به نام خدا

گزارش تمرین ۱ درس MLSD

نام استاد:

علی زارع‌زاده

نام دانشجو:

علی عبداللهی

شماره دانشجویی:

۴۰۱۲۱۲۳۳۵

۱) خزش

در این مرحله با استفاده از chromeDriver , selenium لینک نزدیک به ۵۰۰۰۰ آگهی در شهر تهران از وب‌سایت دیوار استخراج شد و سپس داده‌های این ۵۰۰۰۰ آگهی یک به یک استخراج شد.
که این داده های برای هر آگهی به صورت : توضیحات، دسته‌بندی آگهی، نام آگهی و ویژگی‌های هر کالا یا خدمات بود.
سپس از بین آن‌ها آگهی‌هایی که نتوانسته بود داده‌های آن را استخراج کند را حذف کرده و آگهی‌هایی که نام کالا و یا دسته‌بندی نداشتند را نیز حذف کردیم.
سپس داده‌ها را داخل یک dataFrame ذخیره کرده و آن را به فرمت فایل csv ذخیره کردیم.
که کد مربوط به این قسمت در پوشه crawl موجود می‌باشد.

۲) آماده سازی داده

در این بخش همان‌طور که خواسته شده بود پایگاه داده LPostgreSQ را نصب کرده و داده‌های اولیه را در آن ذخیره کردیم و در ادامه کار نیز داده‌ها که به صورت dataFrame بود را درون این پایگاه داده ذخیره کرده و در قسمت‌های مختلف پروژه این جداول را از پایگاه داده خوانده و از آن‌ها استفاده میکردیم. همچنین همان‌طور که خواسته شده بود با استفاده از ابزار DVC داده‌ها را در هر مرحله ورژن‌گذاری کردیم و با استفاده از git سایر فایل‌های تمرین‌را ورژن‌گذاری کردیم.

که پوشه گیت مورد نظر را میتوانید در لینک زیر مشاهده نمایید:
https://github.com/aliabdollahi024/Mlops_Hw1

۲) تحلیل اکتشافی داده

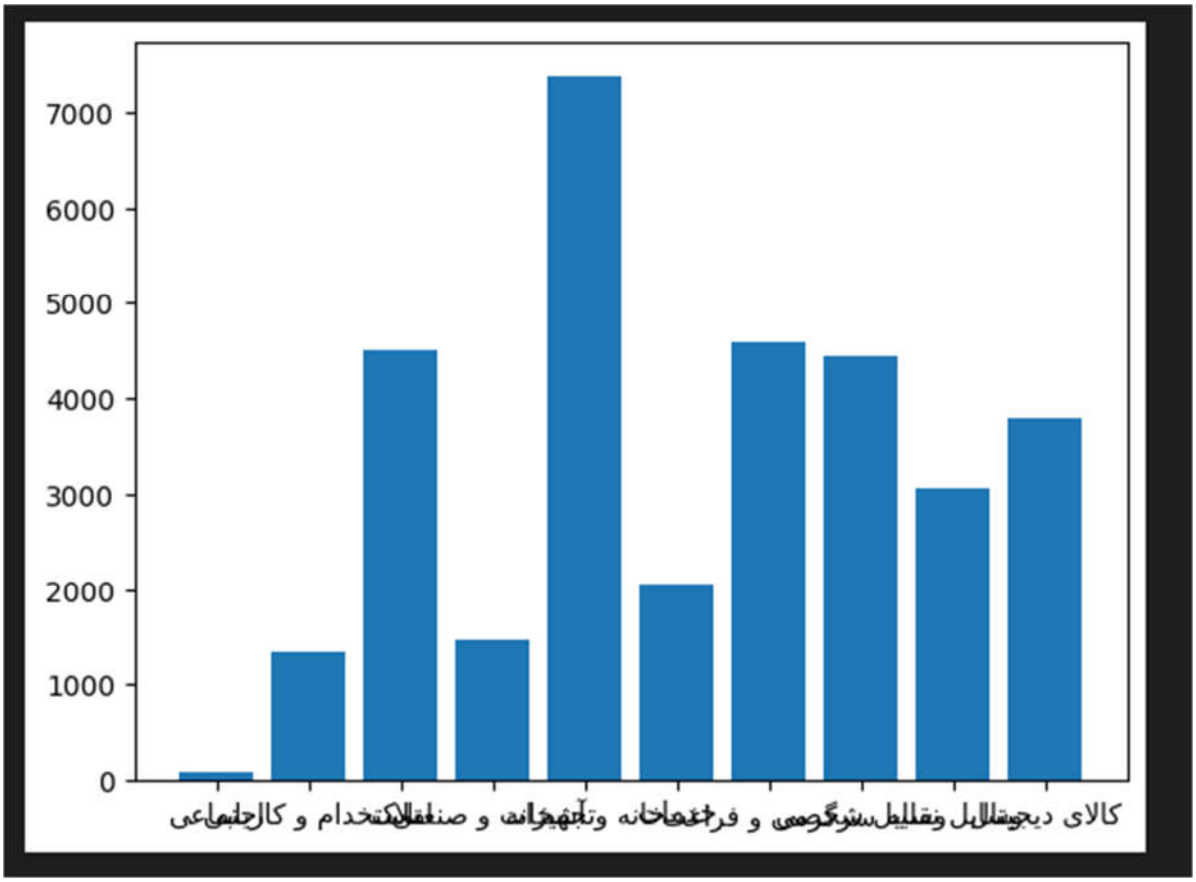
ویژگی‌هایی که برای هر آگهی استخراج شده‌اند برای اکثر آگهی های دیگر وجود ندارند بنابراین در داده‌های ما مقادیر آن‌ها nan می‌باشد بنابراین ما برای ساده‌سازی و به صورت عددی درآوردن داده‌ها ویژگی‌هایی که مقدار nan داشتند را با صفر جایگزین کرده و به سایرین مقدار یک اختصاص دادیم.

