بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

درس سامانه‌های یادگیری ماشین

تمرین اول (تمرین تحلیل دادگان)

پیش‌بینی دسته‌بندی آگهی دیوار

نام و نام خانوادگی:

محمدحسین موثقی‌نیا

شماره دانشجویی:

۴۰۰۲۰۰۹۱۹

فروردین ۱۴۰۲

۱. بخش استخراج دادگان (Crawling)

در این بخش ابتدا چند دسته خاص برای استخراج انتخاب شده است. دسته‌های خرید آپارتمان، خرید ویلا، خرید زمین و ملک کلنگی، خرید واحد تجاری، اتومبیل، موتورسیکلت، موبایل، لپ‌تاپ، یخچال و فریزر و ماشین لباس‌شویی انتخاب شده اند. همچنین در نظر گرفته شده است که فقط دادگانی استخراج شوند که تصویر داشته باشند و همچنین به عنوان معاوضه آگهی نشده باشند تا فضای دادگان کمی محدودتر باشد.

سپس با استفاده از کتابخانه requests آدرس آگهی‌های هر صفحه از مجموعه آگهی‌ها برای چندین شهر مختلف استخراج شده و در فایل url.csv مبتنی بر این که در چه دسته‌ای بوده است ذخیره شده است. پس از این با استفاده از کتابخانه requests و همچنین BeautifulSoup محتویات آگهی‌های هر کدام از این صفحه‌های به دست آمده در مرحله قبل استخراج شده است. با توجه به این که هرکدام از دسته‌های محصولات ویژگی‌های متفاوتی دارند و در نتیجه محل قرارگیری این ویژگی‌ها در صفحه مربوط به هر دسته متفاوت است؛ ابتدا چند نمونه از هر دسته به صورت انسانی بررسی شد و محل قرارگیری هر ویژگی به صورت تقریبی به دست آمد که در چه تگ‌هایی قرار گرفته اند. سپس با استفاده از شروطی که گذاشته شد؛ از جمله پیدا کردن تگ‌های رنگ، قیمت و ... متناسب با هر دسته برنامه نوشته شده توانسته است اطلاعات را استخراج کند.

در نهایت در بخش استخراج نیز با توجه به محدودیت درخواست به سایت دیوار؛ که در هر ثانیه فقط یک درخواست می‌توان ارسال کرد؛ یک تایمر رندم در نظر گرفته شده است تا به صورت تصادفی تاخیرهایی در اجرای حلقه ایجاد کند. البته با این وجود در بعضی موارد جلوی درخواست‌ها گرفته شد که به همین دلیل کد به صورت بازه بازه اجرا شد. (در حلقه بازه‌ای از آدرس‌ها اجرا می‌شد و سپس به صورت دستی بازه جدید در نظر گرفته می‌شد و دوباره اجرا می‌شد.) به دلیل نگهداری دادگان در حین استخراج تعداد زیادی فایل tmp در حین اجرا ایجاد شده است که در دایرکتوری data/tmp ذخیره شده اند. فایل نهایی دادگان استخراج شده با نام raw_data.csv ذخیره شده است. کدهای مربوط به این بخش در نوت‌بوک crawling.ipynb ذخیره شده است.

۲. بخش طراحی و ذخیره سازی در پایگاه‌داده

در این بخش ابتدا دادگان استخراج شده نرمال شده اند که شامل نرمال‌سازی اعداد و متون و دسته‌هاست که در این بخش دادگانی که وجود ندارند نیز مقدار ناشناس یا ۱- گرفته اند. بخشی از نرمال‌سازی نیز در قسمت تحلیل اکتشافی انجام شده است.

