

# GENRE گزارش پروژه CLASSIFICATION

## آنچه بررسی خواهیم کرد..

### مقدمه

در این بخش، به معرفی کلی پروژه و هدف آن پرداخته می شود. توضیح داده می شود که هدف اصلی از تحلیل آنها چیست.

### توضیحات کد

این بخش شامل توضیح هر قطعه کد به تفکیک می باشد. هر بخش از کد شامل توضیحات مختصر و مفید درباره عملکرد آن بخش است.

### تحلیل و تفسیر داده های اولیه

در این قسمت، داده های اولیه بارگذاری و بررسی می شوند. تحلیل اولیه شامل بررسی ویژگی ها، توزیع داده ها و هر گونه نکته مهمی است که از داده ها استخراج می شود.

### پردازش و تجزیه و تحلیل داده ها

این بخش شامل تمامی مراحل پردازش داده ها از جمله پاکسازی، نرمال سازی و هر گونه عملیات پیش پردازشی است که روی داده ها انجام می شود.

## مدل سازی و ارزیابی

در این قسمت، مدل های مختلفی که برای تحلیل داده ها استفاده شده اند، معرفی و ارزیابی می شوند. نتایج هر مدل و معیارهای ارزیابی مختلف توضیح داده می شود.

## نتایج و نتیجه گیری

این بخش شامل تفسیر نهایی نتایج به دست آمده از مدل ها و تحلیل های انجام شده است. همچنین نقاط قوت و ضعف مدل ها و روش های به کار گرفته شده بررسی می شود.

## نتیجه گیری نهایی

در این قسمت، نتیجه گیری کلی از پروژه و یافته های اصلی آن ارائه می شود.

## مقدمه

کلاس بندی یکی از مسائل مهم حوزه هوش مصنوعی است. در کلاس بندی، هدف این است که مدل بتواند هر نمونه ورودی را در یکی از کلاس های از پیش تعیین شده جای دهد. یک مثال معروف در کلاس بندی دودویی، تشخیص اسپم بودن یا نبودن یک متن است. مثال دیگر، مدلی است که با دریافت یک عکس تشخیص دهد که در عکس چه حیوانی وجود دارد؛ یک گربه، یک سگ یا یک موش.

در این پروژه می خواهیم فیلم ها را بر اساس ژانرشان کلاس بندی کنیم. برای این کار به یک مدل نیاز داریم که با دریافت توضیحات مربوط به یک فیلم، ژانر آن فیلم را پیش بینی کند و به عنوان خروجی برگرداند.

## توضیحات کد

**پیش پردازش:** ابتدا داده هایی را که در اختیارمان قرار می گیرد بررسی می کنیم. توزیع نمونه ها

را می بینیم و ژانرهای موجود را استخراج می کنیم. در صورت نیاز، بعضی نمونه ها را اصلاح یا

حذف خواهیم کرد. سپس پیش پردازش را روی متن های ستون **description** انجام خواهیم

داد. و سپس توکن بندی جملات را هم انجام می دهیم همچنین داده های آموزشی را به دو داده

آموزش و ارزیابی تقسیم می کنیم و از آنها در مراحل بعدی استفاده می کنیم.

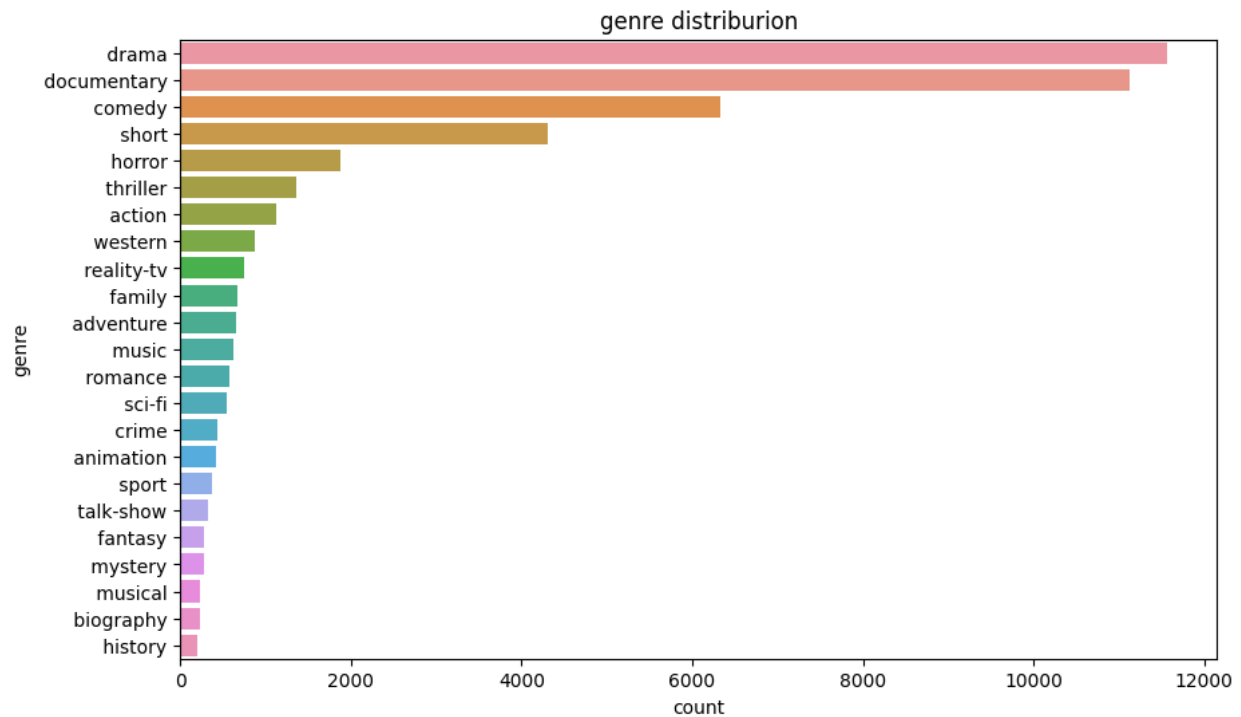
جاسازی : به کمک یک روش دلخواه، متن‌های خام description را به ساختاری قابل فهم برای مدل‌تان تبدیل میکنیم. این جاسازی وابسته به مدلی که استفاده می‌کنیم، می‌تواند متفاوت باشد. برای مثال در این پروژه از روش TF-IDF برای یک بخش استفاده خواهد شد.

طراحی و آموزش مدل : بر اساس روش Naïve Bayes مدل خود را طراحی میکنیم و روی داده‌های آموزشی، آموزش می‌دهیم.

ارزیابی و اصلاح : در این مرحله عملکرد مدل خود را روی داده‌های ارزیابی خواهیم کرد..

تست و گزارش : در مرحله آخر نیاز است که مدل خود را روی داده‌های تست ، تست میکنیم و نتایج را گزارش میکنیم. مدل خود را با معیارهای دقت (accuracy) ، فراخوانی (recall) و امتیاز F1 ارزیابی میکنیم. این کار را با گرفتن میانگین micro روی معیارهای ارزیابی مربوط به هر کلاس انجام میدهیم.

