**Data Preprocessing**

1. Load two datasets into DataFrame.
2. Combine two datasets into one.
3. Remove unnecessary columns.
4. Data cleaning on columns which contain missing value.
5. Inspection of duplicate data.
6. Applying data transformation of each column.
7. Combine all the columns into one column.

一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述

In the beginning of data preprocessing, we first load in two csv files and save it into the DataFrame we will use later.

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

To have an efficient data preprocessing, we merge two datasets into one based on the column “title”. Before merging, the two datasets contain 20 and 4 columns. After merging two datasets, the new DataFrame contains 23 columns which is the sum of two datasets minus one duplicate columns “title”.

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

In all 23 columns, we only needed 7 of them for our model training. We removed all the other unnecessary columns and saved the columns we needed into a new DataFrame.

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

Before doing data transformation, we inspected each column of dataset to see if there is any missing value. We found that there are 3 missing values in column “overview”. In order to avoid error in our model training, we removed those 3 data with missing values. It can be seen that the original dataset contains 4809 pieces of data and the dataset after applying data cleaning contains 4806 pieces of data.

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

We also inspected if there are duplicate data in the dataset. The result showed that all the data are unique.

一張含有 文字, 軟體, 字型, 數字 的圖片

自動產生的描述

After doing data cleaning, we started on data transformation. First, we applied a transformation function to column “genres”. In this column, the original data format is “{"id": 28, "name": "Action"}”, the target value we needed in this column is “name”.

一張含有 文字, 字型, 數字, 行 的圖片

自動產生的描述

In column “keywords”, we applied the same function in column “genres”. The original formant of this column is “{"id": 1463, "name": "culture clash"}”, the target value we needed is “name”.

一張含有 文字, 軟體, 螢幕擷取畫面, 數字 的圖片

自動產生的描述

In column “cast”, it contains all the characters of each actor. The original data format is “{"cast\_id": 242, "character": "Jake Sully", "credit\_id": "5602a8a7c3a3685532001c9a", "gender": 2, "id": 65731, "name": "Sam Worthington", "order": 0}”, the target value we needed in this column is “name”, which is the name of actors. Also, in our model training, we only wanted the first three actor’s name, they are the main characters in this movie.

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

In column “crew”, it contains all the crew members of this film. The original data format is “{"credit\_id": "52fe48009251416c750aca23", "department": "Editing", "gender": 0, "id": 1721, "job": "Editor", "name": "Stephen E. Rivkin"}”, the target value we needed is “job”. Since the director is the most discussed among the crew members of a movie, we wanted only director name.

一張含有 文字, 字型, 數字, 行 的圖片

自動產生的描述

In column “overview”, it contains the introduction to the movie, the type of the data is string. To convert its type into list, we applied a lambda function which split all the words in the string.

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

In order to avoid confusion, we also applied a lambda function to remove all the space in all the columns.

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

After applying all the transformations functions, we combined the values in all the columns into one column “tags”.

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述  
We then created a new DataFrame and saved only three columns which are “movie\_id”, “movie title”, and “tags” into it.

一張含有 文字, 字型, 數字, 行 的圖片

自動產生的描述

We converted the type of the column “tags” in new DataFrame from list to string.

一張含有 文字, 字型, 行, 數字 的圖片

自動產生的描述  
The last step in data preprocessing is converting all the characters into lower case.