دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

یادگیری ماشین

پاییز ۱۴۰۴

استاد: دکتر علی شریفی زارچی
مسئول پروژه: امیررضا آذری

پروژه مهلت ارسال نهایی: ۳۰ دی

- پروژه، ۱+۳ نمره درس می‌باشد.
- ابتدا، با دقت کامل، تمامی پروژه‌ها را مطالعه بفرمایید، سپس در فرم قرار داده شده، اولویت خود را از یک تا نه مشخص بفرمایید تا یک پروژه به شما تعلق بگیرد. توجه بفرمایید ظرفیت هر پروژه، **تنها ۷ تیم** می‌باشد.
- **معیار تصمیم‌گیری نهایی برای هر تیم، زمان پر کردن فرم می‌باشد. بنابراین این کار را به زمان دیگر موکول نکنید زیرا در این صورت اولویت‌های پایانی به شما خواهند رسید و جای اعتراضی وجود نخواهد داشت.**
- یک نمره امتیازی پروژه، شامل 0.5 نمره **بخش رقابتی پروژه** و 0.5 نمره دیگر برای **بخش خلاقیت** می‌باشد.
- **بخش رقابتی:** هر پروژه، یک معیار رقابت دارد. تیم اول 0.5، تیم دوم 0.3 و تیم سوم 0.2 نمره اضافه بر روی نمره پروژه و درس دریافت خواهد کرد. تیم‌های برتر بر اساس نظر مسئول مربوط به هر پروژه انتخاب خواهند شد.
- **بخش خلاقیت:** در این بخش هرگونه خلاقیت و ایده نو و جدید که قابل بحث و تاثیرگذار باشد، نمره خواهد گرفت. تشخیص میزان خلاقیت با مسئول پروژه انتخاب شده می‌باشد.
- هر پروژه مسئول مشخص دارد که بعد از مشخص شدن پروژه‌های تیم‌ها، معرفی خواهند شد. شما در حین انجام دادن پروژه با ایشان در ارتباط خواهید بود و هر گونه ابهام و سوالی را می‌توانید با ایشان در میان بگذارید.
- **گزارش نهایی:** گزارش نهایی هر پروژه متفاوت می‌باشد. اما در تمامی گزارش‌ها باید میزان همکاری اثرگذار هر شخص مشخص شده باشد. همچنین در صورت استفاده از ابزارهای LLM ، بخشی که از ابزارها استفاده شده است باید مشخص شود و همچنین پرامپت استفاده شده ذکر شود.
- هر تیم موظف است یک مخزن گیت‌هاب برای پروژه خود ایجاد کند. نکته حیاتی این است که هر عضو تیم باید منحصراً با اکانت شخصی خود فعالیت کند و تمام تغییرات و ارسال کد (Commit) خود را به صورت جزئی و مستمر در مخزن ثبت نماید. میزان همکاری اعضا در تاریخچه کامیت‌ها به دقت بررسی خواهد شد و انتظار می‌رود که مشارکت تمامی اعضای تیم در پیشبرد پروژه عادلانه و متوازن باشد؛ چرا که این شفافیت، بخشی از فرآیند ارزیابی نهایی خواهد بود.
- **توجه بفرمایید هرگونه عدم تطابق تحویل و گزارش تیم‌ها با نکات ذکر شده توسط مسئول پروژه، سبب نمره منفی خواهد شد.**
- دقت بفرمایید نمره هر پروژه از ۱۰۰ می‌باشد، اما بعضی پروژه‌ها بنا به سختی آن‌ها، نمرات امتیازی دارند که این نمره امتیازی، تنها بر روی خود نمره پروژه اعمال خواهد شد. به این معنی که نمره ۱۱۰ با ۱۰۰ فرقی نخواهد داشت.

پروژه‌ها

پروژه شماره ۱ (۱۰۵ نمره)

تعریف مسئله

سیگنال‌های الکتروانسفالوگرام (EEG)

سیگنال‌های EEG ثبت‌کننده‌ی فعالیت الکتریکی مغز هستند که از طریق الکترودهایی روی پوست سر اندازه‌گیری می‌شوند. این سیگنال‌ها اطلاعات ارزشمندی درباره‌ی وضعیت‌های ذهنی، فعالیت‌های شناختی و عملکرد بخش‌های مختلف مغز فراهم می‌کنند و به‌طور گسترده در علوم اعصاب، تشخیص‌های بالینی و واسط‌های مغز–رایانه مورد استفاده قرار می‌گیرند.

در این پروژه، با داده‌های EEG آشنا می‌شویم و روش‌های پیش‌پردازش، حذف نویز و تمیز سازی داده‌ها را بررسی می‌کنیم. سپس به استخراج ویژگی‌های مناسب از سیگنال‌ها پرداخته و داده‌ها را دسته بندی و خوشه‌بندی خواهیم کرد.

تصویرسازی حرکتی (Motor Imagery)
تصویرسازی حرکتی فرآیندی است که در آن از فرد خواسته می‌شود بدون انجام یک حرکت، صرفاً انجام آن حرکت را در ذهن خود تصور کند؛ برای مثال تصور حرکت دست یا پا. در این پروژه، از داده‌های EEG ثبت شده از افراد سالم در حین انجام تصویرسازی حرکتی استفاده می‌کنیم. سیگنال‌های EEG با قرار دادن ۵۹ الکترود روی نواحی مختلف جمجمه اندازه‌گیری شده اند و تغییرات پتانسیل الکتریکی مغز را به‌صورت تابعی از زمان نشان می‌دهند. مجموعه داده‌های مورد استفاده، دارای سه کلاس «دست چپ»، «دست راست» و «پا» می‌باشد. در طی ثبت داده‌ها، از نشانه‌های بصری (مانند فلش‌ها) برای مشخص‌کردن نوع حرکت مورد تصور استفاده شده و فرد مورد آزمایش، در بازه‌های زمانی مشخص، تصویرسازی حرکت نمایش داده شده را انجام داده است.
انتظار می‌رود هنگام تصور یک حرکت خاص، الگوهای متمایزی در فعالیت نواحی مختلف مغز ایجاد شود که بتوان آن‌ها را در سیگنال‌های EEG تشخیص داد. هدف این پروژه، پردازش سیگنال‌های EEG و پیش‌بینی نوع حرکت تصور شده در هر پنجره‌ی زمانی با استفاده از روش‌های یادگیری ماشین است.

ساختار داده‌های ورودی
فایل دیتاست را از این لینک دانلود کنید. در ادامه از فایل BCICIV_calib_ds1a.mat استفاده می‌کنیم که فرآیند تصویرسازی حرکتی را توسط دو کلاس «دست چپ» و «پا» انجام داده است. برای درک ساختار داده‌های مورد استفاده به این لینک مراجعه کنید.

