بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

فاز اول پروژه درس سامانه‌های یادگیری ماشین

عنوان:

قسمت تحلیل و پیش‌پردازش دادگان

پروژه SentiMovie

مدرس:

دکتر علی زارع‌زاده

اعضای گروه:

محمدحسین موثقی‌نیا (۴۰۰۲۰۰۹۱۹)

مهدی منوچهری (۴۰۰۲۱۱۵۹۲)

حمید مهتدی جعفری (۴۰۱۲۱۲۳۴۶)

۷ اردیبهشت ۱۴۰۲

چکیده

در این پروژه بناست تا با استفاده از دیتاست مربوط به سایت rottentomatoes تلاش کنیم دو مدل یادگیری ماشین آموزش دهیم که یکی بناست به ازای هر جمله ورودی، احساسات مثبت یا منفی آن را تحلیل کند و مدل دوم بناست مبتنی بر تعدادی از جملات ورودی که مرتبط به یک فیلم هستند، یک امتیاز که میزان محبوبیت فیلم را تعیین کند به عنوان خروجی ارائه کند. در این فاز از پروژه به تحلیل دادگان، نرمال‌سازی، حذف دادگان خارج از محدوده، تکمیل یا اصلاح دادگان از دست رفته، استخراج ویژگی، تقسیم بندی دادگان به مجموعه آموزش، اعتبارسنجی و تست و همچنین داده‌افزایی می‌پردازیم. در تک تک بخش‌های این فاز از پروژه تلاش شده است تا کدها به صورت ماژولار و مستقل توسعه پیدا کنند تا به راحتی بتوان از آن‌ها به عنوان ماژول‌های مستقل در پایپلاین پردازشی نهایی پروژه استفاده نمود.