## تحلیل و تفسیر داده‌های اولیه



نمودار توزیع ژانرها

در نمودار ارائه شده، توزیع تعداد فیلم‌ها و برنامه‌ها بر اساس ژانرهای مختلف نشان داده شده است. محور افقی تعداد فیلم‌ها یا برنامه‌ها و محور عمودی ژانرهای مختلف را نشان می‌دهد. داده‌های ارائه شده به شرح زیر است:

- درام: ۱۱۵۷۱

- مستند: ۱۱۱۳۲

- کمدی: ۶۳۳۰

- کوتاه: ۴۳۱۲

- ترسناک: ۱۸۷۳

- هیجان‌انگیز: ۱۳۵۲

- اکشن: ۱۱۱۸

- وسترن: ۸۷۷

- ۷۵۱: (Real i ty-TV) - واقعیت نما

- خانوادگی: ۶۶۷

- ماجراجویی: ۶۵۹

- موسیقی: ۶۲۱

- عاشقانه: ۵۷۱

- علمی-تخیلی: ۵۵۰

- جنایی: ۴۲۹

- انیمیشن: ۴۲۳

- ورزشی: ۳۶۷

- تاک شو: ۳۳۲

- فانتزی: ۲۷۵

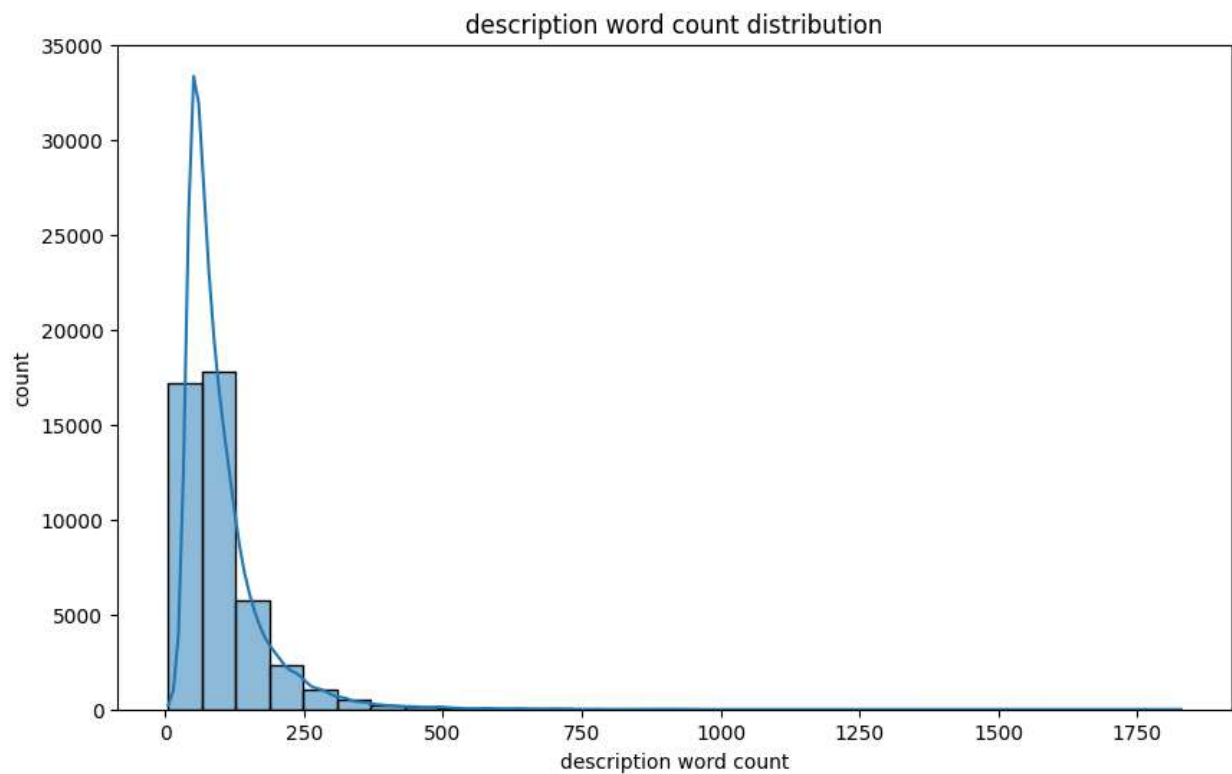
- معمایی: ۲۷۱

- موزیکال: ۲۳۶

- زندگینامه‌ای: ۲۲۵

- تاریخی: ۲۰۷

در این نمودار، ژانر درام با بیشترین تعداد فیلم و برنامه، رتبه اول را دارد و پس از آن ژانر مستند قرار گرفته است. ژانرهای کمدی و کوتاه نیز به ترتیب در رتبه‌های بعدی قرار دارند. ژانرهای تاریخی، زندگینامه‌ای و موزیکال کمترین تعداد فیلم و برنامه را دارند.



توزیع تعداد کلمات در توضیحات فیلم

۴۵۱۴۹	-تعداد مشاهدات
۱۰۲,۱۳	-میانگین
۷۶,۷۰	-انحراف معیار
۶	-کمینه
۵۵	-چارک اول (۲۵٪)
۷۹	-میانه (۵۰٪)
۱۲۱	-چارک سوم (۷۵٪):
۱۸۲۹	-بیشینه

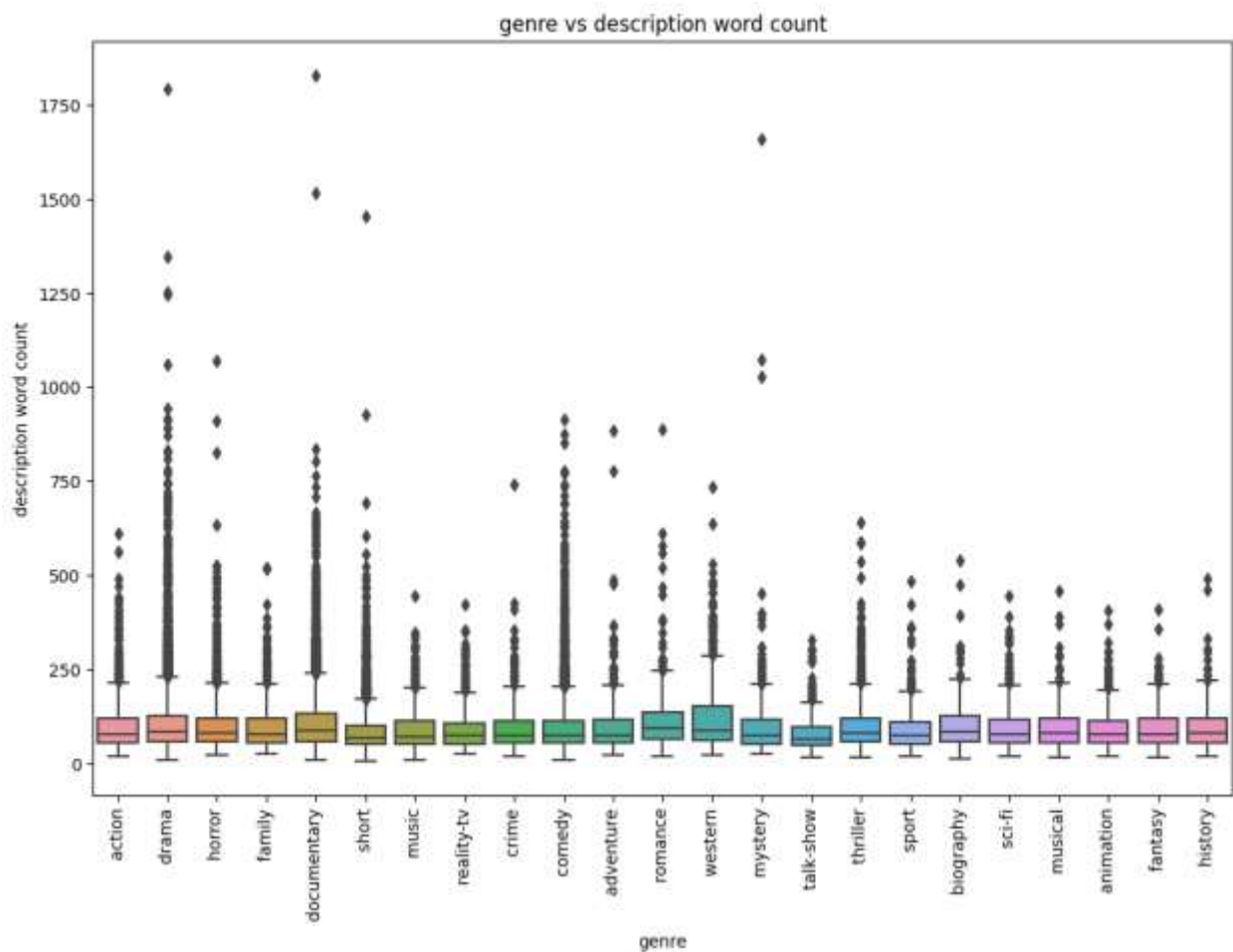
تفسیر نمودار:

این نمودار توزیع طول توصیفات (تعداد کلمات در هر توصیف) را نشان می‌دهد. همان‌طور که مشاهده می‌شود، بیشتر توصیفات شامل تعداد کلمات کمتری هستند و با افزایش تعداد کلمات، تعداد توصیفات کاهش می‌یابد. - میانگین طول توصیفات ۱۰۲ کلمه و میانه ۷۹ کلمه است. این نشان می‌دهد که بیشتر توصیفات کمتر از ۱۰۲ کلمه دارند.