۱. پیش پردازش داده (۳۰ نمره)
۱.۱. (۱۵ نمره) بارگذاری داده
فرآیند تصویرسازی حرکتی برای یک فرد به صورت پشت سر هم و مطابق توضیحات ارائه شده در لینک توضیحات صورت می‌گیرد. به منظور تفکیک بازه‌های زمانی متناسب با هر حرکت، نیاز است که سیگنال پیوسته‌ی ورودی را پنجره بندی کنیم. طول هر پنجره‌ی زمانی را با توجه به بازه‌ی نمایش نشانه‌های بصری و فرکانس نمونه برداری به دست آورده و از متغیر pos به عنوان نقطه‌ی شروع هر پنجره استفاده کنید.
کدی بنویسید که بخش‌های مورد نیاز از دیتاست را خوانده و پردازش کند. خروجی این مرحله باید یک دیتاست در قالب معمول مسائل یادگیری ماشین باشد که فیچرها و برچسب‌های متناظر با هر نمونه را شامل می‌شود. ۷۵٪ داده‌ها را به آموزش و ۲۵٪ را به تست اختصاص دهید.
نمودار مربوط به یکی از نمونه‌های دیتاست به دست آمده را به ازای چنل‌های ۰، ۱۵، ۳۰، ۴۵ و ۵۹ به صورت تابعی از زمان رسم کنید.
۲.۱. (۵ نمره) فیلترهای زمانی
سیگنال‌های مغزی بر اساس بازه‌های فرکانسی به چندین دسته تقسیم می‌شوند. در کاربرد فعلی، سیگنال‌های Mu، در بازه‌ی فرکانسی ۸ تا ۱۳ هرتز و سیگنال‌های Beta در بازه‌ی فرکانسی ۱۳ تا ۳۰ هرتز بیشترین کاربرد را دارند. با اعمال یک فیلتر band-pass، اطلاعات فرکانسی مربوط به این دو دسته را حفظ کرده و باقی فرکانس‌ها را حذف کنید.
۳.۱. (۱۰ نمره) استخراج ویژگی‌ها
با توجه به نزدیکی مکانی الکترودها، سیگنال‌های چنل‌های مختلف، در سطح جمجمه با یکدیگر ترکیب می‌شوند. به این منظور از فیلترهای مکانی مختلفی برای استخراج ویژگی‌ها و کاهش بُعد استفاده می‌شود. یکی از روش‌های متداول در کاربرد فعلی الگوریتم common spatial patterns (CSP) می‌باشد. روش کار این الگوریتم را به صورت کامل توضیح داده و آن را بر روی داده‌ها اعمال کنید.
توسط TSNE، نمودار scatter plot نمونه‌ها را در فضای دو بُعدی، قبل و بعد از اعمال این الگوریتم رسم کرده و عملکرد این گام را در استخراج ویژگی‌ها بررسی کنید.

۲. دسته بندی (۶۰ نمره)
۱.۲. (۴۰ نمره) پیاده سازی Kernel SVM
در این بخش، دیتاست پردازش شده‌ی حاصل از مراحل قبل را دسته بندی می‌کنیم. به این منظور روش SVM با کرنل RBF را پیاده‌سازی کرده و توسط آن دسته بندی دو کلاسه انجام دهید. برای اطمینان از صحت پیاده سازی خود می‌توانید عملکرد نهایی مدل خود را با نسخه‌ی آماده از scikit-learn مقایسه کنید.
۲.۲. (۱۰ نمره) ارزیابی

به منظور ارزیابی عملکرد، متریک‌های precision, recall, F1, accuracy و همچنین نمودار ROC و confusion matrix را گزارش کنید.

۳.۲. (۱۰ نمره) مقایسه با روش‌های دسته بندی دیگر

به انتخاب خود سه روش یادگیری ماشین دیگر را انتخاب کرده و توسط آن‌ها نیز دسته بندی را انجام داده و نتایج را مقایسه کنید. دقت کنید که برای این بخش نیازی به پیاده سازی وجود ندارد و می‌توانید از مدل‌های آماده‌ی scikit-learn استفاده کنید.

۳. (۱۵ نمره) خوشه بندی

توسط الگوریتم k-means داده‌ها را خوشه بندی کرده و scatter plot آن را در فضای دو بُعدی رسم کنید. تعداد خوشه‌های بهینه را توسط رسم کردن نمودار معیارهای silhouette score و WCSS به ازای k های مختلف به دست آورید.

۴. معیار رقابت

علاوه بر فایل BCICIV_calib_ds1a.mat که تا به این جا از آن استفاده کردیم، توسط پایپلاین پیاده سازی شده، فایل BCICIV_calib_ds1c.mat را نیز دسته بندی کرده و نتایج را مقایسه کنید. میانگین دقت به دست آمده روی این دو فایل به عنوان معیار رقابت در نظر گرفته می‌شود و به تیمی که بالاترین دقت را به دست آورده باشد نمره‌ی امتیازی تعلق خواهد گرفت.

به منظور بالاتر بردن دقت نهایی، مجاز به استفاده از هر تکنیک دلخواه برای پیش پردازش، تنظیم هایپر پارامتر ها و یا انتخاب مدل می‌باشید.

پروژه شماره ۲ (۱۲۰ نمره)

هدف پروژه

هدف این پروژه، پیاده‌سازی یک سامانه‌ی تشخیص اشیاء (Object Detection) برای شناسایی و مکان‌یابی موجودات زیر آب در تصاویر، با تکیه بر معماری Faster R-CNN است. شما باید یک آشکارساز دو مرحله‌ای طراحی کنید که (۱) نواحی کاندید (Region Proposals) را تولید کند و (۲) برای هر ناحیه، کلاس شیء و مختصات جعبه‌ی مرزی (Bounding Box) را پیش‌بینی نماید. در تمام مراحل، همه چیز باید از پایه پیاده‌سازی شود و استفاده از مدل آماده/از پیش آموزش‌دیده مجاز نیست.

