Raport MSID

Katsiaryna Viarenich

June 2023

Treść

1 Opis problemu

1.1 Opis problemu:

Problemem, który ma być rozwiązany, jest przewidywanie oceny filmu na stronie TMDB (The Movie Database) na podstawie różnych kolumn zawartych w zbiorze danych. Kolumny te obejmują takie informacje jak:

1. `adult` (czy film jest dla dorosłych),
2. `backdrop path` (ścieżka do tła filmu),
3. `belongs to collection` (informacja o przynależności do kolekcji),
4. `budget` (budżet filmu),
5. `genres` (gatunki filmu),
6. `homepage` (strona domowa filmu),
7. `id` (identyfikator filmu),
8. `imdb id` (identyfikator filmu w bazie IMDb),
9. `original language` (oryginalny język filmu),
10. `original title` (oryginalny tytuł filmu),
11. `overview` (opis filmu),
12. `popularity` (popularność filmu),
13. `poster path` (ścieżka do plakatu filmu),
14. `production companies` (firmy produkcyjne),
15. `production countries` (kraje produkcji),
16. `release date` (data premiery),
17. `revenue` (dochody z filmu),
18. `runtime` (czas trwania),
19. `spoken languages` (używane języki),
20. `status` (status filmu),
21. `tagline` (slogan filmu),
22. `title` (tytuł filmu),
23. `video` (czy film ma zwiastun).

1.2 Wartość tego problemu:

Przewidywanie oceny filmu na stronie TMDB ma wiele potencjalnych zastosowań i wartości. Oto kilka powodów, dla których jest to warte uwagi:

1. Pomoc w podejmowaniu decyzji:

   Przewidywanie oceny filmu może pomóc osobom decydującym o dystrybucji filmów, inwestorom lub twórcom w ocenie potencjalnego sukcesu filmu na tej platformie. Na podstawie przewidywanej oceny mogą podejmować decyzje dotyczące budżetu reklamy, dystrybucji kinowej lub wyboru rynków docelowych.

2. Udoskonalenie systemu rankingowego:

   Wysoka jakość przewidywania oceny filmu może pomóc w doskonaleniu systemu rankingowego TMDB. Uwzględnienie przewidywanej oceny w algorytmach rankingu może poprawić trafność wyników i pomóc użytkownikom w znalezieniu filmów, które najbardziej ich zainteresują.

3. Badania rynkowe i analiza trendów:

   Przewidywanie oceny filmu może być wykorzystane do przeprowadzania badań rynkowych i analizy trendów wśród filmów. Może to dostarczyć cennych informacji na temat preferencji i gustów widzów, co może być wykorzystane do planowania strategii marketingowych lub tworzenia nowych produkcji.

1.3 Wnioski:

Przewidywanie oceny filmu na stronie TMDB ma wiele korzyści i wartości zarówno dla użytkowników, jak i dla branży filmowej. Dzięki dokładnym przewidywaniom oceny, można dostarczać rekomendacje filmów, wspierać proces podejmowania decyzji dotyczących dystrybucji filmów oraz prowadzić badania rynkowe i analizę trendów.

2 Zebranie danych

2.1 Opis procesu pobierania danych:

W celu pobrania danych, został stworzony kod, który wykorzystuje interfejs API TMDB (The Movie Database) do pobierania informacji o filmach. Poniżej znajdują się kluczowe punkty tego kodu:

1. **Funkcja fetch_movie_data(num_pages):**

   Funkcja "fetch_movie_data" jest odpowiedzialna za pobranie podanej liczby stron z filmami z API TMDB. Wykorzystuje pętlę "for" w zakresie od 1 do num_pages, aby pobrać dane z kolejnych stron API. Tworzony jest URL z uwzględnieniem numeru strony i sortowania według popularności.

Następnie, za pomocą żądania GET, dane są pobierane w formacie JSON i dodawane do obiektu DataFrame. Funkcja zwraca finalny DataFrame zawierający wszystkie pobrane filmy.

2. **Funkcja enrich_movie_data(movie_df):**

   Funkcja "enrich_movie_data" służy do wzbogacenia pierwotnego DataFrame (df) o dodatkowe informacje o filmach. Dla każdego identyfikatora (id) filmu z pierwotnego DataFrame, generowany jest odpowiedni URL, aby pobrać szczegóły filmu z API TMDB. Dane są pobierane w formacie JSON i dodawane jako nowy wiersz do obiektu DataFrame "movie_info_df". Dodatkowo, funkcja kontroluje częstotliwość żądań, używając zmiennej "request_counter" oraz opóźnienia sleep(1) co 10 żądań, aby uniknąć zbyt częstego wysyłania żądań do API.

   Przez porównanie kolumn z pierwotnego DataFrame i "movie_info_df", wyklucza się powtarzające się kolumny. Następnie, wykluczone kolumny zostają usunięte z pierwotnego DataFrame (df), a utworzony DataFrame (movie_info_df) jest dołączany do niego za pomocą funkcji concat(). Ostatecznie, przetworzony DataFrame zostaje zapisany do pliku CSV o nazwie "movies.csv".

W ten sposób, kod pobiera dane o filmach z API TMDB, wzbogaca je o dodatkowe szczegóły i zapisuje do pliku CSV.

