Uczenie maszynowe w Python
Projekt zaliczeniowy

Oskar Legner - 24517
Jakub Macioszek - 24875

Spis treści

Kod źródłowy

Repozytorium GIT z kodem źródłowym projektu jest dostępne w serwisie GitHub:

https://github.com/Hardcoded-inc/machine-learning-author-recognition

Problem badawczy i opis projektu

Założeniem projektu był przeanalizowanie dataset'u tekstów piosenek i wytrenowanie modelu w celu predykcji wykonawcy.

W początkowej fazie budowania algorytmu procesującego dane, skorzystaliśmy z dataset'u (https://www.kaggle.com/neisse/scrapped-lyrics-from-6-genres) zawierającego również oetykietowanie gatunków poszczególnych piosenek.
Dzięki temu, już przy określaniu założeń projektu wzięliśmy pod uwagę "plan B": Wytrenowanie modelu w celu przewidywania gatunku, na podstawie tekstu piosenki - co wydaje się być nieco mniejszym wyzwaniem. Plan B dawał nam bezpieczeństwo, gdyby pierwotnie określony przez nas cel był zbyt trudny do osiągnięcia.

Z czasem zdaliśmy sobie jednak sprawę z dużego zanieczyszczenia wybranego dotasetu i sporej presji czasu. Finalnie w projekcie wykorzystaliśmy mniej zanieczyszczony dataset (https://www.kaggle.com/edenbd/150k-lyrics-labeled-with-spotify-valence), rezygnując jednak z etykietowania gatunków i decydując się na postawienie na jedną kartę - predykcję wykonawcy piosenki.
Spory wpływ na podjęcie decyzji o zmianie dataset'u miał fakt, iż pierwszy zawierał również piosenki w języku brazylijskim. Sprawiło nam to znaczne trudności przy podejściu do usuwania stopwords'ów oraz taggowania.

Procesowanie danych

Kroki podjęte w ramach przygotowania tekstów piosenek:

1. Zmiana wielkości wszystkich liter na małe.

2. Usunięcie wszelkiego tekstu wewnątrz nawiasów (zarówno okrągłych, jak i kwadratowych i klamrowych). Tego typu wartości wskazywały najczęściej refreny, zwrotki, czy czasem np. chórki.

3. Usunięcie rekordów zawierających klauzulę “written by”. Oczyszczenie tekstów z tego typu wtrąceń byłoby bardziej czasochłonnym zabiegiem - przez wzgląd na brak spójności w umieszczaniu klauzuli w tekstach), natomiast nie dającym zbyt wielkiej przewagi w finalnym wykorzystaniu danych. Klauzula “written by” występowała w niewielkiej grupie tekstów (zaledwie 141 wystąpień pośród 150 tyś rekordów: 0.094%).

4. Usunięcie interpunkcji

5. Usunięcie fraz zawierających liczby

6. Usunięcie pustych rekordów - które mogły powstać w wyniku powyższych działań. (Na przykład jeśli cały tekst był otoczony nawiasami, w drugim kroku pozostawilibyśmy pusty rekord)

7. Usunięcie słów z listy dropwords. Lista uzyskana przy pomocy biblioteki nltk.

8. Usunięcie najczęściej występujących słów

9. Usunięcie najrzadziej wystepujących słów

10. Stemming. Wykorzystaliśmy porterStemmer

11. Tokenizacja. W tym wypadku wykorzystaliśmy bibliotekę textBlob

Przeprocesowane dane zapisywane są do pliku "*checkpoint*.csv*" określonego w pliku *process_data.py*. Aby uruchomić algorytm, należy uruchomić plik *process_data.py* poprzez interpreter python.

Dostępne flagi:
-v: Tryb verbose
—full: Operuje na 10tyś rekordach. Nie ustawienie flagi oznacza działanie na 100 rekordach.

Powyższe wartości procesowanych danych wynikają bezpośrednio z wniosków Jake wyciągnęliśmy w fazie trenowania modelu.

Trenowanie modelu

Dla przeprocesowanych danych przygotowaliśmy algorytm trenowania modelu ML. Trening przebiega z użyciem przekazywanego w parametrze klasyfikatora, a słowa tekstów piosenek są odpowiednio wektoryzowane.

Aby uruchomić algorytm, należy uruchomić plik *train.py* poprzez interpreter python. Program wykorzystuje dane z pliku "*checkpoint*.csv*", ustawianego wewnątrz kodu. Trening przebiega kolejno z użyciem 4 różnych klasyfikatorów (kolejno: Regresja Logiczna, SVC, Random Forest, XGBClassifier), z wykorzystaniem różnych parametrów wektoryzacji.

Wyniki i analiza

Dokładność "wytrenowanego" modelu - niezależnie od użytego klasyfikatora - jest zbyt niska, aby można było mówić jakimkolwiek sukcesie. Wyniki wahają się w okolicach prawdopodobieństwa losowego trafienia poprawnej odpowiedzi.

Możliwym powodem może być zbyt mała ilość danych na jednego wykonawcę. Dataset zawiera 150 tyś rekordów, jednak na konkretnego artystę wypada średnio jedynie około 100 utworów (15 tyś wykonawców).

Po uzyskaniu pierwszych, dość niskich wyników, podjęliśmy próby dokładniejszego procesowania tekstu, jednak nie ukazało to praktycznie żadnej poprawy.

Również trafność przewidywań modelu spada wraz ze zwiększeniem ilości danych, wykorzystywanych w trakcie treningu. Wynika to prawdopodobnie z faktu iż nasze wyniki są przypadkowe, natomiast wraz ze zwiększeniem ilości danych - zwiększa się również ilość autorów, co obniża prawdopodobieństwo trafienia.