WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ CYBERNETYKI

SPRAWOZDANIE
Metody Eksploracji Danych

Temat laboratorium: **KLASYFIKACJA NA PODSTAWIE KLASYFIKATORA BAYESOWSKIEGO I NAJBLIŻSZEGO SĄSIEDZTWA**

INFORMATYKA

(kierunek studiów)

INŻYNIERIA SYSTEMÓW – ANALIZA DANYCH

(specjalność)

Zespół:

Michał ŚLĘZAK
Szymon OLEŚKIEWICZ

Prowadzący laboratorium:

Dr inż. Romuald Hoffmann, prof. WAT

Warszawa 2025

Spis treści

Wstęp

Celem niniejszego sprawozdania jest prezentacja wyników klasyfikacji danych dotyczących decyzji zawodników siatkówki plażowej o rozegraniu meczu treningowego w zależności od warunków pogodowych. W pracy zastosowano dwie metody klasyfikacji: naiwny klasyfikator Bayesa oraz metodę k-najbliższych sąsiadów (k-NN) z dwoma funkcjami odległości - euklidesową i miejską (Manhattan).

Rozdział I. Podstawy teoretyczne

I.1. Naiwny klasyfikator Bayesa

Naiwny klasyfikator Bayesa jest probabilistyczną metodą klasyfikacji opartą na twierdzeniu Bayesa z założeniem niezależności warunkowej cech. Zasada działania klasyfikatora opiera się o rozwiązanie zadania optymalizacyjnego w postaci:

$$C^* = arg \max_{\{C_i:\, i \geq 1\}} \left(Pr\{C_i\} \prod_{k=1}^{n} Pr\{x_k | C_i\} \right) \tag{1}$$

, gdzie

- $Pr\{C_i\}$ – prawdopodobieństwo wystąpienia klasy i, które można estymować poprzez iloczyn liczby wystąpień danej klasy do liczność zbioru testowego,
- $Pr\{x_k|C_i\}$ – prawdopodobieństwo warunkowe wystąpienia atrybutu x_k pod warunkiem wystąpienia klasy C_i.

Dla każdego atrybutu prawdopodobieństwo $Pr\{x_k|C_i\}$ można estymować w oparciu o zbiór uczący Z, wykorzystując wzór:

$$Pr\{x_k|C_i\} = \frac{\left|C_i^{x_k}\right|}{|C_i|} \tag{2}$$

, gdzie

- $\left|C_i^{x_k}\right|$ – oznacza liczbę „przykładów” (w zbiorze Z) z klasy C_i, dla których atrybut o numerze $k = 1, \dots, n$ przyjmuje wartość x_k.

I.2. Metoda k-najbliższych sąsiadów (k-NN)

Metoda k-NN opiera się na klasyfikacji obiektu w oparciu o jego k najbliższych sąsiadów, w sensie przyjętej miary odległości. Algorytm nie buduje jawnego modelu klasyfikacji, lecz opiera decyzję bezpośrednio na podobieństwie nowego obiektu do obiektów ze zbioru treningowego.

Klasyfikacja nowego obiektu y przebiega według następującego schematu:

1. Obliczenie odległości między obiektem y a wszystkimi obiektami ze zbioru treningowego
2. Wybór k najbliższych sąsiadów
3. Przypisanie klasy na podstawie głosowania (prostego lub ważonego)

Dwie metryki odległości wykorzystane w pracy to:

1. Odległość euklidesowa

$$d(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{3}$$

2. Odległość miejska (Manhattan)

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{4}$$

Dla danych jakościowych konieczne jest przeprowadzenie kodyfikacji kategorii na wartości numeryczne, aby umożliwić obliczenie odległości w przestrzeni cech.

Rozdział II. Opis problemu

II.1. Treść zadania

Mieszkańcy pewnej małej kamienicy, mieszczącej się zaraz tuż przy plaży morskiej, zaczęli od paru dni z ciekawością obserwować treningi grupy zawodników siatkówki plażowej. Mieszkańcy zauważyli, że ta grupa graczy niekiedy dzieli się na dwa zespoły w celu rozegrania meczu. Niestety obserwatorzy nie znają zamiarów trenujących, ale odnotowali, że gracze grają w różnych warunkach pogodowych. Swoje obserwacje odnotowali w tabeli.