قواعد و محدودیت‌های مهم (برای کل پروژه)

- پیاده‌سازی از پایه: همه‌ی بخش‌های اصلی سیستم (پیش‌پردازش، دیتالودر، معماری، زیان‌ها، متریک‌ها، پس‌پردازش مثل NMS، و حلقه آموزش) باید توسط شما پیاده‌سازی شود.
- ممنوعیت استفاده از مدل آماده: استفاده از مدل‌های از پیش آموزش‌دیده (وزن‌های آماده) و ماژول‌های آماده‌ی تشخیص شیء (مثل torchvision.models.detection) ممنوع است؛ اما استفاده از لایه‌های پایه مانند Conv2d، BatchNorm و ReLU مجاز است.
- ممنوعیت کد آماده: استفاده از کدهای آماده‌ی اینترنتی برای بخش‌های اصلی پروژه مجاز نیست و کد باید محصول کار خودتان باشد.
- زبان برنامه‌نویسی: فقط Python مجاز است.
- تحویل کد: کدها باید در قالب ipynb. باشد و همه‌ی سلول‌ها اجرا شده و خروجی‌ها ذخیره شده باشد.
- گزارش‌نویسی: تمرکز گزارش روی نتایج، تحلیل و تصمیمات طراحی است (نه توضیح خط‌به‌خط کد).

مجموعه‌دادگان (Dataset)

از مجموعه‌دادگان Underwater Object Detection استفاده کنید (قابل دریافت از Kaggle). این مجموعه شامل ۶۳۸ تصویر از موجودات زیر آب در ۷ کلاس است و برای هر شیء در تصویر، جعبه‌ی مرزی (Bounding Box) ارائه شده است.

ساختار داده و تقسیم‌بندی

- از زیرپوشه‌های train، validation و test همان‌طور که در دیتاست وجود دارد استفاده کنید.
- فرمت Annotation (مثلاً XML/JSON/TXT) را شناسایی کرده و تبدیل/استانداردسازی لازم را انجام دهید.
- تمام مراحل خواندن و تبدیل برچسب‌ها باید از پایه پیاده‌سازی شود.

تعریف مسئله (Problem Definition)

ورودی

یک تصویر RGB یا BGR از صحنه‌ی زیر آب.

خروجی

برای هر تصویر، مجموعه‌ای از پیش‌بینی‌ها شامل کلاس هر شیء (۷ کلاس)، مختصات جعبه‌ی مرزی $(x_{min}, y_{min}, x_{max}, y_{max})$، و نمره اطمینان (confidence score). تمام محاسباتِ پس‌پردازش (مثل NMS(-Soft)) باید از پایه پیاده‌سازی شود.

مراحل انجام پروژه و روش نمره‌دهی

مرحله ۱: پرسش‌های تشریحی و مبانی نظری (۱۰ نمره)

۱. (۳ نمره) مقایسه خانواده Region-based CNN: معماری‌های R-CNN، Fast R-CNN و Faster R-CNN را مختصر توضیح دهید و بگویید هرکدام چه مشکل/گلوگاهی را حل کرده‌اند.

۲. (۳ نمره) مقایسه آشکارسازهای تک‌مرحله‌ای و دومرحله‌ای: مزایا/معایب Stage-one (مثل YOLO/SSD) و Stage-two (مثل Faster R-CNN) را بیان کنید و کاربردهای رایج هرکدام را مثال بزنید.

۳. (۴ نمره) مفاهیم GIoU، Soft-NMS و OHEM: هرکدام را تعریف کنید، مزایا و کاربردشان را بگویید و توضیح دهید چرا ممکن است روی داده‌های زیر آب مفید باشند.

مرحله ۲: معرفی دیتاست، EDA و پیش‌پردازش (۱۰ نمره)

۱. نمایش نمونه‌ها: حداقل ۱۰ تصویر به همراه Bounding Boxهای واقعی و نام کلاس‌ها را نمایش دهید.

۲. EDA آماری: موارد زیر را محاسبه و به صورت نمودار/جدول گزارش کنید: تعداد تصاویر در هر کلاس، توزیع تعداد اشیاء در هر تصویر، توزیع اندازه‌ی Bounding Boxها (عرض/ارتفاع/نسبت تصویر).

۳. پیش‌پردازش: تصمیمات پیش‌پردازشی (مثلاً تغییر اندازه، نرمال‌سازی، تبدیل رنگ، حذف تصاویر خراب) را توضیح دهید و علت هر تصمیم را بنویسید.

مرحله ۳: تقویت داده (۱۰ نمره)

۱. تقویت‌های پایه: با توجه به ماهیت تصاویر زیر آب، چند روش مناسب (چرخش، برش، تغییر روشنایی/کنتراست، Color Jitter، Blur، نویز و ...) انتخاب کنید. برای هر روش دلیل انتخاب، نحوه اثرگذاری، و نحوه اعمال روی Bounding Boxها (به‌روزرسانی صحیح مختصات) را بنویسید و چند نمونه قبل/بعد را نمایش دهید.

۲. Mosaic Augmentation: ایده Mosaic را توضیح دهید (مزایا/معایب/کاربردها) و سپس آن را برای دیتاست خود پیاده‌سازی کنید. خروجی چند نمونه موزاییکی را همراه با جعبه‌های مرزی صحیح نمایش دهید.

مرحله ۴: ساخت دیتالودر و collate_fn اختصاصی (۱۰ نمره)

۱. به دلیل متفاوت بودن تعداد اشیاء در تصاویر، طول برچسب‌ها در یک batch یکسان نیست. یک collate_fn بنویسید که برای هر batch: لیست boxes و labels را تا طول بیشترین تعداد شیء در آن batch، pad کند؛ mask مناسب برای تشخیص داده‌های واقعی از padding تولید کند؛ و همه چیز را به tensorهای مناسب تبدیل کند.

۲. یک batch نمونه را چاپ/نمایش دهید و نشان دهید padding درست انجام شده است.

مرحله ۵: تعریف مسئله و طراحی معماری Faster R-CNN (۲۵ نمره)

۱. Backbone (استخراج ویژگی): یک شبکه‌ی کانولوشنی مناسب را خودتان طراحی کنید (می‌تواند سبک‌تر از ResNet باشد). استفاده از وزن‌های آماده/پیش‌آموزش‌دیده مجاز نیست.

۲. RPN + Head: معماری کلی Faster R-CNN را توضیح دهید: تولید Proposals، ROI Pooling/ROI Align، سپس سرِ تشخیص (کلاس‌بندی + رگرسیون جعبه).

۳. استفاده از GIoU: در بخش‌های مرتبط با همپوشانی و زیانِ رگرسیون جعبه‌ها، از GIoU به جای IoU استفاده کنید و دقیق توضیح دهید در کدام قسمت‌ها (مثلاً برچسب‌گذاری anchors یا زیان رگرسیون).

۴. جزئیات پیاده‌سازی: توضیح دهید ابعاد ورودی/خروجی هر ماژول چیست، و چگونه tensorها را بین اجزا منتقل می‌کنید.