3 Wstępna analiza danych

3.1 Kolumny, które nie zawierają przydatnych informacji

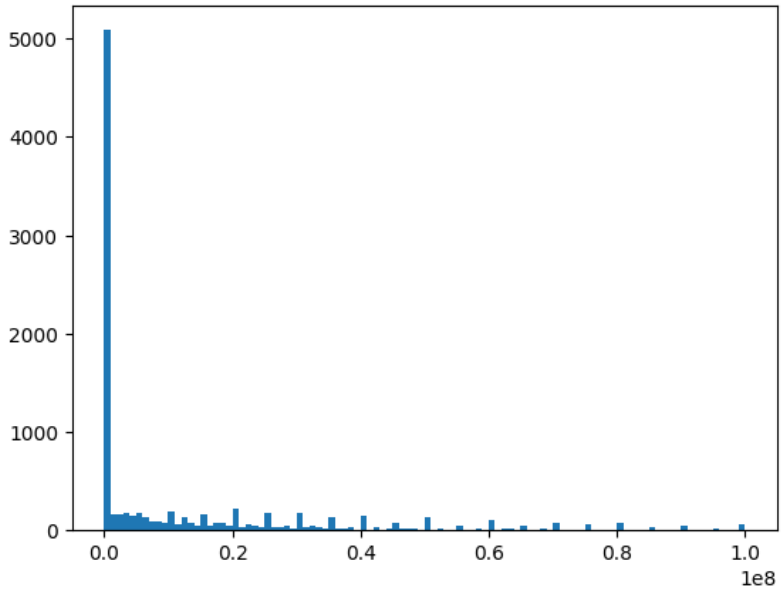
Przeglądając kolumny tabeli, można zauważyć, że niektóre z nich nie posiadają użytecznych informacji.

1. Kolumny "backdrop path" i "poster path" przechowują ścieżki do obrazów, jednak analiza obrazów nie będzie wykorzystywana w tym projekcie.
2. Kolumny "id" i "tmdb id" zawierają indeksy, które nie będą używane.
3. Pary kolumn "genres" i "genres id", "original title" i "title" są duplikatami.
4. Kolumny "adult" i "video" zawierają jedynie wartości False.

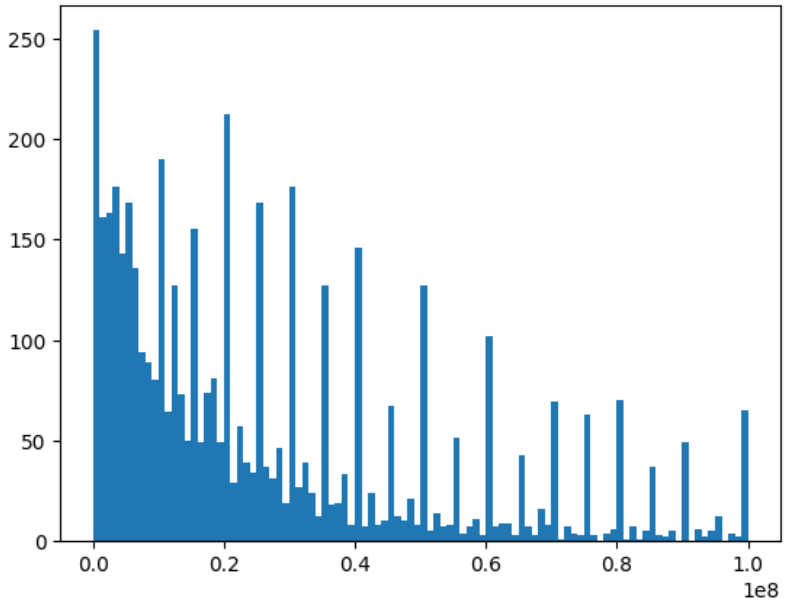
3.2 Numeryczne kolumny

3.2.1 Budżet

Jedną z kolumn numerycznych jest budżet. Przyjrzyjmy się dystrybucji wartości:



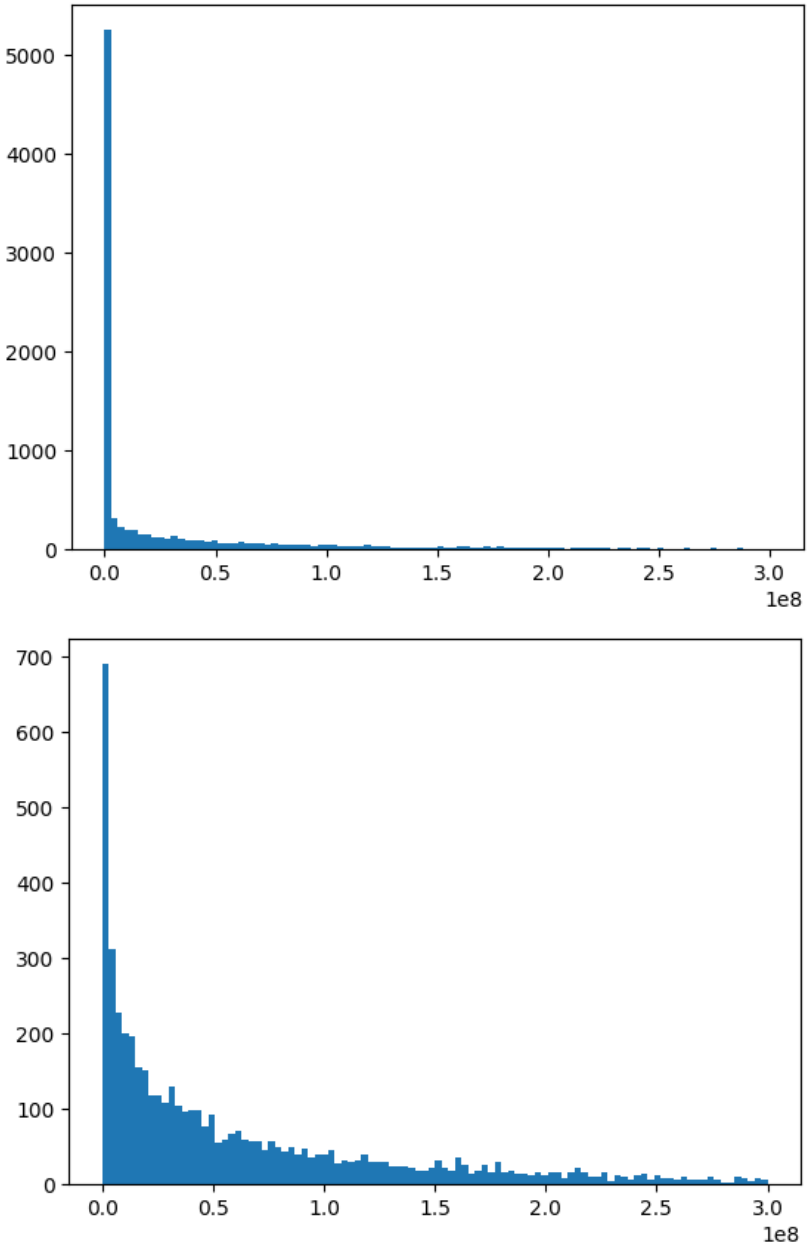
Duża liczba zer utrudnia analizę wykresu. Dlatego warto stworzyć wykres bez zer:



Późniejsze eksperymenty wykazały, że warto pozostawić kolumny z zerowym budżetem jako zera, zamiast zastępować je średnią wartością. Najprawdopodobniej filmy o zerowym budżecie rzeczywiście miały bardzo niskie budżety.

3.2.2 Przychody

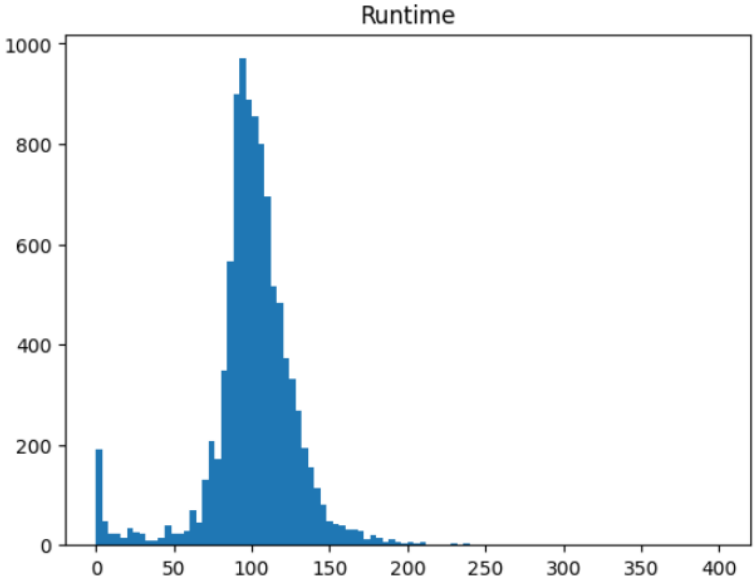
Przeprowadzono podobną analizę dla kolumny przychodów.

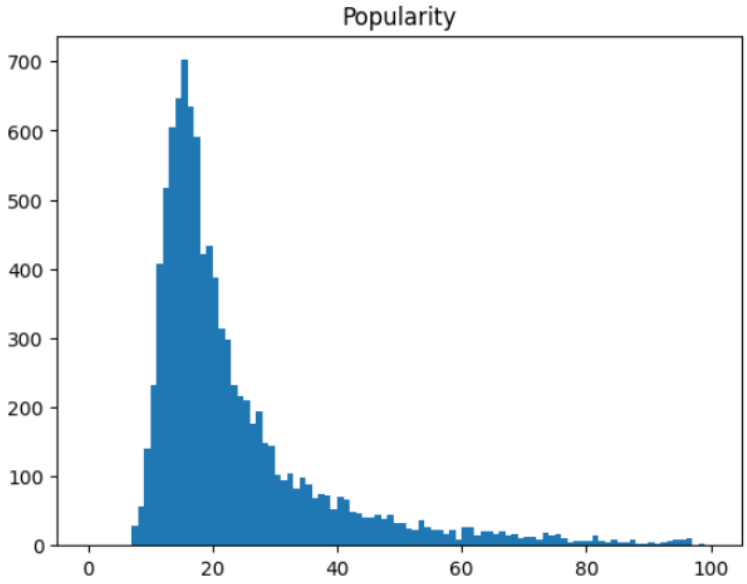


Eksperymenty wykazały, że w tym przypadku wartości zerowe również najlepiej pozostawić jako zera.

3.2.3 Czas trwania i popularność

Kolumny czasu trwania i popularności zostały poddane analogicznej analizie.

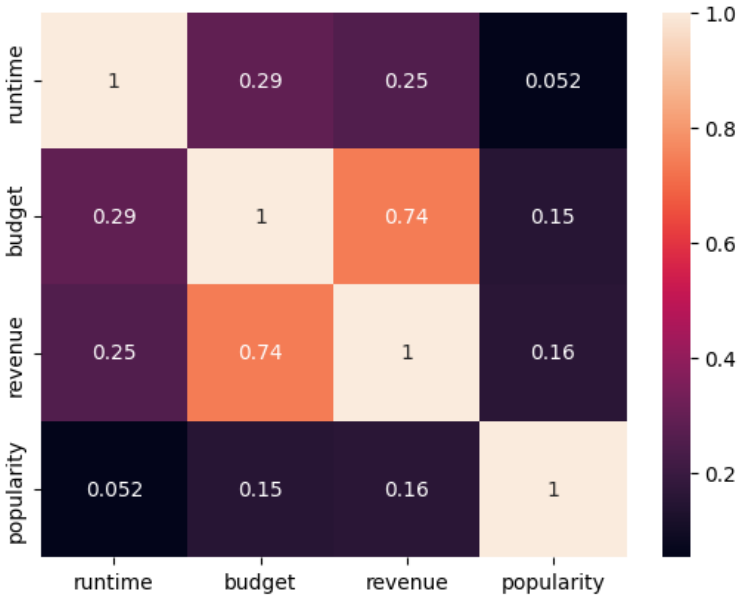




Widać, że rozkłady w tych przypadkach są dość ładne.

