysunek 2 Załadowanie i przedstawienie danych przy użyciu biblioteki pandas*

Jak wcześniej wspomniałem zawsze przed zaczęciem pracy z tworzeniem modelu należy zrozumieć dane, z którymi się pracuje. Do tego można zastosować kilka technik.

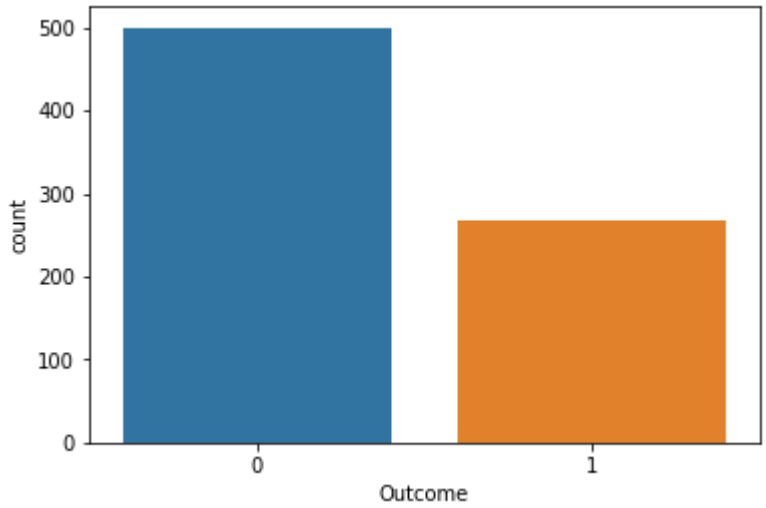
*Pandas* dostarcza wiele metod, między innymi takie umożliwiające uzyskanie opisu danych i kilka statystyk.

Warto także często zwizualizować atrybuty aby np. poznać dokładniejszy rozkład danych.




*Rysunek 3 Histogramy atrybutów danych zbioru Pima diabetes*

A także rozkład klas.




## Tworzenie modelu naiwnego Bayes’a:

Scikit-learn umożliwia bardzo prostu sposób tworzenia różnego rodzaju klasyfikatorów. Jednym z nich jest Gaussowski model naiwnego Bayes’a.

**Model = Gaussian Naive Bayes classifier**

```
In [133]: model = GaussianNB()
```

Model ten zakłada, że atrybuty będą zawsze w rozkładzie normalnym i działa wtedy najlepiej. Jak widać na powyższych wykresach nie jest to jednak zawsze prawdą.

## Cross-validation:

Często zdarza się, że mamy do czynienia z małymi zbiorami danych. Gdy taki zbiór podzielimy na zbiór treningowy i testowy, a czasem jeszcze walidacyjny nasze dane stają się zbyt małe aby poprawnie wyuczyć model. W takich przypadkach należy podjąć pewne kroki aby zapewnić, że wielkość zbioru będzie odpowiednio duża do wytrenowania modelu.

Jedną z nich jest kroswalidacja. Polega ona na podziale całego zbioru na określoną ilość podzbiorów, a następnie przeprowadzeniu na nich predykcji jak celny będzie nasz model.

Jednymi z podstawowych rodzajów walidacji modelu są:

- **Walidacja prosta** – mająca na celu podzielenie zbioru na zbiór testowy i treningowy. Najczęściej zbiór testowy nie stanowi więcej niż 20-30% wielkości całego zbioru.
- **K-Fold validation** – zbiór dzielony jest na K części. Następnie kolejno każdy z podzbiorów brany jest jako zbiór testowy, a pozostałe jako uczący. Analiza

jest wykonywana tyle razy na ile części został podzielony zbiór. Po czym wszystkie wyniki się sumuje i uśrednia.

- **Stratyfikowana K-Fold validation** – zasada działania jest taka sama jak w przypadku zwykłego K-Fold validation z dodatkowym zachowaniem oryginalnych proporcji między klasami (labels) w podzielonych zbiorach.

