Klasyfikacja kategorii filmów na youtube w zależności od właściwości filmu

MSID Lab środa 15:15TP

Igor Wojciech Banaszak[1]

4 czerwca, 2022

1. Wstęp

Problemem wybranym do badań jest zależność klasyfikacji z kategorii filmów na youtubie od ich właściwości. Inspiracją do tego rodzaju rozważań było pytanie, w jaki sposób algorytm youtube klasyfikuje filmy. Celem Projektu jest zbadanie kategorii od dostępnych parametrów:

- Ilość wyświetleń
- Ilość komentarzy
- Ilość polubień
- Czy dla dzieci
- Czas trwania
- Data dodania
- Godzina dodania

oraz stworzenie modelu przewidującego kategorię z podanych powyżej parametrów.

2. Zbiór danych i jego przetwarzanie

2.1. Zbiór danych

Zbiór wykorzystany w pracy został zescrapowny za pomocą youyube Data api v3, które dostarcza nam youtube, proces ten został podzielony na:

- Pobieranie id filmów z podanego kanału[1]
- Pobieranie informacji o filmie z podanego id filmu[2]

Metoda z wybieraniem kanału została wybrana po nieudanej próbie z użyciem szukania filmów [3]. Metoda ta polega na szukaniu przez api filmów, które spełniają podane do metody filtry. Próba ta nie powiodła się z przyczyny zbyt dużego zużycia limitu, który nakłada youyube api v3. Kanały użyte do metody z kanałami zostały wybrane ręcznie (28 kanałów), co pozwoliło na zebranie 26965 rekordów.

Spośród wszystkich kategorii jakie, dostarcza nam youtube, wybrałem do projektu pięć najpopularniejszych kategorii:

- 2 - Motoryzacja
- 10 - Muzyka
- 17 - Sport
- 24 - Rozrywka
- 25 - Wiadomości i polityka

Po zescrapowaniu otrzymujemy zbiory, które łączymy,[4] aby utworzyć jeden zbiór, który składa się z poniższych atrybutów:

- Id filmu
- Opublikowany o
- Ilość wyświetleń
- Ilość komentarzy
- Ilość polubień
- Czy dla dzieci
- Czas trwania
- Id kategorii

2.2. Przetwarzanie wstępne

Podczas przetwarzania zbioru danych zostały usunięte rekordy [5] dla innych kategorii niż wybrane 2.1.

Atrybut "Opublikowany o"został podzielony na liczbę dni po 01.01.2012 oraz na minuty od północy.[6]

Brakujące dane w ilości komenatrzy zostały uzupełnione o wartość 0, ponieważ komentarze w tych filamach zostały wyłączone.

[1] ScrapChanel.py
[2] ScrapDetails.py
[3] ScrapSearch.py

[4] merge.py
[5] Select_category.py
[6] Select_category.py

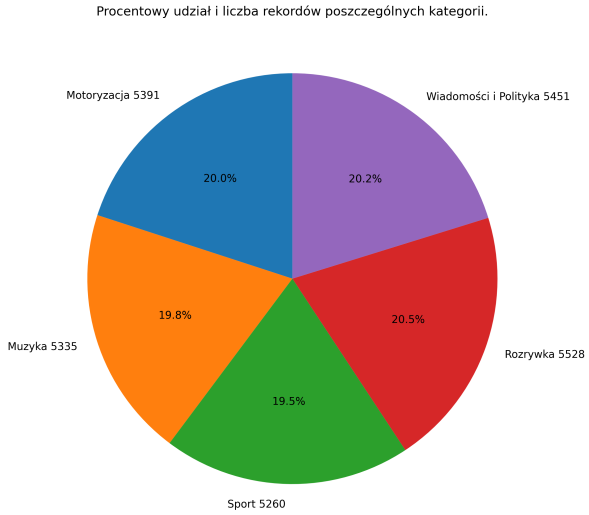
Brakujące dane w ilości polubień zostały uzupełnione o wartość średnią z danej kategorii dla filmów znajdujących się w widełkach [95% ; 105%] pod względem ilości wyświetleń.

Rekordy z brakującymi danymi w ilości wyświetleń zostały usunięte, ponieważ był tylko jeden taki.

Czas trwania zmieniamy z ISO 8601 na sekundy.