- انحراف معیار ۷۶,۷۰ کلمه است که نشان‌دهنده پراکندگی بالای طول توصیفات است.

- نمودار نشان‌دهنده توزیع نامتقارن به سمت راست است (چولگی به راست)، به این معنی که تعداد کمی از توصیفات بسیار طولانی هستند.

به طور کلی، بیشتر توصیفات در محدوده کوتاه تا متوسط قرار دارند و تعداد کمی توصیف بسیار طولانی وجود دارد. این توزیع نشان‌دهنده الگوی رایجی است که در بسیاری از متون و اسناد مشاهده می‌شود، جایی که اکثریت محتوا کوتاه و مختصر است و تعداد کمی از محتواها طولانی و مفصل هستند.



در این نمودار جعبه‌ای، ژانرهای مختلف فیلم‌ها در محور افقی قرار دارند و تعداد کلمات توضیحات در محور عمودی.

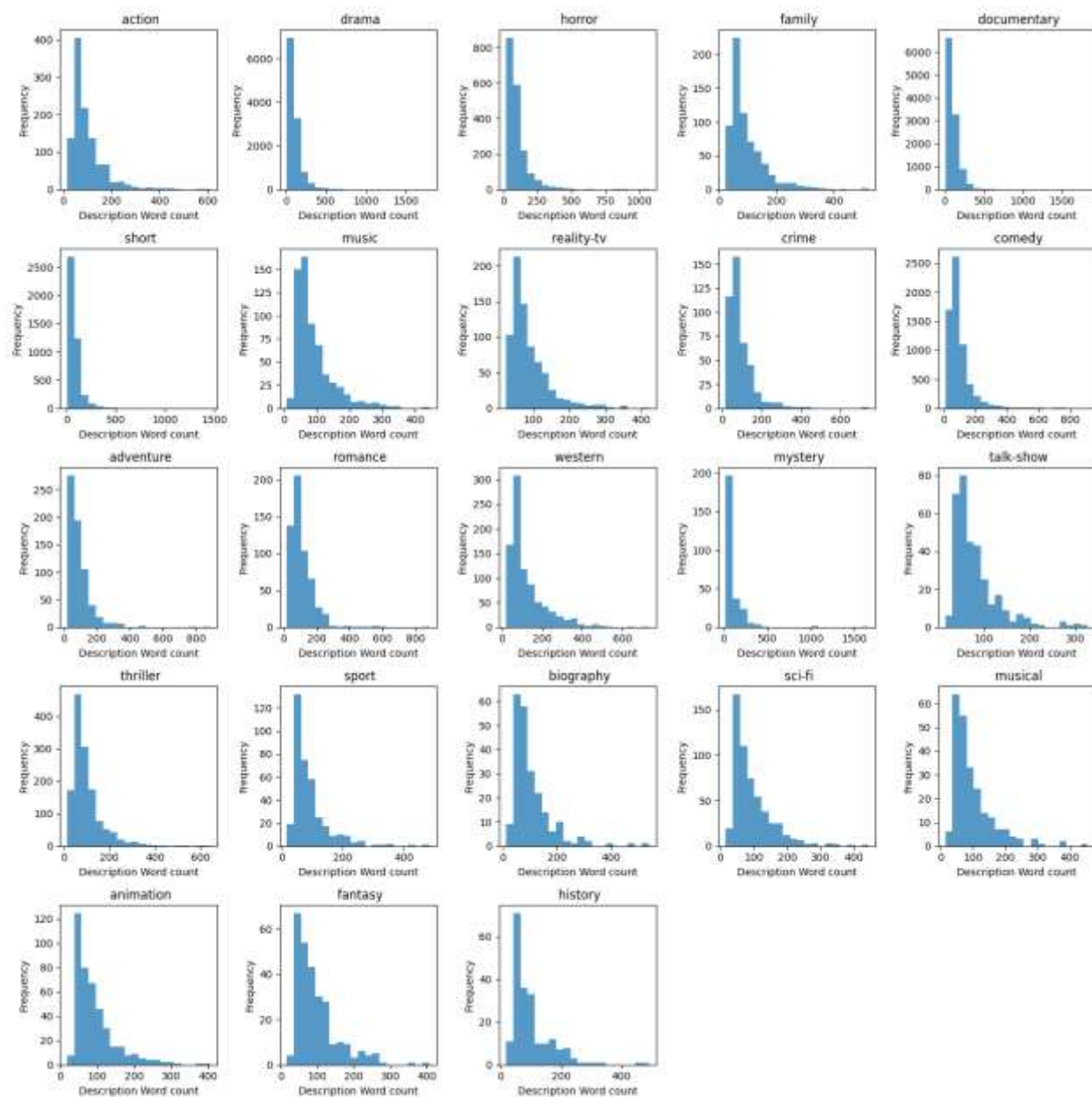
برای هر ژانر یک جعبه نمایش داده شده که محدوده بین چارک اول و سوم (IQR) را نشان می‌دهد و خط وسط جعبه میانگین داده‌ها را نشان می‌دهد.

نقاط داده‌ای که خارج از محدوده نرمال قرار دارند به صورت نقاطی جداگانه (Outliers) نمایش داده شده‌اند.

ژانرها شامل اکشن، درام، ترسناک، خانواده، مستند، کوتاه، موسیقی، واقعیت‌نمایی (Reality-TV)، جنایی، کمدی، ماجراجویی، عاشقانه، وسترن، معمایی، تاک‌شو، هیجان‌انگیز، ورزشی، زندگی‌نامه، علمی-تخیلی، موزیکال، انیمیشن، فانتزی و تاریخی هستند.

بیشترین تعداد کلمات توضیحات در ژانرهای درام، مستند و جنایی مشاهده می‌شود.





مجموعه‌ای از هیستوگرام‌ها برای هر ژانر است که توزیع تعداد کلمات توضیحات را نشان می‌دهند.

هر هیستوگرام تعداد تکرار تعداد کلمات مختلف را در توضیحات فیلم‌های هر ژانر نشان می‌دهد.

به عنوان مثال، در ژانر اکشن بیشتر توضیحات کمتر از ۲۰۰ کلمه دارند، در حالی که در ژانر درام نیز توضیحات عمدتاً کمتر از ۴۰۰ کلمه هستند.

