Michał Libura

Klasyfikator KLJ - dokumentacja

Spis treści

[Standaryzacja nazewnictwa zmiennych 3](#_Toc445838981)

[Opis preprocessingu 4](#_Toc445838982)

[Warunki logiczne 4](#_Toc445838983)

[Budowa klasyfikatora 5](#_Toc445838984)

# Standaryzacja nazewnictwa zmiennych

W celu zachowania spójności analiz zdecydowałem się ujednolicić nazewnictwo zmiennych, wykorzystywanych w późniejszych analizach. Nowe nazewnictwo zostało zastosowane we wszystkich bazach. Wprowadzono też kilka nowych zmiennych mojego autorstwa, których celem jest umożliwienie bardziej precyzyjnej klasyfikacji. Krótki opis zmiennych:

1. *Zdanie* – zmienna tekstowa, zawierająca surowe cytaty (uzasadnienia lub skojarzenia)
2. *Kategoria* – zmienna numeryczna, zawierająca id kategorii produktowej
3. *Kategoria\_name* – zmienna tekstowa, zawierająca nazwę kategorii produktowej
4. *Marka* – zmienna numeryczna, zawierająca id marki
5. *Marka\_name* – zmienna tekstowa, zawierająca nazwę marki
6. *Liczba\_znakow* – zmienna numeryczna, określająca ilość znaków w cytacie (po preprocessingu)
7. *Vulg* – zmienna binarna, określająca, czy w danym cytacie wystąpiło przynajmniej jedno słowo wulgarne (TRUE – tak, FALSE – nie)
8. *Jest\_Marka* – zmienna binarna, określająca, czy w danym cytacie wystąpił przynajmniej jeden element nazwy marki (np. samo ING z ING Bank Śląski), do której cytat się odnosi (TRUE – tak, FALSE – nie)
9. *Lematy* – zmienna tekstowa, zawierająca poddane przetworzone i zlematyzowane cytaty (słowa niemożliwe do zlematyzowania pozostają w niezlematyzowanej formie)
10. *Tagi* – zmienna tekstowa, zawierająca tagi gramatyczne (słowa niemożliwe do otagowania zostają pominięte)
11. *Long* – zmienna binarna, identyfikująca cytaty, które są pojedynczym ciągiem znaków nierozdzielonych spacją (TRUE – tak, FALSE – nie)
12. *Flaga* – zmienna binarna, identyfikująca, czy w danym cytacie któreś słowo (przynajmniej jedno) nie mogło zostać zlematyzowane (TRUE – tak, FALSE – nie)
13. *Min\_One\_Lem* – zmienna binarna, określająca, czy w danym cytacie pojawiło się przynajmniej jedno możliwe do zlematyzowania słowo (TRUE – tak, FALSE – nie)

# Opis preprocessingu

W celu zwiększenia porównywalności poszczególnych cytatów (co jest warunkiem koniecznym dla budowy skutecznego klasyfikatora), surowe dane tekstowe w postaci cytatów zostały przetworzone. Lista wykonanych operacji:

1. Zastąpienie polskich znaków diakrytycznych ich łacińskimi odpowiednikami (np. ą na a, format „latin-ascii”)
2. Zamiana wszystkich liter na małe
3. Usunięcie znaków interpunkcyjnych
4. Usunięcie liczb
5. Usunięcie słów powszednich (ja, ty itd.)
6. Usunięcie zbędnej przestrzeni (spacji)

Na tak przetworzonym zbiorze danych tekstowych przeprowadzana jest lematyzacja oraz tagowanie gramatyczne w oparciu o słownik fleksyjny Polimorf (<http://zil.ipipan.waw.pl/PoliMorf>). Tagi gramatyczne sprowadzone zostają do najbardziej podstawowych form (np. rzeczownik, czasownik itd.).

# Warunki logiczne

W bazach danych znajduje się wiele obserwacji, które już na tym etapie można wykluczyć z dalszej analizy pod kątem przydatności do raportu. Stworzyłem kilka warunków logicznych, które eliminują niektóre obserwacje. Wyglądają one następująco:

1. Wykluczone zostają wszystkie „puste” obserwacje (Liczba\_znakow == 0)
2. Wykluczone zostają wszystkie obserwacje, w których pojawił się przynajmniej jeden wulgaryzm (Vulg == TRUE)
3. Wykluczone zostają wszystkie obserwacje, w których nie pojawiło się przynajmniej jedno słowo możliwe do zlematyzowania (Min\_One\_Lem == FALSE)

# Budowa klasyfikatora

Zlematyzowane i otagowane gramatycznie dane tekstowe przetworzone zostały do postaci wektorowej, która stanowi podstawę późniejszych analiz. Próba ważenia tf-idf nie wpłynęła pozytywnie na jakość klasyfikatora, więc zdecydowałem się pozostać przy surowej liczbie wystąpień słów.