۱ مقدمه

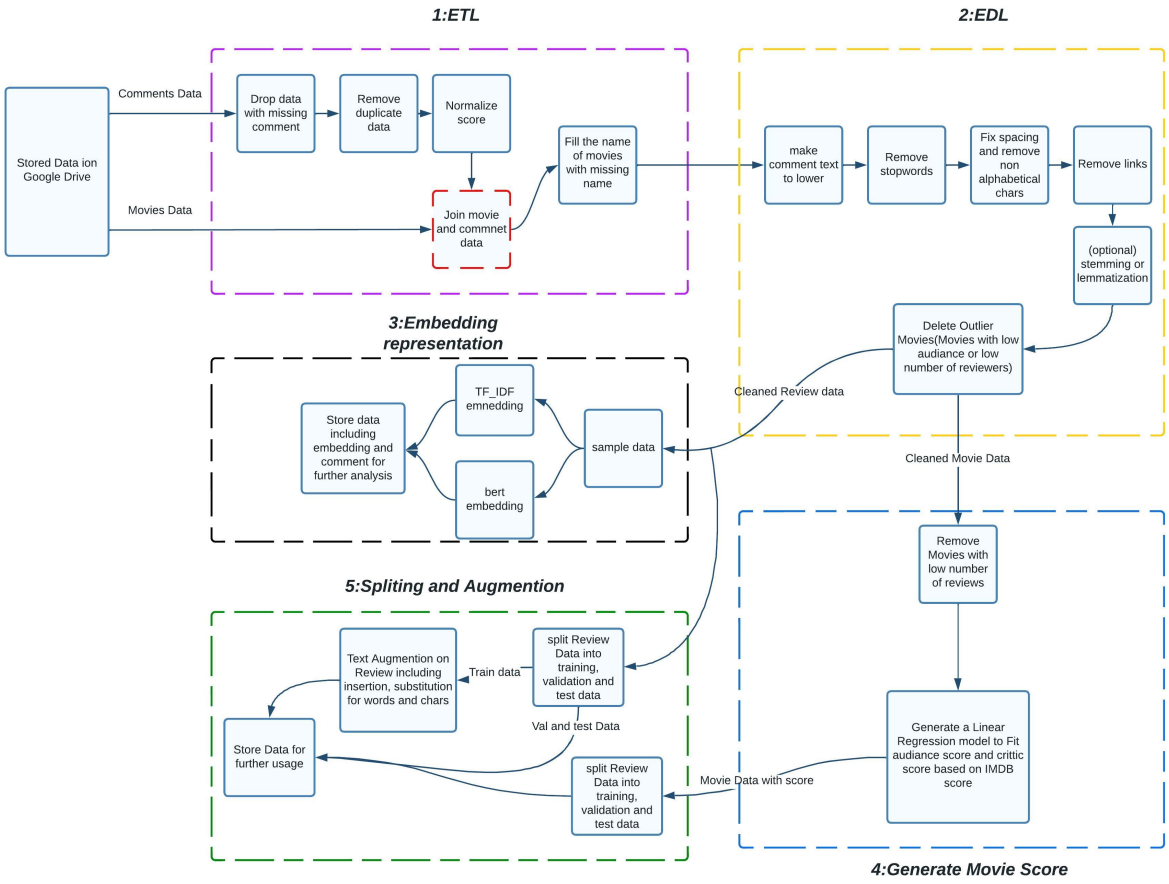
۱-۱ فرآیند پردازش

در این فاز از پروژه همانطور که در شکل۱ مشخص شده است، ابتدا یک سیستم مدیریت ورژن فایل با استفاده از DVC راه اندازی شده و همچنین ورژن کنترل git تا تمامی فایل‌ها و کدها به صورت ساختارمند و مرتبط ورژن گذاری شده و روند پردازش قابل بررسی و ذخیره سازی باشد. سپس فرآیند استخراج و تبدیل و بارگذاری داده انجام شده و وارد فاز پردازش و تحلیل شده است. در نهایت پس از انجام تحلیل‌های لازم و همچنین پاک‌سازی و نرمال‌سازی دادگان؛ فرآیند تقسیم به بخش‌های آموزش، اعتبارسنجی و تست انجام شده و بر روی دادگان بخش آموزش، عملیات داده‌افزایی[1] نیز اجرا شده است. فایل‌ها و کدهای مربوط به هرکدام از قسمت‌ها در بخش کد و داده قابل دسترس است که در ادامه به صورت مفصل توضیح داده شده است.

۱-۲ معرفی دادگان

در این پروژه از دادگان مربوط به سایت rottentomatoes استفاده شده است که در سایت kaggle در این لینک به صورت یک دیتاست استخراج شده از سایت مربوطه قرار داده شده است. سیاست سایت rottentomatoes به این شکل است که یک سری از کاربران را به عنوان کاربران عادی و یک سری را به عنوان کاربران سطح بالاتر در نظر می‌گیرد. به کاربران عادی audience و به کاربران سطح بالاتر که معمولا

[1] Augmentation



شکل ۱: پایپلاین پردازشی داده

منتقدینی هستند که تجربه بیشتری نسبت به کاربران عادی دارند، tomatometer می‌گوید. این دیتاست شامل نظرات کاربران به تفکیک فیلم، امتیازات کاربران عادی و سطح بالا به فیلم‌ها، تعداد کابران عادی و سطح بالایی که امتیاز داده اند و همچنین مثبت یا منفی بودن هرکدام از نظرات می‌باشد. در این سایت به نظرات مثبت (Fresh) و به نظرات منفی (Rotten) گفته می‌شود. این مجموعه دادگان شامل حدود یک میلیون کامنت و ۱۷ هزار فیلم متفاوت است.

۲ استخراج، تبدیل، بارگذاری داده

۲-۱ مقدمه

در این بخش هدف اصلی دریافت داده‌ها از منبع خود و سپس تبدیل و بهبود داده‌ها و در نهایت بارگذاری آن است تا در بخش‌های بعدی مورد استفاده قرار گیرد.

۲-۲ استخراج

در این بخش به عنوان داده اصلی از سایت kaggle و از این لینک استفاده می‌کنیم. برای نگهداری داده‌ها از google Drive استفاده شده است که برای نگهداری و بروزرسانی داده‌ها از ابزار کنترل DVC استفاده شده است. این داده‌ها شامل ۲ فایل csv است که فایل اول شامل کامنت‌های کاربران برای فیلم‌ها و اطلاعات مربوط به کامنت‌ها و فایل دوم شامل فیلم‌ها و جزییات آن است.

۲-۳ تبدیل

در ابتدا باید داده مربوط به کامنت‌ها خوانده شده و سطرهایی که خود متن کامنت وجود ندارد حذف شود (با توجه به اینکه امکان پر کردن کامنت‌های خالی ممکن نبوده و همچنین تعداد آن به نسبت تعداد کل محدود است) سپس کامنت‌هایی که تکراری هستند حذف شده و فقط یکی از آن‌ها باقی می‌ماند. حال باید امتیاز کامنت‌ها نرمالایز شود (با توجه به اینکه نمرات در مبناهای متفاوتی و برخی نیز به حروف مانند A+ است) در صورتی که نمره عددی باشد به مبنای ۱۰ تغییر می‌کند و اگر به حروف باشد (از F- تا A+) به عدد تبدیل شده و سپس به مبنای ۱۰ می‌رود. سپس باید دو فایل داده فیلم‌ها و داده کامنت‌های تغییر داده شده را با یکدیگر ترکیب کرد که برای این کار یک join بر روی داده مشترک این ۲ سری داده که شماره یکتای فیلم است اتجام می‌شود که خروجی یک دیتافریم شامل اطلاعات فیلم مانند امتیاز کاربران و منتقدان و کامنت‌ها و اطلاعات کامنت شامل امتیاز کامنت است. دیتافریم خروجی شامل مقادیریست که با نام فیلم در آن‌ها خالی است که در این قسمت نام فیلم پیدا شده و در دیتافریم قرار داده می‌شود این مقادیر به کمک id یکتای فیلم در اینترنت پیدا شده است و در قسمت نام فیلم قرار گرفته است.