Poniżej przedstawione zostaną wyniki predykcji modelu używając 3 powyższych sposobów. Przebadany zostanie wpływ wielkości podzbiorów na otrzymywany wynik modelu. W przypadku Walidacji prostej przedział zostanie podzielony na następujące części: 90/10, 80/20 (zwane zasadą Pareta) i 50/50 gdzie pierwsza liczba to zbiór treningowy a druga to zbiór testowy.

Do wszystkich poniższych testów użyto funkcji *train_test_split* z biblioteki scikit-learn z opcjonalnym parametrem shuffle=True, który odpowiada za przetasowanie danych przed ich podziałem. Wszystkie próby zostały wykonane 10 krotnie, a wyniki uśredniono.

Oprócz accuracy wyliczono także F1 score. Jest to średnia ważona precyzji oraz miary recall.

```
F1 = 2 * (precision * recall) / (precision + recall)
```

# 4. Analiza klasyfikatorów bez dyskretyzacji

## Analiza Wine dataset

### Walidacja prosta

**Split 90/10**

In [66]:
```
sum = 0
sumf1 = 0
for i in range(10):
    train_X, test_X, train_Y, test_Y = train_test_split(X, y, test_size=0.5)
    model = GaussianNB().fit(train_X, train_Y)
    y_pred = model.predict(test_X)
    errors = (test_X.shape[0],(test_Y != y_pred).sum())
    accuracy = 100-errors[1]/errors[0]*100
    sum += accuracy
    sumf1 += f1_score(test_Y, y_pred, average='macro')
print(sum/10)
print(sumf1/10)
```

```
93.59550561797752
0.937441864641132
```

*Rysunek 4 Podział danych na część testową i treningową oraz trenowanie modelu naiwnego Bayes'a*

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Podział zbiorów treningowy/testowy** | 90/10 | 70/30 | 50/50 |
| **Średnia predykcja** | 0.88 | 0.93 | 0.92 |
| **F1 score** | 0.91 | 0.93 | 0.90 |

*Tabela 1 Wyniki predykcji modelu Wine przy różnych proporcjach podziału zbioru treningowego i testowego*

Podział całego zbioru, który jest mały, na zbiory gdzie zbiór treningowy jest mały nie jest zawsze dobrym pomysłem. Przy małym zbiorze treningowym nasze estymacje parametrów będą miały większą wariancję, natomiast przy zbyt małym zbiorze testowym nasze statystyki pomiarowe będą miały większą wariancję. Powinno się dążyć do tego aby obie te wariancje nie były zbyt wysokie.

### Walidacja K-Fold

Biblioteka scikit-learn umożliwia tworzenie takiej walidacji w prosty sposób. Należy podać liczbę podzbiorów, a następnie zaaplikować otrzymane indeksy do predykcji modelu. Przebadano podział na 10, 5 i 2 podzbiory.

**KFolds**

In [247]:
```
folds = 10
kf = KFold(n_splits=folds, shuffle=True)
print(kf)
```

```
KFold(n_splits=10, random_state=None, shuffle=True)
```

In [248]:
```
sum = 0
for train_index, test_index in kf.split(x):
    x_train, x_test = x[train_index], x[test_index]
    y_train, y_test = y[train_index], y[test_index]
    model = GaussianNB().fit(x_train, y_train)
    y_pred = model.predict(x_test)
    errors = (x_test.shape[0],(y_test != y_pred).sum())
    accuracy = 100-errors[1]/errors[0]*100
    sum += accuracy
average = sum/folds
print('Accuracy:', average)
```

```
Accuracy: 97.77777777777779
```

*Rysunek 5 Predykcja modelu za pomocą walidacji KFold*

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.97 | 0.96 | 0.94 |
| **F1 score** | 0.95 | 0.95 | 0.93 |

*Tabela 2 Wyniki predykcji modelu Wine przy różnych proporcjach podziału zbioru przy walidacji K-Fold*

Wyniki nieco różnią się od zwykłego podziału. Jako, że nasze zbioru są małe nie możliwe jest zastosowanie zwykłego podziału np. 80/20. Stracilibyśmy w ten sposób zbyt dużo cennych danych. Jak widać walidacja K-Fold zwróciła wyższe wyniki w prawie każdym podziale. Z tabeli 2 można także wyczytać, że najlepiej działa walidacja gdzie fold ustawiony jest na 10. Jest to także często zalecana wielkość podzbiorów.

