آرین قره محمد زاده قشقایی

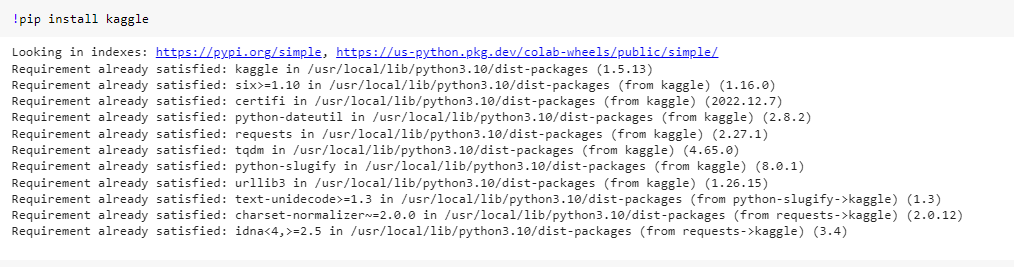
4017724001

محمد برزگر

4010724001

تمرین 5

دیتاست movie\_5000\_t به وسیله api کگل به کولب اضاف شده و دیتاست را دانلود کردیم در ابتدا.





دیتاست حاوی 2 فایل tmdb\_5000\_credits.csv و tmdb\_5000\_movies.csv بود که فایل tmdb\_5000\_movies.csv حاوی ستون های زیر بود

'budget', 'genres', 'homepage', 'id', 'keywords', 'original\_language', 'original\_title', 'overview', 'popularity', 'production\_companies', 'production\_countries', 'release\_date', 'revenue', 'runtime', 'spoken\_languages', 'status', 'tagline', 'title', 'vote\_average', 'vote\_count'], dtype='object

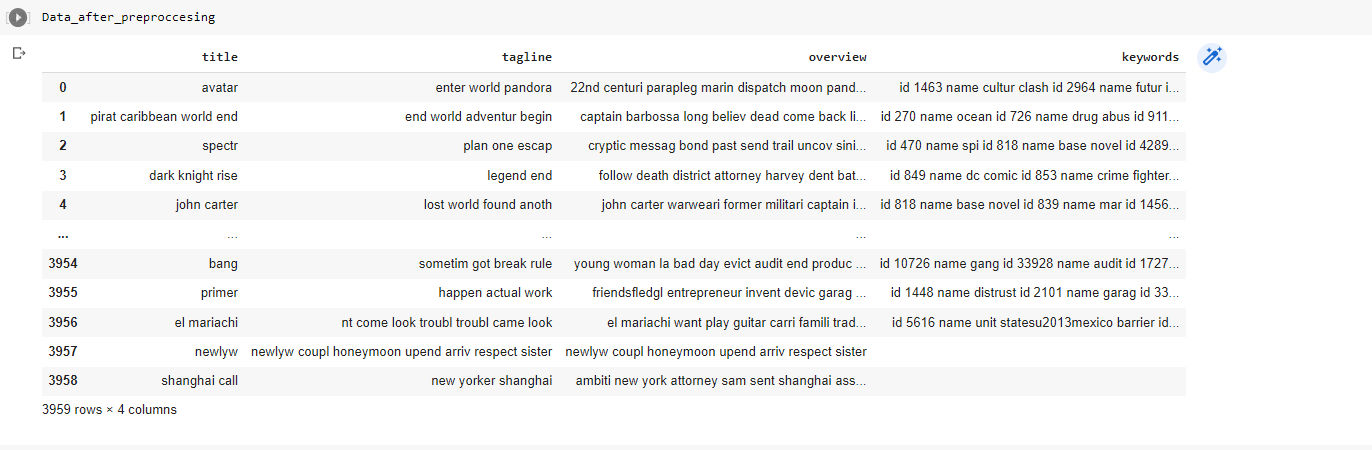
از بین این ستون ها، ستون های tagline , overview , title , keywords را انتخاب و عمل پری پراکسسینگ را انجام میدیم.

دو تابع زیر برای حذف کلمات stem و stopwords ها ساخته شده.



در زیر جدول داده های فیلم ها قبل از پری پراکسسینگ و بعد از آن را نمایش داده.