مرحله ۶: طراحی و پیاده‌سازی Region Proposal Network (RPN) (۲۵ نمره)

۱. تولید Anchor Points و Anchor Boxes: روی feature map نقاط anchor ایجاد کنید و برای هر نقطه k جعبه با اندازه/نسبت‌های مختلف بسازید. مقدار k و مجموعه مقیاس‌ها/نسبت‌ها را خودتان انتخاب کنید و دلیل انتخاب را بنویسید.

۲. نمایش بصری: anchors را روی ۲ تصویر از دیتاست رسم و نمایش دهید.

۳. برچسب‌گذاری مثبت/منفی: با یک/چند آستانه مناسب، anchors را به positive و negative تقسیم کنید. سپس نمونه‌های مثبت/منفی را برای ۲ تصویر نمایش دهید و تمام پارامترهای به‌کاررفته (آستانه‌ها، تعداد نمونه‌برداری و ...) را گزارش کنید.

مرحله ۷: آموزش مدل کامل و تابع زیان (۱۵ نمره)

۱. تابع زیان: درباره زیان‌های رایج Faster R-CNN تحقیق کنید و یک زیان چندبخشی پیاده‌سازی کنید (مثلاً زیان objectness برای RPN، زیان رگرسیون جعبه برای RPN (ترجیحاً مبتنی بر (G)IoU یا Smooth-L1)، زیان کلاس‌بندی برای ROI head، و زیان رگرسیون جعبه برای ROI head).

۲. گزارش تنظیمات: نرخ یادگیری، بهینه‌ساز، batch size، تعداد epoch، وزن‌دهی بخش‌های زیان، و هر تنظیم مهم را کامل گزارش کنید.

۳. نمودارهای آموزش: نمودار روند آموزش (زیان کل و زیان‌های اجزا، و در صورت امکان mAP روی اعتبارسنجی) را رسم کنید.

مرحله ۸: ارزیابی و نمایش نتایج (۱۵ نمره)

۱. ارزیابی کمی: حداقل یک معیار مناسب تشخیص شیء را گزارش کنید (پیشنهاد: mAP@0.5 و/یا mAP@[0.5:0.95]). روش محاسبه را توضیح دهید و از پایه پیاده‌سازی کنید.

۲. ارزیابی کیفی: ۱۰ نمونه از تصاویر تست را نمایش دهید به همراه جعبه‌های واقعی، جعبه‌های پیش‌بینی‌شده (پس از (Soft-)NMS)، نام کلاس و نمره اطمینان.

۳. تحلیل خطا: حداقل ۵ نمونه‌ی شکست/اشتباه را انتخاب کنید و دلیل احتمالی را تحلیل کنید.

خلاصه ورودی/خروجی

مولفه	توضیح
ورودی	تصویر زیر آب
خروجی	مجموعه‌ای از (کلاس، Bounding Box، نمره اطمینان)
مدل	Faster R-CNN پیاده‌سازی‌شده از پایه
پس‌پردازش	(Soft-)NMS پیاده‌سازی‌شده از پایه

نحوه‌ی ارزیابی و رتبه‌بندی نهایی (نسخه‌ی ساده)

خروجی مدل.
برای هر تصویر، مدل مجموعه‌ای از تشخیص‌ها به صورت $\{(b_i, c_i, s_i)\}$ تولید می‌کند. ارزیابی روی خروجی نهایی پس از NMS انجام می‌شود.

تعریف IoU.

$$\mathrm{IoU}(A, B) = \frac{\mathrm{area}(A \cap B)}{\mathrm{area}(A \cup B)}.$$

قانون تطبیق.
اگر $\text{IoU} \geq ۰/۵$ باشد و آن گراندتروث قبلاً تطبیق داده نشده باشد، تشخیص درست محسوب می‌شود.

محاسبه‌ی امتیازها.

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \quad \text{Recall}_c = \frac{TP_c}{TP_c + FN_c}$$

$$\text{Score} = \frac{۱}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F۱_c.$$

پروژه شماره ۳ (۱۱۵ نمره)

تعریف مسئله
در این پروژه، هدف طراحی و پیاده‌سازی یک مدل یادگیری عمیق مبتنی بر معماری Transformer برای تحلیل احساسات[1] متون مالی است. شما باید مدلی بسازید که بتواند اخبار و جملات مالی را پردازش کرده و آن‌ها را در سه دسته مثبت، منفی یا خنثی طبقه‌بندی کند.
نکته کلیدی این پروژه این است که نباید از کلاس‌های آماده مدل استفاده کنید، بلکه باید لایه‌های اصلی ترنسفورمر (به‌ویژه مکانیزم Self-Attention) را خودتان پیاده‌سازی کنید.

دیتاست: برای آموزش و ارزیابی اولیه، از دیتاست استاندارد Financial PhraseBank استفاده کنید. می‌توانید این داده‌ها را مستقیماً از لینک زیر دریافت کنید:

Financial PhraseBank on Hugging Face

نکته مهم: برای داده‌های Test نهایی جهت ارزیابی در سیستم داوری، یک فایل جداگانه در اختیار شما قرار می‌گیرد که برچسب (Label) ندارد. لینک دانلود این داده در بخش مربوطه آمده است.

قسمت اول: آماده‌سازی داده برای مدل و پیش پردازش (۲۰ نمره)
در این گام باید داده‌ها را برای ورود به شبکه عصبی آماده کنید.

۱. (۳ نمره) دانلود و بارگذاری صحیح داده‌ها از فایل ZIP (با توجه به ساختار خاص فایل‌های Financial PhraseBank که شامل چندین فایل متنی است) و تجمیع آن‌ها در یک DataFrame واحد. به همراه رسم توزیع داده های هر دسته.

۲. (۱۵ نمره) پیش پردازش درست بر داده ها و دسته بندی ۸۰-۲۰ بین داده train و validation.

۳. (۲ نمره) پیاده‌سازی کلاس Dataset در PyTorch و ایجاد DataLoaders با استفاده از یک Tokenizer استاندارد (مانند BERT Tokenizer) برای تولید input ids و attention mask.

قسمت دوم: ساخت مدل (۴۵ نمره)
شما باید در این بخش معماری ترنسفورمر را بدون استفاده از ماژول‌های آماده‌ی سطح بالا پیاده‌سازی کنید.

۱. (۱۵ نمره) پیاده‌سازی کلاس SelfAttention از صفر. باید محاسبات Q, K, V را انجام دهید و فرمول زیر را با استفاده از عملیات ماتریسی (مثل torch.einsum یا matmul) پیاده‌سازی کنید:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

پشتیبانی از Multi-Head Attention در این بخش الزامی است.

۲. (۱۵ نمره) پیاده‌سازی کلاس TransformerBlock شامل لایه‌های Attention، LayerNorm، Feed For-ward Network و اتصال‌های باقیمانده (Residual Connections).