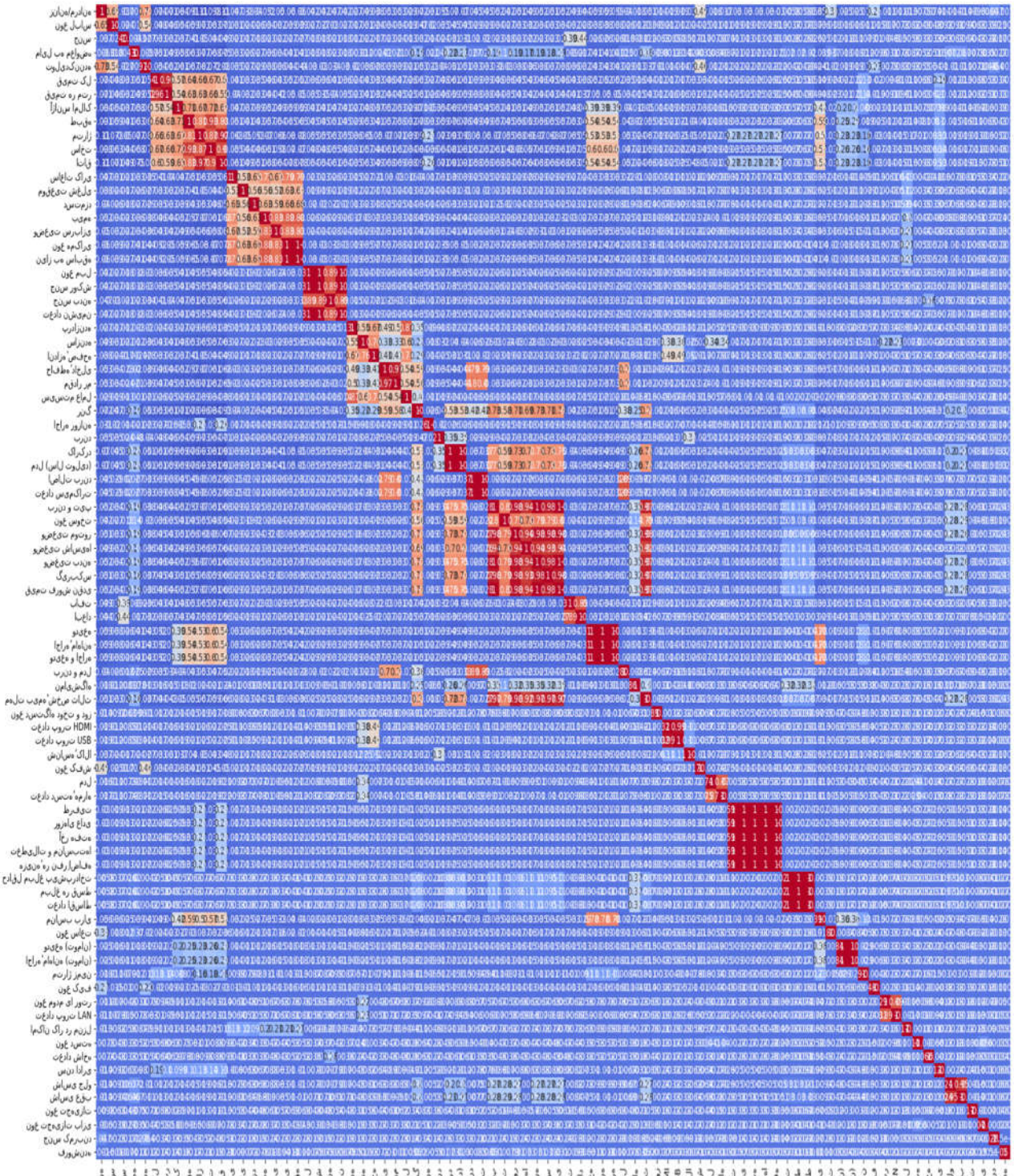
تعدادی از ویژگی ها که به نظر در دسته‌بندی کاربردی نداشتند و با سایر ویژگی‌ها correlation کمتری داشتند را حذف کردیم به مانند: لینک، نوع کالا، شیوه پرداخت، آخرین به روزرسانی آگهی، نوع کالا، وضعیت، آگهی دهنده.