3. Wstępna analiza danych

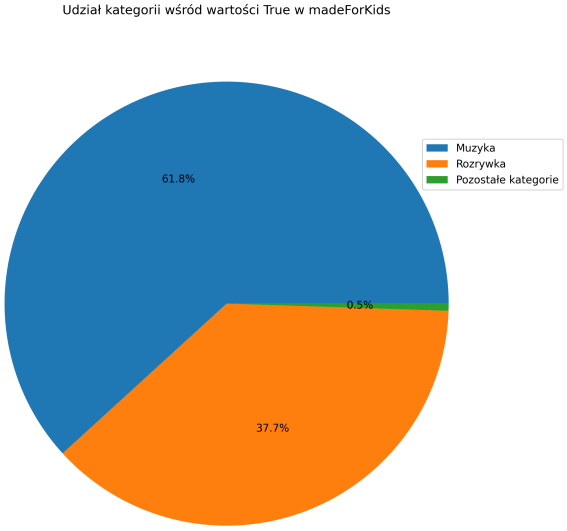
Po wstępnym oczyszczeniu naszych danych mamy następujący stosunek poszczególnych kategorii do wszystkich rekordów:[7]



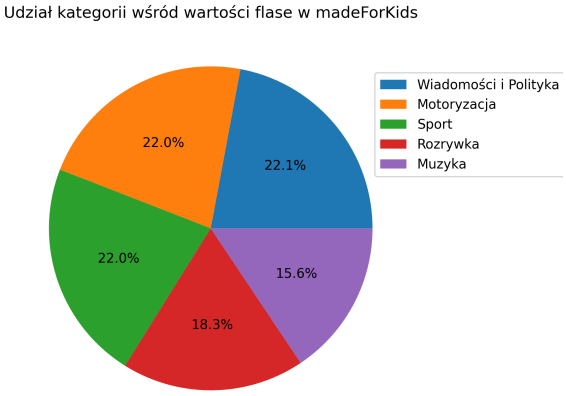
Jak widzimy na diagramie, kategorie są prawie w identycznych proporcjach, co pozwoli na lepsze wytrenowanie modeli

Sprawdzmy teraz zależność między poszczególnymi atrybutami a kategoriami:

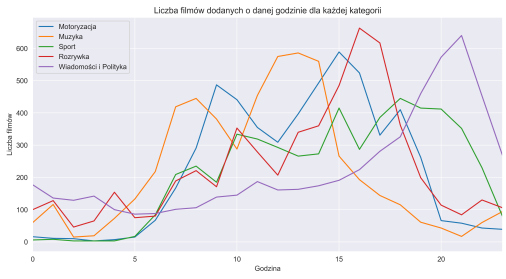
[7] classification.ipynb



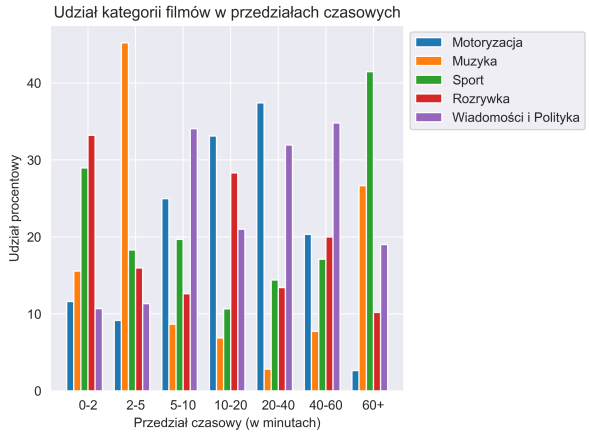
Widzimy, że wartość true w atrybucie "Czy dla dzieci"pozwala nam w 99.5% określić, że będą to dwie kategorie z pięciu, dokładniej mamy 61.8% szans, że będzie to muzyka i 37.7%, że będzie to rozrywka.



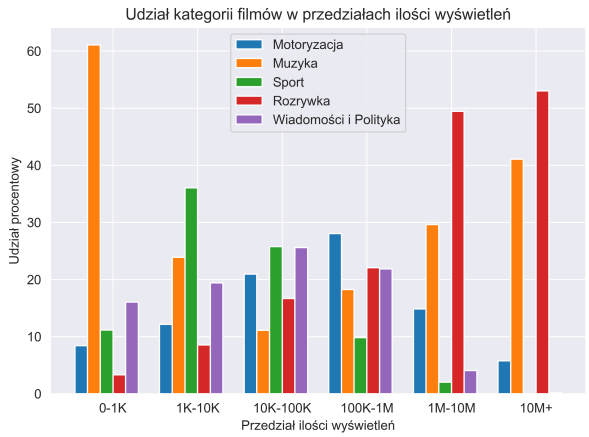
Niestety dla wartości false widzimy, że nie ma tak znacznych różnic pomiędzy kategoriami.



