Sprawozdanie z Ćwiczenia 7 z WSI

Adam Szostek nr. indeksu 331443

22 stycznia 2025

1 Implementowany Algorytm

Naiwny klasyfikator Bayesa to prosty, lecz skuteczny algorytm stosowany w zadaniach klasyfikacyjnych. Opiera się na twierdzeniu Bayesa, które opisuje zależność pomiędzy prawdopodobieństwami zdarzeń i ich warunkami:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}, \tag{1}$$

gdzie:

- $P(A|B)$ - prawdopodobieństwo zdarzenia A, gdy zaszło zdarzenie B,
- $P(B|A)$ - prawdopodobieństwo zdarzenia B, gdy zaszło zdarzenie A,
- $P(A)$ - aprioryczne prawdopodobieństwo zdarzenia A,
- $P(B)$ - aprioryczne prawdopodobieństwo zdarzenia B.

Algorytm zakłada, że cechy wejściowe są od siebie niezależne (założenie naiwności), co znacznie upraszcza obliczenia. Klasyfikator wybiera tę klasę C_k, dla której prawdopodobieństwo warunkowe $P(C_k|\mathbf{x})$ jest największe, gdzie $\mathbf{x}$ to wektor cech. Można to zapisać jako:

$$C_k = \arg\max_{C_k} P(C_k) \prod_{i=1}^{n} P(x_i|C_k), \tag{2}$$

gdzie x_i to i-ta cecha wektora $\mathbf{x}$, a n to liczba cech.

W implementacji klasyfikatora w języku python przyjęto następujące założenia:

- **Estymacja apriorycznych prawdopodobieństw klas**: Wartość $P(C_k)$ dla każdej klasy obliczana jest jako stosunek liczby próbek należących do tej klasy do całkowitej liczby próbek w zbiorze uczącym:

$$P(C_k) = \frac{\text{liczba próbek klasy } C_k}{\text{liczba wszystkich próbek}}$$

- **Estymacja parametrów rozkładu cech**: Założono, że cechy dla każdej klasy są rozkładane zgodnie z rozkładem normalnym. Dla każdej klasy obliczane są średnia (μ) i wariancja (σ^2) dla każdej cechy:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$

- **Obliczanie prawdopodobieństwa warunkowego**: Aby przypisać próbce klasę, obliczane są logarytmy prawdopodobieństw warunkowych dla wszystkich klas. Prawdopodobieństwo warunkowe $P(C_k|\mathbf{x})$ jest proporcjonalne do iloczynu prawdopodobieństw cech w danej klasie pomnożonych przez aprioryczne prawdopodobieństwo klasy:

$$\log P(C_k|\mathbf{x}) \propto \log P(C_k) + \sum_{i=1}^{n} \log P(x_i|C_k)$$

  Klasa z najwyższym wynikiem jest przypisywana do próbki.

2 Planowane eksperymenty numeryczne

2.1 Zbiór danych

Implementację algorytmu naiwnego klasyfikatora Bayesa przetestowano na zbiorze Iris Data Set. Jest to zbiór, który zawiera informacje o 150 próbkach z trzech gatunków irysów. Celem jest nauczenie modelu umiejętności klasyfikacji próbek do odpowiedniego gatunku na podstawie czterech cech zawartych w próbkach.

2.2 Ocena modelu

Do oceny modelu posłużono się mechanizmem walidacji krzyżowej. Jest to technika oceny modelu, która polega na podziale danych na k grup (folds) i iteracyjnym testowaniu modelu na jednej grupie, trenując go na pozostałych. Proces ten można opisać następująco:

1. Podziel zbiór danych $\mathcal{D}$ na k równych (lub prawie równych) części: $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$.
2. Dla każdej części i (od 1 do k):
   (a) Użyj $\mathcal{D} \setminus \mathcal{D}_i$ jako zbioru treningowego.
   (b) Użyj $\mathcal{D}_i$ jako zbioru testowego.
   (c) Wytrenuj model i oblicz metrykę wydajności (np. dokładność, F_1, itp.).
3. Średnia metryka z k iteracji to końcowa ocena modelu:

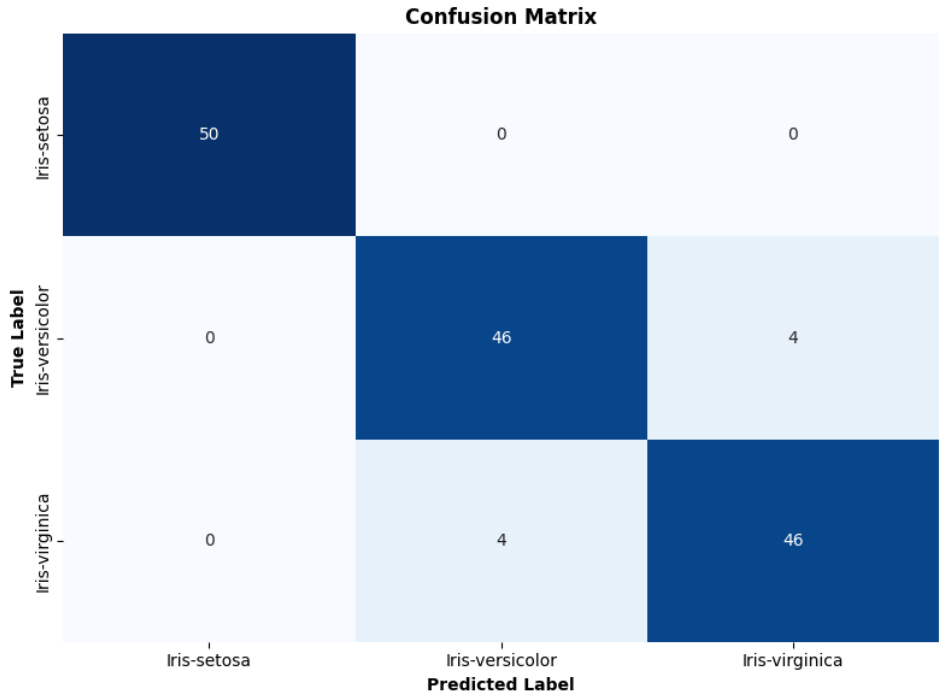
$$\text{Score} = \frac{1}{k} \sum_{i=1}^{k} \text{Score}_i$$

2.3 Porównanie z innymi modelami

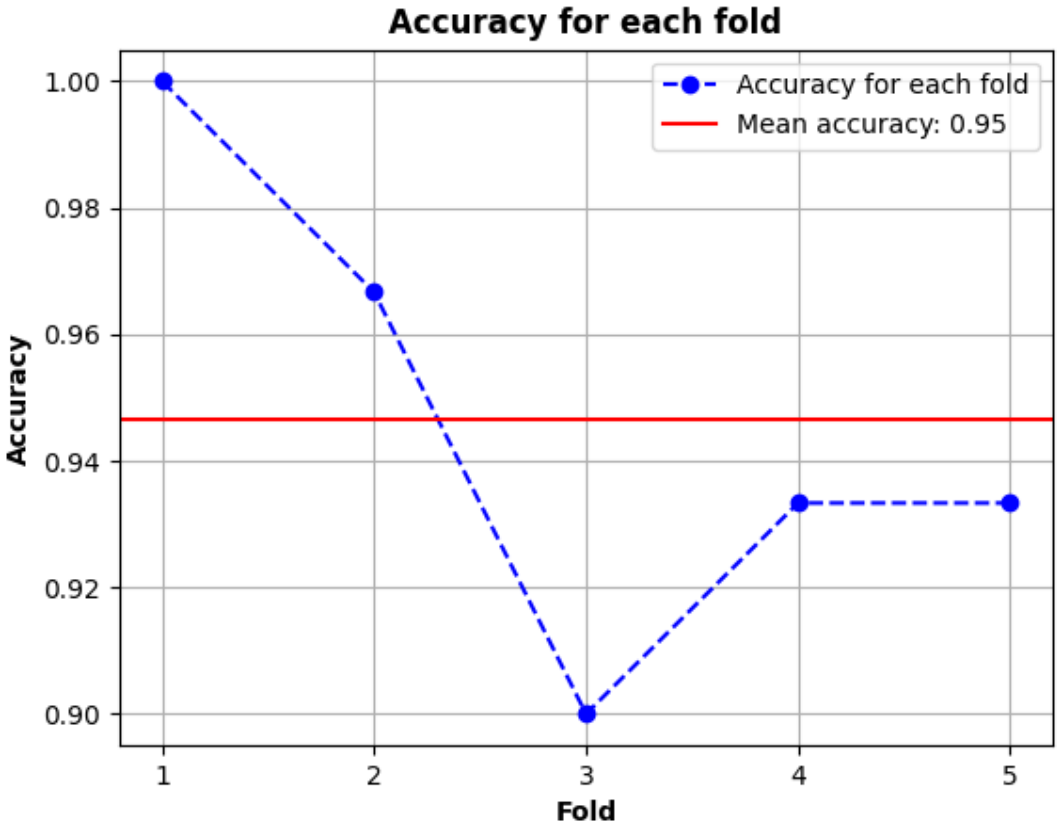
Dodatkowo, sporządzono wykres porównujący wyniki średniej dokładności uzyskanej z 5-krotnej walidacji krzyżowej dla trzech innych modeli używanych w problemach klasyfikacji. Są to SVC (z jądrem rbf), drzewo decyzyjne i "k-nearest neighbours". Do badań użyto implementacji modeli z biblioteki sklearn.