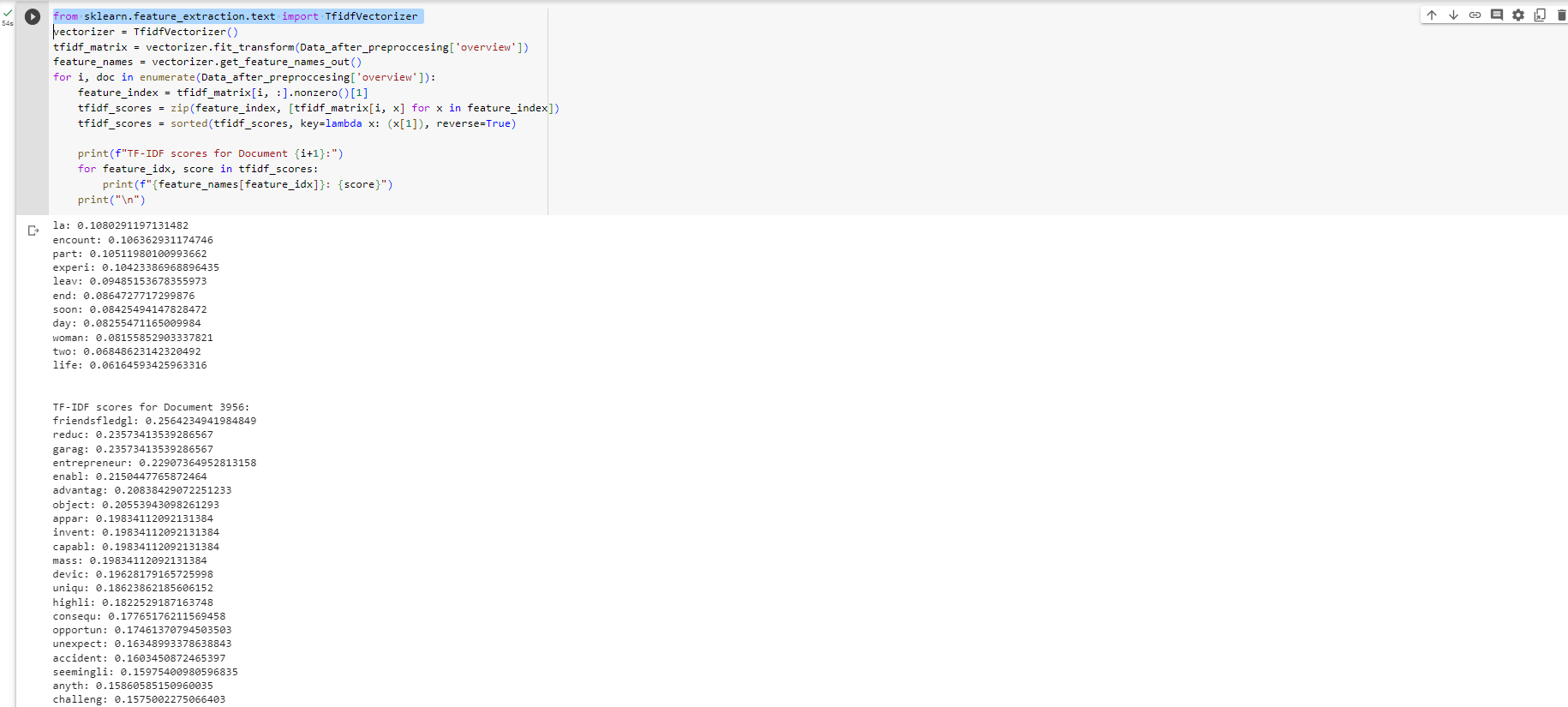




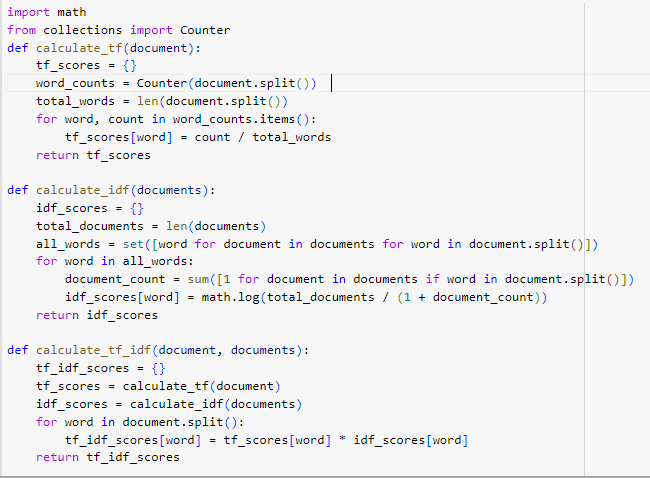
بعد از این که داده ها آماده شد. به وسیله کد زیر

from sklearn.feature\_extraction.text import TfidfVectorizer

از sklearn متدی برای محاسبه TfidF به پروژه اضاف میکنیم. و به وسیله آن وکتور های کلمات را بدست می آوریم.

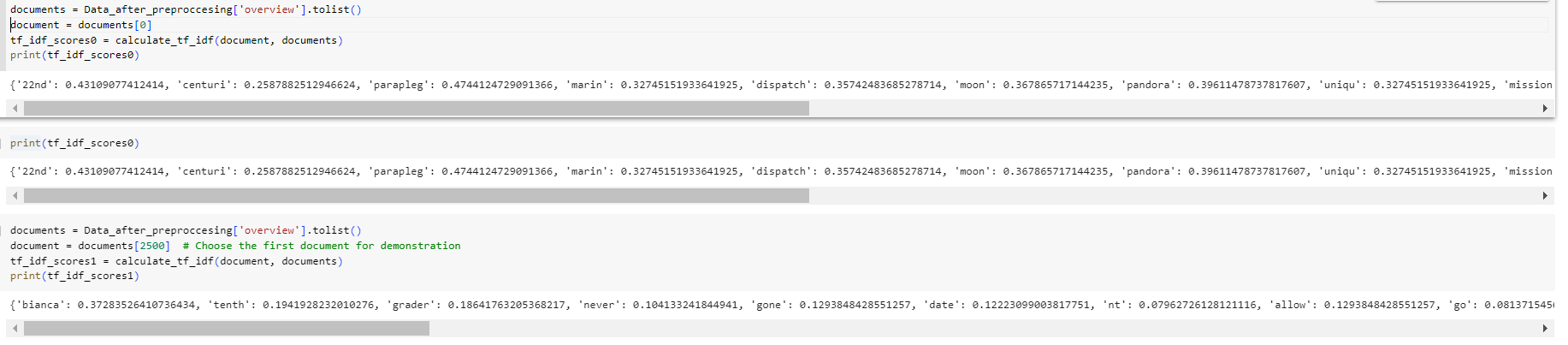


همین طور این متد به صورت دستی توسط ما نیز پیاده سازی شد.

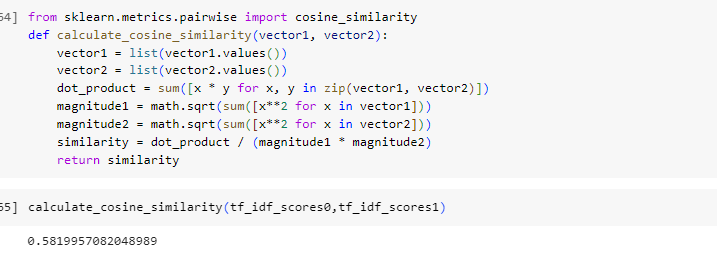


در این پیاده سازی در ابتدا متد tf برای بدست آوردن ترم فریکوانسی کلمات یا همان تعداد تکرار کلمات در داکیومنت را محاسبه میکند و idf نیز متدی برای بدست آوردن معیار اهمیت کلمه بر حسب rarity هست. و در نهایت متد tf- idf برای در نظر گرفتن اسکور نهایی ساخته شده.

در اینجا دو دامیومنت را به صورت رندوم انتخاب و tf idf آن را محاسبه کردیم



وکتور های محاسبه شده را به تابع زیر دادیم



که یک معیار شباهت برحسب cos similarity هست دادیم و مقدار شباهت این دو وکتور را بدست آوردیم.