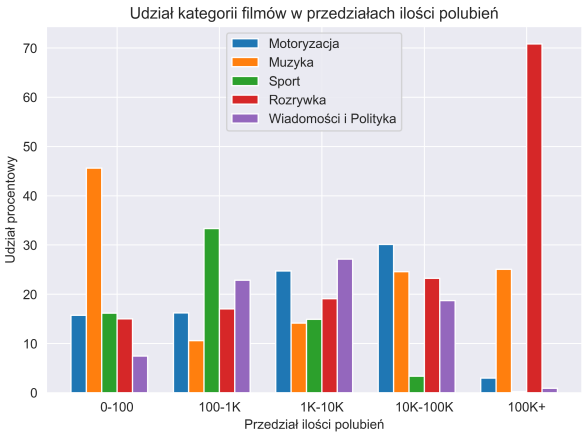
Ten diagram bardzo dużo nam mówi o kategoriach, możemy się z niego dowiedzieć, że filmy z kategorii "Wiadomości i polityka"wrzucane są zazwyczaj w okolicach godziny 21. Natomiast kategoria sport nie ma jednej godziny, w której najczęściej wrzucane są filmy. Pozostałe trzy kategorie mają bardzo podobne wykresy z delikatnym przesunięciem.



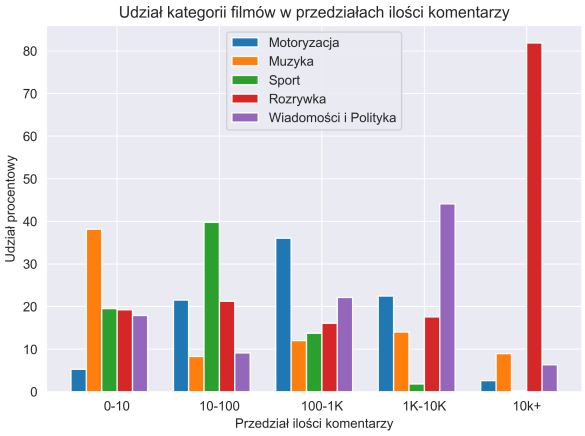
Wykres przedstawia udział poszczególnych kategorii w konkretnych przedziałach czasowych, co pozwala nam na przykład określić, że jeśli film trwa pomiędzy 2-5 minut, to na 45% będzie to film z kategorii muzyka.



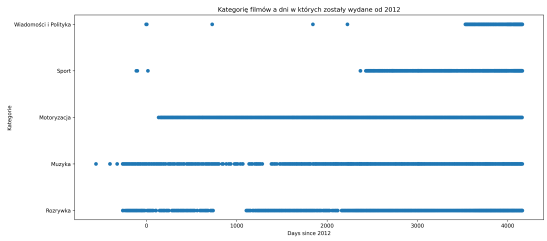
Następny diagram przedstawia zależność między kategoriami, a ilością wyświetleń w przedziałach. Widzimy, że środkowe przedziały dużo nam nie mówią, bo wszystkie kategorie mają podobną ilość wyświetleń, natomiast skrajne, takie jak 0-1k pozwalają nam dostrzec, że za 60% wyświetleń w tym przedziale odpowiada kategoria muzyka lub, że w przedziale 10M+ nie występują kategorie Wiadomości i polityka oraz sport, a motoryzacja odpowiada tylko za 5%.



Ten diagram jest bardzo podobny do poprzedniego, ponieważ tutaj również nie widzimy dużych różnic pomiędzy kategoriami w środkowych przedziałach. Natomiast skrajne kategorie również dostarczają nam bardzo zróżnicowanych wyników, takich jak 70% udziału kategorii rozrywka dla ilości polubień powyżej 100000.



Przedostatni diagram przedstawiający zależność między atrybutami a kategoriami jest bardzo ciekawy, ponieważ tutaj w każdym przedziale inna kategoria dominuje o ponad 15%. Jednak tak jak w dwóch ostatnich diagramach widać największą różnicę między kategoriami w górnym skrajnym przedziale, tak w tym przypadku jest to przedział ponad 10000, w którym kategoria rozrywka ma aż 80% udziału.



Ostatni diagram przedstawia nam zależność między opublikowaniem filmu a kategorią. Możemy tutaj dostrzec, że zgromadzony przez nas zbiór nie jest idealny. Kategorie powinny się pokrywać, ponieważ codziennie są publikowane filmy z tych kategorii. Dlatego do uczenia naszego modelu nie będziemy stosować tego atrybutu, tak aby nasz model był możliwy do wykorzystania dla najnowszych filmów i jak najlepiej działał na innych zbiorach.