۲-۴ بارگذاری

این بخش صرفا شامل استفاده از داده‌های خروجی بخش قبل در قالب یک دیتافریم است. در نهایت خروجی این بخش شامل یک فایل دادگان است که شامل اطلاعات فیلم‌ها و کامنت‌ها می‌باشد که در ادامه مورد استفاده قرار می‌گیرد.

۳ تحلیل اکتشافی دادگان

۳-۱ مقدمه

در این بخش به بررسی و تحلیل دادگان پرداخته شده است و همچنین پاکسازی دادگان شامل حذف دادگان خارج از محدوده [1]، تصحیح و جایگزینی برخی از دادگان از دست رفته، اصلاح برخی از ویژگی‌ها، نرمال‌سازی دادگان، استخراج ویژگی‌های جدید پرداخته شده است.

با توجه به این که دادگان این پروژه شامل دو بخش دادگان نظرات و فیلم‌ها می‌باشد، بعضی از بخش‌ها به صورت مجزا روی این دادگان انجام شده است.

برای این منظور یک کلاس مجزا با نام EDA در نظر گرفته شده است که اکثر کدهای این بخش در این کلاس پیاده سازی شده است. این کلاس در فایل EDA.py قرار گرفته است. از این کلاس در نوت‌بوک Data-processing استفاده شده است و همچنین برخی موارد بیشتر نیز در این نوت‌بوک به صورت جداگانه انجام شده است که در ادامه توضیح می‌دهیم.

۳-۲ پاکسازی دادگان مربوط به نظرات

به منظور پاکسازی دادگان متنی نظرات، ابتدا stop-wordsها را حذف کردیم و سپس با استفاده از یک ساختار مبتنی بر عبارات منظم متون را پاکسازی از علائم نگارشی و برخی ساختارهای اضافه می‌کنیم. این موارد با استفاده از clean-data به ازای هر نظر تابع clean-text اعمال شده است. در این تابع می‌توان با فعال سازی حالت ریشه‌یابی [2] یا تقلیل واژه [3]، از هرکدام از این نرمال‌سازی‌ها نیز استفاده کرد. در این فرآیند از هیچ کدام از این دو مورد استفاده نشده است؛ اما این دو فرآیند را عنوان یک هایپرپارامتر برای مسئله می‌توان در نظر گرفت.

۳-۳ پاک‌سازی دادگان فیلم‌ها

در این دادگان با توجه به این که بعضی از فیلم‌ها دارای تعداد تماشاگر بسیار کمی بوده اند یا تعداد منتقدان بسیار کمی داشته اند؛ براساس یک حد آستانه برای این موارد، که به صورت پیشفرض عدد ۵ برای حداقل تعداد منتقد و عدد ۱۰۰ برای حداقل تعداد تماشاگر در نظر گرفته شده است؛ دادگان فیلم‌ها فیلتر شده و مواردی که این حد آستانه را نداشته باشند به عنوان داده خارج از محدوده در نظر گرفته شده و حذف

[1] outlier

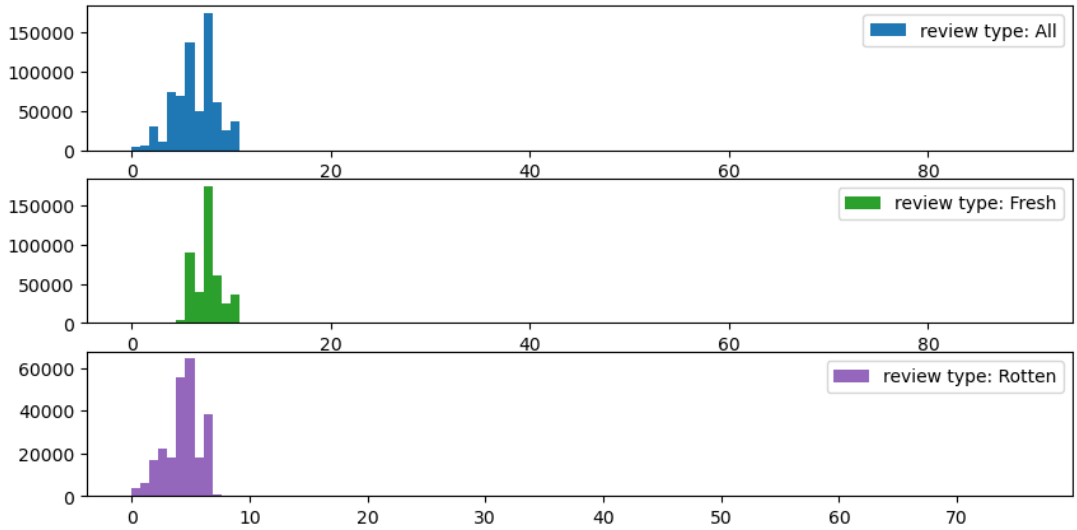
[2] Lemmatization

[3] Stemming

شده اند. این مورد با استفاده از تابع delete-outlier-movie انجام شده است. همچنین براساس معیارهای دیگری نیز این فرآیند انجام شده است که در بخش تحلیل آماری دادگان فیلم‌ها توضیحات بیشتر ارائه شده است.