همچنین در این بخش دادگان به نحوی تغییر کرده که آماده ورود به جداول پایگاه‌داده شوند. که شامل جداول آگهی، دسته‌بندی، جزئیات آگهی، رنگ، نوع پردازنده، رم هستند. کدهای مربوط به آماده سازی و نرمال‌سازی در نوت‌بوک data_preparing.ipynb قابل دسترس است. و همچنین فایل‌های ساخته شده در دایرکتوری /data/db موجود است. همچنین تصویر مربوط به ارتباط جداول در فایل data/db/diagram.png قابل دسترس است. همچنین به منظور این که نتایج قابل بررسی باشد نیز در دایرکتوری /data/db/exported قابل دسترس است. جزئیات سرور دیتابیس بالا آمده نیز در فایل جیسون data/db/database.json موجود است.

دیتابیس مورد استفاده postgres بوده و با استفاده از داکر از آن از طریق pgadmin استفاده شده است. تک تک کامندهای استفاده شده برای دانلود امیج و راه اندازی پایگاه داده در فایل postgres_command.txt در کنار گزارش قرار داده شده است. همچنین کامند sql ساخت جداول و ارتباطات آن‌ها نیز در فایل sql_command.txt در کنار گزارش موجود است. همچنین کامند ورود دادگان دسته‌بند با توجه به این که ساختار سلسله مراتبی داشتند نیز در فایل category_sql_command.txt در کنار گزارش موجود است. به منظور ورود دادگان به جداول نیز از قابلیت import from csv استفاده شده است.

نوع ارتباط دسته‌بندی با توجه به این که سه سطح دارد به این صورت است که هر سطح به سطح قبلی خود متصل است و یک ساختار سلسله مراتبی می سازد. هرکدام از این دسته‌ها به صورت چند به چند به آگهی متصل است؛ به این ترتیب هر آگهی برچسب دسته‌بندی از سطوح مختلف خود را دارد.

برای مقدار رم، نوع پردازنده و رنگ نیز جداول جدایی طراحی شده تا ساختار دیتابیس نرمال باشد و همچنین جزئیات آگهی با توجه به این که لزوما در هر درخواستی نیاز نمی باشد؛ در جدول جدایی ذخیره شده است.

۳. بخش تحلیل اکتشافی دادگان (EDA)

در این بخش ابتدا دادگان متنی با استفاده از hazm نرمال شده [۱] و همچنین اعداد نیز که به صورت فارسی وارد شده اند به اعداد انگلیسی تغییر داده شدند. همچنین در میان نرمال‌سازی تمامی ایموجی‌ها از متون حذف شده است و تمامی new line ها حذف شده اند.

در بخش تحلیل دادگان در سه سطح از دسته‌بندی نمودار برحسب تعداد هر دسته رسم شده است. همانطور که مشاهده می شود در سطح اول یک نامتوازنی وجود دارد و سطح دوم نیز به همین صورت است، اما در سطح سوم همه دادگان یکنواخت است.

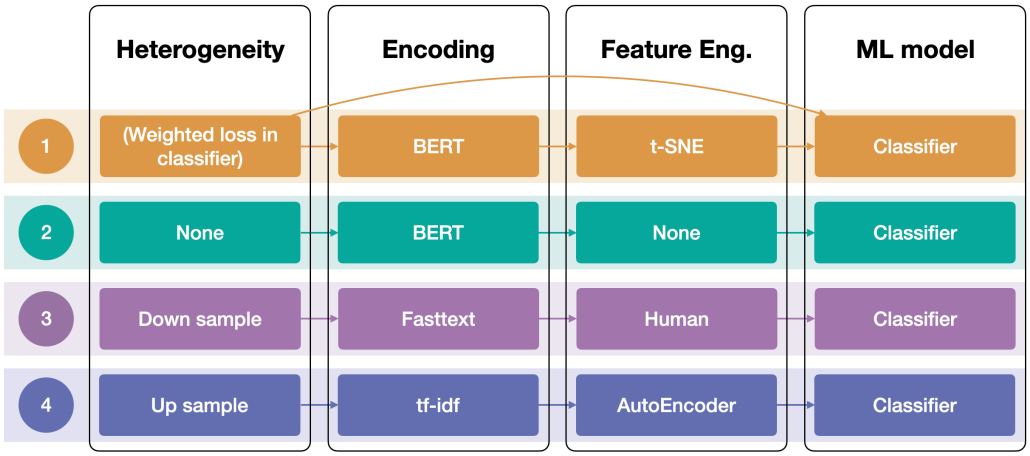
از نظر هزینه نیز توزیع به شکلی است که در بازه بیشتر از ۵ میلیارد اکثر دادگان قرار گرفته اند و بقیه موارد نیز در دسته های دیگر وجود دارد. دلیل این که این هزیه از همه بیشتر است این است که اکثر دادگان از خرید ویلا و خانه و زمین هستند که باعث شده اکثر مبالغ هزینه در این بازه عددی قرار گیرند. همچنین بازه های کمینه، بیشینه، میانگین و ... برای هزینه به دست آورده شده است. همین فرآیند برای تک تک دسته های سطح اول تکرار شده است تا بررسی شود که در دسته‌ها ناهنجاری خاصی وجود نداشته باشد. همانطور که انتظار می رفت هزینه بالای ۵ میلیارد اکثرا مرتبط با دسته خرید خانه و ویلا و ... بوده است.