4. Przygotowanie modelu

4.1. DecisionTreeClassifier

DecisionTreeClassifier jest algorytmem uczenia maszynowego, który opiera się na drzewach decyzyjnych do problemów klasyfikacji. Jest to jeden z najprostszych i najbardziej intuicyjnych algorytmów uczenia maszynowego.

Drzewo decyzyjne składa się z węzłów i krawędzi, które reprezentują decyzje oparte na wartościach cech. Algorytm uczący DecisionTreeClassifier tworzy drzewo decyzyjne na podstawie dostępnych danych treningowych, które składają się z przykładów oznaczonych etykietami klas.

Algorytm działa następująco:

1. Wybór najlepszej cechy, która podzieli zbiór danych na sposób, który najlepiej separuje przykłady klas.
2. Podział danych na podstawie wartości wybranej cechy, tworząc węzeł decyzyjny.
3. Powtarzanie kroków 1-2 rekurencyjnie dla każdego nowo utworzonego węzła, aż zostaną spełnione pewne kryteria stopu.
4. Przypisanie etykiety klasy do liści drzewa na podstawie większościowych etykiet przykładów uczących w danym liściu.

4.2. RandomForestClassifier

RandomForestClassifier jest algorytmem uczenia maszynowego, który opiera się na zasadzie ensemble learning, czyli łączeniu wyników wielu modeli w celu uzyskania lepszej jakości predykcji. Jest oparty na metodzie drzew decyzyjnych.

RandomForestClassifier tworzy wiele drzew decyzyjnych na podstawie losowych podzbiorów danych treningowych. Każde drzewo jest trenowane niezależnie na różnych podzbiorach danych. Podczas prognozowania klasyfikacji każde drzewo decyzyjne w lesie przewiduje wynik, a ostateczna predykcja jest dokonywana na podstawie głosowania większości.

Ważną cechą RandomForestClassifier jest wprowadzenie losowości poprzez losowe wybieranie podzbiorów cech do trenowania każdego drzewa. Dzięki temu zapobiega się overfittingowi (przeuczeniu), gdy model zbytnio dopasowuje się do danych treningowych.

Dodatkowo, podczas trenowania każdego drzewa, stosuje się losowanie ze zwracaniem, co oznacza, że każdy podzbiór danych ma możliwość zawierać duplikaty próbek. Ta technika znana jako bagging (bootstrap aggregating) pomaga zwiększyć różnorodność drzew w lesie i zmniejszyć wariancję modelu.

4.3. Przygotowanie modelu dla naszego zbioru

Nasze dane podzielimy na zbiór treningowy i testowy w stosunku 1:3, co pozwoli na odpowiednie nauczenie modelu.

Po wytrenowaniu naszego modelu oraz wytestowaniu go otrzymujemy następujące wyniki:

Tabela 1: Wyniki F1-score dla różnych modeli

Model	F1-score			
	Motoryzacja	Muzyka	Sport	Rozrywka
DTC	0.67	0.75	0.69	0.68
RFC	0.76	0.84	0.79	0.79

Model	F1-score			
	WiP	Dokładność	Arytmetyczna	Ważona
DTC	0.85	0.73	0.73	0.73
RFC	0.9	0.82	0.82	0.82

WiP[8]

F1-score jest miarą oceny jakości klasyfikacji. Jest to średnia harmoniczna precyzji (precision) i czułości (recall) klasyfikatora. Czułość to miara, która mierzy zdolność klasyfikatora do wykrywania rzeczywiście pozytywnych przypadków, podczas gdy precyzja ocenia, ile zidentyfikowanych jako pozytywne przypadków jest faktycznie poprawnych.

Dokładność (accuracy) to miara, która mierzy ogólną skuteczność klasyfikatora poprzez porównanie liczby poprawnie sklasyfikowanych przypadków do całkowitej liczby przypadków. Średnia arytmetyczna (macro avg) to miara dla każdej klasy, niezależnie od rozmiaru klasy. Średnia ważona (weighted avg) to miara dla każdej klasy, z wagami pro-