۴-۳ تحلیل آماری دادگان نظرات

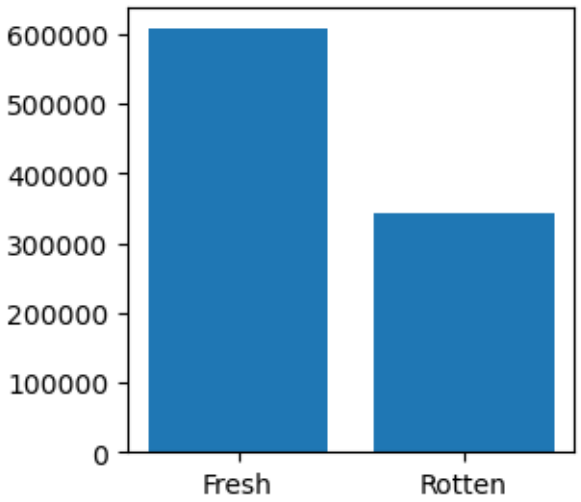
ابتدا از نظر توزیع امتیازدهی کاربران به هر فیلم متناسب با دسته مثبت یا منفی آن توزیع بررسی شده است که در نمودار ۲ قابل مشاهده است. همانطور که مشاهده می‌شود یک تفاوت نسبی بین دسته مثبت و منفی در امتیازدهی وجود دارد و می‌توان بر این اساس گفت که کیفیت نظرات به نسبت خوب هستند.



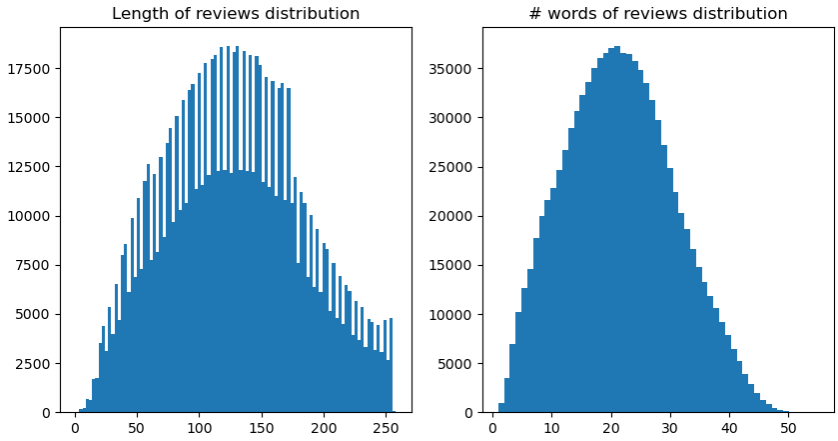
شکل ۲: نمودار توزیع امتیازدهی فیلم‌ها - نمودار بالا توزیع تمامی امتیازات می‌باشد، نمودار میانی توزیع امتیاز در دسته مثبت [۱] می‌باشد، نمودار پایین توزیع امتیاز در دسته منفی [۲] می‌باشد.

همچنین از نظر تعداد هرکدام از دسته‌های مثبت [۳] و منفی [۴] نیز بررسی انجام شده و در نمودار ۳ قابل مشاهده است که حدود ۶۴ درصد دادگان مربوط به کلاس مثبت و ۳۶ درصد دادگان مربوط به کلاس منفی هستند. از طرف دیگر از نظر تعداد کلمات و حروف نیز بررسی کلی روی دادگان انجام شده است که در نمودار۴ قابل مشاهده است. مطابق این نمودارها می‌توان نتیجه گرفت که اکثریت نظرات شامل حدود ۲۳ کلمه و حدود ۱۳۰ حرف می‌باشند. به منظور بررسی بهتر جزئیات نیز یک سری از تحلیل‌های کلی نیز بر روی کلیه دادگان و همچنین به تفکیک کلاس انجام شده است که شامل بیشینه، کمینه، انحراف معیار، میانگین و میانه تعداد کلمات و حروف می‌باشد که در قسمت Some statistic نوت‌بوک Data-processing قابل مشاهده است. همچنین چند نمونه از نظرات نیز در ادامه آن نمایش داده شده است.