3.2.4 Macierz korelacji

W celu lepszej analizy warto również sprawdzić macierz korelacji.



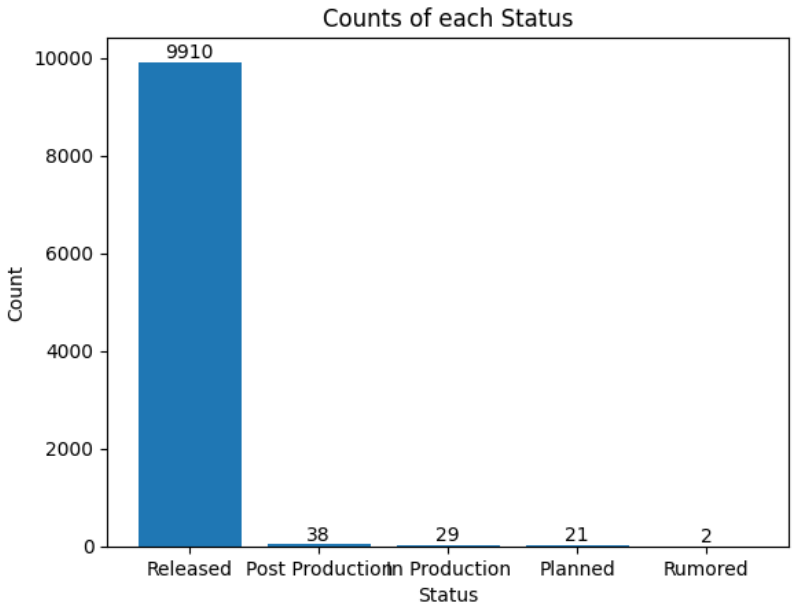
Widać, że istnieje dość duża korelacja między wartościami budżetu i przychodów. Ogólnie wartości korelacji są odpowiednie i zgadzają się z oczekiwanymi zależnościami.

3.3 Kolumny kategorii

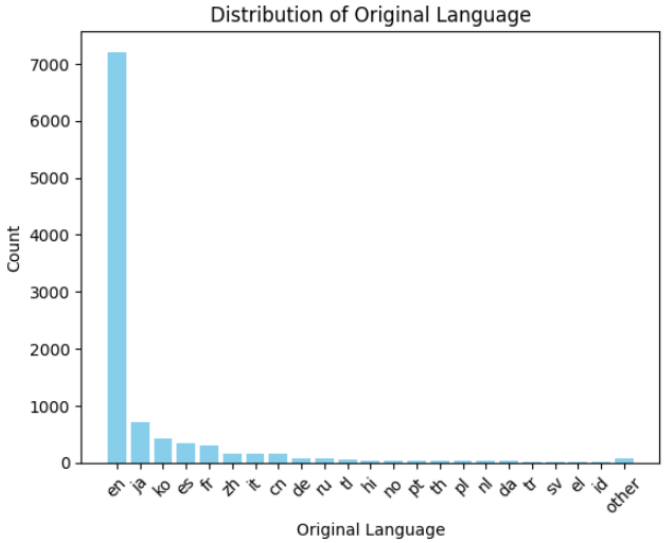
3.3.1 Kategorie

W danych występują pewne zmienne kategoryczne, takie jak status i oryginalny język filmu.

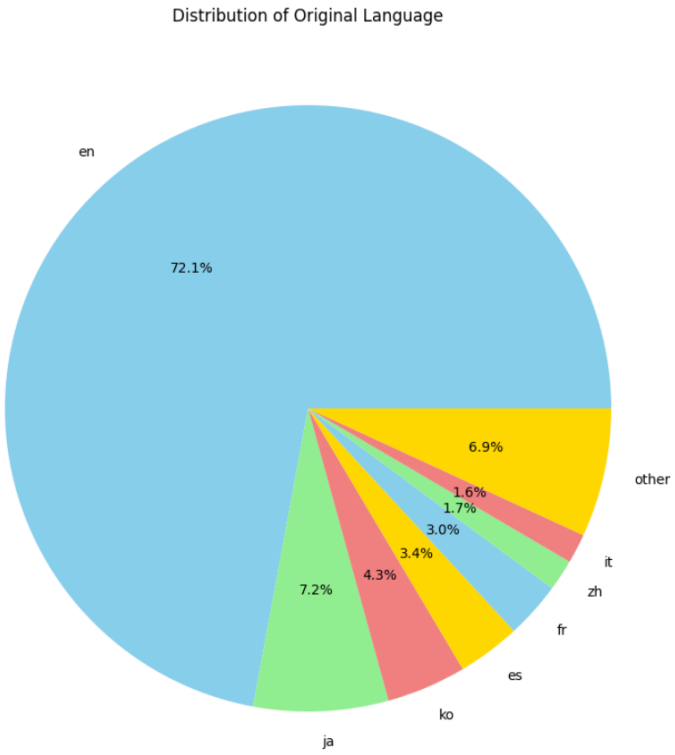
Sytuacja ze zmienną status wydaje się dość prosta: można ją łatwo przekształcić za pomocą kodowania "one-hot":



Zmienna "oryginalny język" posiada wiele języków, co nie jest szczególnie informacyjne.

