MLOPS HW1

Alireza Taherian
401206196

گزارش تمرین اول

یادگیری مفاهیم پیشرفته یادگیری ماشین

استاد

دکتر زارع زاده

گردآورنده

علیرضا طاهریان

شماره دانشجویی

۴۰۱۲۰۶۱۹۶

دانشگاه صنعتی شریف

بهار ۱۴۰۲

فهرست

خزش داده

ابتدا به منظور جمع‌آوری داده از روش crawl کردن از سایت دیوار به آدرس https://divar.ir استفاده شده که شامل مراحل زیر می‌باشد:

- جمع‌آوری لینک آگهی‌ها

بدین منظور حدود ۵۰۰۰۰ لینک از آگهی‌های صفحه‌ی نخست شهر تهران استخراج شد

- مشخص کردن ویژگی‌های استخراجی برای آگهی‌ها

در این مرحله ابتدا آگهی‌ها در دسته‌بندی‌های مختلف شناسایی و ویژگی‌های هریک مورد بررسی قرارگرفت و ویژگی‌های مهم مثل نام‌آگهی، دسته‌بندی آگهی، ویژگی‌ها، توضیحات و ... شناسایی شدند.

- استخراج ویژگی‌های شناسایی شده برای هر یک از آگهی‌ها

در این مرحله ویژگی‌های استخراج شده در مرحله‌ی قبل به کمک خزشگر استخراج و در قالب dataframe که هر آگهی یک سطر از آن باشد قرار گرفت.

کدهای مربوط به این بخش در فایل به آدرس زیر قابل مشاهده است.

https://github.com/AlirezaTH79/MLOps_HW-1/tree/main/models/crawl

آماده‌سازی داده

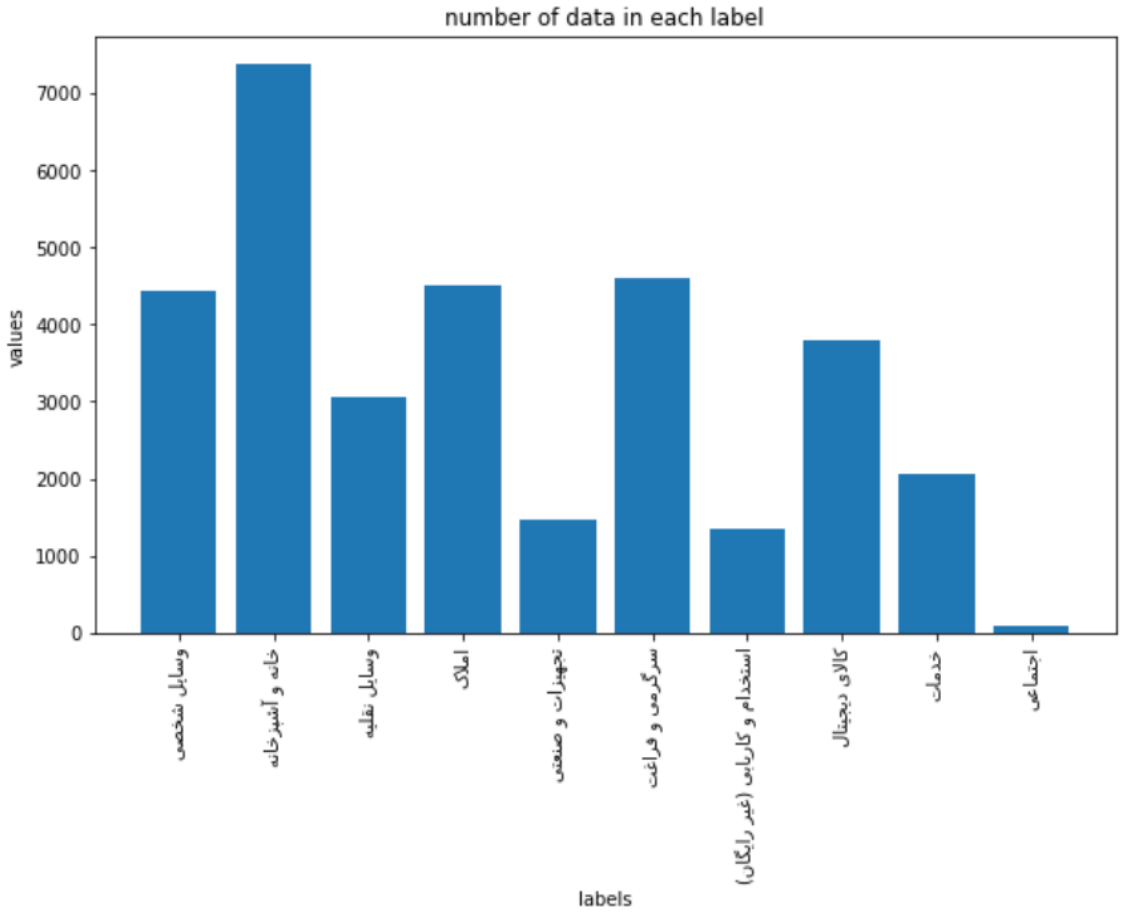
در این مرحله طبق مراحل گفته شده ابتدا PostgreSQL را نصب کرده و در هر مرحله داده‌ها را به کمک کتابخانه psycopg2 در دیتابیس ذخیره کردیم که این کتابخانه با دریافت dataframe آن را در PostgreSQL ذخیره می‌کند و تغییرات داده‌ها و مدل‌ها را به کمک ابزار DVC و git ورژن‌گذاری شده‌است.