### Stratyfikowana walidacja K-Fold

Tak jak w przypadku zwykłego K-Fold, zostały użyte te same podziały. Zmianą jest natomiast to, że gdy zamierzamy dokonać podziału stratyfikowanego nie jest możliwe ustawienie opcji tasowania zbioru.

**StratifiedKFolds**

In [289]:
```
folds = 10
skf = StratifiedKFold(n_splits=folds)
print(skf)
```

```
StratifiedKFold(n_splits=10, random_state=None, shuffle=False)
```

In [312]:
```
sum = 0
for train_index, test_index in skf.split(x,y):
    x_train, x_test = x[train_index], x[test_index]
    y_train, y_test = y[train_index], y[test_index]
    model = GaussianNB().fit(x_train, y_train)
    y_pred = model.predict(x_test)
    errors = (x_test.shape[0],(y_test != y_pred).sum())
    accuracy = 100-errors[1]/errors[0]*100
    sum += accuracy
average = sum/folds
print('Accuracy:', average)
```

```
Accuracy: 96.16959064327486
```

*Rysunek 6 Predykcja modelu za pomocą stratyfikowanej walidacji KFold*

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.96 | 0.95 | 0.94 |
| **F1 score** | 0.95 | 0.95 | 0.94 |

*Tabela 3 Wyniki predykcji modelu Wine przy różnych proporcjach podziału zbioru przy stratyfikowanej walidacji K-Fold*

Tabela 3 w porównaniu z tabelą 2 pokazuje także przewagę kroswalidacji stratyfikowanej nad zwykłą bez stratyfikacji. Dzięki zachowaniu oryginalnych proporcji danych jesteśmy w stanie wyuczyć nasz model wszystkich rodzajów klas. Dzięki temu w naszym zbiorze testowym nigdy nie będzie tak, że znajdą się tam jedynie dane opisane klasą, której nasz model nie widział w trakcie uczenia.

## Analiza Glass dataset

### Walidacja prosta

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Podział zbiorów treningowy/testowy** | 90/10 | 70/30 | 50/50 |
| **Średnia predykcja** | 0.95 | 0.96 | 0.96 |
| **F1 score** | 0.95 | 0.96 | 0.95 |

*Tabela 4 Wyniki predykcji modelu Glass przy różnych proporcjach podziału zbioru treningowego i testowego*

### Walidacja K-Fold

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.89 | 0.91 | 0.94 |
| **F1 score** | 0.90 | 0.90 | 0.93 |

*Tabela 5 Wyniki predykcji modelu Glass przy różnych proporcjach podziału zbioru przy walidacji K-Fold*

### Stratyfikowana walidacja K-Fold

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.94 | 0.97 | 0.96 |
| **F1 score** | 0.94 | 0.96 | 0.94 |

*Tabela 6 Wyniki predykcji modelu Glass przy różnych proporcjach podziału zbioru przy stratyfikowanej walidacji K-Fold*

## Analiza Pima diabetes dataset

### Walidacja prosta

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Podział zbiorów treningowy/testowy** | 90/10 | 70/30 | 50/50 |
| **Średnia predykcja** | 0.74 | 0.75 | 0.74 |
| **F1 score** | 0.70 | 0.73 | 0.73 |

*Tabela 7 Wyniki predykcji modelu Pima diabetes przy różnych proporcjach podziału zbioru treningowego i testowego*