[3] Fresh
[4] Rotten



شکل ۳: توزیع تعداد نظرات از دسته مثبت (Fresh) و منفی (Rotten)

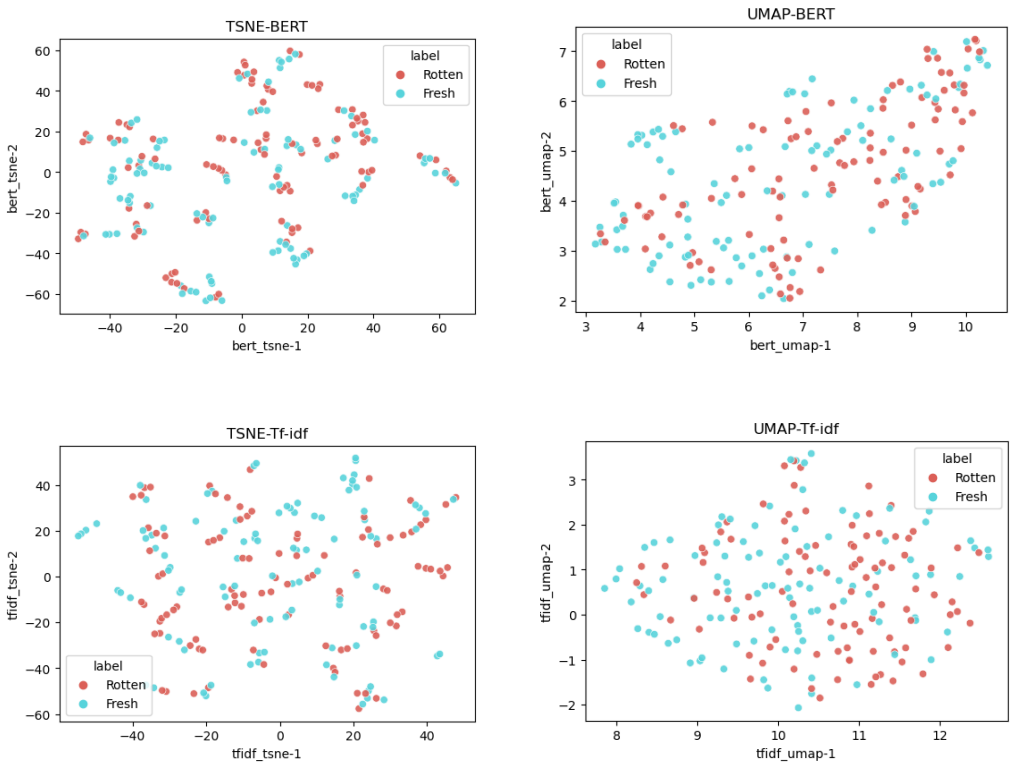


شکل ۴: نمودار توزیع کلمات و کاراکتر در تمامی نظرات موجود - نمودار سمت راست توزیع کلمات و نمودار سمت چپ توزیع حروف می‌باشد.

۳-۵ بررسی امبدینگ دادگان نظرات

به منظور بررسی امکان تفکیک پذیری دادگان کلاس مثبت و منفی با استفاده از بازنمایی بدون آموزش مدل، از دو مدل BERT از پیش آموزش دیده[1] و tf-idf استفاده شده است. هر دو مدل در نوت‌بوک Embedding قابل مشاهده است. در قسمت Bert Embedding به دلیل محدودیت در سیستم، کد را بر روی Google Colab اجرا کردیم. از هر دسته مثبت Fresh و منفی Rotten تعداد ۱۰۰ نظر نمونه برداری شده است.همچنین از مدل bert-base-uncased برای امبدینگ نظرات استفاده شده است[۱]. در قسمت tf-idf Embedding برای امبدینگ نظرات از کلاس TfidfVectorizer از کتابخانه scikit-learn استفاده شده است. پارامتر analyzer در حالت "char" تنظیم شده است که به معنی استفاده از کاراکترn-gram

[1] pre-trained



شکل ۵: نمایش بازنمایی استخراج شده با استفاده از دو مدل مختلف BERT و tf-idf

است.همچنین پارامتر n-gram محدوده اندازه‌ی n-gram را مشخص می‌کند که در اینجا بر روی (2,7) تنظیم شده است که می‌تواند در ثبت الگوهای پیچیده‌تر در متن کمک کند، مانند توالی نویسه‌های رایج یا غلط املایی. به منظور تحلیل کیفیت امبدینگ‌های خروجی با استفاده از دو روش tSNE و UMAP نمایش داده شده است (شکل ۵). همان طور که مشاهده می‌شود امبدینگ‌ها به خوبی نتوانسته اند تفکیک دو کلاس را ایجاد کنند که نتیجه اولیه این است که بایستی این مدل‌ها برای این کار fine-tune شود. در نهایت امبدینگ حاصل از هر دو مدل با فرمت json در فایل sample-embedding.json ذخیره شده است.