توزیع تعداد کلمات در همه ژانرها تمایل به کاهش نمایی دارد، به این معنی که تعداد کلمات بیشتر در توضیحات کمتر مشاهده می‌شود.



wordcloud for drama

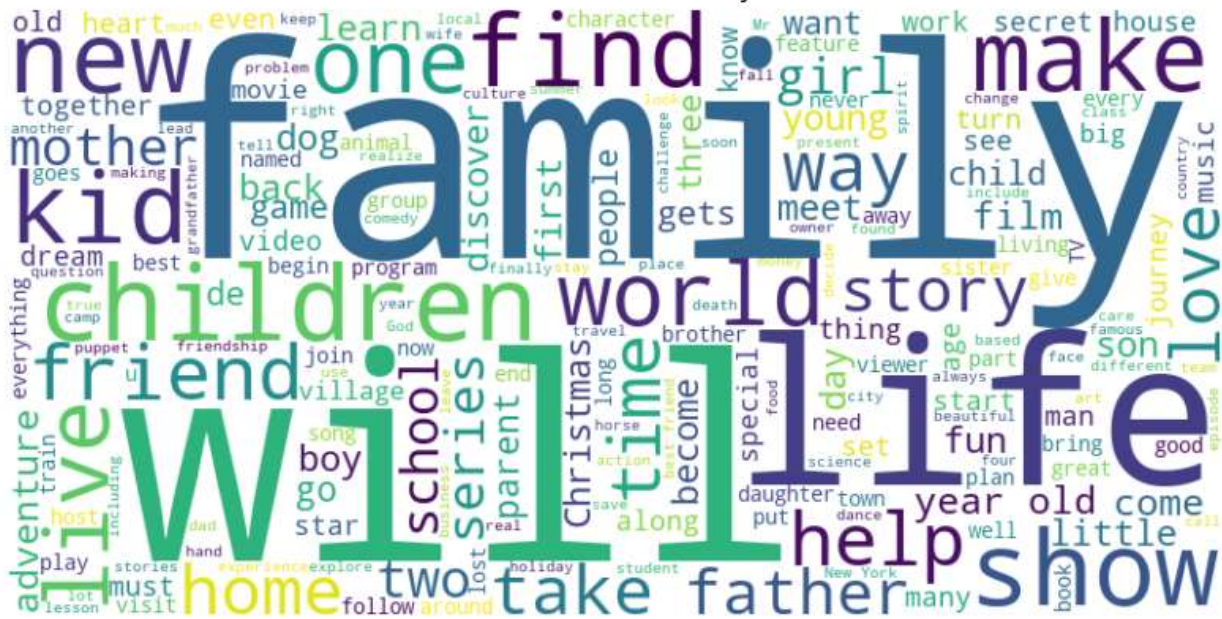


wordcloud for horror

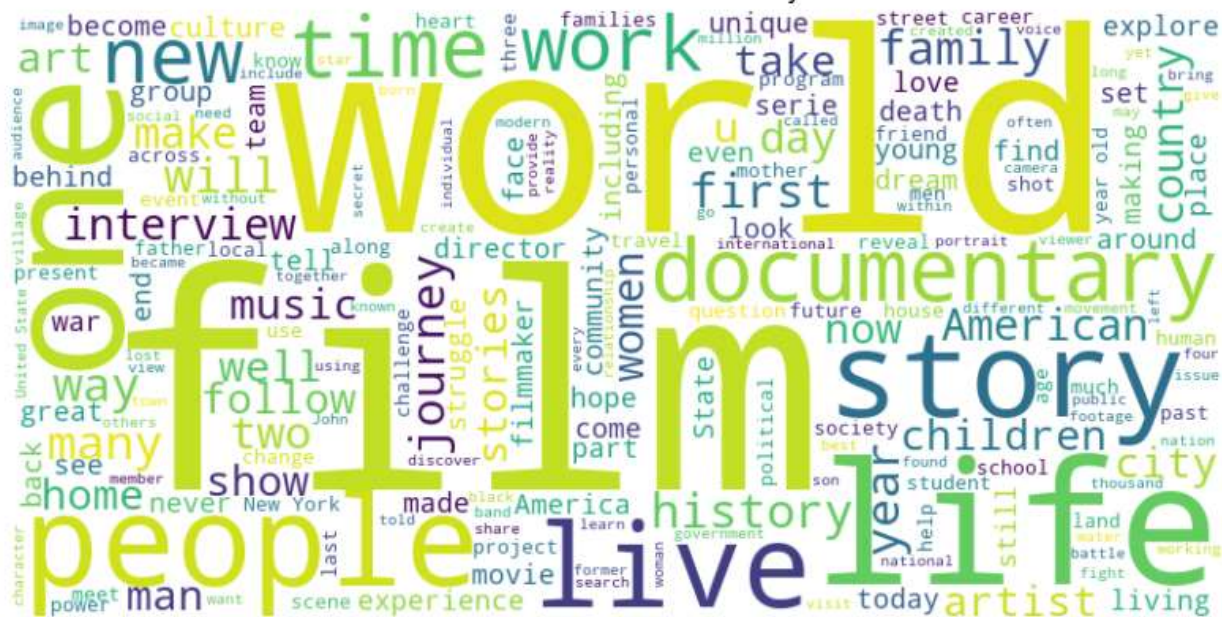




wordcloud for family



wordcloud for documentary



wordcloud for short



wordcloud for music

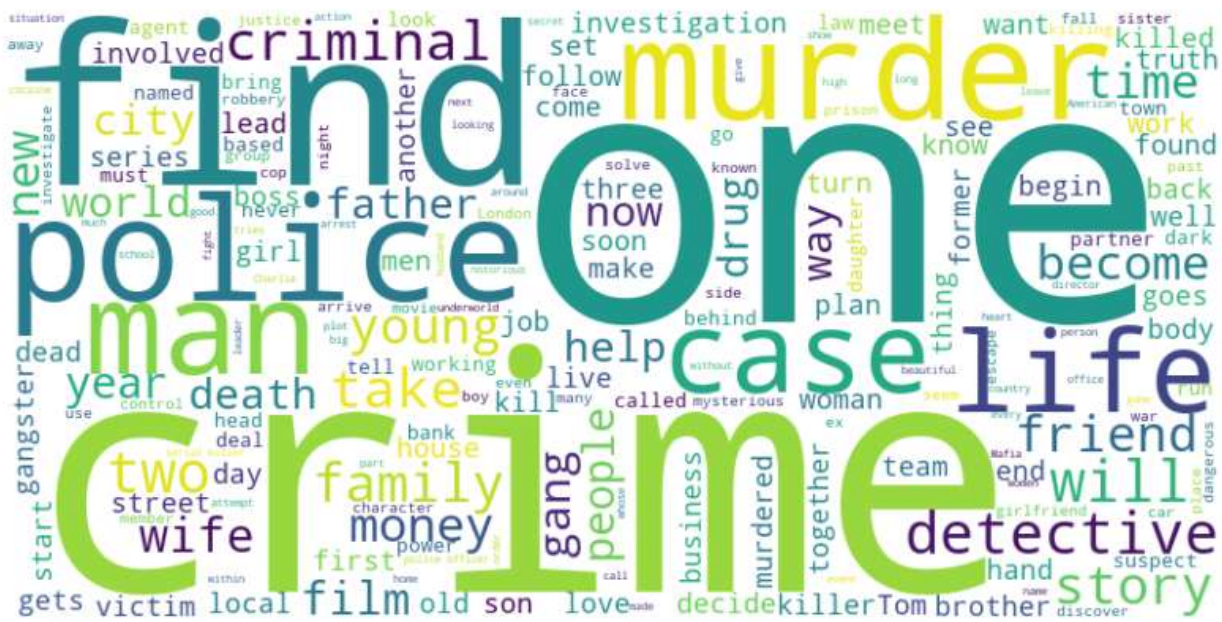




wordcloud for reality-tv



wordcloud for crime



wordcloud for comedy



wordcloud for adventure