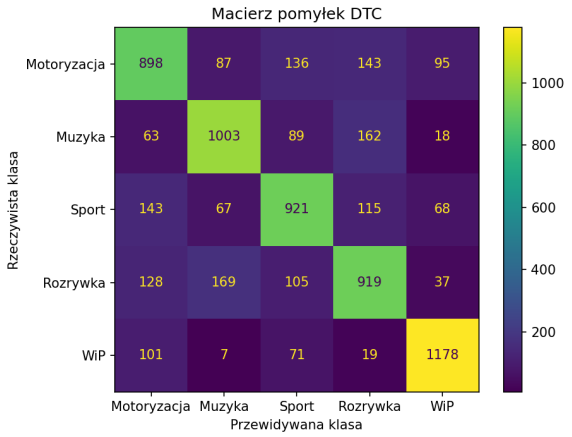
[8] Wiadmości i polityka

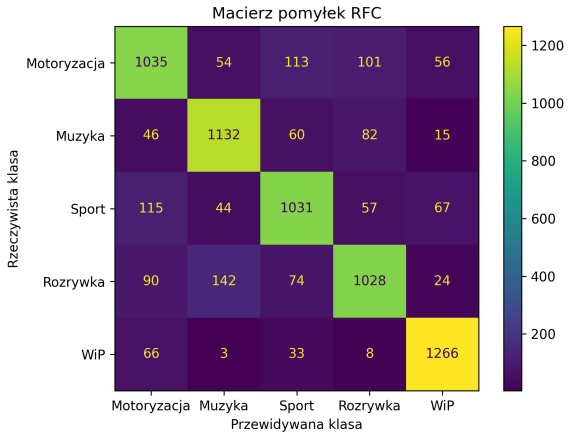
porcjonalnymi do liczności klas. Ponieważ w zbiorze mamy bardzo zbliżoną liczbę rekordów poszczególnych kategorii, dlatego wyniki w tych polach są w przybliżeniu taki same.

Jak widzimy w tabelce nasze modele najlepiej radzą sobie w przewidywaniu wiadomości i polityki oraz nieco gorzej radzi sobie z muzyką.

Ogólnie również możemy zaobserwować, że model RFC radzi sobie lepiej aż o 9 punktów procentowych niż model DTC, który uzyskuje wynik na poziomie 73%.

Nasze dane możemy również zaprezentować za pomocą macierzy pomyłek:





Na macierzy pomyłek możemy zobaczyć dokładnie ile razy model przewidywał jakąś kategorię, a jaka była naprawdę.

Warto sprawdzić czy, nasz model nie poradzi sobie lepiej bez jakiegoś atrybutu przy klasyfikacji:

Model	Bez	F1-score					Dokładność
		Motoryzacja	Muzyka	Sport	Rozrywka	Wip	
DTC	Liczba Wyświetleń	0.62	0.73	0.63	0.66	0.78	0.69
	Liczba komentarzy	0.58	0.74	0.61	0.65	0.62	0.64
	Liczba polubień	0.59	0.72	0.63	0.67	0.83	0.69
	Czy dla dzieci	0.67	0.73	0.69	0.67	0.83	0.72
	Czas trwania	0.57	0.69	0.63	0.59	0.81	0.66
	Minuty po północy	0.61	0.71	0.66	0.62	0.82	0.68
RFC	Liczba Wyświetleń	0.73	0.81	0.72	0.75	0.85	0.77
	Liczba komentarzy	0.69	0.8	0.71	0.75	0.71	0.73
	Liczba polubień	0.68	0.79	0.7	0.76	0.89	0.77
	Czy dla dzieci	0.76	0.82	0.78	0.77	0.89	0.8
	Czas trwania	0.68	0.75	0.73	0.66	0.87	0.74
	Minuty po północy	0.69	0.79	0.73	0.71	0.88	0.76

Z tabelki możemy odczytać, że każdy z argumentów polepsza naszą klasyfikację, nawet "czy dla dzieci", chociaż są to tylko 2 punkty procentowe, ale bardzo dla nas ważne.

5. Wnioski

Przedstawione eksperymenty wskazały, że nasze modele bardzo dobrze sobie radzą z klasyfikacją kategorii od parametrów filmów. Raport wskazuje również na zależność między wszystkimi atrybutami a kategorią.

Warto również zauważyć, że nasze badania uwzględniały różne metody klasyfikacji, takie jak Random Forest Classifier 4.2 oraz Decision Tree Classifier 4.1. Porównując wyniki obu modeli, stwierdziliśmy, że model RFC radzi sobie znacznie lepiej niż model DTC. Różnica w wynikach między nimi wynosi aż 9 punktów procentowych, co sugeruje, że model RFC jest bardziej efektywny w klasyfikacji kategorii filmów na podstawie parametrów, bo otrzymujemy wynik klasyfikacji na poziomie 82%.

Na podstawie tych obserwacji, możemy wnioskować, że nasz model ma potencjał do dalszego rozwoju i zastosowania w przyszłych badaniach. Może być również używany do klasyfikacji innych zbiorów danych związanych z YouTube.