### Walidacja K-Fold

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.76 | 0.74 | 0.75 |
| **F1 score** | 0.75 | 0.73 | 0.73 |

*Tabela 8 Wyniki predykcji modelu Pima diabetes przy różnych proporcjach podziału zbioru przy walidacji K-Fold*

### Stratyfikowana walidacja K-Fold

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.76 | 0.75 | 0.73 |
| **F1 score** | 0.75 | 0.73 | 0.70 |

*Tabela 9 Wyniki predykcji modelu Pima diabetes przy różnych proporcjach podziału zbioru przy stratyfikowanej walidacji K-Fold*

## Dyskretyzacja:

Dyskretyzacja ma na celu zamianę ciągłych zmiennych losowych na dyskretne zmienne losowe. Gdy dokonuje się dyskretyzacji zawsze ma się do czynienia z pewnego stopnia błędem dyskretyzacji. Gdy dokonana zostaje zamiana liczb ciągłych na dyskretne tracone są niektóre informacje o danych, np. ucięcie liczb po przecinku, lub nawet całkowita strata informacji o zmiennej poprzez wrzucenie liczby do kubełka w którym traci swoje oryginalne znaczenie i zostaje zastąpiona inną liczbą.

Jedną z użytych metod dyskretyzacji jest kubełkowanie (ang. *binning*). Binning tworzy określoną ilość interwałów na danym zbiorze. Następnie sprawdza, każdą wartość po kolei i jeżeli wpada ona w dany przedział jest ona tam wtedy dodawana a liczba wystąpień wzrasta, np. interwał (0.0, 2.05> i atrybut z wartością 1.42 zostanie tam przydzielony. Następnie takim przedziałom nadawana może być jakaś klasa i na tej bazie zostaje dokonana klasyfikacja.

Dyskretyzacji dokonuje się na każdej kolumnie osobno ze względu na różne wartości atrybutów zatem także inne przedziały, do których dane zostaną przydzielone.

## Pandas.cut

Pierwsza z wybranych metod dyskretyzacji jest dostarczana przez bibliotekę pandas. Do metody należy podać przedział danych (u nas kolumna), liczbę kubełków na ile ma zostać podzielony dany przedział oraz kilka opcjonalnych argumentów. Metoda tworzy daną ilość przedziałów na danym zbiorze jedynie ze względu na minimalną i maksymalną wartość atrybutów. Tzn. przedziały zostają utworzone w równomiernych wielkościach. Metoda zwraca utworzone przedziały oraz liczbę atrybutów, które zostały do danego interwału przydzielone.

```
def discretization(mode, column, bin_count):
    if mode == 'CUT':
        return pd.cut(column, bin_count, labels=False)
    elif mode == 'QCUT':
        return pd.qcut(column, bin_count, labels=False, duplicates='drop')
    else:
        hist, bins_edges = np.histogram(column, bins=mode)
        return np.digitize(column, bin_edges)
```

*Rysunek 7 Sposób tworzenie różnych sposobów dyskretyzacji danych*

## Pandas.qcut

Drugą z wybranych metod jest metoda qcut, także dostarczane przez bibliotekę pandas. Qcut tak jak cut zwraca interwały oraz liczbę przypisanych tam danych, jednakże różni się w sposobie tworzenia interwałów. Metoda ma w swojej nazwie „q" które z angielskiego oznacza kwantyle (ang. *quantile*). Oznacza to, że interwały są tworzone z uwzględnieniem rozkładu ilościowego oryginalnych danych ciągłych. Utworzone przedziały będą miały taką samą lub bardzo zbliżoną liczbę danych przypisanych do siebie. Dzięki temu sposobowi możemy zapewnić lepsze rozłożenie danych w każdym kubełku. Jest bardziej prawdopodobne, że nie uzyskamy przedziałów, w których znajduje się większość danych i takich, w których wartości w ogóle nie ma.