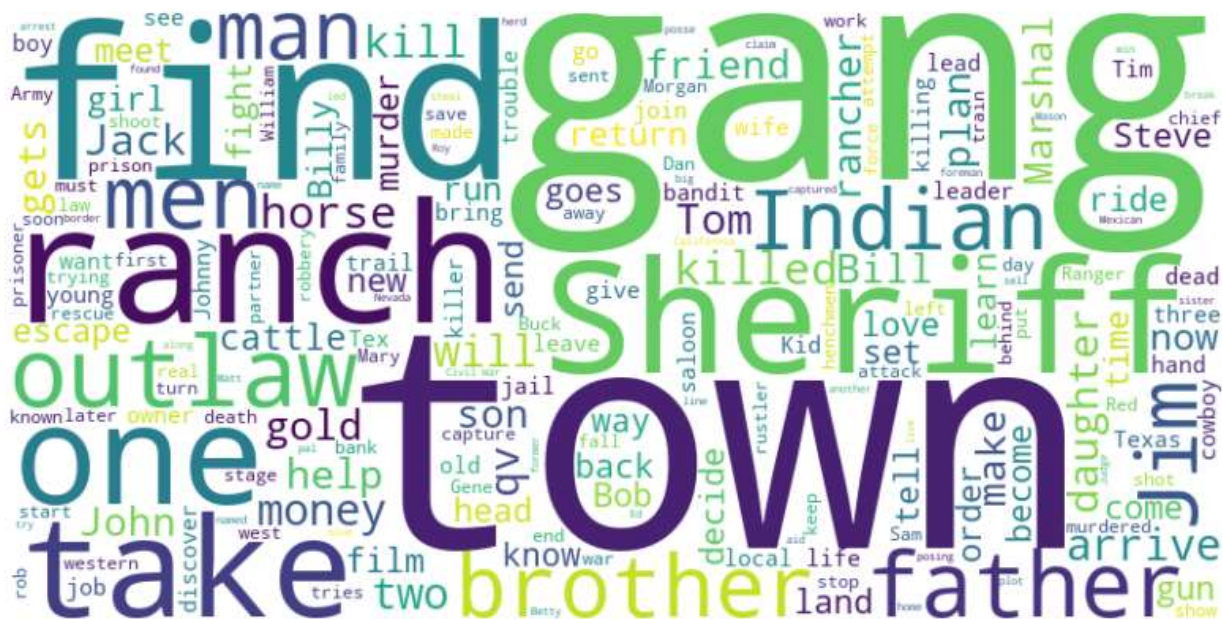




wordcloud for romance



wordcloud for western

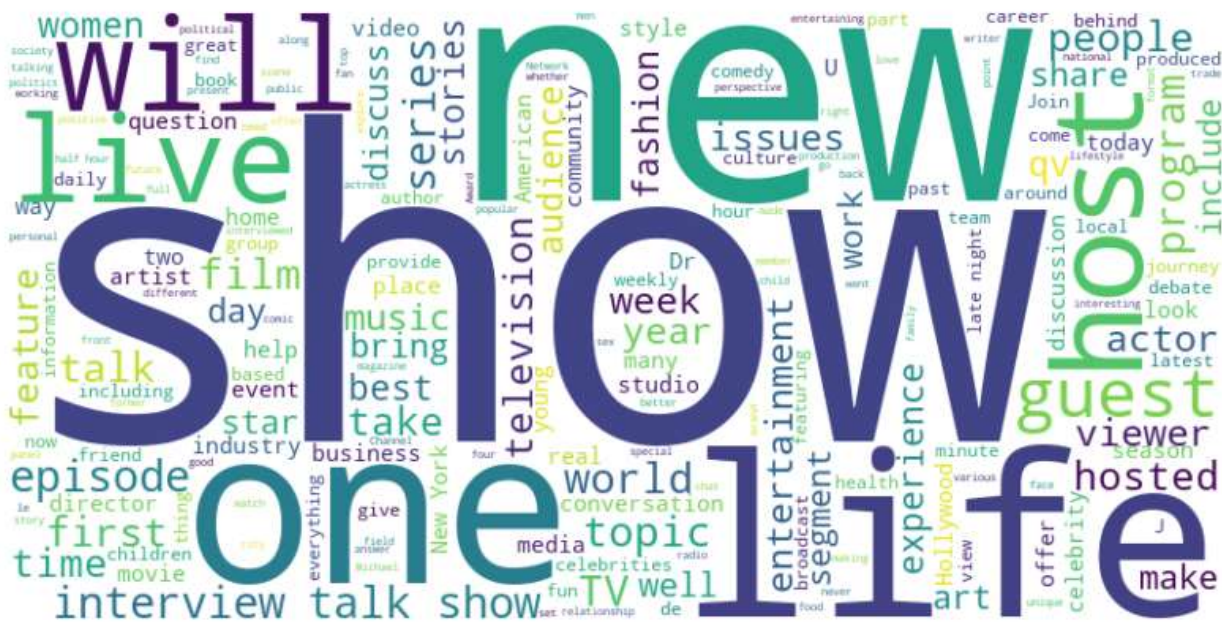




wordcloud for mystery



wordcloud for talk-show



wordcloud for thriller



wordcloud for sport





wordcloud for biography



wordcloud for sci-fi





wordcloud for musical



wordcloud for animation





wordcloud for fantasy



wordcloud for history

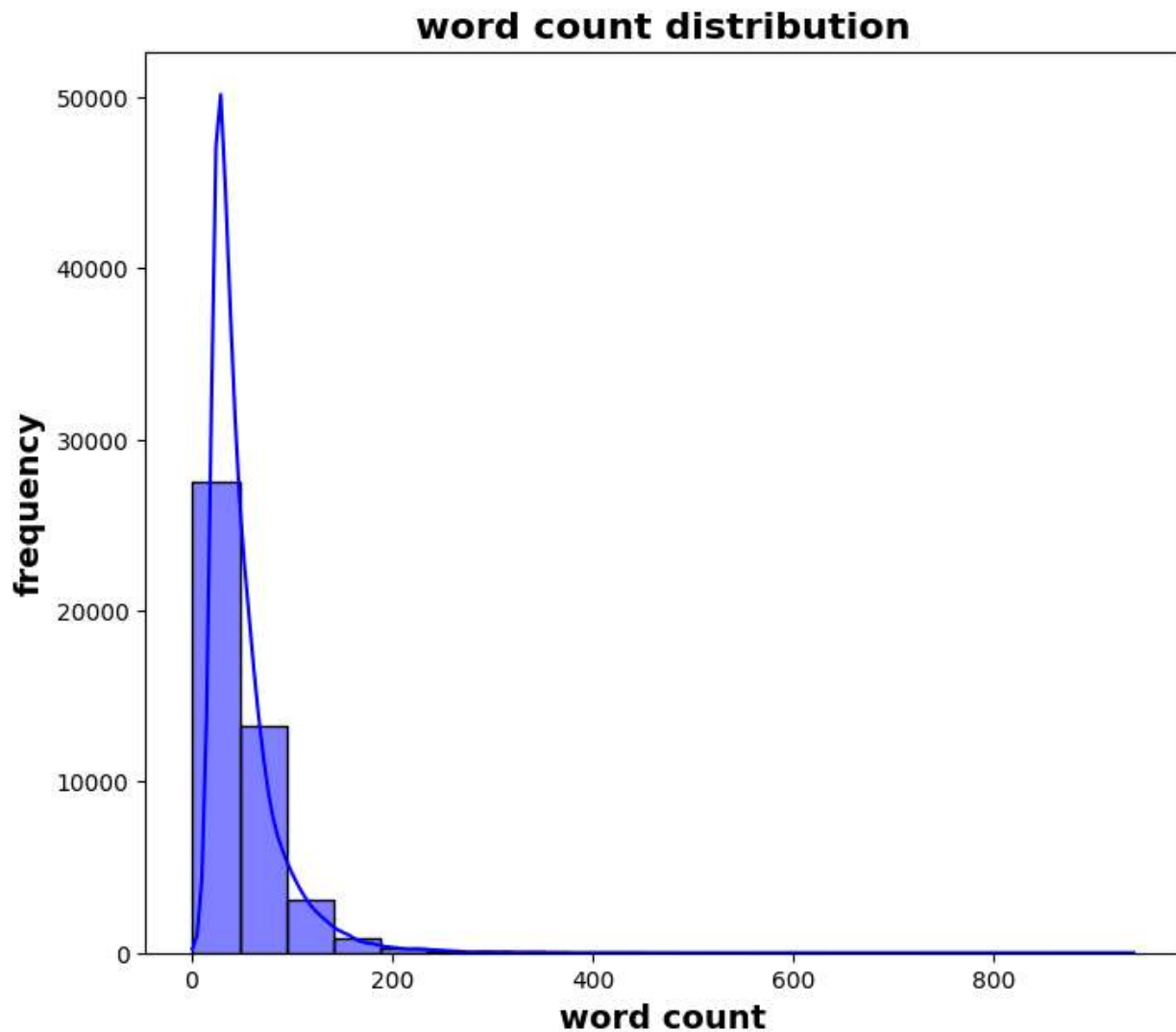


این تصاویر نمایان کننده کمالاتی هستند که بیشترین تکرار را در ژانرهای متفاوت داشته اند.

