Systemy Sztucznej Inteligencji

Dokumentacja Projektu

Porównanie algorytmu KNN oraz Naiwnego Klasyfikatora Bayesa przy klasyfikacji odręcznie pisanych cyfr.

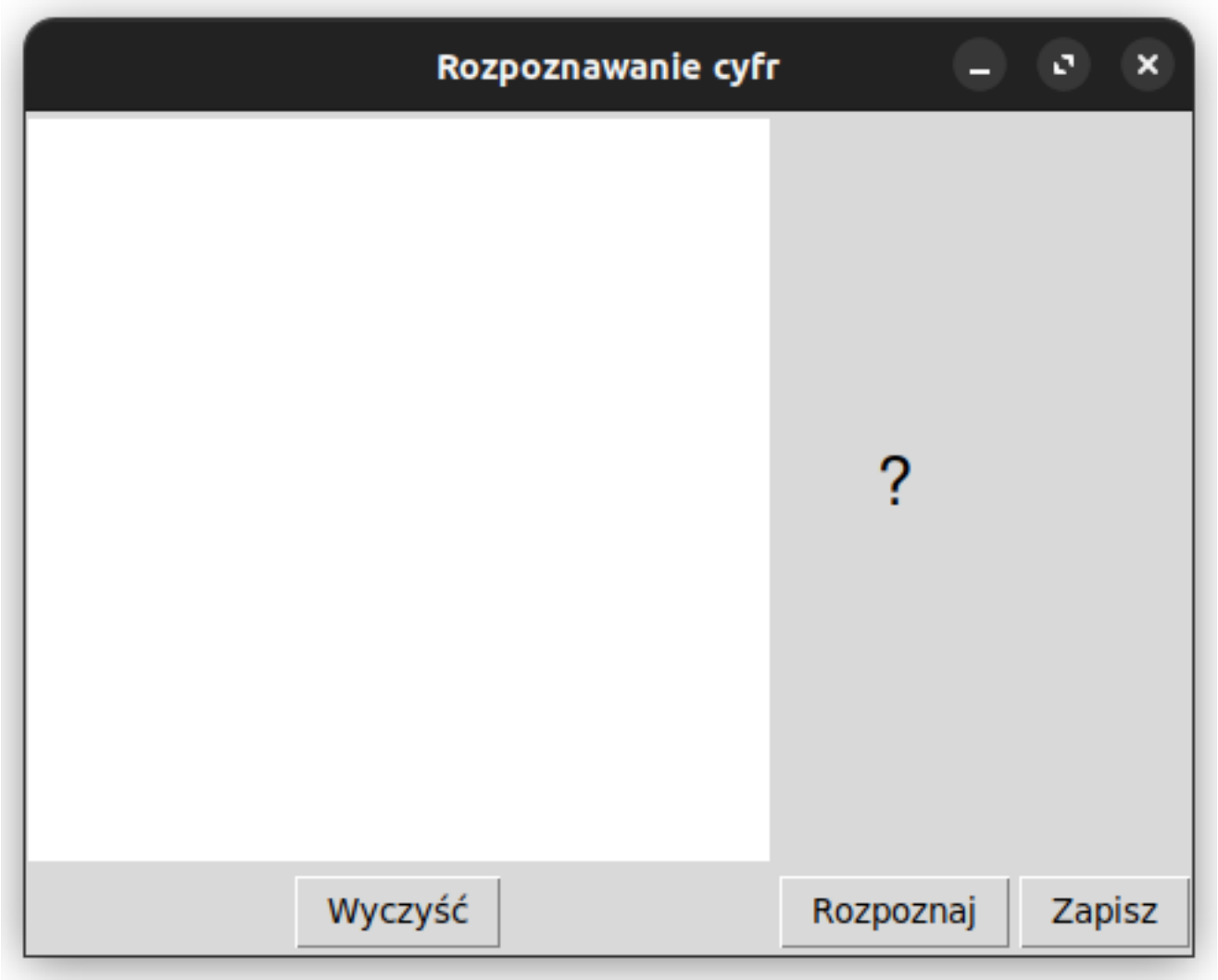
Piotr Skowroński gr. 3/6
Krzysztof Czuba gr. 4/7
Jakub Poreda gr. 3/6

26 czerwca 2023

1 Wstęp

1.1 Opis programu

Celem programu jest klasyfikowanie odręcznie pisanych cyfr przez użytkownika. Aplikacja zawiera proste GUI, które pozwala użytkownikowi narysować cyfrę na płótnie, a następnie po wciśnięciu przycisku 'Rozpoznaj' program klasyfikuje cyfrę za pomocą jednego z klasyfikatorów w celu rozpoznania narysowanej cyfry. Otrzymane wyniki klasyfikacji są wyświetlane w bloku po prawej stronie interfejsu użytkownika.