Tab. 1. Obserwacje mieszkańców

Nr obserwacji	Siła wiatru	Zachmurzenie	Odczuwalna temperatura	Zagrano mecz
1	silny	pochmurnie	zimno	nie
2	silny	pochmurnie	ciepło	nie
3	brak	słonecznie	ciepło	tak
4	brak	słonecznie	gorąco	nie
5	słaby	pochmurnie	gorąco	tak
6	słaby	słonecznie	ciepło	tak
7	brak	pochmurnie	zimno	nie
8	silny	słonecznie	zimno	tak
9	brak	pochmurnie	gorąco	tak
10	silny	pochmurnie	ciepło	tak

Dzisiaj jest ciepło i słonecznie, ale wieje silny wiatr. Wobec tego mieszkańcy kibicujący grze zastanawiają się czy gracze pojawią się na plaży, aby rozegrać swój mecz treningowy, czy tylko wykonać ćwiczenia.

II.2. Opis problemu badawczego

Problem badawczy dotyczy przewidywania decyzji zawodników siatkówki plażowej o rozegraniu meczu treningowego na podstawie warunków pogodowych. Mieszkańcy kamienicy położonej przy plaży przez dziesięć dni obserwowali zachowanie grupy zawodników i odnotowali warunki pogodowe oraz podjęte decyzje o grze.
Zebrane dane obejmują następujące atrybuty:

- **Siła wiatru** - zmienna jakościowa o trzech wartościach: silny, słaby, brak
- **Zachmurzenie** - zmienna jakościowa o dwóch wartościach: pochmurnie, słonecznie
- **Odczuwalna temperatura** - zmienna jakościowa o trzech wartościach: zimno, ciepło, gorąco
- **Zagrano mecz** - zmienna decyzyjna binarna: tak, nie

Zadanie polega na przewidzeniu, czy w danych warunkach pogodowych:

siła wiatru = silny,
zachmurzenie = słonecznie,
odczuwalna temperatura = ciepło,

zostanie rozegrany mecz treningowy.

Rozdział III. Implementacja algorytmów

III.1. Struktura danych

Implementację przeprowadzono w języku Python z wykorzystaniem biblioteki pandas do zarządzania danymi. W celu zapewnienia spójności wartości kategorycznych zastosowano klasy wyliczeniowe (StrEnum), które definiują dopuszczalne wartości dla każdego atrybutu.

Kod. 1. Struktura danych

```
class SilaWiatru(StrEnum):
    silny = 'silny'
    slaby = 'słaby'
    brak = 'brak'

class Zachmurzenie(StrEnum):
    pochmurnie = 'pochmurnie'
    slonecznie = 'słonecznie'

class Temperatura(StrEnum):
    zimno = 'zimno'
    cieplo = 'ciepło'
    goraco = 'gorąco'

class Zagrano(StrEnum):
    tak = 'tak'
    nie = 'nie'
```

Dane treningowe oraz testowe zostały zorganizowane w strukturze DataFrame biblioteki pandas, co umożliwia efektywne operacje na danych i łatwą integrację z algorytmami klasyfikacji.

III.2. Implementacja naiwnego klasyfikatora Bayesa

Funkcja naive_bayess implementuje algorytm naiwnego klasyfikatora Bayesa zgodnie z przedstawionymi podstawami teoretycznymi. Algorytm działa następująco:

1. Wyznaczenie unikalnych klas decyzyjnych ze zbioru treningowego
2. Obliczenie prawdopodobieństw a priori $Pr\{C_i\}$ dla każdej klasy
3. Dla każdego atrybutu obliczenie prawdopodobieństw warunkowych $Pr\{x_k|C_i\}$
4. Wyznaczenie klasy maksymalizującej iloczyn $Pr\{C_i\}\prod_{k=1}^{n} Pr\{x_k|C_i\}$

Kod. 2. Funkcja implementująca algorytm naiwnego klasyfikatora Bayesa

```
1           def naive_bayess(X_train: pd.DataFrame, y_train: pd.Series, X_test:
            pd.DataFrame):
2               y_uniques = y_train.unique()
3               prob = [len(y_train[y_train == y]) / len(y_train) for y in
            y_uniques]
4
5               for col in X_train:
6                   matching = X_train[col][X_train[col] == X_test[col][0]]
7                   ys = (y_train[matching.index])
8
9                   for i in range(len(prob)):
10                      prob[i] *= len(ys[ys == y_uniques[i]]) / len(ys)
11
12              result = y_uniques[argmax(prob)]
13
14              return result
```

III.3. Implementacja metody k-NN

Funkcja knn implementuje metodę k-najbliższych sąsiadów z możliwością parametryzacji funkcji odległości. Algorytm wykonuje następujące kroki:

1. Kodyfikacja danych jakościowych na wartości numeryczne
2. Obliczenie odległości między obiektem testowym a wszystkimi obiektami treningowymi
3. Sortowanie obiektów według odległości rosnąco
4. Wybór k najbliższych sąsiadów
5. Głosowanie proste - wybór klasy występującej najczęściej wśród k sąsiadów

Dla obsługi danych jakościowych zaimplementowano funkcję codify_df, która przypisuje kolejne liczby naturalne kategoriom każdego atrybutu w kolejności ich występowania w zbiorze referencyjnym.