Podsumowując te dwie metody: cut dobiera przedziały ze względu na to aby były one równomierne rozłożone na całej wielkości przedziału danych, a qcut uwzględnia częstość występowania danych i dobiera kubełki tak aby liczba danych w nich była taka sama.

## Histogram z algorytmem Doane's formula

Ostatnią z wybranych do przetestowania metod dyskretyzacji jest tworzenie histogramu i tworzenie odpowiednich kubełków za pomocą algorytmu Doeane'sa. Jest to modyfikacja formuły Sturges'a, która zakłada że atrybuty będą w rokzładzie normalnym. Formuła Doane'sa ulepsza poprzednią formułe o zastosowanie do danych nie z rozkładu normalnego.

$$k = 1 + \log_2(n) + \log_2\left(1 + \frac{|g_1|}{\sigma_{g_1}}\right)$$

*Rysunek 8 Formuła Doane'sa do tworzenia binów w histogramach*

# 5. Analiza dyskretyzacji i kroswalidacji

## Zbiór Pima diabetes

### Bins = 10

Krosswalidacja: Kfold. Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.66 | 0.65 |
| **F1 score** | 0.59 | 0.59 | 0.58 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.65 | 0.65 | 0.66 |
| **F1 score** | 0.58 | 0.58 | 0.59 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.66 | 0.66 |
| **F1 score** | 0.60 | 0.59 | 0.60 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.66 | 0.67 |
| **F1 score** | 0.60 | 0.59 | 0.61 |

Krosswalidacja: Kfold, Dyskretyzacja: Doane’s formula

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.61 | 0.61 | 0.62 |

| | | | |
|---|---|---|---|
| **F1 score** | 0.55 | 0.55 | 0.56 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: Doane’s formula

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.62 | 0.61 | 0.61 |
| **F1 score** | 0.56 | 0.56 | 0.56 |

**Bins = 5**

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.67 | 0.66 | 0.66 |
| **F1 score** | 0.56 | 0.55 | 0.54 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.67 | 0.66 | 0.66 |
| **F1 score** | 0.52 | 0.50 | 0.51 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.66 | 0.68 |
| **F1 score** | 0.48 | 0.48 | 0.48 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.67 | 0.66 | 0.66 |
| **F1 score** | 0.48 | 0.47 | 0.45 |

**Bins = 2**

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.65 | 0.65 | 0.66 |
| **F1 score** | 0.43 | 0.43 | 0.47 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.65 | 0.66 | 0.66 |
| **F1 score** | 0.43 | 0.44 | 0.44 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.66 | 0.67 |
| **F1 score** | 0.48 | 0.48 | 0.48 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.67 | 0.66 | 0.66 |
| **F1 score** | 0.48 | 0.47 | 0.45 |

Jak widać z powyższych tabel najlepsze wyniki dla zbioru Pima Diabetes uzyskujemy przy krosswalidacji Kfold jak i StratifiedKfold przy podziale na 10 lub 5 foldów i 10 binów. Najlepsze wyniki uzyskano przy dyskretyzacji QCUT lecz są one bardzo porównywalne z dyskretyzacją CUT.

## Zbiór Wine

### Bins = 10

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.90 | 0.87 | 0.35 |
| **F1 score** | 0.61 | 0.64 | 0.19 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.91 | 0.91 | 0.91 |
| **F1 score** | 0.92 | 0.92 | 0.92 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.90 | 0.87 | 0.36 |
| **F1 score** | 0.64 | 0.52 | 0.20 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.92 | 0.91 | 0.89 |
| **F1 score** | 0.92 | 0.91 | 0.90 |

Krosswalidacja: Kfold, Dyskretyzacja: Doane’s formula

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.80 | 0.71 | 0.26 |
| **F1 score** | 0.48 | 0.42 | 0.17 |

StratifiedKfold, Dyskretyzacja: Doane’s formula

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.87 | 0.86 | 0.86 |
| **F1 score** | 0.87 | 0.86 | 0.67 |

