Jan Jędrzejewski
28.01.2022r

WSI - zadanie 7

Modele bayesowskie

Naiwny klasyfikator Bayesa

Treść zadania:
Zaimplementuj naiwny klasyfikator Bayesa, oraz zbadaj działanie algorytmu w zastosowaniu do zbioru danych https://archive.ics.uci.edu/ml/datasets/iris

Zbiór danych:
W zbiorze Iris Data Set jest 150 skategoryzowanych kwiatków. Każdy z nich przynależy do jednej z trzech klas. Wszystkie klasy są w równych proporcjach (po 50 przykładów każdej klasy). Kwiaty kategoryzujemy na podstawie czterech atrybutów. Dodatkowo mamy informację że jedna z klas jest liniowo separowalna od pozostałych dwóch.

Implementacja:
W pliku bayes.py zaimpolementowano klasę naiwnego klasyfikatora Bayesa. Przy konstruktorze należy podać dwie wartości: liczbę klas i liczbę atrybutów danych. Funkcja fit zapisuje zbiór trenujący jako zmienne klasy, w odpowiedni sposób, dzieląc go na klasy i na konkretne atrybuty, jest to podział, który w prosty sposób można wykorzystać dane do predykcji. W pliku read_dataset.py znajduje się funkcja do pobierania danych oraz do rysowania macierzy ze sprawnością algorytmu.

Działanie algorytmu:
Przeprowadziłem badanie działania algorytmu w zależności od rozmiaru zbioru testowego. Test_size to część zbioru danych która została wybrana jako zbiór testujący, pozostałe dane są zbiorem trenującym. Poniższe wartości zostały uśrednione ze 100 wykonanych prób.

```
Test_size=0.05 Average accuracy : 0.942

Test_size=0.1 Average accuracy : 0.953

Test_size=0.2 Average accuracy : 0.954

Test_size=0.3 Average accuracy : 0.952

Test_size=0.4 Average accuracy : 0.959
```

Jak obserwujemy rozmiar zbioru trenującego nie wpływa negatywnie na działanie algorytmu, dla każdej wartości obserwujemy wysoką skuteczność na poziomie około 95%. Dlatego przeprowadziłem jeszcze serię doświadczeń dla bardziej radykalnych wartości Test_size.

```
Test_size=0.5 Average accuracy : 0.952

Test_size=0.7 Average accuracy : 0.945

Test_size=0.8 Average accuracy : 0.945
```

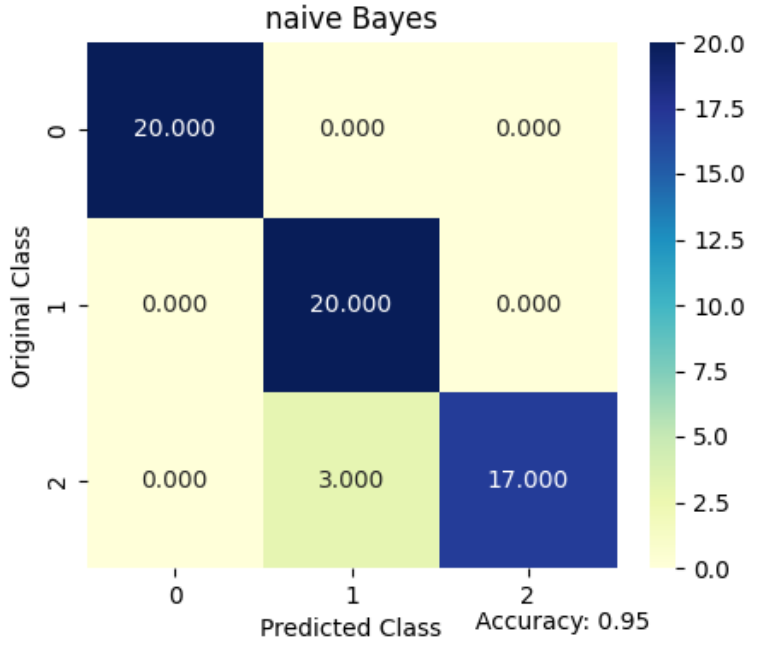
Algorytm dalej prezentuje świetnie wyniki, nawet dla bardzo małego zbioru trenującego.

Podsumowanie:

Uruchamiałem wiele razy algorytm i oglądałem macierze obrazujące działanie algorytmu, wartość accuracy wahała się od 0,9 do 1,0 w najlepszych przypadkach. Na poprzedniej stronie widzimy, że średnia skuteczność określania klasy algorytmu to 95%.

Na macierzach zaobserwowałem, że algorytm najczęściej myli się pomiędzy klasami 1 i 2 (prawie nigdy dla klasy 0) to najprawdopodobniej oznacza że klasa 0 jest liniowo separowalna od pozostałych i algorytm nie ma problemu z odróżnieniem jej.

Algorytm działa bardzo dobrze nawet dla zbioru trenującego o małej liczebności (20% całego zbioru) co wskazuje na to że dane są bardzo wysokiej jakości i nawet niewielka ich ilość pozwala przewidywać dokładnie klasy.



Oto przykładowa macierz wygenerowana dla zbioru testującego o rozmiarze 0.3