YetAnotherSpamFilter

Tomasz Kosmulski

1. Wyniki testowe i treningowe.

|  |  |  |
| --- | --- | --- |
|  | Train | Test |
| accuracy\_score | 0.9965 | 0.9872 |
| precision\_score | 0.9851 | 0.9751 |
| recall\_score | 1.0 | 0.9751 |
| roc\_auc\_scor | 0.9977 | 0.9832 |
| r2\_score | 0.9804 | 0.9328 |

2. Wybór techniki i modelu.

Jako preprocessing wiadomości wybrałem CountVectorizer. Umożliwia on na dalszym etapie klasyfikację na podstawie słownictwa użytego w wiadomości. Jako klasyfikator wybrałem rodzinę klasyfikatorów Bayes’owskich. W porównaniu z innymi klasyfikatorami, działają one lepiej dla danych dyskretnych, w szczególności dla wektora liczb naturalnych. Z tego samego powodu nie dokonuję skalowania danych wejściowych uzyskanych z preprocessingu.

3. Strategia podziału danych.

Dane testowe stanowią 30% zestawu danych. Na pozostałych danych dokonana jest cross-walidacja dla podziału na 5 zbiorów.

4. Opis danych wejściowych.

Dane wejściowe pochodzą z <https://www.kaggle.com/datasets/karthickveerakumar/spam-filter>

Zawierają 5695 wartości, z czego 1368 reprezentują spam.

5. Analiza wyników.

Model wykazuje zadowalające metryki jak na swoją prostotę. Model może być ulepszony w następujący sposób:

* Stworzenie Ensemble na większej liczbie modeli
* Utworzenie sieci neuronowej między preprocessing’iem a klasyfikatorem.

Ze względu na dodanie sieci neuronowej, możliwa będzie konieczność zmiany klasyfikatora.