همچنین توزیع دادگان برای متون عنوان و توضیحات نیز به صورت مجزا، از نظر تعداد کاراکتر و تعداد کلمات بررسی شده است و نسبت به میانگین و میانه مشخص شده است. همچنین برای بیشتر از ۲۰ کاراکتر و کلمه نیز بررسی شده است و نمودار مربوط به آن رسم شده است.

همه کدهای این موارد در نوت‌بوک EDA.ipynb قابل دسترس است. همچنین با توجه به این که بعضی از قسمت‌ها به روز رسانی شده و نرمال شده است؛ خروجی مربوطه در فایل data/ normalized_raw_data.csv ذخیره شده است.

٤. بخش آزمایش‌ها

در این بخش ٤ آزمایش مختلف به نحوی طراحی شده است که شامل تمامی حالات گفته شده در صورت تمرین باشد. همانطور که در شکل (۱) آورده شده است.



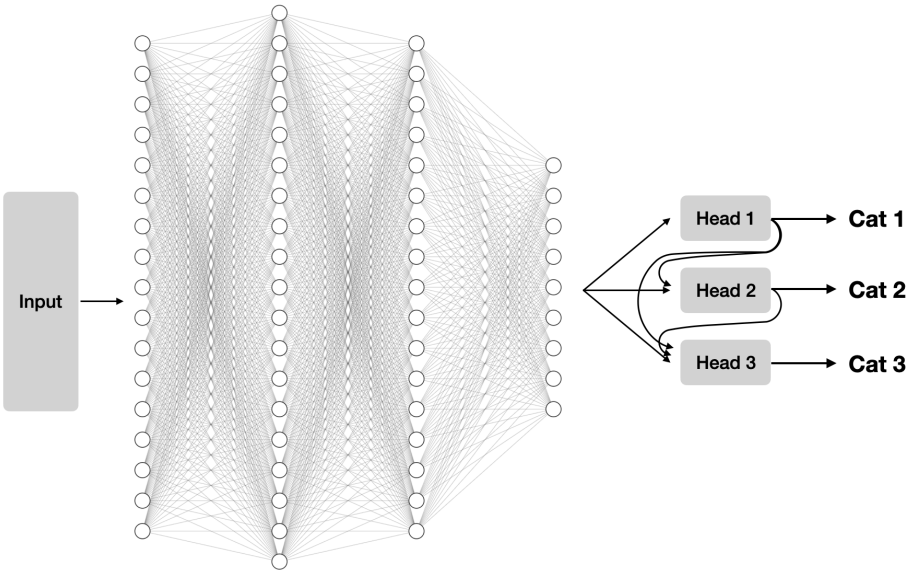
شکل (۱) – روند آزمایش‌ها؛ در ستون اول شماره آزمایش آورده شده است. در ستون دوم روش برطرف کردن ناهمگنی بیان شده است. در ستون سوم روش تبدیل دادگان مبتنی به عددی توضیح داده شده است. در ستون چهارم روش مهندسی ویژگی بیان شده است و در ستون پنجم نیز مدل دسته‌بند می‌باشد.