## پردازش و تجزیه و تحلیل داده‌ها

پاکسازی شامل موارد زیر است:

- حذف نام‌های اشخاص و مکان‌ها از متن با استفاده از `متد`.
  - گسترش اختصارات موجود در متن از طریق `متد` `expand_contractions`.
  - پاکسازی متن از علائم نگارشی، آدرس‌های اینترنتی، اعداد و فاصله‌های اضافی با استفاده از `متد` `clean_text`.
  - حذف کلمات پرتکرار (`stop-words`) از متن با استفاده از `متد` `remove_stopwords`.
  - لماتایز کردن کلمات به کمک `متد` `lemmatize_text`.
- سپس داده‌های پاکسازی شده به منظور حذف داده‌های تکراری بر اساس ستون `'description'` در `df_train` اعمال می‌شود. نتیجه نهایی این تمیز شده‌ها به یک فایل `csv` با نام `clean.csv` ذخیره می‌شوند.



این نمودار هیستوگرام توزیع تعداد کلمات را نشان می‌دهد. محور افقی (X) تعداد کلمات و محور عمودی (Y) فراوانی (تعداد دفعات تکرار) را نشان می‌دهد.

۱. **توزیع:** تعداد زیادی از نمونه‌ها تعداد کلمات کمی دارند (بین ۰ تا ۵۰ کلمه)، و به تدریج با افزایش تعداد کلمات، تعداد نمونه‌ها کاهش می‌یابد.
۲. **پیک:** بیشترین فراوانی در محدوده کمترین تعداد کلمات مشاهده می‌شود.
۳. **دم بلند:** دم نمودار به سمت راست کشیده شده است که نشان‌دهنده وجود تعدادی نمونه با تعداد کلمات بالاست، اما تعداد این نمونه‌ها کم است.

۴. منحنی تراکم: منحنی آبی که روی هیستوگرام قرار دارد، تراکم احتمال را نشان می‌دهد که شکل توزیع داده‌ها را بهتر نشان می‌دهد. این منحنی نیز نشان‌دهنده تجمع داده‌ها در تعداد کلمات کم و کاهش تدریجی فراوانی با افزایش تعداد کلمات است.

اکثریت نمونه‌ها کوتاه هستند: اکثر توضیحات دارای تعداد کمی کلمه هستند، که ممکن است نشان‌دهنده مختصر بودن توضیحات باشد.

نمونه‌های بلندتر کم‌تر هستند: توضیحات با تعداد کلمات زیاد بسیار کم هستند.

توزیع غیر متقارن: توزیع داده‌ها به شدت به سمت چپ متقارن نیست و دم بلندی به سمت راست دارد که نشان‌دهنده وجود نمونه‌هایی با تعداد کلمات بسیار بالا است.

## مدل‌سازی و ارزیابی

محاسبه TF-ID :

ابتدا با استفاده از توابع `compute_tf` و `compute_idf`، فراوانی تکرار کلمات در هر سند (TF) و شاخص DF معکوس (IDF) محاسبه می‌شود.

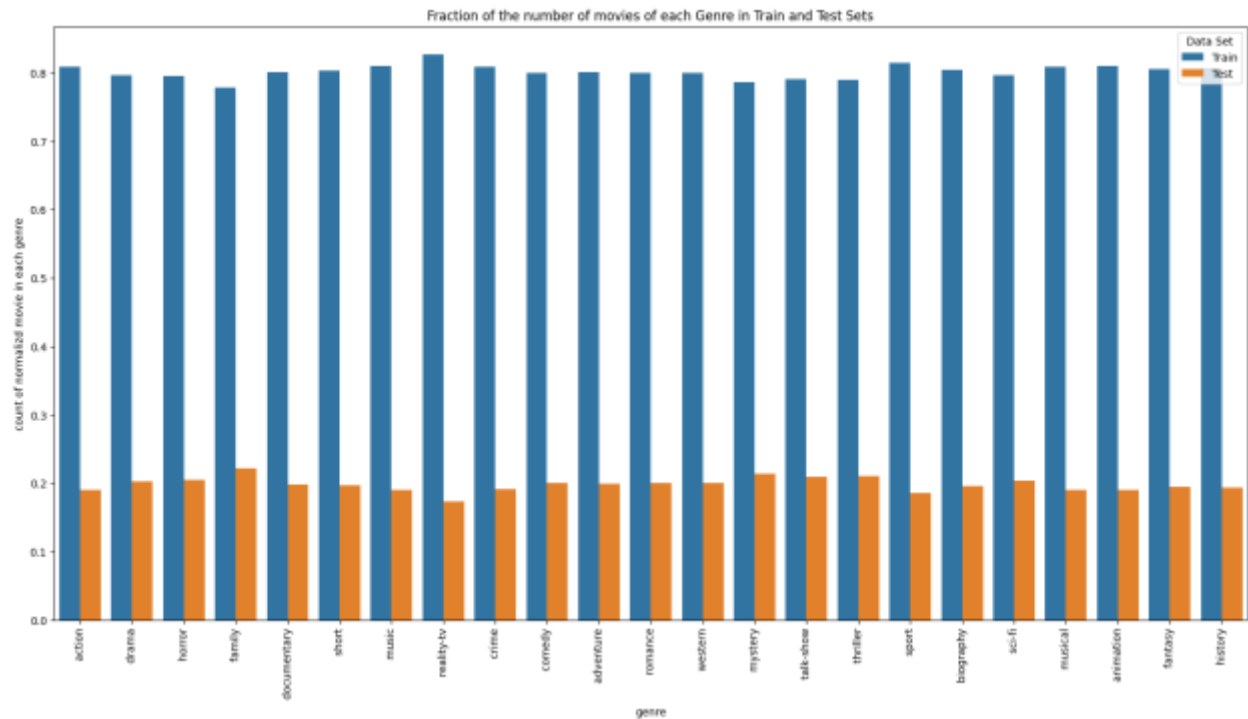
سپس با استفاده از `compute_tfidf`، مقادیر TF-IDF برای هر واژه در اسناد محاسبه و ذخیره می‌شود.

ذخیره و بارگذاری داده‌ها:

داده‌های TF-IDF به صورت Sparse Matrix ذخیره و با استفاده از فرمت `npz` ذخیره می‌شوند، تا استفاده بهینه‌تری از حافظه‌ی سیستم داشته باشند.

تقسیم داده‌گان به داده‌های آموزشی و **validation**:





این نمودار تقسیم داده هایی از ژانر های مختلف را در داده های آموزش و ارزیابی نشان میدهد.

## آموزش مدل Naive Bayes :

یک کلاس به نام `NaiveBayesClassifier` پیاده سازی شده است که مدل Naive Bayes را بر اساس

داده های TF-IDF آموزش می دهد.

محاسباتی مانند احتمال پیشین (prior) و احتمال شرطی (likelihood) برای هر کلاس محاسبه می شوند.

## ارزیابی مدل:

مدل با استفاده از داده های آزمایشی ارزیابی می شود.