۳. (۱۵ نمره) تجمیع بلوک‌ها در کلاس اصلی مدل (FinancialTransformer) شامل لایه‌های Embedding (برای کلمات و موقعیت‌ها) و یک لایه Classifier نهایی برای تولید خروجی ۳ کلاسه.

قسمت سوم: آموزش مدل (۲۰ نمره)
در این مرحله مدل طراحی شده را آموزش می‌دهید.

[1] Sentiment Analysis

۱. (۱۰ نمره) نوشتن حلقه آموزش (Training Loop) استاندارد با استفاده از Optimizer و تابع هزینه مناسب.

۲. (۱۰ نمره) ثبت و رسم نمودار Loss و Accuracy برای هر دو مجموعه Train و Validation در پایان هر Epoch جهت پایش فرآیند یادگیری.

قسمت چهارم: ارزیابی دقیق مدل بر داده‌ی validation (۱۰ نمره)

در این مرحله باید عملکرد نهایی مدل آموزش‌دیده را با معیارهای استاندارد روی داده‌های اعتبارسنجی بسنجید تا کیفیت مدل فراتر از نمودار خطا مشخص شود.

۱. (۴ نمره) محاسبه معیارهای ارزیابی کمی: برای کل مجموعه داده Validation، معیارهای زیر را محاسبه و گزارش کنید. با توجه به اینکه مسئله چندکلاسه (۳ کلاس) است، در صورت استفاده از کتابخانه‌هایی مثل scikit-learn، حتماً نوع میانگین‌گیری (مثلاً weighted یا macro) را مشخص کنید:

- Accuracy
- Precision
- Recall
- F1-Score

۲. (۶ نمره) رسم ماتریس درهم‌ریختگی (Confusion Matrix): ماتریس درهم‌ریختگی را محاسبه کرده و آن را به شکل گرافیکی رسم کنید. محورهای نمودار باید دارای برچسب‌های واضح کلاس‌ها (Negative, Neutral, Positive) باشند تا مشخص شود مدل بیشترین خطا را در تشخیص کدام کلاس‌ها دارد.

قسمت پنجم: ارزیابی خروجی مدل بر داده‌ی test (۵ نمره)

در این بخش عملکرد نهایی مدل خود را روی داده‌های دیده نشده می‌سنجید.

۱. (۵ نمره) تولید فایل خروجی برای مسابقه: مدل خود را روی این جملات اجرا کرده و پیش‌بینی‌ها را در یک فایل CSV ذخیره کنید. متعاقباً روش ارزیابی این بخش اعلام می‌شود و شما باید فایل csv را در مکانی که گفته می‌شود آپلود کنید تا نمره نهایی داوری و محاسبه شود.

قسمت ششم: خروجی بصری مکانیزم توجه (۱۵ نمره)

برای درک بهتر عملکرد مدل، باید نشان دهید مدل به کدام کلمات توجه بیشتری دارد.

۱. (۱۵ نمره) استخراج ماتریس وزن‌های توجه (Attention Weights) از آخرین لایه مدل و رسم Heatmap برای حداقل ۵ جمله اول از داده‌های تست بخش قبل. خروجی باید نشان دهد که مدل برای پیش‌بینی حس جمله، روی چه کلماتی تمرکز کرده است مثلاً کلماتی مثل "risk", "loss", "growth".

-

پروژه شماره ۴ (۱۰۰ نمره)

تعریف مسئله:
هدف این پروژه طراحی و پیاده‌سازی یک مدل یادگیری ماشین برای پیش‌بینی فروش آتی فروشگاه‌های زنجیره‌ای Rossmann است. علاوه بر پیش‌بینی دقیق فروش، دانشجویان باید بتوانند ویژگی‌های پیشرفته‌تر مدل‌سازی، مانند برآورد عدم قطعیت، دسته‌بندی فروش، و تحلیل مدل را پیاده‌سازی کرده و تحلیل دقیقی از عملکرد ارائه دهند.

مشخصات دیتاست:

- داده‌ها از رقابت فروش فروشگاه‌های Rossmann استخراج شده است.
- تعداد نمونه‌ها: بیش از ۱ میلیون ردیف فروش روزانه برای بیش از ۱۱۰۰ فروشگاه
- ستون هدف: Sales
- فایل‌های جانبی: store.csv (حاوی مشخصات هر فروشگاه شامل نوع فروشگاه، فاصله تا رقیب، نوع سرویس‌دهی و ...)
- ویژگی‌ها: ترکیبی از عددی، باینری و زمانی (تاریخ، روز هفته، ماه و ...)

مراحل پروژه و نمره‌بندی:

فاز ۱: Data Visualization - تحلیل اکتشافی و شناخت داده‌ها – ۵ نمره

- رسم نمودارهای روند فروش در طول زمان برای چند فروشگاه منتخب
- بررسی تأثیر ویژگی‌های مختلف (Promo، SchoolHoliday، StateHoliday و ...) بر فروش
- نمودار همبستگی متغیرها و تحلیل آن
- مقایسه روند فروش در فروشگاه‌های مختلف

فاز ۲: Feature Engineering - مهندسی ویژگی و آماده‌سازی داده – ۱۵ نمره

- ترکیب جداول sales و store بر اساس Store ID
- ایجاد ویژگی‌های زمانی جدید (روز هفته، ماه، تعطیلی، میانگین متحرک و ...)
- ایجاد lag features و آماره‌های متحرک (فروش روز قبل، میانگین ۷ روز اخیر، انحراف معیار و ...)
- استخراج ویژگی‌های دوره‌ای (مثل Fourier terms برای فصل‌ها) و فاصله تا تعطیلات رسمی
- نرمال‌سازی و مدیریت داده‌های گمشده
- تفکیک داده به آموزش/اعتبارسنجی/تست با حفظ ترتیب زمانی

فاز ۳: Learn & Estimate - آموزش و ارزیابی مدل‌ها – ۳۰ نمره
توجه کنید در این فاز باید کامل مدل را پیاده‌سازی کرده و استفاده از کتابخانه‌ها مجاز نیست.