همانطور که در شکل (۱) نشان داده شده است؛ چهار آزمایش به این صورت طراحی شده اند که در هر بخش از تمرین شامل تمامی حالات باشند. فقط در بخش کدگذاری متن‌ها با توجه به این که سه مدل در نظر گرفته شده بود؛ در دو آزمایش اول از مدل BERT استفاده شده است. همچنین در آزمایش اول بخش برطرف کردن ناهمگنی با توجه به این که استفاده از تابع هزینه وزن دار برای کلاس‌های مختلف خود جزئی از دسته‌بند نهایی است عملا کاری که انجام شده است در بخش دسته‌بند بوده است که یک تابع هزینه وزن دار مبتنی بر اندازه هر دسته تعریف شده است. در ادامه هر کدام از آزمایش‌ها با جزئیات بیشتری توضیح داده شده است.

به صورت کلی در تمامی آزمایش‌ها برای کدگذاری رنگ، نوع پردازنده از بردار one-hot استفاده شده است.

۲.٤ مدل دسته‌بند

به صورت کلی مدل دسته‌بندی که طراحی شده است یک مدل مبتنی بر شبکه عصبی کاملا متصل[1] است که با توجه به این که دادگان دارای ۳ سطح دسته‌بندی هستند؛ این مدل نیز دارای ۳ سر خروجی است. به این شکل که سر اول خروجی تعیین کننده دسته‌بندی سطح ۱ و سر دوم دسته‌بندی سطح ۲ و سر سوم دسته‌بندی سطح ۳ را برای دادگان مشخص می‌کند. شکل (۲) نمایش شماتیک مدل طراحی شده است. البته اندازه لایه‌های مخفی با توجه به ابعاد ورودی متفاوت بوده است که در هر آزمایش به صورت مجزا توضیح داده خواهد شد.



شکل (۲) – مدل شماتیک دسته‌بند طراحی شده

هرکدام از سرها بعدی به اندازه سطح دسته مربوطه دارند، یعنی برای مثال دسته‌های سطح اول که ٤ مورد

[1] Fully connected

هستند، سر اول نیز ٤ بعد دارد و به همین ترتیب برای سرهای دیگر شبکه. همچنین ورودی سر اول صرفا از لایه مخفی قبلی است اما ورودی سر دوم با توجه به این که به نتیجه سر اول وابسته است، علاوه بر لایه مخفی قبلی، خروجی سر اول نیز با لایه مخفی قبلی کانکت شده و وارد سر دوم می‌شود و به همین ترتیب برای سر سوم هم خروجی سر اول و هم خروجی سر دوم به همراه لایه مخفی قبلی، به صورت کانکت شده وارد سر سوم می‌شوند. علاوه براین یک خروجی دیگر نیز از هر سر گرفته می‌شود که بر روی آن یک سافت‌مکس گرفته می‌شود تا به صورت احتمالاتی به دست بیاید و کلاس نهایی را مشخص کند.

همچنین تابع هزینه نیز برای هر سر به صورت مجزا تعریف شده است و در نهایت تابع هزینه کلی مطابق عبارت ۱ به صورت جمع با ضریب از هرکدام از این توابع هزینه نوشته شده است.

$$L_T = a \,.\, L_{head1} + b \,.\, L_{head2} + c \,.\, L_{head3} \qquad (۱)$$

همانطور که در عبارت ۱ بیان شده است؛ تابع هزینه کلی از جمع با ضرایب از کراس انتروپی هرکدام از سرهای شبکه عصبی به دست می‌آید که مقادیر ضرایب به عنوان هایپرپارامتر در نظر گرفته شده اند.