**Bins = 5**

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.88 | 0.88 | 0.35 |
| **F1 score** | 0.55 | 0.55 | 0.20 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.93 | 0.91 | 0.92 |
| **F1 score** | 0.93 | 0.92 | 0.92 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.89 | 0.87 | 0.34 |
| **F1 score** | 0.62 | 0.44 | 0.20 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.92 | 0.91 | 0.89 |
| **F1 score** | 0.92 | 0.92 | 0.90 |

**Bins = 2**

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.80 | 0.73 | 0.28 |
| **F1 score** | 0.51 | 0.44 | 0.15 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.84 | 0.83 | 0.85 |
| **F1 score** | 0.84 | 0.82 | 0.86 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.85 | 0.79 | 0.29 |
| **F1 score** | 0.53 | 0.42 | 0.17 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.89 | 0.89 | 0.87 |
| **F1 score** | 0.89 | 0.88 | 0.87 |

Zdecydowanie najlepsze wyniki w zbiorze Wine uzyskano w przypadku krosswalidacji stratyfikowanej wraz z dyskretyzacją QCUT przy podziale na 10 binów. Ciekawym jest że krosswalidacja stratyfikowana miała około 30% lepsze wyniki w przypadku F1 Score w porównaniu do zwykłej krosswalidacji Kfold, podczas gdy wyniki średniej celności były podobne.

## Zbiór Glass

### Bins = 10

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.62 | 0.47 | 0.34 |
| **F1 score** | 0.39 | 0.25 | 0.11 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.74 | 0.70 | 0.61 |
| **F1 score** | 0.56 | 0.54 | 0.48 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.51 | 0.31 | 0.26 |
| **F1 score** | 0.29 | 0.16 | 0.07 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.78 | 0.76 | 0.55 |
| **F1 score** | 0.73 | 0.68 | 0.56 |

Krosswalidacja: Kfold, Dyskretyzacja: Doane’s formula

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.08 | 0.10 | 0.16 |
| **F1 score** | 0.07 | 0.07 | 0.08 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: Doane’s formula

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.51 | 0.50 | 0.51 |
| **F1 score** | 0.31 | 0.30 | 0.32 |

**Bins = 5**

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.57 | 0.34 | 0.35 |
| **F1 score** | 0.41 | 0.15 | 0.11 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.68 | 0.63 | 0.58 |
| **F1 score** | 0.45 | 0.42 | 0.35 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.44 | 0.27 | 0.24 |
| **F1 score** | 0.24 | 0.13 | 0.07 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.76 | 0.72 | 0.57 |
| **F1 score** | 0.66 | 0.60 | 0.53 |

**Bins = 2**

Krosswalidacja: Kfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.47 | 0.28 | 0.18 |
| **F1 score** | 0.27 | 0.13 | 0.06 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: CUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.64 | 0.61 | 0.48 |
| **F1 score** | 0.43 | 0.37 | 0.32 |

Krosswalidacja: Kfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.38 | 0.33 | 0.18 |
| **F1 score** | 0.22 | 0.19 | 0.06 |

Krosswalidacja: StratifiedKfold, Dyskretyzacja: QCUT

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.63 | 0.51 |
| **F1 score** | 0.37 | 0.34 | 0.33 |

W przypadku zbioru Glass najlepszy wynik uzyskano w przypadku dyskretyzacji za pomocą metody QCUT i krosswalidacji stratyfikowanej przy 10 kubełkach. Metoda Doena, która ma na celu pomóc w przypadku wartości nie z dystrybucji normalnej miała najgorsze wyniki. Jej celność spadła poniżej 10% w przypadku Kfold. Gdy użyto krosswalidacji bez stratyfikacji celność spadała aż o około 40% w porównaniu ze stratyfikacją.