معیارهای ارزیابی مانند دقت، بازخوانی، فراخوانی و اسکور  $F_1$  برای هر کلاس و به طور وزن دار برای کل داده ها محاسبه می شوند.

## تیون پارامترها:

با استفاده از تابع `tune_alpha`، پارامترهای مدل) مانند `alpha` در (Naive Bayes بهینه‌سازی می‌شوند تا عملکرد بهتری داشته باشند.

### ذخیره و بارگذاری مدل:

مدل نهایی با استفاده از کتابخانه‌ی `pickle` ذخیره و در زمان‌های دیگر مورد استفاده قرار می‌گیرد. این فرآیندها به صورت کامل و خودکار انجام می‌شوند تا یک مدل دقیق برای دسته‌بندی ژانر فیلم‌ها ارائه شود که قابلیت پیش‌بینی و ارزیابی دقیقی را دارد.

### توابع محاسباتی $TF-IDF$

#### `compute_tf(doc)`

این تابع برای محاسبه فراوانی نرمال شده ( $TF$ ) واژگان در یک سند (متن) استفاده می‌شود. متن ورودی (`doc`) را به واژگان تقسیم می‌کند و برای هر واژه تعداد تکرار آن را محاسبه می‌کند. سپس مقدار  $TF$  را برای هر واژه به ازای کل واژگان در سند محاسبه کرده و در یک دیکشنری بازمی‌گرداند.

#### `compute_idf(corpus)`

این تابع برای محاسبه شاخص  $DF$  معکوس ( $IDF$ ) واژگان بر اساس مجموعه‌ای از اسناد (`corpus`) استفاده می‌شود. متن‌های ورودی را به واژگان تقسیم می‌کند و برای هر واژه تعداد اسنادی که حاوی آن واژه هستند را محاسبه می‌کند.

$IDF$  برای هر واژه را با استفاده از تابع لگاریتم و به ازای تعداد کل اسناد وارد شده، محاسبه کرده و در یک دیکشنری بازمی‌گرداند.

#### `compute_tfidf(tf, idf)`

این تابع برای محاسبه مقدار  $TF-IDF$  برای هر واژه از ترکیب  $TF$  و  $IDF$  استفاده می‌شود.

TF و IDF دو دیکشنری هستند که ورودی‌های تابع می‌باشند و مقدار TF-IDF برای هر واژه را محاسبه می‌کند و در یک دیکشنری بازمی‌گرداند.

ذخیره و بارگذاری Sparse Matrix TF-IDF

### ذخیره sparse matrix

این بخش از کد برای ذخیره کردن یک Sparse Matrix حاوی مقادیر TF-IDF استفاده می‌شود.

استفاده از ماتریس Sparse برای ذخیره داده‌های TF-IDF که بسیار بزرگ هستند، بهینه است.

ماتریس را با استفاده از کتابخانه‌ی scipy.sparse ذخیره می‌کند.

### بارگذاری sparse matrix

این بخش از کد برای بارگذاری Sparse Matrix TF-IDF از فایل ذخیره شده استفاده می‌شود.

ماتریس Sparse را از فایل npz بارگذاری می‌کند و به یک ماتریس Sparse در Python تبدیل می‌کند.

آموزش و ارزیابی مدل Naive Bayes

### NaiveBayesClassifier

این کلاس برای پیاده‌سازی و آموزش یک مدل Naive Bayes بر اساس TF-IDF برای تشخیص ژانر فیلم‌ها استفاده می‌شود.

متغیرهای prior و likelihood برای محاسبه احتمال پیشین و احتمال شرطی به ترتیب آماده می‌کند.

### توابع محاسبه:

`calculate_priors(y_train)` محاسبه احتمال پیشین برای هر کلاس بر اساس تعداد نمونه‌های آموزشی.

`calculate_likelihoods(tfidf_df_train, y_train, alpha)` محاسبه احتمال شرطی برای هر

ویژگی و کلاس با استفاده از پارامتر `alpha` به عنوان smoothing.

`calculate_posteriors(tfidf_vector, priors, likelihoods)` محاسبه احتمال پسین برای هر

کلاس بر اساس ورودی‌های محاسبه شده.

### آموزش و ارزیابی:

`fit(tfidf_df_train, y_train)` آموزش مدل بر اساس داده‌های آموزشی.

`predict(tfidf_df_test)` پیش‌بینی برچسب‌ها بر اساس داده‌های آزمایشی.

`score(tfidf_df_test, y_test)` ارزیابی عملکرد مدل بر اساس دقت.

`evaluate(tfidf_df_test, y_test)` محاسبه معیارهای دقت، بازخوانی، فراخوانی و  $F1$  برای ارزیابی مدل.

### تیون پارامتر: $\alpha$

`tune_alpha(tfidf_df_train, y_train, tfidf_df_test, y_test, alpha_values)` تیون

پارامتر  $\alpha$  برای مدل Naive Bayes با استفاده از بهترین دقت.

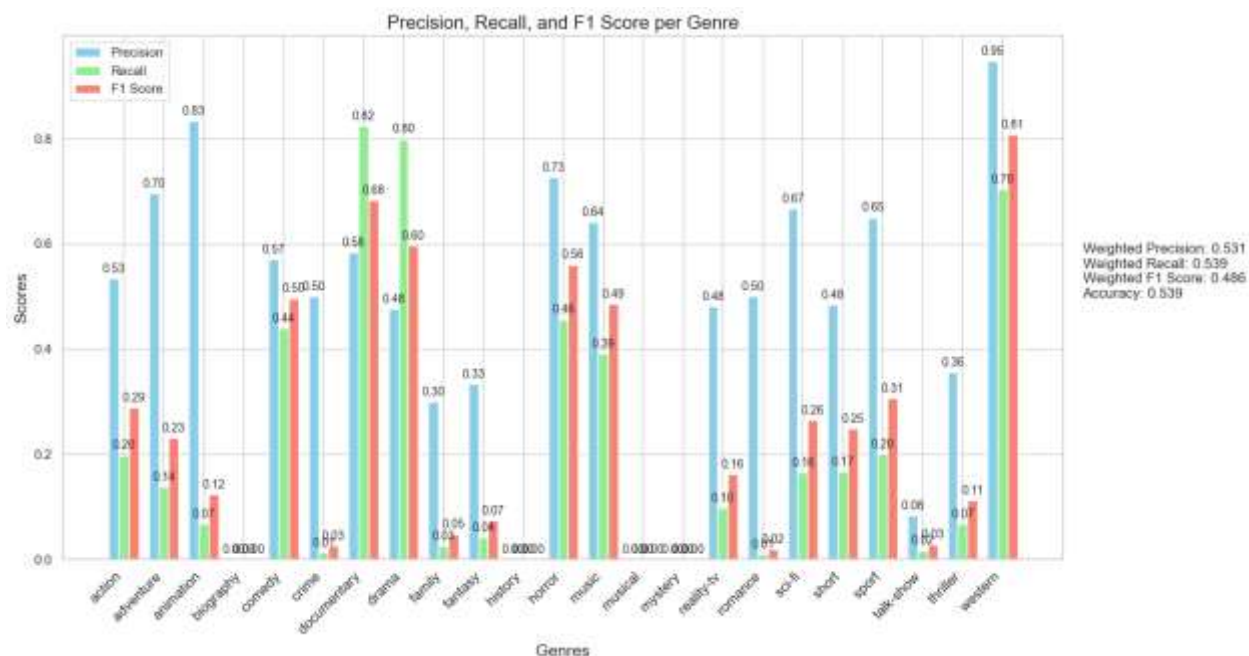
ذخیره و بارگذاری مدل

ذخیره و بارگذاری مدل:

`save_model(model, filename)` ذخیره مدل به صورت فایل با استفاده از کتابخانه `pickle`.

`load_model(filename)` بارگذاری مدل از فایل ذخیره شده به منظور استفاده بعدی.