۲.٤ آزمایش اول

در این آزمایش با توجه به این که قرار است از تابع هزینه وزن‌دار در دسته‌بند استفاده شود؛ عملا در بخش برطرف کردن ناهمگنی فرآیند خاصی طی نشده و در بخش دسته‌بند توضیح داده می‌شود. بنابراین مستقیما به سراغ مدل تبدیل متون به داده عددی می‌رویم.

۱.۲.٤ کدگذاری دادگان متنی

در این آزمایش از مدل از پیش آموزش دیده BERT بر روی زبان فارسی استفاده شده است که تحت عنوان ParsBERT ارائه شده است[۲]. این مدل در چند ورژن مختلف ارائه شده است که در این تمرین از ورژن سوم آن استفاده شده است. خروجی این مدل ۷٦۸ بعد دارد که برای بخش‌های عنوان، زیرعنوان و توضیحات مورد استفاده قرار گرفته است و هر کدام را به صورت مجزا کدگذاری کرده است. بخش رنگ و نوع پردازنده نیز همانطور که در قسمت قبل گفته شد به صورت one-hot کدگذاری شده است. کد این بخش در نوت‌بوک Encoding.ipynb بخش BERT قابل دسترس است. همچنین خروجی این کدگذاری برای دادگان در فایل encoding/bert_em_data.json ذخیره شده است.

۲.۲.٤ مهندسی ویژگی

در بخش مهندسی ویژگی در این آزمایش از روش t-SNE استفاده شده است که با توجه به این که ما ابعاد خروجی را ۱٦ در نظر گرفته ایم؛ از الگوریتم پیش‌فرض کتابخانه sklearn یعنی الگوریتم barnes_hut نمی‌توان استفاده کرد و می‌بایست بر روی الگوریتم exact تنظیم نماییم که نکته منفی این

کار، زمانبر بودن الگوریتم exact است. در نهایت خروجی این روش برای دادگان دارای ۱۶ بعد خواهد بود. کدهای مربوط به این فرآیند در نوت‌بوک Feature_Eng.ipynb در دسترس است. همچنین خروجی این فرآیند نیز در فایل feature_eng/ex1_tSNE_data.csv ذخیره شده است.

۳.۲.٤ برطرف کردن ناهمگنی

برای برطرف کردن ناهمگنی در این آزمایش از تابع هزینه وزدار استفاده شده است که وزن مبتنی بر تعداد هر کلاس نسبت به بقیه کلاس‌ها محاسبه شده است و با کمک تابع کراس انتروپی وزن دار این فرآیند در دسته‌بند در نظر گرفته شده است. همچنین برای هرکدام از سطوح کلاس نیز تابع مجزایی تعریف شده است که براساس آن سطح وزن دهی می‌شود. کدهای مربوط به وزن دهی در نوت‌بوک آزمایش ۱ که دسته‌بند آموزش دیده است، فایل ex1_classifier.ipynb قابل دسترس است.

٤.۲.٤ مدل یادگیری ماشین (دسته‌بند)

در این بخش مدل دسته‌بند با توجه به این که بعد داده ورودی ۱۶ است، دارای سه لایه مخفی به ترتیب ٦٤، ٦٤ و ۳۲ می‌باشد و بعد از آن مستقیما لایه ۳۲ بعدی به سر اول خروجی و همین ۳۲ بعد به همراه سر اول خروجی به سر دوم و همین ۳۲ بعد به همراه سر اول و دوم به سر سوم متصل شده اند. کد مربوط به آموزش این مدل و تست آن در نوت‌بوک ex1_classifier.ipynb قابل دسترس است.