# 6. Porównanie najlepszych wyników

## Zbiór Pima diabetes

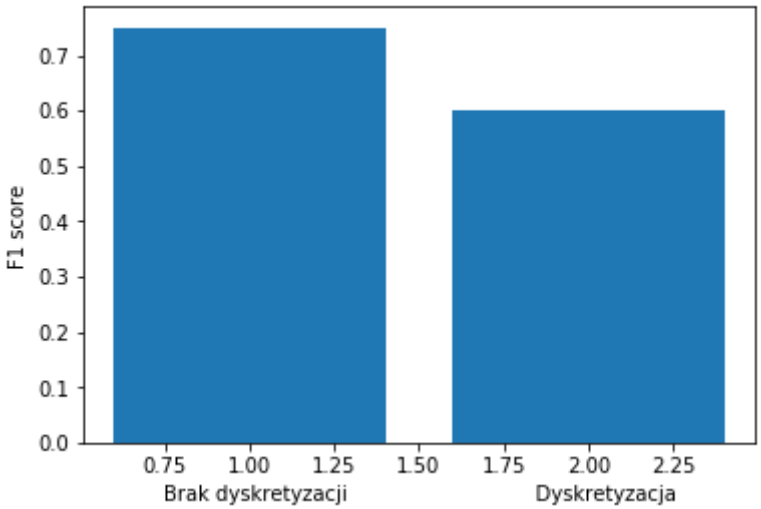
Brak dyskretyzacji, Krosswalidacja: StratifiedKfold

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.76 | 0.75 | 0.73 |
| **F1 score** | 0.75 | 0.73 | 0.70 |

Dyskretyzacja QCUT, Krosswalidacja: Kfold, bins: 10

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.66 | 0.66 | 0.66 |
| **F1 score** | 0.60 | 0.59 | 0.60 |

Przy dyskretyzacji najlepsze wyniki uzyskano przy dyskretyzacji metodą QCUT oraz przy krosswalidacji Kfold. Jednakże krosswalidacja stratyfikowana miała bardzo podobne wyniki. W przypadku Pima diabetes przez dyskretyzacje uzyskane wyniki zostały zmniejszone o około 10-15%.




*Rysunek 9 Porównanie wyników dyskretyzacji i jej braku dla zbioru Pima Diabetes przy 10 foldach*

## Zbiór Wine

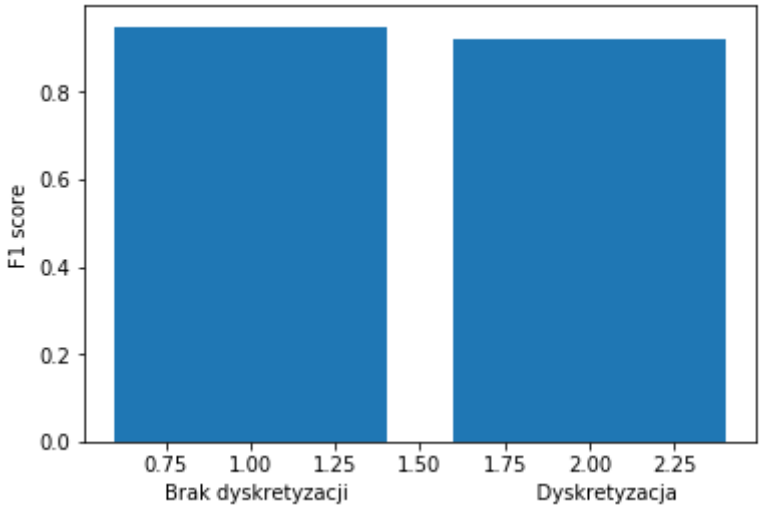
Brak dyskretyzacji, Krosswalidacja: StratifiedKfold

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.96 | 0.95 | 0.94 |
| **F1 score** | 0.95 | 0.95 | 0.94 |

Dyskretyzacja: QCUT, Krosswalidacja: StratifiedKfold, bins: 10

| | Podział 1 | Podział 2 | Podział 3 |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.92 | 0.91 | 0.89 |
| **F1 score** | 0.92 | 0.91 | 0.90 |

W przypadku dyskretyzacji jak i jej braku wyniki są bardzo porównywalne. Są one na poziomie 90-95%, czyli bardzo wysoko i różnią się o około 3%.