- پیاده‌سازی مدل baseline ساده (رگرسیون خطی یا میانگین متحرک) و ارزیابی آن
- آموزش مدل‌های کلاسیک مانند XGBoost یا LightGBM با تنظیم hyperparameter
- استفاده از اعتبارسنجی زمانی مبتنی بر فروشگاه (store-aware CV) و گزارش RMSE در هر مرحله

- تحلیل feature importance با استفاده از SHAP
- تحلیل خطا: بررسی مواردی که مدل در آن‌ها عملکرد ضعیفی داشته به طور دقیق با تفسیر و مدرک

فاز ۴: Uncertainty & Sales Classification - **تحلیل عدم قطعیت و طبقه‌بندی فروش – ۲۵ نمره**

- پیاده‌سازی پیش‌بینی چندگامی (multi-step forecasting) (فروش ۷ روز آینده) با مدیریت انتشار خطا
- تخمین uncertainty با روش‌هایی مثل quantile regression یا ensemble
- دسته‌بندی فروش به سه کلاس (کم، متوسط، زیاد) و آموزش مدل طبقه‌بند چندکلاسه با ROC-AUC و F1-score

فاز ۵: Deep - **قسمت عمیق پروژه – ۲۵ نمره**

- پیاده‌سازی دقیق uncertainty estimation + multi-step forecasting + SHAP
- استفاده از مدل‌های یادگیری عمیق LSTM و TCN با مستندات کامل
- بهینه‌سازی هایپرپارامترها با Optuna یا Ray Tune

فاز ۵ + ۱ : مستندسازی و ارائه نتایج

- گزارش کامل نتایج پیش‌بینی، تحلیل خطاها، نمودارها و تفسیر نهایی
- مستندسازی مرتب، ساختارمند و قابل فهم

معیار رقابت بین تیم‌ها:

- معیار اصلی: Root Mean Squared Error مدل نهایی بر روی مجموعه تست
- معیارهای مکمل: Mean Absolute Percentage Error و R^2
- برای طبقه‌بندی: ROC-AUC و F1 Score

-

پروژه شماره ۵ (۱۲۰ نمره)

تعریف مسئله
در این پروژه هدف این است که یک مدل مولد مبتنی بر latent space برای تصاویر بسازید و سپس آن را بهبود دهید تا (۱) نمونه‌های تولیدی باکیفیت‌تری ایجاد کند، (۲) فضای نهفته رفتار قابل‌تفسیرتری داشته باشد، و (۳) بتوان تولید را با یک ورودی کنترلی (برچسب کلاس) هدایت کرد.

دیتاست این پروژه: Fashion-MNIST (۱۰ کلاس، تصاویر خاکستری ۲۸ × ۲۸)

فاز اول: آماده‌سازی داده‌ها و تحلیل اکتشافی (۱۰ نمره)
هدف این فاز آماده کردن ورودی استاندارد برای مدل و ارائه‌ی یک تصویر کلی از توزیع داده‌ها است.
آماده‌سازی داده و ساخت ورودی مدل (۷ نمره)
دیتاست Fashion-MNIST را دریافت کنید و ورودی مناسب برای آموزش مدل را بسازید. موارد زیر باید در نوت‌بوک شما به‌صورت شفاف وجود داشته باشد:

- دانلود/بارگذاری داده و گزارش تعداد نمونه‌های train/test/validation و شکل هر نمونه.
- تعریف یک pipeline پیش‌پردازش.
- ساخت DataLoader برای train/test/validation.
- یک sanity check: نمایش یک batch (حداقل ۲۵ تصویر) به‌همراه برچسب کلاس.
- ثبت seed برای بازتولیدپذیری نتایج.

تحلیل اکتشافی (EDA) و نمودارها (۳ نمره)

- نمودار میله‌ای توزیع کلاس‌ها (۱۰ کلاس) در مجموعه‌ی آموزش.
- هیستوگرام شدت پیکسل‌ها (برای کل داده یا چند کلاس منتخب).
- یک هیستوگرام آماری ساده در سطح تصویر (یکی از موارد زیر):
  - میانگین شدت پیکسل هر تصویر، یا
  - تعداد پیکسل‌های غیرصفر در هر تصویر.

فاز دوم: پیاده‌سازی VAE و بهبود کیفیت خروجی (۵۰ نمره)
پیاده‌سازی VAE از ابتدا (۳۰ نمره)
یک VAE را from scratch پیاده‌سازی و آموزش دهید (reparameterization trick + Decoder + Encoder).
تابع هزینه باید شامل موارد زیر باشد:

- reconstruction loss (یکی از BCE یا MSE را انتخاب و ثابت نگه دارید).
- یک ترم regularization روی latent space (در حالت پایه با وزن ۱).

خروجی مورد انتظار از این بخش:

- نمایش نمونه‌های reconstruction (حداقل ۲۰ تصویر ورودی/خروجی).
- نمایش نمونه‌های sampling (حداقل ۵۰ تصویر).
- گزارش روی test set: مقادیر reconstruction، regularization و total.

بهبود کیفیت خروجی و معیار رقابت (۲۰ نمره)
حداقل یک تغییر برای بهبود کیفیت خروجی اعمال کنید (نام روش مهم نیست). یکی از موارد زیر را انجام دهید:

- تغییر architecture (مثلاً convolutional یا عمیق‌تر).
- تغییر latent dimension.
- تغییر شدت یا برنامه‌ی regularization در طول آموزش.

مقایسه با مدل پایه باید شامل موارد زیر باشد:

- جدول متریک‌های test set.
- چند نمونه reconstruction و sampling (مقایسه قبل و بعد از بهبود).

معیار رقابت بین تیم‌ها:
معیار اصلی رقابت استفاده از FID (Frechet Inception Distance) روی ویژگی‌های استخراج‌شده از classifier ارائه‌شده در درس است. برای این کار:

- حداقل 10,000 تصویر تولید کنید.
- FID را بین تصاویر تولیدی و تصاویر test set محاسبه کنید (با استفاده از classifier ارائه‌شده).

تیم‌های برتر که کمترین FID را ثبت کنند، امتیاز اضافی دریافت می‌کنند.

فاز سوم: کنترل رفتار latent space با تغییر وزن regularization (۲۰ نمره)
در این فاز هدف این است که با تنظیم دقیق شدت جریمه بر فضای نهفته، به مدلی دست یابید که ویژگی‌های بصری داده‌ها را به شکلی تفکیک‌شده و معنادار در ابعاد خود ذخیره کند.

تحلیل اثر وزن‌دهی روی latent space (۲۰ نمره)
در این فاز باید شدت ترم regularization را تغییر دهید تا رفتار latent space قابل‌تفسیرتر شود. منظور این است که با تغییرات کوچک و کنترل‌شده در بردار نهفته، تغییرات معنادار و پیوسته در تصویر خروجی مشاهده شود.
موارد الزامی که باید در گزارش شما مستند شود:

- حداقل سه مقدار متفاوت برای وزن ترم regularization انتخاب کنید (مثلاً 0.5، 1، 4).
- برای هر مقدار، مدل را آموزش داده و روی test set گزارش کنید: مقادیر reconstruction، regularization و total.
- برای هر مقدار، عملیات latent traversal را به شرح زیر انجام دهید:
  - انتخاب حداقل ۵ بعد متفاوت از بردار نهفته (latent vector).
  - برای هر بعد، تولید حداقل ۷ مقدار پیوسته (مثلاً در بازه 3- تا 3+).
  - نمایش خروجی‌های بازسازی‌شده به صورت یک شبکه (grid) منظم.
- نگارش یک پاراگراف جمع‌بندی که تحلیل کند کدام وزن، بهترین توازن بین کیفیت بازسازی (reconstruction) و قابلیت تفسیر فضای نهفته را ایجاد کرده است.