۳-۶ تحلیل آماری دادگان فیلم‌ها

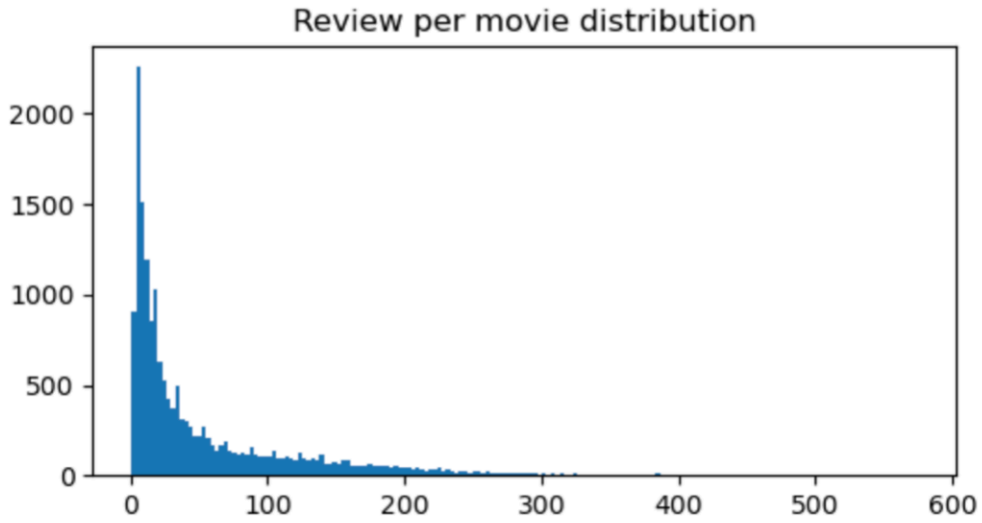
دادگان فیلم‌ها شامل نام فیلم، امتیاز کاربران عادی، امتیاز منتقدان، وضعیت امتیاز برای کاربران، وضعیت امتیاز برای منتقدین (که مشابه امتیاز است ولی شامل سه حالت دسته مختلف است) می‌باشد. در این بخش که در نوت‌بوک Data-processing بخش Exploring Movies می‌باشد، به کمک توابعی که در کلاس EDA نوشته شده است، تعداد نظرات هر فیلم را شمارش کرده و یک آمار کلی به دست می‌آید. این بخش با توجه به زمان‌بر بودن محاسبات به نحوی نوشته شده است که درصورتی که فایل محاسبات قبلی موجود باشد، دیگر محاسبات را تکرار نکرده و آن را خوانده و برمیگرداند. در غیر این صورت محاسبات را از ابتدا انجام

می‌دهد؛ همچنین یک پارامتر بازنویسی نیز می‌گیرد که این اجازه را می‌دهد که در صورت نیاز به صورت اجباری حتی در صورت وجود فایل محاسبات از قبل، محاسبات جدید را انجام دهد و جایگزین فایل قبلی کند. پس از این فرآیند با توجه به این که برخی از فیلم‌ها در قسمت‌های قبل با توجه به حد آستانه تعداد امتیازات توسط منتقدین و کاربران حذف شده اند، در این جا حدود ۱۷ هزار فیلم باقی مانده است. که توزیع تعداد کامنت‌ها برای آن‌ها مطابق شکل اول در شکل ۶ می‌باشد. همان طور که در شکل نیز مشخص است تعدادی از فیلم‌ها دارای تعداد نظرات نزدیک به صفر هستند. به همین منظور یک تابع حذف دادگان خارج از محدوده اعمال شده و فیلم‌هایی که تعداد نظرات آن‌ها کمتر از حدآستانه ورودی که به صورت پیش‌فرض عدد ۱۰ است، باشند، حذف می‌شوند. پس از حذف این دادگان خارج از محدوده، محاسبات تعداد نظرات به ازای هر فیلم دوباره انجام شده و خروجی آن در شکل دوم شکل ۶ قابل بررسی است؛ تعدادی از دادگان حذف شده اند و در نهایت تعداد فیلم‌های باقی مانده حدود ۱۲ هزار مورد شد.

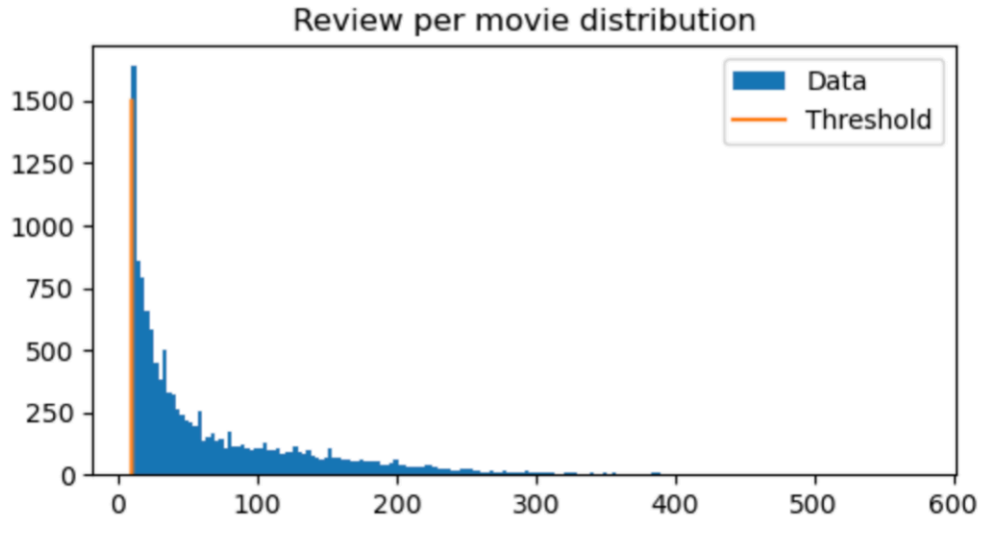
۳-۷ استخراج ویژگی جدید (امتیاز دهی به فیلم‌ها)