3 Uzyskane wyniki

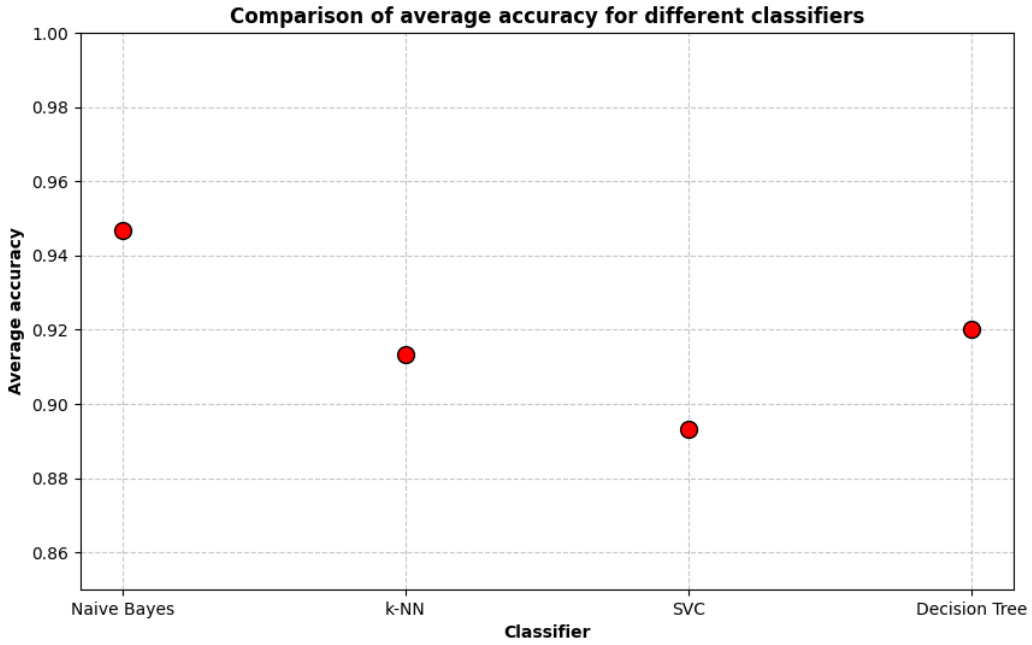
Poniżej zaprezentowano macierz pomyłek uzyskaną poprzez wykonanie 5-krotnej walidacji krzyżowej dla algorytmu naiwnego klasyfikatora Bayesa.



Poniższy wykres prezentuje dokładność uzyskaną dla każdego folda walidacji krzyżowej oraz średnią z tych wyników.



Poniżej widać porównanie średniej dokładności z 5-krotnej walidacji krzyżowej dla czterech różnych rodzajów klasyfikatorów w tym badanego algorytmu naiwnego Bayesa.



4 Wnioski z uzyskanych wyników

Na podstawie wyników przedstawionych wyżej, można śmiało stwierdzić, że naiwny klasyfikator Bayesa osiąga bardzo wysoką dokładność na testowanym zbiorze danych. Kluczową obserwacją jest bardzo mała wielkość Iris Dataset, który całościowo zawiera jedynie 150 próbek. Taka ilość danych dobrze wpasowuje się do mniej skomplikowanych algorytmów takich jak właśnie naiwny klasyfikator Bayesa. Może to powodować znaczące zawyżenie otrzymanych wyników, co tłumaczy dokładność w pierwszym foldzie równą 1,0. Inną istotną kwestią jest niezależność cech, która jest kluczowym założeniem algorytmu (pozwala prawidłowo wykorzystać twierdzenie Bayesa). Jest ona bardzo rzadko w pełni spełniona, ale w przypadku testowanego zbioru danych prawdopodobne jest, że dane są bliskie niezależności co ponownie tłumaczy bardzo wysoką dokładność. Jednocześnie porównanie z innymi klasyfikatorami popiera tę tezę, ponieważ naiwny klasyfikator Bayesa osiąga wyższą skuteczność poprawnej klasyfikacji od pozostałych modeli co wysuwa wniosek, że dobór algorytmu był poprawny dla badanego zbioru, co natomiast pomaga potwierdzić poprawność założeń. Ponadto założenie rozkładu normalnego dla cech irysów może być szczególnie trafne z uwagi na naturalny rozkład ich cech spotykany w rzeczywistości.
Podsumowując, naiwny klasyfikator Bayesa to prosty, lecz skuteczny algorytm, który dobrze nadaje się do klasyfikacji prostych danych. Jednocześnie należy podchodzić z ostrożnością do otrzymanych wyników z uwagi na niewielką liczbę próbek.