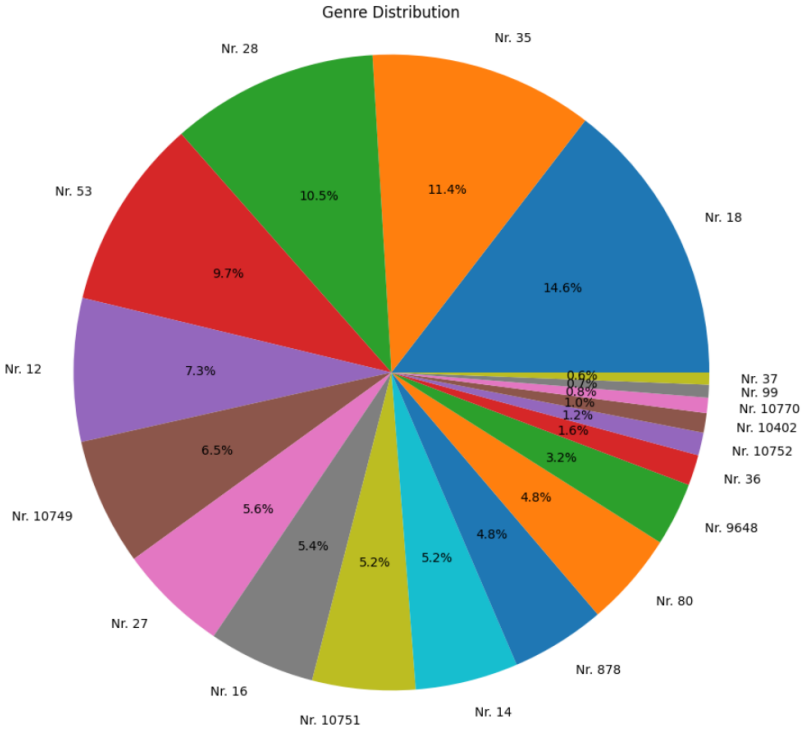


Stworzę kategorie dla siedmiu najpopularniejszych języków, a pozostałe umieszczę w kategorii "inne".



3.3.2 Kategoria w postaci listy

Informacje o przynależności do konkretnego gatunku filmu występują w postaci listy: [28, 53, 80]

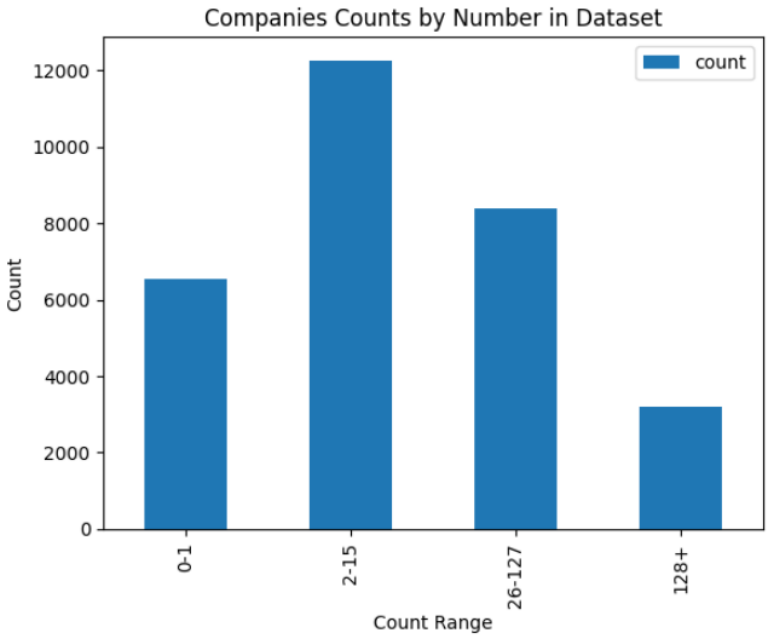


W przeciwieństwie do zmiennej "oryginalny język", stworzę kolumny dla każdego z gatunków.

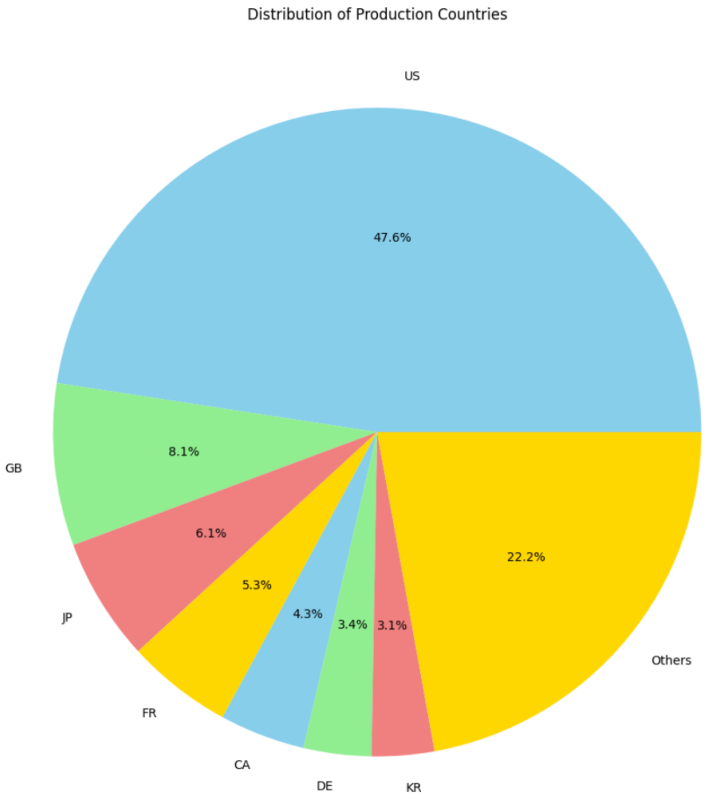
3.3.3 Kategoria w postaci słownika

Niektóre dane występują w postaci słownika, takie jak firma filmowa, dostępne języki i kraj pochodzenia.