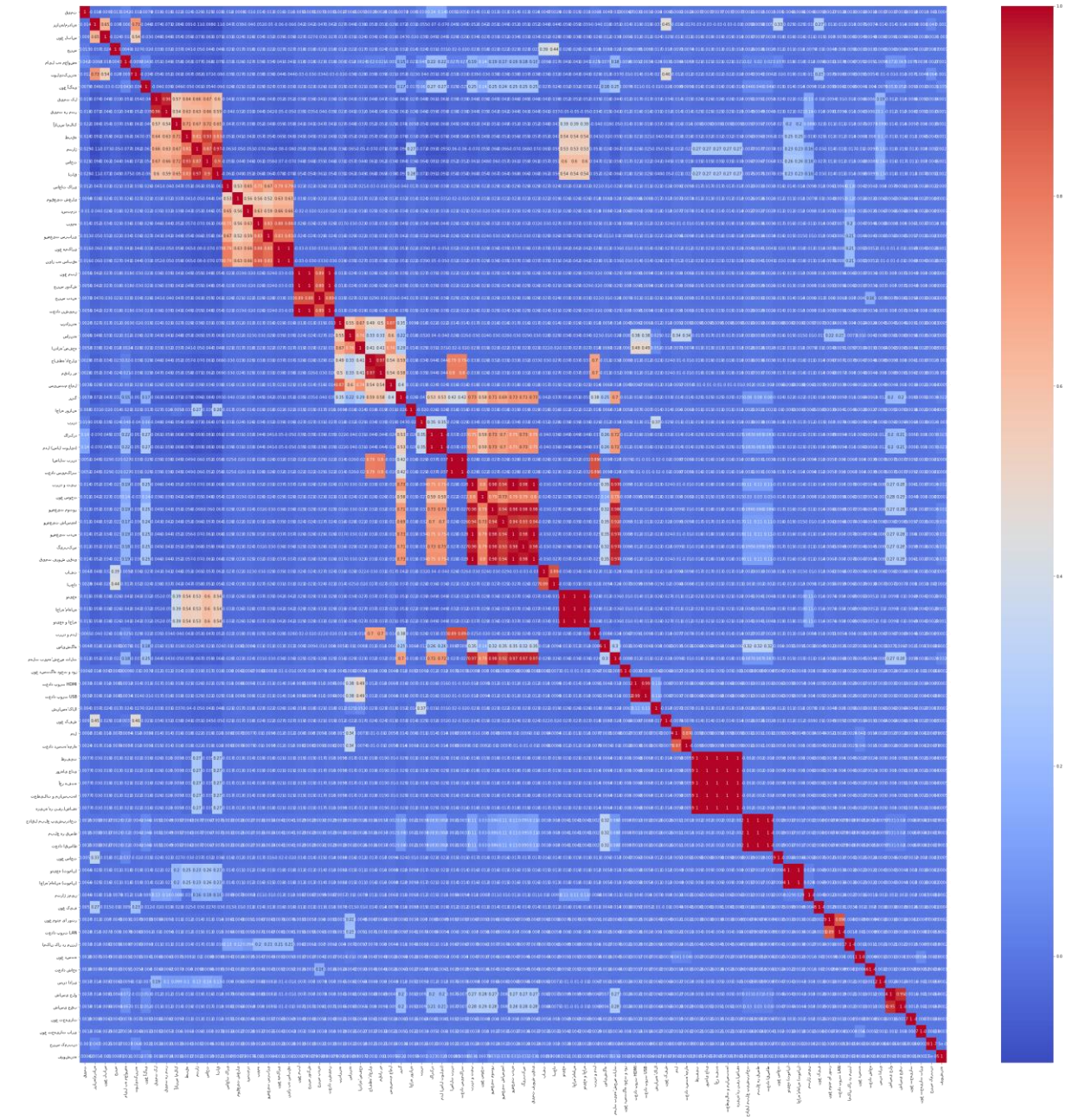
تحلیل اکتشافی داده و پاکسازی

در این مرحله ابتدا ویژگی‌هایی که در مراحل قبل crawl کرده‌بودیم را مورد بررسی قرار دادیم و داده‌هایی که دسته‌بندی آن‌ها یا عنوان آن‌ها به درستی دریافت نشده‌بودند را حذف کردیم و یکسری از ویژگی‌هایی که برای یادگیری مناسب نبودند از جمله آخرین بروزرسانی، سابقه‌ی فعالیت در خدمات دیوار و ... را حذف کردیم. سپس مقادیر ویژگی‌ها را بسته به زیربخش‌بودن به دسته‌ی cat1، cat2 و cat3 تقسیم کرده و در ۳ ستون مجزا قرار دادیم. از طرفی ویژگی قیمت که جمع‌آوری شده‌بود شامل واحد بود که آن را نیز به کمک regex به مقدار عددی تبدیل کردیم.