Tagi gramatyczne analizowane były zarówno w formie prostej, jak i jako bi-gramy. Porównanie modeli budowanych w oparciu o obie formy wykazało, że reprezentacja bi-gramowa powodowała w tym przypadku mniejszą dokładność predykcyjną modeli, dlatego postanowiłem pozostać przy reprezentacji prostej.

W procesie budowy klasyfikatora testowałem różne techniki uczenia statystycznego, które następnie były porównywane ze sobą pod kątem ich trafności w przewidywaniu zmiennej zależnej (zmienna Wynik) na zbiorze testowym. Modele trenowane były na danych z roku 2014, natomiast testowane na danych z roku 2015. Kryterium porównawczym, mierzącym „zdolność predykcyjną” danego modelu, została miara AUC (*Area Under Curve*) – pole pod krzywą ROC (*Receiver Operating Characteristic curve*). Krzywa ROC obrazuje relacje między stosunkiem wartości pozytywnych (Wynik == TRUE) poprawnie zakwalifikowanych przez model (*True Positive Rate*) a stosunkiem wartości negatywnych (Wynik == FALSE), błędnie zakwalifikowanych jako pozytywne przez model dla danego progu (*treshold)*. Próg oznacza tutaj wartość graniczną, która decyduje o zakwalifikowaniu danego przypadku jako „dobry cytat” lub „ zły cytat”. AUC przyjmuje wartości od 0 do 1, interpretowane jest jako prawdopodobieństwo, że wybrany losowo z populacji „dobry cytat” zostanie wyżej zakwalifikowany (przypisane mu zostanie wyższe prawdopodobieństwo przynależenia do klasy „dobrych cytatów”) niż losowo wybrany „zły cytat”[[1]](#footnote-1). Najgorszym możliwym wynikiem jest zatem AUC równe 0.5 – w takiej sytuacji trafność przewidywania danego modelu jest tożsama z trafnością, jaką osiągnęlibyśmy przypisując obserwacjom klasy wynikowe w sposób losowy (np. przez rzut monetą). Im wartość jest bliższa 0 lub 1, tym doskonalszy jest model predykcyjny (w przypadku wartości poniżej 0.5 należy po prostu odwrócić predykcje). Zdecydowałem się na wybór tego kryterium, ponieważ rozkład zmiennej zależnej (który w tym przypadku bardzo nierówny, gdyż „dobre cytaty” stanową bardzo niewielką frakcję populacji) nie ma wpływu na wartości przyjmowane przez AUC – jest to miara często stosowana przy ocenie modeli, mających przewidywać zajście jakiegoś rzadkiego zjawiska (vide[[2]](#footnote-2)). Modele trenowane były osobno dla danych dotyczących skojarzeń i danych dotyczących uzasadnień.

W procesie budowy klasyfikatora testowałem następujące metody:

1. Regresja logistyczna
2. Regresja logistyczna karana typu Lasso
3. Drzewa decyzyjne (CART)
4. Support Vector Machines
5. Naiwny Klasyfikator Bayesowski

Ze względu na obiecujące wyniki (kryterium – AUC), łatwość interpretacji, szybkość działania i małe wymagania w zakresie mocy obliczeniowych komputera zdecydowałem się skupić na metodzie regresji logistycznej typu Lasso. Metoda ta, dzięki swojej wydajności, mogła być trenowana na całym zbiorze treningowym, zawierającym znaczną cześć zmiennych – z analizy wykluczono te słowa i tagi gramatyczne, które pojawiły się w mniej niż jednym promilu wypowiedzi respondentów. Dodatkową zaletą regresji typu Lasso jest fakt, iż w tym modelu wartości niektórych ze współczynników regresji zostają sprowadzone do zera, co oznacza automatyczną selekcję istotnych zmiennych.

W przypadku pozostałych modeli analiza odbywała się na mniejszym zakresie zmiennych – zachowano słowa i tagi, które pojawiały się przynajmniej w jednym procencie wypowiedzi. Następnie jeszcze bardziej zawężono ten zbiór zmiennych poprzez zastosowanie algorytmu Boruta, służącego do selekcji istotnych (dla predykcji) zmiennych niezależnych (w oparciu o bootstrap).

1. http://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it [↑](#footnote-ref-1)
2. http://www.jmlr.org/proceedings/papers/v7/miller09/miller09.pdf [↑](#footnote-ref-2)