فاز چهارم: تولید کنترل‌پذیر با استفاده از label (۲۰ نمره)
در این فاز، هدف ارتقای مدل به یک خودرمزگذار متغیر شرطی (Conditional VAE) است تا فرآیند تولید تصویر تحت کنترل برچسب‌های کلاس قرار گیرد.

تولید تصویر با کنترل کلاس (۲۰ نمره)
در این فاز باید کاری کنید که مدل بتواند با دریافت label (کلاس تصویر)، نمونه‌هایی مطابق همان کلاس تولید کند.
موارد الزامی که باید رعایت شوند:

- برچسب (label) را به‌عنوان ورودی کنترلی به مدل اضافه کنید (طراحی دقیق معماری با تیم).
- مدل را آموزش دهید و حداقل برای هر یک از 10 کلاس:
  - حداقل 20 تصویر تولید کنید.
  - خروجی‌ها را به صورت یک شبکه (grid) نمایش دهید.
- برای سنجش کنترل‌پذیری، از classifier ارائه شده در درس استفاده کنید و گزارش دهید:
  - دقت دسته‌بندی (classification accuracy) روی تصاویر تولیدی (به تفکیک هر کلاس و میانگین کل).
- یک مقایسه کوتاه با حالت بدون label ارائه دهید (شامل چند نمونه تصویری و یک پاراگراف توضیح).

فاز پنجم: گزارش نهایی و جمع‌بندی (۲۰ نمره)
این مرحله شامل تجمیع یافته‌ها و ارائه مستندات نهایی پروژه به شکلی ساختاریافته است.

گزارش و ارائه نتایج (۲۰ نمره)
در این فاز باید یک گزارش نهایی تهیه کنید که تمام مراحل پروژه را به‌صورت خلاصه، شفاف و قابل ارزیابی ارائه دهد.
موارد الزامی گزارش:

- ارائه یک جدول نهایی از نتایج روی test set برای همه مدل‌ها و تنظیمات اجرا شده شامل مقادیر: reconstruction، regularization و total.
- نمایش نمونه‌های کلیدی (به صورت grid) برای هر فاز شامل:
  - بازسازی (reconstruction) و نمونه‌برداری (sampling) برای مدل پایه.
  - مقایسه قبل و بعد برای بهبود کیفیت خروجی.
  - پیمایش فضای نهفته (latent traversal) برای وزن‌های مختلف.
  - تولید کنترل‌پذیر با برچسب برای تمامی کلاس‌ها.
- جمع‌بندی کوتاه (حداکثر یک صفحه) شامل شناسایی بهترین تنظیمات از نظر کیفیت تولید و توازن با قابلیت تفسیر، به همراه بیان محدودیت‌ها و پیشنهادات برای توسعه آتی.
- کدها باید کاملاً قابل اجرا بوده و تنظیمات بازتولیدپذیری (seed) در ابتدای نوت‌بوک درج شده باشد.

-

پروژه شماره ۶ (۱۱۰ نمره)

بیان مسئله: تحلیل احساسات جنبه‌محور (Aspect-Based Sentiment Analysis**)**
در بسیاری از کاربردهای واقعی، دانستن اینکه یک نظر به طور کلی «مثبت» یا «منفی» است، کافی نیست. برای مثال در جمله «قیمت محصول مناسب بود ولی کیفیت ساخت پایینی داشت»، با دو حس متضاد روبرو هستیم. هدف این پروژه پیاده‌سازی سیستمی مبتنی بر یادگیری عمیق است که بتواند احساسات را نسبت به جنبه‌های خاص موجود در متن تشخیص دهد.

فاز اول: آماده‌سازی داده‌ها و تحلیل اکتشافی (۱۵ نمره)
مجموعه داده استاندارد SemEval-2014-Task-4 (زیرمجموعه رستوران یا لپ‌تاپ) را دریافت کنید.

- **استخراج داده (۱۰ نمره):** از مجموعه داده‌ی دریافتی داده‌ی ورودی مدل خود را استخراج کنید. توجه کنید که جملات ممکن است دارای چند جنبه باشند. به همین دلیل ممکن است هر جمله به چند نمونه آموزشی مجزا تبدیل شود.
- **تحلیل داده (۵ نمره):** نمودارهای زیر را برای تحلیل داده رسم کرده و در گزارش بیاورید:
  - نمودار میله‌ای توزیع کلاس‌ها (مثبت، منفی، خنثی).
  - نمودار هیستوگرام تعداد کلمات در جملات.
  - ابر کلمات (Word Cloud) برای کلمات مربوط به جنبه‌های مختلف.

فاز دوم: طراحی و پیاده‌سازی مدل (۶۰ نمره)
در این بخش معماری مدل مبتنی بر ترنسفورمرها را برای شناسایی قطبیت هر جنبه طراحی می‌کنید.

- **ورودی مدل (۱۰ نمره):** از آنجا که ورودی مدل شامل «جمله» و «کلمه جنبه» است، باید آن‌ها را به گونه‌ای ترکیب کنید که ورودی قابل قبول برای مدل فراهم شود (معمولاً در پیاده‌سازی کلاس Dataset).
- **آموزش مدل (۴۰ نمره):** یک مدل زبانی پیش‌آموزش دیده دلخواه (مانند BERT-Base یا RoBERTa) را بارگذاری کرده و لایه آخر آن را برای کلاس‌بندی ۳ کلاسه تغییر دهید. مدل را روی داده‌هایی که آماده کرده‌اید آموزش دهید.
- **تنظیمات و هایپرپارامترها (۵ نمره):** تمامی تنظیمات استفاده شده برای آموزش (مانند Learning Rate، Batch Size و ...) باید دقیقاً گزارش شوند.
- **نمودارهای آموزش (۵ نمره):** نمودار تغییرات loss و accuracy را برای داده‌های آموزشی و اعتبارسنجی در طول اپاک‌های مختلف رسم کنید.