ویژگی‌هایی که مخصوص دسته‌های خاصی بودند و برای سایر دسته‌ها مقدار nan داشتند را نیز فارغ از مقدار آن‌ها تبدیل به ۰ و ۱ کردیم زیرا مساله‌ی ما دسته‌بندی آگهی‌ها بود که نیازی به مقدار آن ویژگی‌ها برای تشخیص نداشت.

سپس در یک نمودار تعداد آگهی‌های هریک از دسته‌ها را به نمایش درآوردیم و به منظور بررسی میزان وابستگی ویژگی‌ها به یکدیگر، ماتریس correlation را چاپ کردیم. که نمودارهای مربوطه در زیر آمده‌اند.


number of data in each label
values
7000
6000
5000
4000
3000
2000
1000
0
وسایل شخصی
خانه و آشپزخانه
وسایل نقلیه
املاک
تجهیزات و صنعتی
سرگرمی و فراغت
استخدام و کاریابی (غیر رایگان)
کالای دیجیتال
خدمات
اجتماعی
labels



کدهای مربوط به این بخش در فایل به آدرس زیر قابل مشاهده است.

https://github.com/AlirezaTH79/MLOps_HW-1/blob/main/models/Preprocessing_divar.ipynb

مهندسی ویژگی و کدگذاری داده

در مرحله‌ی مهندسی ویژگی به منظور ایجاد ویژگی ترکیبی، ویژگی‌های ۰ و ۱ ایجاد شده در مرحله‌ی قبل را به کمک قالب‌های مشخص بصورت جمله درآورده و به عنوان ویژگی description2 به داده‌ها اضافه کردیم. سپس در مرحله‌ی کدگذاری داده، ویژگی‌های description و description2 را به هم متصل کرده و کدگذاری به کمک TF-idf، parseBERT و FastText روی آن‌ها صورت گرفت. (در روش کدگذاری به کمک TF-idf، ابتدا مراحل normalization، sentence tokenization ، word tokenization، حذف stopword ها و حذف emoji ها بر روی descriptionها به کمک کتابخانه hazm صورت گرفت. در نهایت به دلیل حجم زیاد embedding خروجی TF-idf، کلماتی که دارای فرکانس کمتر از ۱۰ در تمام description ها بودند را حذف کردیم.)

پس از کدگذاری به منظور کاهش ابعاد، از روش‌های PCA، LDA و TNSE به شرح زیر استفاده شده‌است.

- PCA بروری embedding خروجی TF-idf
- LDA برروی embedding خروجی parseBERT
- TNSE برروی embedding خروجی FastText

نتایج دسته‌بند برروی flow های مختلف

به منظور جبران ناهمگنی توزیع داده روش‌های زیر ارائه شده‌است که نتایج اعمال آن‌ها در flow های مختلف آمده‌است.

- کاهش داده

در این روش ابتدا داده‌های دسته آخر که دارای فراوانی بسیارکمی (۹۴ تا) بودند را حذف کرده و سپس از هر دسته به اندازه‌ی داده‌های دسته با کمترین فراوانی، نمونه‌برداری کردیم.

- افزایش داده

در این روش ابتدا داده‌های دسته آخر که دارای فراوانی بسیارکمی (۹۴ تا) بودند را حذف کرده و سپس تعداد داده‌های هر دسته را به اندازه‌ی داده‌های دسته با بیشترین فراوانی، به کمک روش interpolation افزایش دادیم.

- تغییر تابع loss

در این روش برای هر دسته متناسب با احتمال آن دسته، وزنی برای تابع loss دسته‌بند logistic regression در نظر گرفتیم.

Flow های مختلفی که در این پروژه انجام گرفته به شرح زیر هستند:

- Flow1: TF-idf + PCA + logistic & random forest & xgboost
- Flow2: TF-idf + PCA + logestic & random forest & xgboost + upsampling
- Flow3: Parsbert + LDA + Logistic & random forest & xgboost
- Flow4: Parsbert + LDA + Logistic & random forest & xgboost + downsampling

- Flow5: FastText & TSNE & Logistic & random forest & xgboost
- Flow6: FastText & TSNE & weigthed loss logistic regression

که دقت های آنها در جدول زیر آمده است.

	Logistic Regression	Random Forest	XGBoost
Flow1	0.842	0.895	0.903
Flow2	0.818	0.891	-
Flow3	0.833	0.874	0.868
Flow4	0.814	0.829	0.824
Flow5	0.689	0.935	0.932
Flow6	0.649	-	-