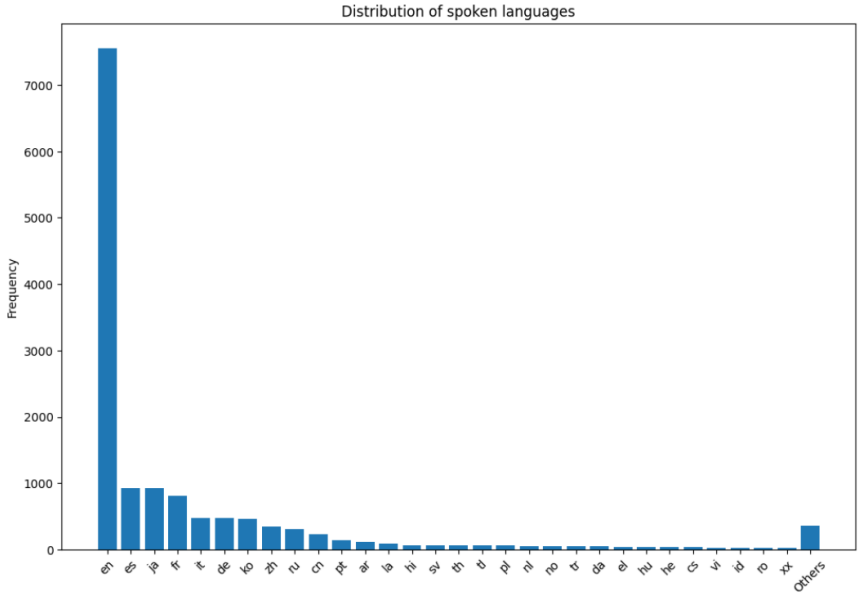
Można zauważyć, że cecha "firma filmowa" nie ma sensu. W zbiorze danych z dziesięcioma tysiącami filmów mamy około trzydziestu tysięcy różnych firm. Większość z nich pojawia się mniej niż piętnaście razy w naszym zbiorze.

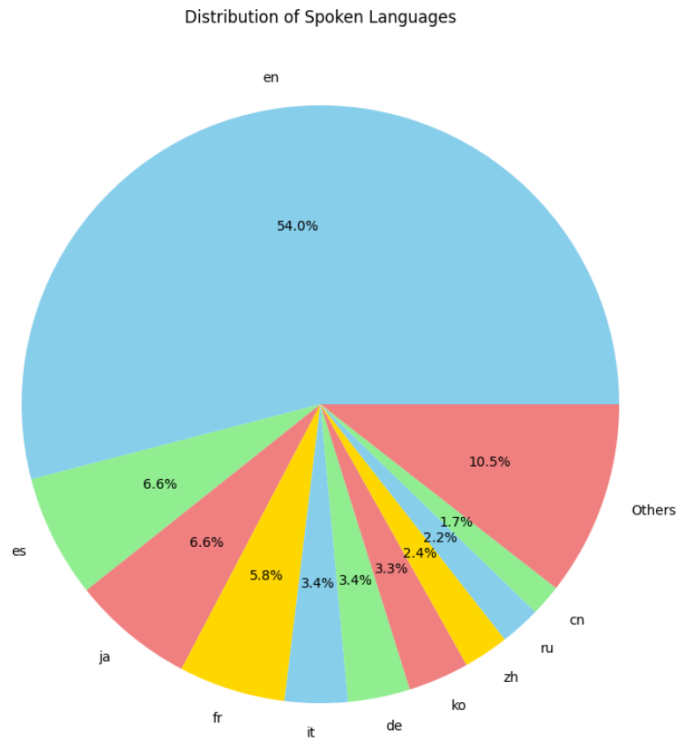


Dla "krajów produkcji" wybierzemy siedem najpopularniejszych krajów. Pozostałe kraje zostaną zakodowane jako "inne".



Dla "używanych języków" wybierzemy dziesięć najpopularniejszych języków. Pozostałe zostaną zakodowane jako "inne".