و سه دسته اول هر کالا را نیز از یک‌دیگر جدا کرده و هر کدام از این سه را درون یک ستون قرار می‌دهیم.

و پس از آن دو نمودار را از روی این دادگان ساخته شده به نمایش گذاشتیم که یکی از آن‌ها تعداد آگهی‌های متعلق به هر یک از دسته‌های اصلی ده‌گانه سایت دیوار بود و نمودار دیگر correlation بین ویژگی‌های ساخته شده را نمایش میداد.

کد مربوط به این قسمت را میتوانید در پوشه plotFeautures مشاهده نمایید.


7000
6000
5000
4000
3000
2000
1000
0



۳) مهندسی ویژگی

در این قسمت برخی از ویژگی‌های آگهی را انتخاب کرده و با استفاده از آن ویژگی و مقدار آن ویژگی برای آگهی مورد نظر جملاتی را برای هر آگهی تولید کرده و این جملات را به یک‌دیگر چسبانده و یک توضیحات دیگر برای هر آگهی تولید کردیم. که کد این قسمت را میتوانید در پوشه preproccesingAndBert و فایل makeDataSet.ipynb مشاهده نمایید. و سپس با استفاده از روش‌های قسمت بعد که کدگذاری داده میباشد توضیحات اصلی و توضیحات ساخته شده را کدگزاری کرده و سپس با روش‌های TNSE , lda , pca ابعاد کدگزاری‌ها را کاهش دادیم که در قسمت بعد ترکیباتی که از این دو ساخته شده اند را توضیح خواهیم داد.

۴) کدگزاری داده

در این قسمت ابتدا دو بخش توضیحات هر آگهی که در بالا ذکر شد که به صورت متنی میباشند را نرمالایز کرده و تعدادی از preprocces های مرسوم مثل حذف stopword را روی آن اعمال کرده و به Bert، FastText ،TF-IDF داده و از خروجی هر کدام داده‌های کدگزاری شده به ازای هر آگهی را استخراج میکنیم. در مورد FastText چون به ازای هر توکن یک امبدینگ تولید میکند و ما برای کل آگهی یک

امبدینگ میخواهیم امبدینگ تمام توکن‌ها را به صورت وزن‌دار با هم جمع میزنیم که وزن آن از امبدینگ TF-IDF استخراج میشود.

و سپس روی امبدینگ‌های خروجی گرفته شده سه روش کاهش ابعاد PCA , LDA , TNSE , را اعمال میکنیم.

که ترکیباتی که در این آزمایش به کار برده شده‌اند به شرح زیر میباشند:

- کاهش بعد با pca بر روی امبدینگ خروجی bert بر روی توضیحات ساخته شده
- کاهش بعد با pca بر روی امبدینگ خروجی bert بر روی توضیحات هر آگهی
- کاهش بعد با TNSE بر روی امبدینگ خروجی bert بر روی توضیحات ساخته شده
- کاهش بعد با TNSE بر روی امبدینگ خروجی bert بر روی توضیحات هر آگهی
- کاهش بعد با PCA بر روی امبدینگ خروجی TF-IDF بر روی مجموع توضیحات هر آگهی
- کاهش بعد با PCA بر روی ویژگی‌های صرفا صفر و یک
- کاهش بعد با lda بر روی ویژگی‌های صرفا صفر و یک
- کاهش بعد با TNSE بر روی ویژگی‌های صرفا صفر و یک
- کاهش بعد با PCA بر روی امبدینگ خروجی FastText بر روی مجموع توضیحات هر آگهی
- کاهش بعد با TNSE بر روی امبدینگ خروجی FastText بر روی مجموع توضیحات هر آگهی

کد کاهش ابعاد روی خروجی bert در پوشه dimentionReductionBert میباشد و کد مربوط به خروجی گرفتن از برت در پوشه preproccesingAndBert و انتهای فایل makeDataSet.ipynb می‌باشد.
کد خروجی گرفتن و کاهش ابعاد TF-IDF در پوشه TFIDF می‌باشد.
کد خروجی گرفتن و کاهش ابعاد FastText در پوشه FastText می‌باشد.