Rysunek 1: Wygląd aplikacji

Rysunek 2: Rozpoznawanie

1.2 Użyte biblioteki

Program korzysta z następujacych zewnętrznych bibliotek:
- Pillow
    - Do transformacji zapisanych cyfr na macierz
    - Do transformacji narysowanej cyfry na macierz
- numpy
- seaborn

1.3 Baza danych

Baza danych składa się z 1000 obrazów cyfr narysowanych przez nas (100 dla każdej cyfry). Każdy piksel obrazu jest reprezentowany w skali szarości (ma wartość od 0 do 255, gdzie 0 to biały, a 255 to czarny kolor). Obrazy są przechowywane w formacie png.

Rysunek 3: Przykładowy obraz z bazy danych

Rysunek 4: Ten sam obraz po zmianie rozdzielczości na 28x28 pikseli

2 Opis działania

2.1 Normalizacja danych

Piksele wczytanego obrazu są konwertowane na skalę szarości tj. 0 - kolor biały, 255 - kolor czarny oraz są normalizowane do przedziału od 0 do 1 co ułatwia modelowi dopasowanie cyfr. Wzór na normalizację pojedyńczego piksela:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

gdzie:

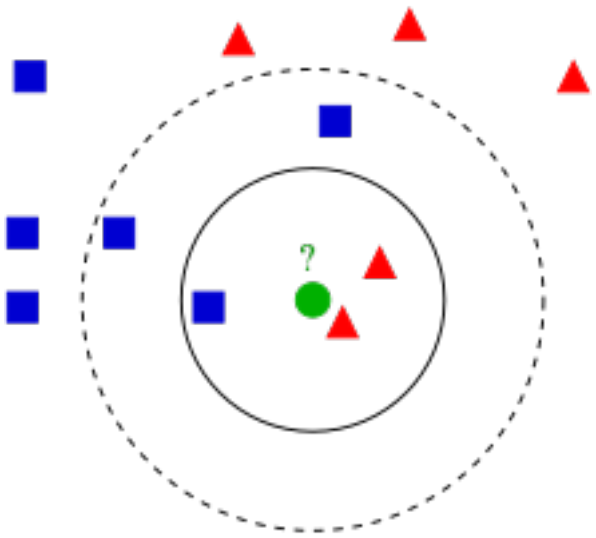
z_i - znormalizowany piksel
x_i - piksel
x - zbiór wszystkich pikseli

Ostatecznie wzór ma postać:

$$z_i = \frac{x_i}{255}$$

2.2 Algorytm k najbliższych sąsiadów

Klasyfikator kNN to jedna z ważniejszych nieparametrycznych metod klasyfikacji. W tej metodzie klasyfikowany obiekt przydzielamy do tej klasy, do której należy większość z k sąsiadów.



Rysunek 5: Przykład klasyfikacji metodą kNN

W przypadku k=3 (mniejszy okrąg), zielona kropka zostanie zakwalifikowana do czerwonych trójkątów. W przypadku k=5 (większy okrąg) - do niebieskich kwadratów.

2.3 Metryka odległości

Użyta została odległość Minkowskiego określona wzorem:

$$L_m(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

gdzie:

L_m - odległość między punktami x i y
x, y - punkty w przestrzeni n wymiarowej
x_i, y_i - i'ta współrzędna punktów x i y
p - parametr określający rodzaj metryki

Przetestowaliśmy skuteczność klasyfikatora kNN dla liczby sąsiadów $k \in \{1, 3, 5, 7, 9, 11, 13, 15\}$ oraz dla wartości parametru $p \in \{1, 2, 3\}$.