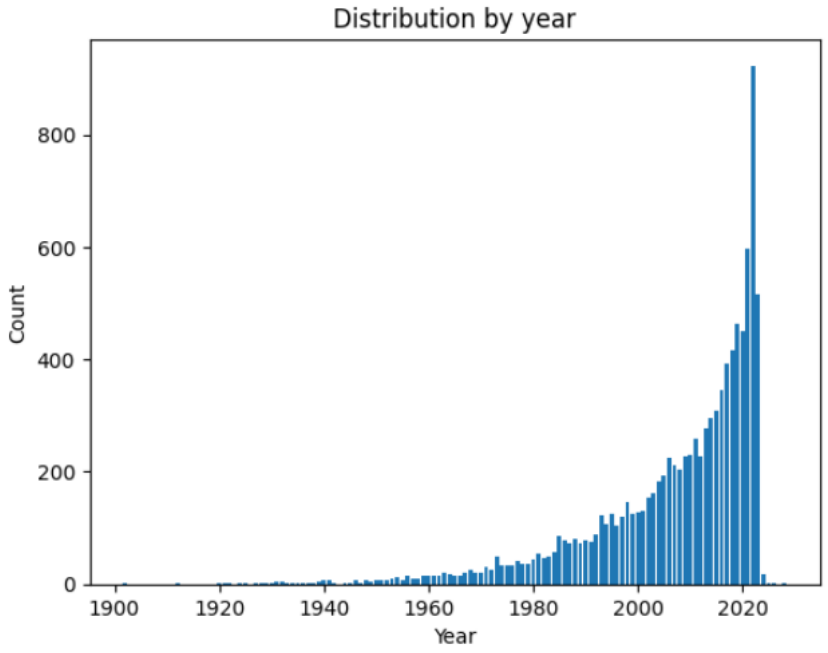

Distribution of spoken languages
Frequency
0
1000
2000
3000
4000
5000
6000
7000
en
es
ja
fr
it
de
ko
zh
ru
cn
pt
ar
la
hi
sv
th
tl
pl
nl
no
tr
da
el
hu
he
cs
vi
id
ro
xx
Others


Distribution of Spoken Languages
en
54.0%
es
6.6%
ja
6.6%
fr
5.8%
it
3.4%
de
3.4%
ko
3.3%
zh
2.4%
ru
2.2%
cn
1.7%
Others
10.5%

3.4 Kolumny wymagające prostych transformacji

3.4.1 Rok produkcji

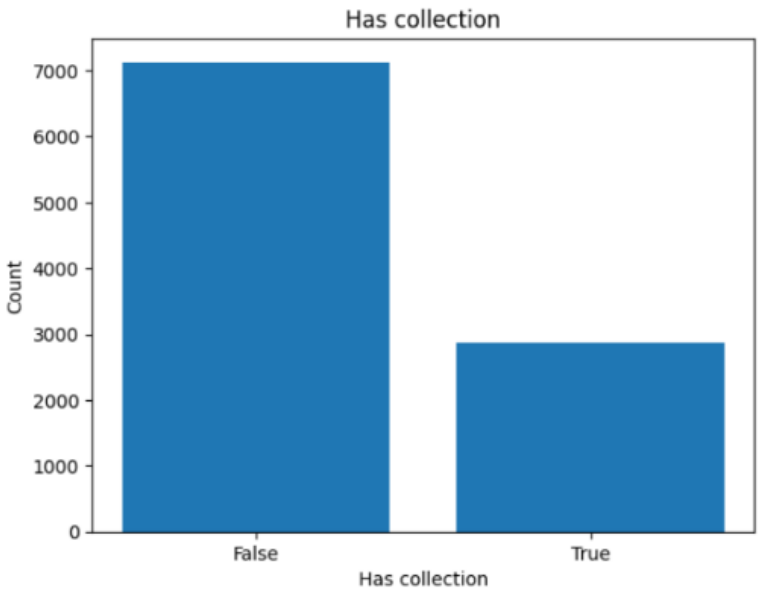
Uważam, że sensowne jest zamienienie daty na rok produkcji.
Dane wyglądają następująco:



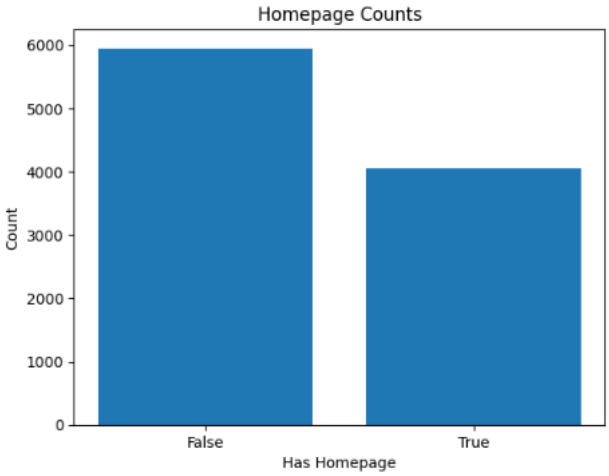
3.4.2 Wartości logiczne

Niektóre zmienne, takie jak kolekcja i strona internetowa, są dość trudne do przetworzenia. Jednak one mają wiele wartości NaN, co można wykorzystać: jeśli film należy do kolekcji lub ma stronę internetową, zostanie to oznaczone jako True, w przeciwnym razie jako False.

Czy należy do kolekcji:



Czy ma stronę internetową:



4 Działanie modeli

4.1 Zaimplementowane modele

Zostały uruchomione następne modele:

1. Regresja liniowa: Linearny sposób modelowania zależności między zmiennymi zależnymi a niezależnymi poprzez dopasowanie równania liniowego do obserwowanych danych.

2. Regresja Ridge: Model regresji liniowej, który stosuje regularyzację L2 w celu zapobieżenia nadmiernemu dopasowaniu poprzez dodanie składnika kary do funkcji straty.

3. Regresja Lasso: Podobna do regresji Ridge, ale używa regularyzacji L1, co może prowadzić do rzadkich modeli przez wyzerowanie niektórych współczynników.

4. Regresja Elastic Net: Kombinacja regresji Ridge i Lasso, która stosuje zarówno regularyzację L1, jak i L2, zapewniając równowagę między selekcją cech a zapobieżeniem nadmiernemu dopasowaniu.

5. Las Losowy: Metoda uczenia zespołowego, która konstruuje wiele drzew decyzyjnych i zwraca średnią predykcję poszczególnych drzew.

6. Gradient Boosting: Inna metoda uczenia zespołowego, która buduje sekwencję modeli, gdzie każdy kolejny model koryguje błędy poprzedniego modelu, prowadząc do poprawionych predykcji.

7. XGBoost: Skalowalna i wydajna implementacja gradient boosting, która zapewnia lepszą wydajność i szybkość w porównaniu do tradycyjnych implementacji gradient boosting.