با توجه به این که یکی از مدل‌های یادگیری ماشینی که هدف این پروژه است، تعیین یک امتیاز محبوبیت برای فیلم‌ها مبتنی بر تعدادی از نظرات ورودی می‌باشد؛ در این بخش تلاش کردیم تا با استفاده از دو امتیاز دهی مربوط به کاربران عادی و منتقدین یک امتیاز دهی واحد به دست آوریم. دلیل این که مستقلا فقط از یکی از این امتیازدهی‌ها استفاده نکردیم این است که لزوما نظرات کاربران عادی یا منتقدین نمی‌تواند نظر اکثریت مردم باشد. به همین منظور که بررسی کنیم که آیا اختلاف این نظرات چقدر با هم جدی است یک نمودار بر اساس آماره اختلاف امتیاز منتقدین و کاربران عادی رسم شد که در شکل ۷ قابل مشاهده است. مبتنی بر این نمودار نتیجه گرفتیم که بعضا در فیلم‌هایی که تعداد کمی هم ندارند نظرات منتقدان و کاربران عادی حتی تا ۹۰ امتیاز با هم اختلاف دارد. به همین منظور از دو دم توزیع نرمال به دست آمده از هر طرف ۳۰ نمونه و از بین دو آستانه رنگی در تصویر که همان آستانه دم‌های توزیع می‌باشد نیز ۴۰ نمونه به صورت تصادفی انتخاب کردیم به این ترتیب مجموعا ۱۰۰ نمونه انتخاب شد. سپس به ازای تک تک این فیلم‌های استخراج شده امتیاز IMDB را از سایت آن به دست آوردیم و در کنار آن‌ها قرار دادیم. در نهایت یک مدل خطی از دو امتیاز منتقدان و کاربران عادی به امتیاز IMDB فیت کردیم و تابع به دست آمده را به منظور امتیاز دهی به تمامی فیلم‌ها استفاده کردیم. همچنین امتیاز نهایی نیز با توجه به این که حداکثر منتقدان و کاربران عادی عدد ۱۰۰ را در نظر می‌گیرند، بر همین اساس یک عدد بین ۰ و ۱ قرار گرفت، که ۱ به معنی بهترین و ۰ به معنی بدترین خواهد بود.

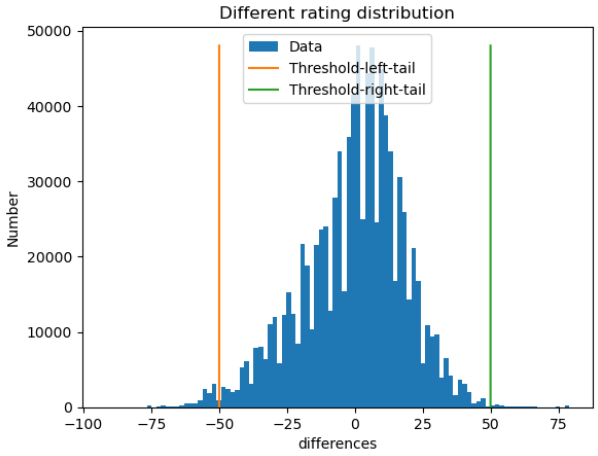
```
Max number of review: 574       [Joker]
Min number of review: 1         [A Brother's Kiss]
Mean number of review: 55
Median number of review: 24
```



```
Max number of review: 574       [Joker]
Min number of review: 10                [1776]
Mean number of review: 71
Median number of review: 41
```



شکل ۶: تصویر تعداد کامنت فیلم‌ها - شکل بالا مربوط به قبل از اعمال حد آستانه و شکل پایین مربوط به بعد از اعمال حد آستانه می‌باشد. همانطور که مشاهده می‌شود با اعمال حد آستانه ۱۰ تمامی مواردی که کمتر از ۱۰ بوده اند حذف شده اند.



شکل ۷: توزیع نرمال اختلاف مقادیر امتیاز به فیلم - شکل مربوط به اختلاف امتیاز tomatometer rating و audience rating می‌باشد. ستون افقی مقدار اختلاف امتیاز دو گروه و ستون عمودی تعداد هر فیلم را نمایش می‌دهد.