2.4 Pseudokod algorytmu kNN

Data: Dane wejściowe: zbiór treningowy *train_data* zbiór testowy *test_data* liczba sąsiadów k metryka p
Result: Zbiór testowy z przewidzianymi etykietami
foreach *test_instance in test_data* **do**
    distances = [];
    foreach *train_instance in train_data* **do**
        distance = *point_distance*(*test_instance*, *train_instance*, p);
        distances.append((*train_instance*, *distance*));
    end
    sorted_distances = *sort*(*distances*, *by* = *distance*);
    k_nearest_neighbors = *sorted_distances*[: k];
    test_instance.set_predicted_label(*predicted_label*);
end
return *test_data*;

Algorithm 1: Algorytm k najbliższych sąsiadów.

2.5 Implementacja

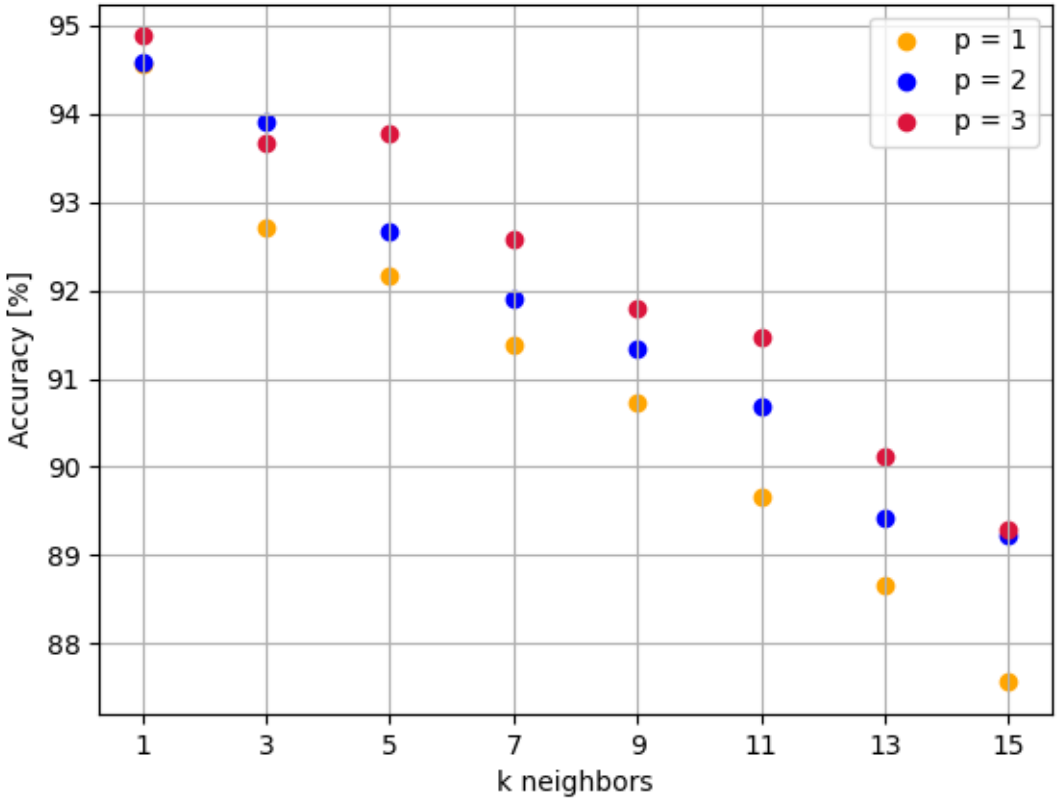
1. main.py - główny plik programu
2. knn.py - plik zawierający implementację algorytmu kNN
3. bayes.py - plik zawierający implementację naiwnego klasyfikatora Bayesa
4. utils.py - plik zawierający funkcje pomocnicze (np. wczytywanie danych)
5. test.py - plik zawierający funkcje testujące klasyfikatory

```
1  print("Hello world!")
```

2.6 Testy

Tutaj powinna pojawić się analiza uzyskanych wyników oraz wykresy/pomiary.

2.7 Eksperymenty



Rysunek 6: Zależność dokładności klasyfikacji od liczby sąsiadów k i parametru p

Najlepsze wyniki klasyfikator kNN uzyskuje dla $k = 1$ oraz $p = 3$, gdzie jego dokładność wynosi 94.9%.

3 Pełen kod aplikacji

```
1 Tutaj wklejamy pelen kod.
```