Kod. 3. Funkcja implementująca klasyfikację metodą k-NN.

```
1             def knn(X_train: pd.DataFrame, y_train: pd.Series, X_test:
              pd.DataFrame, k: int, dist_f: Callable):
2                 def codify_df(df: pd.DataFrame, reference_df: pd.DataFrame =
              None) -> pd.DataFrame:
3                     if reference_df is None:
4                         reference_df = df
5                     new_df = pd.DataFrame({
6                         col: [
7                             list(reference_df[col].unique()).index(df[col][i])
8                             for i in range(len(df[col]))
9                         ]
10                        for col in df
11                    })
12                    return new_df
13
14                Xs_train = codify_df(X_train)
15                Xs_test = codify_df(X_test, X_train)
16                ys_train = pd.Series([
17                    list(y_train.unique()).index(y)
18                    for y in y_train
19                ])
20
21                x_test = list([Xs_test[col][0] for col in Xs_test.columns])
22                distances = [
23                    dist_f(list([xs[1][col] for col in Xs_train.columns]),
              x_test)
24                    for xs in Xs_train.iterrows()
25                ]
26                Xs_train["Y"] = ys_train
27                Xs_train["Distance"] = distances
28                Xs_train.sort_values(by=["Distance"], ascending=True,
              inplace=True)
29
30                k_nearest = []
31                for ki in range(k):
32                    neighbours = Xs_train[Xs_train["Distance"] ==
              min(Xs_train["Distance"])]
33                    Xs_train = Xs_train[Xs_train["Distance"] !=
              min(Xs_train["Distance"])]
```

```
34                  k_nearest.extend(neighbours.Y)
35
36              return y_train.unique()[argmax(len([x for x in k_nearest if x
            == y]) for y in y_train.unique())]
```

Zaimplementowano dwie funkcje odległości.

Kod. 4. Funkcje implementujące dwie metryki odległości - euklidesowa i Manhattan

```
1           def euler_dist(x: list[int], y: list[int]) -> float:
2               return sqrt(sum(abs(_x - _y) for _x, _y in zip(x, y)))
3
4           def manhattan_dist(x: list[int], y: list[int]) -> float:
5               return sum(abs(_x - _y) for _x, _y in zip(x, y))
```

Rozdział IV. Wyniki eksperymentów

IV.1. Wyniki klasyfikacji

IV.1.1. Klasyfikacja metodą naiwnego Bayesa

Dla danych testowych:

siła wiatru = **silny**,
zachmurzenie = **słonecznie**,
temperatura **= ciepło**,

naiwny klasyfikator Bayesa dokonał następującej klasyfikacji:

zagrano mecz = **tak**.

IV.1.2. Klasyfikacja metodą k-NN z odległością euklidesową

Dla tych samych danych testowych, metoda k-NN z odległością euklidesową dała następujący wynik:

zagrano mecz = **nie**.

Klasyfikacji dokonano dla wszystkich istotnych wartości parametru k (wartości od 1 do 5), uzyskując ten sam wynik.

IV.1.3. Klasyfikacja metodą k-NN z odległością miejską

Dla tych samych danych testowych, metoda k-NN z odległością miejską dała taki sam wynik co metoda k-NN z odległością miejską:

zagrano mecz = **nie**.

Klasyfikacji dokonano dla wszystkich istotnych wartości parametru k (wartości od 1 do 5), uzyskując ten sam wynik.

IV.2. Analiza porównawcza wyników

Otrzymane wyniki pokazują istotną rozbieżność między metodami klasyfikacji. Naiwny klasyfikator Bayesa przewiduje rozegranie meczu, podczas gdy obie wersje metody kNN przewidują jego brak.

IV.2.1. Analiza wyników naiwnego klasyfikatora Bayesa

Decyzja klasyfikatora bayesowskiego wynika z analizy prawdopodobieństw warunkowych dla poszczególnych atrybutów. Na podstawie danych treningowych można zaobserwować następujące zależności:
Prawdopodobieństwa *a priori*:

- $Pr\{zagrano\ mecz = tak\} = {}^{6}/_{10} = 0.6$
- $Pr\{zagrano\ mecz = nie\} = {}^{4}/_{10} = 0.4$

Analiza poszczególnych atrybutów:

- Zachmurzenie = **słonecznie**:

Przy słonecznej pogodzie mecze grane częściej: 3 przypadki "tak" vs 1 przypadek "nie"

$$Pr\{słonecznie|tak\} > Pr\{słonecznie|nie\}$$

Ten atrybut silnie wspiera decyzję o grze

- Temperatura = **ciepło**:

Przy ciepłej temperaturze: 3 przypadki "tak" vs 2 przypadki "nie"

$$Pr\{ciepło|tak\} > Pr\{ciepło|nie\}$$

Stosunek nieznacznie sprzyja decyzji "tak"

- Siła wiatru = **silny**:

Przy silnym wietrze: 2 przypadki "tak" vs 2 przypadki "nie"

$$Pr\{silny|tak\} = Pr\{silny|nie\}$$

Atrybut neutralny, nie dyskryminuje między klasami

Łączne prawdopodobieństwo *a posteriori* dla klasy "tak" okazało się wyższe dzięki kumulatywnemu efektowi prawdopodobieństw warunkowych, szczególnie silnemu wpływowi atrybutu "zachmurzenie = słonecznie".

IV.2.2. Analiza wyników metody k-NN

Metoda k-NN opiera decyzję na lokalnym podobieństwie w przestrzeni cech. Po zakodowaniu atrybutów jakościowych na wartości numeryczne i obliczeniu odległości, algorytm zidentyfikował k-najbliższych sąsiadów.
Analiza przestrzeni cech wskazuje, że najbliższymi sąsiadami testowego przypadku są prawdopodobnie obserwacje charakteryzujące się podobną konfiguracją atrybutów. Z danych treningowych wynika, że w najbliższym sąsiedztwie dominują przypadki z decyzją "nie", co zadecydowało o wyniku klasyfikacji.

Fakt, że obie metryki odległości (euklidesowa i miejska) dały identyczne wyniki klasyfikacji, może wynikać z dwóch czynników:

1. **Mały rozmiar zbioru danych** – przy zbiorze treningowym liczącym zaledwie 10 obserwacji, różnice między metrykami są mało zauważalne
2. **Niska wymiarowość** – przy tylko trzech atrybutach różnice geometryczne między metrykami są niewielkie

Podsumowanie

W ramach laboratorium przeprowadzono klasyfikację danych dotyczących decyzji o rozegraniu meczu siatkówki plażowej przy użyciu trzech wariantów algorytmów klasyfikacji. Otrzymano rozbieżne wyniki: naiwny klasyfikator Bayesa przewidział rozegranie meczu (wynik: tak), podczas gdy obie wersje metody k-NN przewidziały jego brak (wynik: nie).

Główne wnioski z przeprowadzonej analizy:

1. **Rozbieżność wyników między metodami** jest naturalnym zjawiskiem wynikającym z fundamentalnych różnic w podejściu do klasyfikacji. Naiwny Bayes opiera się na globalnych prawdopodobieństwach warunkowych, podczas gdy k-NN wykorzystuje lokalne podobieństwo w przestrzeni cech.
2. **Ograniczenia małego zbioru danych** (10 obserwacji) znacząco wpływają na wiarygodność wyników. Obydwie metody są wrażliwe na konkretne przypadki w zbiorze treningowym, co może prowadzić do niestabilnych predykcji.
3. **Naiwne założenie o niezależności atrybutów** w klasyfikatorze Bayesa nie zawsze odpowiada rzeczywistości. W analizowanym problemie cechy pogodowe mogą być skorelowane, co wpływa na jakość klasyfikacji.

Bibliografia

[1] Hoffmann R.: „Metody eksploracji danych - Wykład 5. Klasyfikacja Naive Bayes slajdy”, Materiały dydaktyczne WAT, 2025.

[2] Hoffmann R.: „Metody eksploracji danych - Wykład 5. Klasyfikacja regresja logistyczna Cz.II i K-NN slajdy”, Materiały dydaktyczne WAT, 2025.

Spis tabel

Załączniki

1. Plik źródłowy Lab-3-Zadanie-1-Klasyfikatory.py – implementacja algorytmów klasyfikacji zawierająca funkcje:
    a. `naive_bayes()` – implementacja naiwnego klasyfikatora Bayesa;
    b. `knn()` – implementacja metody k-najbliższych sąsiadów;
    c. `euler_dist()` – funkcja odległości euklidesowej;
    d. `manhattan_dist()` – funkcja odległości miejskiej;
    e. `print_result()` – funkcja pomocnicza do wyświetlania wyników.
2. Plik notebook Lab-3-Zadanie-1-Obliczenia.ipynb – kompletny kod przeprowadzonych eksperymentów obejmujący:
    a. Definicję klas wyliczeniowych dla atrybutów.
    b. Przygotowanie danych treningowych i testowych.
    c. Wykonanie klasyfikacji metodą naiwnego Bayesa
    d. Wykonanie klasyfikacji metodą k-NN z użyciem obu metryk odległości
    e. Wyświetlanie wyników klasyfikacji