Błażej Kapkowski, Konrad Konsek 13.03.2024

**„Laboratorium” 2**

**Metoda najmniejszych kwadratów**

**Dane techniczne:**

Język: Python

Translator: Visual Studio Code

Procesor: AMD Ryzen 7 5800H

System operacyjny: Windows 11

**Realizacja ćwiczenia:**

Celem zadania było zastosowanie metody najmniejszych kwadratów do predykcji, czy nowotwór jest złośliwy czy łagodny. Dane na temat nowotworów zawierają cechy, takie jak promień, tekstura itp. Zadanie polegało na analizie dwóch zbiorów danych (treningowego i walidacyjnego) i zastosowaniu metody najmniejszych kwadratów do stworzenia modelu predykcyjnego, który może klasyfikować nowotwory jako złośliwe lub łagodne.

**Poszczególne kroki:**

1. Otwarcie zbiorów danych:

Wykorzystano funkcje pd.io.parsers.read\_csv z biblioteki pandas do wczytania danych z plików breast-cancer-train.dat i breast-cancer-validate.dat.

data1=pd.io.parsers.read\_csv("breast-cancer-train.dat", usecols={1,2,4,5,10})

data2=pd.io.parsers.read\_csv("breast-cancer-validate.dat", usecols={1,2,4,5,10})

1. Tworzenie histogramu i wykresu:

Stworzono histogram i wykres pierwszej kolumny danych za pomocą funkcji hist i plot w celu wizualizacji rozkładu danych.

1. Tworzenie reprezentacji danych:

Stworzono reprezentację liniową i kwadratową danych za pomocą macierzy A.

A1\_linear = np.vstack([np.ones(len(df1)), df1.iloc[:, 1:].values.T]).T

A1\_quadratic = np.vstack([np.ones(len(df1)), df1.iloc[:, 1:].values.T, (df1.iloc[:, 1:].values\*\*2).T]).T

A2\_linear = np.vstack([np.ones(len(df2)), df2.iloc[:, 1:].values.T]).T

A2\_quadratic = np.vstack([np.ones(len(df2)), df2.iloc[:, 1:].values.T, (df2.iloc[:, 1:].values\*\*2).T]).T

1. Tworzenie wektora b:

Stworzono wektor b, który zawiera informacje o tym, czy nowotwór jest złośliwy (1) czy łagodny (-1). Wykorzystano funkcję np.where do zwięzłego zakodowania wektora b.

b1 = np.where(df1.iloc[:, 0] == 'M', 1, -1)

b2 = np.where(df2.iloc[:, 0] == 'M', 1, -1)

1. Znajdowanie wag:

Zastosowano równanie normalne do znalezienia wag dla reprezentacji liniowej i kwadratowej. Użyto do tego funkcji solve.

w\_linear = solve(A1\_linear.T @ A1\_linear, A1\_linear.T @ b1)

w\_quadratic = solve(A1\_quadratic.T @ A1\_quadratic, A1\_quadratic.T @ b1)

1. Obliczanie współczynnika uwarunkowania:

Wykorzystano cond(AT A) do obliczenia współczynnika uwarunkowania macierzy dla obu reprezentacji.

cond\_A1\_linear = np.linalg.cond(np.dot(A1\_linear.T, A1\_linear))

cond\_A1\_quadratic = np.linalg.cond(np.dot(A1\_quadratic.T, A1\_quadratic))

1. Sprawdzanie skuteczności modelu:

Wykorzystano uzyskane wagi do przewidywania typu nowotworu (złośliwego lub łagodnego) dla zbioru walidacyjnego. Obliczono także liczby fałszywie dodatnich i fałszywie negatywnych przypadków dla obu reprezentacji.

predictions\_linear\_labels = np.where(predictions\_linear > 0, 1, -1)

predictions\_quadratic\_labels = np.where(predictions\_quadratic > 0, 1, -1)

false\_positives\_linear = np.sum((predictions\_linear\_labels > 0) & (b2 == -1))

false\_negatives\_linear = np.sum((predictions\_linear\_labels <= 0) & (b2 == 1))

false\_positives\_quadratic = np.sum((predictions\_quadratic\_labels > 0) & (b2== -1))

false\_negatives\_quadratic = np.sum((predictions\_quadratic\_labels <= 0) & (b2== 1))