۴ تقسیم‌بندی دادگان

به منظور تقسیم‌بندی دادگان به مجموعه آموزش، اعتبارسنجی و تست، به این صورت عمل شده است که با توجه به تعداد بالای دادگان، ۱۰ درصد برای تست و از مابقی دادگان ۱۰ درصد برای اعتبارسنجی و بقیه دادگان برای آموزش استفاده شده است. این فرآیند به صورت مجزا برای دادگان نظرات و فیلم‌ها اجرا شده است و به ترتیب خروجی آن‌ها در فایل‌های مجزا ذخیره شده است. تقسیم بندی در دادگان نظرات به صورت عادی بر روی تمامی دادگان اجرا شده است ولی در دادگان مربوط به فیلم‌ها، مبتنی بر فیلم‌ها تقسیم بندی انجام شده است یعنی تقسیم بندی به شکلی انجام شده است که نظرات یک فیلم فقط در یکی از دسته‌های آموزش، اعتبارسنجی یا تست قرار گیرند و در چند دسته نباشند تا دچار نشت اطلاعات نشویم. کدهای این بخش در نوت‌بوک Spliting-and-Augmentation قابل دسترس است. همچنین جدول ۱ تعداد دادگان هرکدام از دسته‌ها را توضیح می‌دهد. در این جدول تعداد دادگان در حالت تقسیم بندی نظرات و فیلم‌ها آورده شده است و سطر میانی تعداد کلیه نظرات فیلم‌ها می‌باشد که مبتنی بر فیلم تقسیم بندی انجام شده است.

جدول ۱: دادگان تقسیم بندی شده

Test	Validation	Train	
94,919 (10%)	85,427 (9%)	768,835 (81%)	نظرات
90,588 (10%)	84,389 (9%)	749,011 (81%)	فیلم‌ها (نظرات)
1,286 (10%)	1,158 (9%)	10,414 (81%)	فیلم‌ها

۵ داده‌افزایی

با توجه به این که در مقالات مختلف نشان داده شده است که داده‌افزایی[1] باعث بهبود کیفیت عملکرد مدل‌های یادگیری ماشین می‌شود [۲، ۳، ۴] ؛ در این بخش با استفاده از پکیج nlpaug [۵] فرآیند داده‌افزایی طراحی شده است. برای این منظور یک کلاس طراحی شده که شامل توابع مختلفی است که شامل داده‌افزایی‌های مبتنی بر روش‌های مختلف از جمله: جایگزینی با هم‌معنی، جایگزینی با هم‌آوا، اضافه کردن کاراکتر براساس خطای کیبورد، اضافه کردن یا حذف کردن تصادفی بعضی از کاراکترها، ترجمه معکوس، اضافه کردن لغت جدید. می‌باشد. به منظور استفاده از این داده‌افزایی‌ها جریان‌های مختلفی را می‌توان متصور بود که بر روی یک متن خاص به ترتیب چه مواردی اعمال شود، برای همین منظور یک تابع طراحی شد که به ازای پارامتر دریافتی مشخص کند که چه مواردی باید اعمال شود یا نشود. همچنین یک تابع افزایش داده نیز تعریف شده است که در ابتدا دادگان را از کلاس‌های مختلف به ازای یک پارامتر دریافتی با هم برابر کند. به این صورت که ابتدا پارامتر دریافتی را در تعداد کلاس اکثریت ضرب کرده و سپس تمامی کلاس‌ها را به همان اندازه افزایش داده (با استفاده از سمپل گیری) می‌دهد. این نکته قابل ذکر است که داده‌افزایی صرفا بر روی دادگان آموزش اعمال می‌شود تا مدل ای که آموزش می‌بیند مقاوم تر باشد. با توجه به این که تعداد دادگان بسیار زیاد است، دو تابع اجرایی تعریف شده است که یکی به صورت پشت سر هم اجرا شده و تک به تک دادگان را بررسی می‌کند و یک تابع به صورت موازی از پردازنده استفاده می‌کند تا سرعت پردازش بیشتر شود. با توجه به محدودیت پردازنده‌های Colab ما از حالت پشت سر هم استفاده کردیم که در حدود ۴ ساعت زمان اجرا نیاز دارد. ولی از تابع موازی نیز می‌توان در حالتی که پردازنده‌های مناسبی وجود داشته باشد به منظور افزایش سرعت اجرا استفاده کرد. تابع موازی به نحوی نوشته شده است که خودش تعداد هسته‌های موجود را بررسی کرده و متناسب با آن بهترین تقسیم بندی پردازشی را انجام می‌دهد. در نهایت نیز دادگان حاصل از داده‌افزایی در فایل train-aug.pkl ذخیره شده است.

[1] Augmentation

مراجع

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol.abs/1810.04805, 2018.

[2] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol.8, pp.1–34, 2021.

[3] P. Liu, X. Wang, C. Xiang, and W. Meng, "A survey of text data augmentation," in *2020 International Conference on Computer Communication and Network Security (CCNS)*, pp.191–195, IEEE, 2020.

[4] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[5] E. Ma, "Nlp augmentation," https://github.com/makcedward/nlpaug, 2019.