*Rysunek 10 Porównanie wyników dyskretyzacji i jej braku dla zbioru Wine przy 10 foldach*

## Zbiór Glass

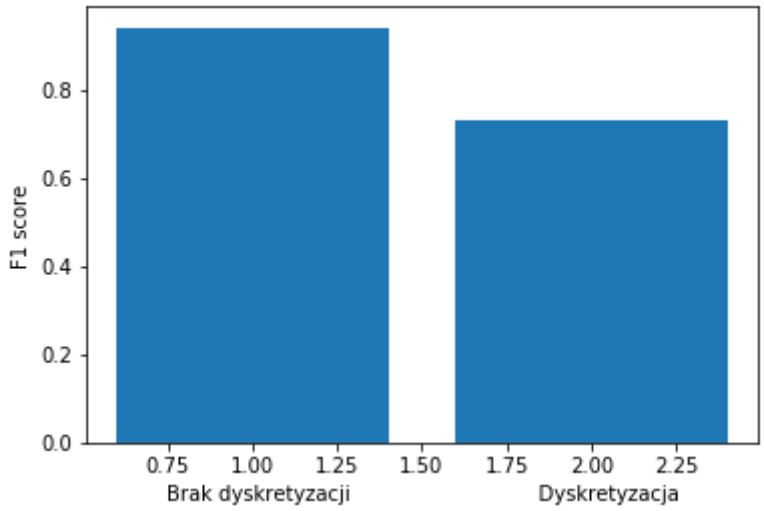
Brak dyskretyzacji, Krosswalidacja: StratifiedKfold

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.94 | 0.97 | 0.96 |
| **F1 score** | 0.94 | 0.96 | 0.94 |

Dyskretyzacja: QCUT, Krosswalidacja: StratifiedKfold, bins: 10

| | **Podział 1** | **Podział 2** | **Podział 3** |
|---|---|---|---|
| **Liczba podzbiorów** | 10 | 5 | 2 |
| **Średnia predykcja** | 0.78 | 0.76 | 0.55 |
| **F1 score** | 0.73 | 0.68 | 0.56 |

Przy dyskretyzacji zbioru Glass wyniki były o wiele niższe niż w przypadku jej braku. Najlepszy wynik uzyskano przy dyskretyzacji metodą QCUT i stratyfikowaną krosswalidacją. Inne wyniki były o wiele niższe. Pomimo tego wynik w porównaniu do braku dyskretyzacji był i tak niższy o około 20-25%.




*Rysunek 11 Porównanie wyników dyskretyzacji i jej braku dla zbioru Glass przy 10 foldach*

## 7. Podsumowanie

W większości przypadków najlepsze wyniki uzyskiwano przy krosswalidacji stratyfikowanej. Najlepszą metodą dyskretyzacji okazała się metoda QCUT bazująca na kwantylach i zachowaniu równomiernego rozłożenia wartości w kubełkach. Najlepszy podział na kubełki okazał się w większości w przypadku 10 binów, jednakże wyniki z podziału na 5 są bardzo porównywalne.

Okazało się także, że w części przypadków dyskretyzacja nie była pomocna, a nawet zmniejszyła ona drastycznie jakość wyników. Jest to często spowodowane stratą wielu wartościowych informacji o atrybutach. Model Gaussowski naiwnego Bayes'a, pomimo założeniu o danych z rozkładu normalnego okazał się lepszy lub porównywalny gdy użyto różnych metod dyskretyzacji na danym zbiorze.

Pomimo podobnych predykcji w przypadku średniej celności, faktyczne wyniki, którymi należy się kierować są pokazane przez F1 score, który daje bardziej realistyczny i dokładniejszy wgląd na predykcję.