5) نتایج خروجی مدل XGBoost بر روی داده‌های مسیرهای مختلف:

بر روی هر یک از مسیرهای بالا برای ساخت ویژگی‌های کاهش ابعاد یافته مدل دسته‌بند XGBoost را تست کرده‌ایم و به نتایج زیر رسیده‌ایم:

- کاهش بعد با pca بر روی امبدینگ خروجی bert بر روی توضیحات ساخته شده : دقت : ۰,۹۰
- کاهش بعد با pca بر روی امبدینگ خروجی bert بر روی توضیحات هر آگهی : دقت : ۰,۸۵۳
- کاهش بعد با TNSE بر روی امبدینگ خروجی bert بر روی توضیحات ساخته شده : دقت : ۰,۸۸۶
- کاهش بعد با TNSE بر روی امبدینگ خروجی bert بر روی توضیحات هر آگهی: دقت : ۰,۸۰۸

- کاهش بعد با PCA بر روی امبدینگ خروجی TF-IDF بر روی مجموع توضیحات هر آگهی
- کاهش بعد با PCA بر روی ویژگی‌های صرفا صفر و یک
- کاهش بعد با lda بر روی ویژگی‌های صرفا صفر و یک

سه مسیر بالا را با یک‌دیگر مرج کرده و خروجی گرفتیم که دقت برابر شد با: ۰,۹۱۸

- کاهش بعد با TNSE بر روی ویژگی‌های صرفا صفر و یک : دقت : ۰,۲۳۴
- کاهش بعد با PCA بر روی امبدینگ خروجی FastText بر روی مجموع توضیحات هر آگهی : دقت : ۰,۸۶۶
- کاهش بعد با TNSE بر روی امبدینگ خروجی FastText بر روی مجموع توضیحات هر آگهی : دقت : ۰,۹۳۱

کد این آزمایش‌ها را میتوانید در پوشه AugmentAndTrainModels و فایل trainAndAugment.ipynb مشاهده نمایید.

5) رفع ناهمگنی توزیع داده:

برای رفع ناهمگنی داده‌ها سه روش پیشنهاد داده شده بود که در این‌جا برای هر کدام دو مسیر از مسیرهای داده‌ای بالا را امتحان کرده‌ایم و سپس بر روی آن‌ها دسته بند XGBoost را پیاده سازی و دقت را گزارش کرده‌ایم:

کاهش داده :

- کاهش بعد با PCA بر روی امبدینگ خروجی FastText بر روی مجموع توضیحات هر آگهی : دقت : ۰,۸۹۲

سه مسیر زیر با یکدیگر مزج کرده و سپس کاهش داده را انجام داده‌ایم:

- کاهش بعد با PCA بر روی امبدینگ خروجی TF-IDF بر روی مجموع توضیحات هر آگهی
- کاهش بعد با PCA بر روی ویژگی‌های صرفا صفر و یک
- کاهش بعد با lda بر روی ویژگی‌های صرفا صفر و یک

دقت : ۰,۸۶۲

افزایش داده :

- کاهش بعد با TNSE بر روی امبدینگ خروجی bert بر روی توضیحات ساخته شده : دقت : ۰,۷۰۳
- کاهش بعد با TNSE بر روی امبدینگ خروجی FastText بر روی مجموع توضیحات هر آگهی: دقت : ۰,۹۲۱

تابع خطای وزن‌دار:

در این مورد از دسته بند logistic regression استفاده شد و به تابع خطای آن وزنی متناسب با احتمال هر یک از دسته‌ها اختصاص داده شد که متناسب با نسبت تعداد داده‌های هر دسته به کل داده‌ها آموزش انجام می‌شود.

که در این‌جا فقط یک آزمایش آن هم بر روی داده کاهش ابعاد یافته خروجی bert بر روی توضیحات ساخته شده انجام شده است که دقت آن معادل است با : ۰,۷۶۴

تمام کدهای این قسمت را می‌توانید در پوشه AugmentAndTraiModels مشاهده نمایید.