۳.٤ آزمایش دوم

در این آزمایش در بخش برطرف کردن ناهمگنی و مهندسی ویژگی هیچ کاری انجام نشده است و در بخش کدگذاری از مدل BERT و در بخش دسته‌بند نیز از مدلی متناسب با ابعاد داده استفاده شده است. در نتیجه عملا تمامی دادگان در کنار دادگان به دست آمده از مدل BERT برای عنوان، زیرعنوان و توضیحات آمده است و داده ورودی ۲۳۷۷ بعد دارد. (به دلیل این که بخش برطرف کردن ناهمگنی و مهندسی ویژگی بر روی حالت بدون انجام در این آزمایش تنظیم شده است، فایل مربوط به مهندسی ویژگی feature_eng/ex2_none_data.csv می‌باشد.

۱.۳.٤ کدگذاری دادگان متنی

در این آزمایش از مدل از پیش آموزش دیده BERT بر روی زبان فارسی استفاده شده است که تحت عنوان ParsBERT ارائه شده است[۲]. این مدل در چند ورژن مختلف ارائه شده است که در این تمرین از ورژن سوم آن استفاده شده است. خروجی این مدل ۷٦۸ بعد دارد که برای بخش‌های عنوان، زیرعنوان و توضیحات مورد استفاده قرار گرفته است و هر کدام را به صورت مجزا کدگذاری کرده است. بخش رنگ و نوع پردازنده نیز همانطور که در قسمت قبل گفته شد به صورت one-hot کدگذاری شده

است. کد این بخش در نوت‌بوک Encoding.ipynb بخش BERT قابل دسترس است. همچنین خروجی این کدگذاری برای دادگان در فایل encoding/bert_em_data.json ذخیره شده است.

۲.۳.٤ مدل یادگیری ماشین (دسته‌بند)

در این بخش مدل دسته‌بند با توجه به این که بعد داده ورودی ۲۳۷۷ است، دارای پنج لایه مخفی به ترتیب ٤۰۹٦، ۱۰۲٤، ۲٥٦، ٦٤ و ۳۲ می‌باشد و بعد از آن مستقیما لایه ۳۲ بعدی به سر اول خروجی و همین ۳۲ بعد به همراه سر اول خروجی به سر دوم و همین ۳۲ بعد به همراه سر اول و دوم به سر سوم متصل شده اند. کدهای مربوط به آموزش این دسته‌بند در نوت‌بوک ex2_classifier.ipynb قابل دسترس است.

٤.٤ آزمایش سوم

در این آزمایش ابتدا کاهش تعداد برای دسته‌های با تعداد بیشتر انجام شده است تا دادگان همگن شوند و سپس از روش fasttext برای کدگذاری دادگان متنی استفاده شده و در نهایت با شیوه انسانی مهندسی ویژگی بر روی دادگان انجام شده و به مدل دسته‌بند داده شده است.

۱.٤.٤ برطرف کردن ناهمگنی

در این بخش از کاهش تعداد نمونه ها[۲] استفاده شده است تا دادگان همگن شوند. با توجه به این که دادگان جمع آوری شده دارای ۳ سطح مختلف از دسته‌بندی هستند؛ و توزیع دادگان در هر سطح متفاوت است نمی توان به حالت توزیع یکنواخت در همه این سطوح دست پیدا کرد برای این منظور دو شکل مختلف کاهش تعداد نمونه انجام شده است اول این که به صورت تقریبی در همه سطوح بتوان به توزیع نزدیک به هم از دسته‌ها دست پیدا کرد و یک بار نیز فقط برای سطح اول دسته‌بندی انجام شد که به صورت دقیق به توزیع یکنواخت دست پیدا کردیم. به این صورت که از دسته‌هایی که تعداد بیشتری داده وجود داشت به صورت رندم نمونه گرفتیم تا به اندازه دسته‌های دیگر برسد و به صورتی که همه دسته‌ها با دسته با کمترین تعداد برابر شوند. در نهایت تعداد کل دادگان برابر ۱۱۳۷ مورد شد و در فایل downsample_cat1_raw_data.csv ذخیره شد. همچنین کدهای این بخش نیز در نوت‌بوک heterogeneity.ipynb قابل دسترسی است.