8. LightGBM: Framework gradient boosting, który używa nowatorskiego algorytmu rozbudowy drzew, co prowadzi do szybszego procesu uczenia i mniejszego zużycia pamięci.

9. CatBoost: Biblioteka gradient boosting, która naturalnie obsługuje cechy kategoryczne i zapewnia wbudowaną selekcję cech, co ułatwia pracę z danymi tabelarycznymi.

10. KNN (K najbliższych sąsiadów): Algorytm nieliniowej regresji, który przewiduje wartość nowej instancji poprzez uśrednienie wartości jej k najbliższych sąsiadów.

4.2 Wyniki

Dla rozszerzenia analizy, ja utworzyłam trzy dataset'y:

1. Zestaw danych 1 - tylko wartości numeryczne:

   Ten zestaw danych może być wykorzystywany do eksploracji relacji pomiędzy wartościami numerycznymi a ocenami filmów.

2. Zestaw danych 2 - wartości numeryczne, gatunek, strona domowa i data premiery:

   Ten zestaw danych jest bardziej rozbudowany i zawiera dodatkowe informacje, takie jak gatunek filmu, strona domowa i data premiery. Może to pomóc w budowie bardziej rozbudowanych modeli predykcyjnych.

3. Zestaw danych 3 - wszystkie dostępne wartości:

   Ten zestaw danych zawiera wszystkie dostępne informacje na temat filmów. Obejmuje zarówno cechy numeryczne, jak i kategoryczne. Ten zestaw danych pozwala na pełniejszą analizę i budowę bardziej kompleksowych modeli predykcyjnych, które uwzględniają szeroki zakres informacji.

Model	Dataset v.1	Dataset v.2	Dataset v.3
Linear Regression	1.35257	1.23178	1.11135
Ridge	1.35257	1.23179	1.11134
Lasso	1.35436	1.32211	1.32211
Elastic Net	1.3527	1.32004	1.32004
Random Forest	1.3862	0.728896	0.704485
Gradient Boosting	1.24652	0.74027	0.733997
XGBoost	1.32971	0.72769	0.710607
LightGBM	1.25507	0.730862	0.71002
CatBoost	1.26563	0.714585	0.69395
KNN	1.36274	1.06599	1.06624

Jako metrykę wybrałam metrykę mean squared error.

Ogólnie modele wykazały dobre wyniki, a ich skuteczność rosła wraz z zwiększaniem ilości dostarczonych danych. Inne wnioski zostaną wyciągnięte w następnej części.

5 Wnioski

Modele oparte na lasach losowych (Random Forest), w tym wszystkie złożone modele, takie jak GradientBoosting, XGBoost, LightGBM i CatBoost osiągnęły najlepsze wyniki dla wszystkich zestawów.

Regresja liniowa, regresja Ridge, regresja Lasso i regresja Elastic Net osiągnęły zbliżone wyniki, ale nie były w stanie dorównać do wyników modeli opartych na lasach losowych i gradient boosting.

Model KNN uzyskał gorsze wyniki niż inne modele w obu zestawach danych. Może to sugerować, że koncepcja KNN nie jest odpowiednia dla tego problemu.

Dodanie niektórych cech do zbioru pierwszego sprawiło, że modele zadziałały znacznie lepiej, ale następne rozszerzenie zbioru danych (do wersji trzeciej) nie przyniosło tak imponujących rezultatów.

6 Komentarze, ulepszenia

6.1 NLP

TF-IDF: Metoda TF-IDF jest powszechnie stosowana do analizy tekstu i wydobycia cech z dokumentów. Polega na przypisaniu wagi słowom w oparciu o ich częstość występowania w dokumencie (TF) oraz odwrotną częstość występowania w całym korpusie dokumentów (IDF). W przypadku dodanej cechy "opis" filmu, zastosowano metodę TF-IDF w celu przetworzenia tekstu i wydobycia istotnych cech. Jednakże, wyniki tej metody okazały się niesatysfakcjonujące.

BERT: BERT to zaawansowany model językowy oparty na transformerach, który może efektywnie modelować kontekst i semantykę tekstu. Zastosowanie BERT-a do analizy tekstu może przynieść lepsze rezultaty w porównaniu do tradycyjnych metod. W tym przypadku, próbowano wykorzystać BERT-a do ekstrakcji cech z dodanej cechy "opis" filmu. Jednak, mimo obiecujących perspektyw, wyniki tej metody również nie były satysfakcjonujące.

W celu wykorzystania cechy "Opis", zastosowano dwie metody: TF-IDF (Term Frequency-Inverse Document Frequency) oraz BERT (Bidirectional Encoder Representations from Transformers). Niestety, obie te metody nie dostarczyły sensownych wyników i ulepszenie bieżacego modelu się nie odbyło.

Z tego wynika, że wydobycie cech z opisów filmów nie pomaga w uzyskaniu sensownych przewidywań oceny. Wartość przewidywań jest znacznie bardziej uzależniona od innych cech filmu, takich jak gatunek, budżet, popularność itp.

6.2 Dalsze badania

W kontekście ulepszeń zaproponowanych modeli, ja proponuję dwie techniki:

Ensemble Learning: dobrym podejściem jest zastosowanie ensemble learning, czyli kombinacja kilku modeli w celu uzyskania lepszych wyników. W tym celu można zastosować już zaimplementowane modele.

Rozszerzenie dataseta: z czasem pozycje filmów będą się zmieniać. Stare filmy stracą swoją pozycję w rankingu, ale pojawią się nowe. W związku z tym za kilka lat możliwe będzie powtórzenie zbierania danych i rozszerzenie tego zbioru danych.