## نتایج و نتیجه گیری



این داده‌ها مربوط به دقت، یادآوری و نمره  $F1$  برای هر دسته است. همچنین دقت وزنی، یادآوری وزنی، نمره  $F1$  وزنی و دقت کلی نیز آورده شده‌اند. نتایج به صورت زیر هستند:

دقت به ازای هر دسته:

- اکشن: ۰,۵۳۴

- ماجراجویی: ۰,۶۹۶

- انیمیشن: ۰,۸۳۳

- زندگی‌نامه: ۰,۰

- کمدی: ۰,۵۷۱

- جنایی: ۰,۵

- مستند: ۰,۵۸۳

- درام: ۰,۴۷۶

- خانوادگی: ۰,۳

- فانتزی: ۰,۳۳۳

- تاریخی: ۰,۰

- ترسناک: ۰,۷۲۶

- موسیقی: ۰,۶۴۲

- موزیکال: ۰,۰

- معمایی: ۰,۰

- واقع نما ۰.۴۸۱ (Real i ty-TV)

- عاشقانه: ۰,۵

- علمی-تخیلی: ۰,۶۶۷

- کوتاه: ۰,۴۸۵

- ورزشی: ۰,۶۵

- تاک شو: ۰,۰۸۳

- هیجانی: ۰,۳۵۶

- وسترن: ۰,۹۴۸

یادآوری به ازای هر دسته:

- اکشن: ۰,۱۹۸

- ماجراجویی: ۰,۱۳۸

- انیمیشن: ۰,۰۶۷

- زندگی نامه: ۰,۰

- کمدی: ۰,۴۳۹

- جنایی: ۰,۰۱۳

- مستند: ۰,۸۲۴

- درام: ۰,۷۹۹

- خانوادگی: ۰,۰۲۶

- فانتزی: ۰,۰۴۲

- تاریخی: ۰,۰

- ترسناک: ۰,۴۵۶

- موسیقی: ۰,۳۹۱

- موزیکال: ۰,۰

- معمایی: ۰,۰

- واقع نما ۰.۰۹۸ (Reality-TV)

- عاشقانه: ۰,۰۱

- علمی-تخیلی: ۰,۱۶۵

- کوتاه: ۰,۱۶۷

- ورزشی: ۰,۲

- تاک شو: ۰,۰۱۷

- هیجانی: ۰,۰۶۷

- وسترن: ۰,۷۰۳

نمره F۱ به ازای هر دسته:

- اکشن: ۰,۲۸۹

- ماجراجویی: ۰,۲۳

- انیمیشن: ۰,۱۲۳

- زندگی نامه: ۰,۰

- کمدی: ۰,۴۹۶

- جنایی: ۰,۰۲۶

- مستند: ۰,۶۸۳

- درام: ۰,۵۹۷

- خانوادگی: ۰,۰۴۷

- فانتزی: ۰,۰۷۴

- تاریخی: ۰,۰

- ترسناک: ۰,۵۶

- موسیقی: ۰,۴۸۶

- موزیکال: ۰,۰



- معمایي: ۰,۰

- واقع نما ۰.۱۶۳ (Reality-TV):

- عاشقانه: ۰,۰۱۹

- علمی-تخیلی: ۰,۲۶۴

- کوتاه: ۰,۲۴۸

- ورزشی: ۰,۳۰۶

- تاک شو: ۰,۰۲۸

- هیجانی: ۰,۱۱۳

- وسترن: ۰,۸۰۷

**دقت وزنی:**

۰,۵۳۱

**یادآوری وزنی:**

۰,۵۳۹

**نمره F۱ وزنی:**

۰,۴۸۶

## نقاط قوت و ضعف مدل

### ۱. نقاط قوت:

- دسته‌هایی مانند "اکشن"، "ماجراجویی"، "انیمیشن"، "ترسناک"، "مستند"، "موسیقی"، و "وسترن" دارای دقت بالایی هستند. این نشان می‌دهد که مدل در تشخیص صحیح نمونه‌های این دسته‌ها نسبتاً خوب عمل کرده است.
- یادآوری بالا برای دسته "مستند" نشان‌دهنده توانایی خوب مدل در شناسایی تعداد زیادی از نمونه‌های واقعی این دسته است.
- نمره F1 بالا برای دسته‌هایی مانند "مستند" و "وسترن" نشان می‌دهد که مدل توانسته است تعادلی خوب بین دقت و یادآوری برقرار کند.

### ۲. نقاط ضعف:

- دسته‌هایی مانند "زندگی‌نامه"، "موزیکال"، و "معمایی" دارای دقت، یادآوری و نمره F1 نزدیک به صفر هستند. این نشان می‌دهد که مدل در تشخیص نمونه‌های این دسته‌ها بسیار ضعیف عمل کرده است.
- یادآوری پایین برای بسیاری از دسته‌ها (مانند "اکشن"، "ماجراجویی"، "انیمیشن"، "کمدی"، "خانوادگی"، و غیره) نشان‌دهنده عدم توانایی مدل در شناسایی تعداد زیادی از نمونه‌های واقعی این دسته‌ها است.
- نمره F1 پایین برای دسته‌هایی مانند "جنایی"، "خانوادگی"، "فانتزی"، "عاشقانه"، و "هیجانی" نشان می‌دهد که مدل نیاز به بهبود بیشتری در این دسته‌ها دارد.

## بهبودهای پیشنهادی

### ۱. افزایش داده‌های آموزشی:

- دسته‌هایی که دارای دقت، یادآوری و نمره F1 پایین هستند، ممکن است نیاز به داده‌های آموزشی بیشتری داشته باشند تا مدل بتواند الگوهای بهتری را از آنها یاد بگیرد.

### ۲. تنظیمات مدل:

- استفاده از تکنیک‌های تنظیم مدل (مثل تنظیم هایپرپارامترها) می‌تواند به بهبود عملکرد مدل کمک کند.

- استفاده از مدل‌های پیچیده‌تر یا ترکیب مدل‌های مختلف (ensemble methods) می‌تواند نتایج بهتری را به همراه داشته باشد.
- ۳. پیش‌پردازش داده‌ها:
  - بهبود فرآیند پیش‌پردازش داده‌ها می‌تواند به مدل کمک کند تا ویژگی‌های مهم‌تری را استخراج کند.
  - استفاده از تکنیک‌های افزایش داده (data augmentation) می‌تواند به تنوع داده‌های آموزشی و بهبود عملکرد مدل کمک کند.
- ۴. ارزیابی بیشتر:
  - ارزیابی دقیق‌تر مدل با استفاده از معیارهای مختلف و بررسی خطاها می‌تواند به شناسایی نقاط ضعف و ارائه راه‌حل‌های مناسب برای بهبود کمک کند.

## تحلیل جامع‌تر:

۱. عملکرد کلی مدل:
  - با توجه به دقت وزنی، یادآوری وزنی و نمره  $F_1$  وزنی، مدل به طور کلی عملکرد متوسطی دارد. این نشان می‌دهد که مدل نیاز به بهبود دارد تا بتواند نتایج بهتری را ارائه دهد.
۲. تمرکز بر دسته‌های بحرانی:
  - دسته‌هایی با عملکرد بسیار ضعیف (مثل "زندگی‌نامه"، "موزیکال"، و "معمایی") نیاز به توجه ویژه دارند. بهبود عملکرد در این دسته‌ها می‌تواند تاثیر قابل توجهی بر عملکرد کلی مدل داشته باشد.

## نتیجه‌گیری نهایی:

- این تحلیل نشان می‌دهد که مدل در برخی دسته‌ها عملکرد خوبی دارد، اما در برخی دیگر نیاز به بهبودهای جدی دارد. تمرکز بر بهبود یادآوری و نمره  $F_1$  برای دسته‌های ضعیف و همچنین افزایش داده‌های آموزشی می‌تواند بهبود عملکرد کلی مدل را به همراه داشته باشد.