۲.٤.٤ کدگذاری دادگان متنی

در این بخش از روش fasttext استفاده شده است و بخش عنوان، زیرعنوان و توضیحات با استفاده از این روش کدگذاری شده است. برای این منظور مدل آموزش دیده شده روی متن فارسی استفاده

[2] down sample

شده است. کدهای این بخش در نوت‌بوک Encoding.ipynb بخش fasttext قابل دسترس است. همچنین نتایج این بخش نیز در فایل encoding/fasttext_em_data.json ذخیره شده است.

۳.٤.٤ مهندسی ویژگی

در این بخش از روش انسانی استفاده شده است و با بررسی‌ای که انجام شده است؛ ستون‌های زیرعنوان، سال و رنگ ارزشی در تعیین دسته‌بندی نداشته اند به همین دلیل این ستون‌ها از دادگان حذف شد. بقیه موارد هرکدام در دسته‌های مختلف از جمله مواردی هستند که بسیار تعیین کننده اند. بنابراین ابعاد داده ورودی ۹۲۱ است که ۳ مورد آن سطوح مختلف دسته‌بندی است که در زمان آموزش از داده ورودی حذف می‌شوند و به عنوان برچسب هر سطح از دسته‌بندی‌ها مورد استفاده قرار می‌گیرند. کدهای مربوطه در نوت‌بوک Feature_Eng.ipynb قابل دسترس است. همچنین داده خروجی این بخش در فایل feature_eng/ex3_human_data.csv ذخیره شده است.

٤.٤.٤ مدل یادگیری ماشین (دسته‌بند)

در این بخش مدل دسته‌بند با توجه به این که بعد داده ورودی ۹۱۸ است، دارای چهار لایه مخفی به ترتیب ۱۰۲٤، ۲۵٦، ٦٤ و ۳۲ می‌باشد و بعد از آن مستقیما لایه ۳۲ بعدی به سر اول خروجی و همین ۳۲ بعد به همراه سر اول خروجی به سر دوم و همین ۳۲ بعد به همراه سر اول و دوم به سر سوم متصل شده اند. همچنین کدهای مربوط به آموزش و تست این مدل در نوت‌بوک ex3_classifier.ipynb قابل دسترس است.

٥.٤ آزمایش چهارم

در این آزمایش ابتدا افزایش تعداد نمونه انجام شده است تا در سطح یک دسته‌بندی همه از لحاظ تعداد توزیع یکنواخت پیدا کنند. سپس با استفاده از tf-idf داده‌های متنی کدگذاری می‌شوند و پس از آن با استفاده از AutoEncoder مهندسی ویژگی انجام شده و در نهایت به دسته‌بند داده شده است.

۱.٥.٤ برطرف کردن ناهمگنی

در این بخش با استفاده از افزایش نمونه‌ها[3] از طریق دوبار ترجمه قسمت‌های متنی آگمنت شده اند و همچنین بقیه بخش‌ها مثل سال یا نوع پردازنده و مواردی از این دست براساس یک بازه و نمونه برداری رندم از این بازه اگمنت شده است. این فرآیند نیز همانند کاهش داده، بر روی دسته سطح اول اجرا شده است. کدهای این بخش نیز در نوت‌بوک heterogeneity.ipynb قابل دسترسی است. همچنین خروجی این بخش در فایل upsample_cat1_raw_data.csv ذخیره شده است.

[3] up sample

۲.۵.٤ کدگذاری دادگان متنی

در این بخش از روش tf-idf استفاده شده است و بخش عنوان، زیرعنوان و توضیحات با استفاده از این روش کدگذاری شده است. برای این منظور مدل آموزش دیده شده روی متن فارسی استفاده شده است. در نهایت ابعاد داده برابر ۲٦٦٥٦ می‌باشد. کدهای این بخش در نوت‌بوک Encoding.ipynb بخش tf-idf قابل دسترس است. همچنین نتایج این بخش نیز در فایل encoding/tf-idf_em_data.json ذخیره شده است.

۳.۵.٤ مهندسی ویژگی

در این بخش با استفاده از AutoEncoder مهندسی ویژگی انجام شده است به این صورت که از لایه میانی این شبکه به عنوان ویژگی‌های داده استفاده شده است. قسمت انکدر این شبکه کاملا متصل از لایه‌های ۸۱۹۲، ٤۰۹٦، ۲۰٤۸، ۵۱۲، ۲۵٦، ۱۲۸ و ۱٦ تشکیل شده است و قسمت دیکدر نیز برعکس همین شبکه است. لایه میانی از بعد ۱٦ می‌باشد که به عنوان انکد شده ویژگی‌های داده مورد استفاده قرار می‌گیرد. کدهای مربوطه در نوت‌بوک Feature_Eng.ipynb قابل دسترس است. همچنین داده خروجی این بخش در فایل feature_eng/ex4_autoencoder_data.csv ذخیره شده است. (کدهای این بخش با استفاده از pytorch نوشته شده است.)

٤.۵.٤ مدل یادگیری ماشین (دسته‌بند)

در این بخش مدل دسته‌بند با توجه به این که بعد داده ورودی ۱٦ است، دارای سه لایه مخفی به ترتیب ۱۲۸، ٦٤ و ۳۲ می‌باشد و بعد از آن مستقیما لایه ۳۲ بعدی به سر اول خروجی و همین ۳۲ بعد به همراه سر اول خروجی به سر دوم و همین ۳۲ بعد به همراه سر اول و دوم به سر سوم متصل شده اند. همچنین کدهای مربوط به آموزش و تست این مدل در نوت‌بوک ex4_classifier.ipynb قابل دسترس است.

٦.٤ مقایسه نتایج آزمایش‌ها

در نهایت در این این بخش به بررسی نتایج هر قسمت می‌پردازیم. همانطور که در جدول (۱) قابل بررسی است میزان دقت دسته‌بند در هر کدام از آزمایش‌ها آورده شده است.

با توجه به جدول (۱) می‌توان نتیجه گرفت که در مواردی که از امبدینگ با بعد پایین استفاده شده است عملکرد به اندازه کافی خوب نبوده است. به همین دلیل یک بار برای آزمایش اول و چهارم با اندازه امبدینگ‌های بزرگتر ٦٤ و همچنین ۱۲۸ تست شد ولی نتایج بهتر از اندازه ۱٦ نبود. به همین دلیل نتایج امبدینگ با ابعاد ۱٦ در جدول بالا گزارش شده است. (همچنین جزئیات بیشتر در نوت‌بوک‌های هرکدام از

آزمایش‌ها اضافه شده است.) از نظر عملکرد دو آزمایش دوم و سوم نتایج بهتری داشته اند؛ از طرفی از آنجایی که دسته سطح اول به دلیل کمتر بودن تعداد عملکرد بهتری داشته است و هرچه تعداد دسته‌ها بیشتر شده است عملکرد کاهش پیدا کرده است. به نظر می آید مهندسی ویژگی در این دادگان بسیار حساس است و وقتی مهندسی ویژگی به صورت دستی انجام شده ویا اصلا انجام نشده است نتایج بهتر بوده است.

	آزمایش اول	آزمایش دوم	آزمایش سوم	آزمایش چهارم
Accuracy Cat1	۰.٦٥	۰.۹۳	۰.۸۲	۰.٥٦
Accuracy Cat2	۰.٤	۰.۹	۰.۷۷	۰.٤٦
Accuracy Cat3	۰.۳۳	۰.۸۳	۰.٦۹	۰.۳۲

جدول (۱) – مقایسه میزان دقت در هر سطح از دسته‌بندی برای آزمایش‌های مختلف

۵. منابع

[1] R. (n.d.). GitHub - roshan-research/hazm: Python library for digesting Persian text. GitHub. https://github.com/roshan-research/hazm

[2] Farahani, M., Gharachorloo, M., Farahani, M., & Manthouri, M. (2021). Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, *53*, 3831-3847.