فاز سوم: ارزیابی و تحلیل خطا (۱۵ نمره)
مدل آموزش دیده را روی داده‌های تست ارزیابی کرده و موارد زیر را در گزارش نهایی درج کنید:

- **گزارش دسته‌بندی (۵ نمره):** محاسبه معیارهای Precision، Recall و F1-Score. با توجه به نامتوازن بودن داده‌ها، معیار Macro-F1 را نیز حتماً گزارش دهید.
- **ماتریس آشفتگی (۵ نمره):** ماتریس آشفتگی (Confusion Matrix) را بر اساس خروجی‌های مدل رسم کنید.
- **تحلیل خطا (۵ نمره):** ۵ نمونه از نظراتی که مدل به غلط پیش‌بینی کرده است را انتخاب کنید. متن جمله، برچسب واقعی و پیش‌بینی مدل را بنویسید و دلیل احتمالی خطا را تحلیل کنید.

فاز چهارم: تست مدل (۲۰ نمره)
بررسی استحکام (Robustness) مدل در مواجهه با جملات جدید و چالش‌برانگیز.

- **تست جملات چالش‌برانگیز (۱۰ نمره):** ۲۰ جمله چالش‌برانگیز بنویسید (ساختارهای شرطی، منفی در منفی و ...) و خروجی مدل را گزارش کنید.
- **حمله تخاصمی ساده (۱۰ نمره):** جمله‌ای بنویسید که مدل آن را درست دسته‌بندی می‌کند، سپس با تغییرات جزئی بدون تغییر معنا، سعی کنید مدل را به اشتباه بیندازید.

ارزیابی رقابتی: شایان ذکر است که در پایان این پروژه، تمامی تیم‌هایی که روی موضوع یکسانی فعالیت کرده‌اند، بر اساس معیارهای ارزیابی (به‌ویژه Macro-F1) با یکدیگر مقایسه خواهند شد. این مقایسه نه تنها برای سنجش کیفیت مدل‌های طراحی شده، بلکه برای رتبه‌بندی نهایی و شناسایی پیاده‌سازی‌های برتر انجام می‌گیرد.

پروژه شماره ۷ (۱۰۵ نمره)

تعریف مسئله
هدف این پروژه ساخت یک مدل Binary Classification کارآمد و دقیق برای شناسایی مشتریان در معرض ریزش (Churn) در حوزه مخابرات است. تیم پروژه باید بر انتخاب بهترین ویژگی‌ها، مدیریت عدم توازن داده‌ها و مقایسه مدل‌های پیشرفته تمرکز کند. تمرکز اصلی ارزیابی مدل‌ها بر بیشینه‌سازی معیار Recall خواهد بود.

دیتاست: Telco Customer Churn Dataset

فاز اول: تحلیل اکتشافی داده‌ها (EDA) — ۱۵ نمره

- **هدف:** شناخت رفتار داده‌ها
- **شناخت اولیه (۵ نمره)**
  - گزارش ابعاد دیتاست (Shape)، نوع متغیرها و مدیریت داده‌های تکراری (Duplicates)
- **تحلیل تک‌متغیره (۵ نمره)**
  - رسم نمودار Histogram برای متغیرهای عددی (مانند Tenure)
  - رسم نمودار Bar Plot برای متغیرهای دسته‌ای
  - تحلیل توزیع‌ها
- **تحلیل دومتغیره و همبستگی (۵ نمره)**
  - رسم Heatmap برای بررسی همبستگی متغیرهای عددی
  - استفاده از نمودارهای مقایسه‌ای مانند Boxplot برای بررسی رابطه مهم‌ترین ویژگی‌های عددی با متغیر هدف (Churn)

فاز دوم: پیش‌پردازش داده‌ها (Preprocessing) — ۱۵ نمره

- **هدف:** تمیزکاری داده‌ها
- **مدیریت مقادیر گم‌شده (۵ نمره)**
  - شناسایی ستون‌های دارای مقدار گم‌شده (NaN)
  - پیاده‌سازی Imputation همراه با توجیه متنی دقیق برای روش انتخابی
- **انکودینگ و مقیاس‌بندی (۱۰ نمره)**
  - اعمال Label Encoding برای متغیرهای دودویی
  - اعمال One-Hot Encoding برای متغیرهای دسته‌ای
  - اعمال StandardScaler یا MinMaxScaler بر تمام ویژگی‌های عددی

فاز سوم: مهندسی و انتخاب ویژگی (Feature Engineering & Selection) — ۲۵ نمره

- **هدف:** افزایش قدرت مدل با استفاده از ویژگی‌های بهتر
- **مهندسی ویژگی (۵ نمره)**

- ایجاد حداقل دو ویژگی جدید معنادار (مانند ترکیب ویژگی‌ها یا بازه‌بندی Tenure)

- **انتخاب ویژگی (۲۰ نمره)**
- **روش فیلترمحور:**
  - استفاده از آزمون‌های آماری مانند ANOVA یا Chi-Squared
  - انتخاب ۱۰ تا ۱۵ ویژگی برتر
- **روش مدل‌محور:**
  - استفاده از Feature Importance در مدل‌هایی مانند:
    - Lasso Regression
    - Random Forest
- **الزامی:**
  - ارائه توضیح متنی مستدل برای انتخاب زیرمجموعه نهایی ویژگی‌ها

فاز چهارم: مدل‌سازی پیشرفته و بهینه‌سازی (Modeling & Optimization) — ۴۰ نمره

- **مدیریت عدم توازن داده‌ها (۱۰ نمره)**
  - اعمال SMOTE روی داده‌های آموزشی
  - مقایسه تأثیر Class Weights و SMOTE
- **مدل‌های پایه (۱۰ نمره)**
  - Logistic Regression
  - KNN یا SVM
  - Random Forest
  - Gradient Boosting (مانند XGBoost یا LightGBM)
- **آنسامبلینگ مدل (۵ نمره)**
  - پیاده‌سازی Soft Voting Classifier با ۳ مدل برتر
- **تنظیم فراپارامتر (۱۰ نمره)**
  - استفاده از GridSearchCV یا RandomizedSearchCV
  - گزارش بهترین پارامترها برای حداقل دو مدل
- **اعتبارسنجی پیشرفته (۵ نمره)**
  - استفاده از Stratified K-Fold Cross Validation
  - گزارش میانگین و انحراف معیار Recall

فاز پنجم: ارزیابی و گزارش نهایی (۱۰ نمره)

- **گزارش معیارها (۴ نمره)**

- ارائه جدول جامع شامل معیارهای Accuracy، Precision، Recall و F1-Score برای تمامی مدل‌های پیاده‌سازی شده.

- **مقایسه ROC-AUC (۳ نمره)**
  - گزارش مقدار عددی AUC برای چهار مدل برتر پروژه جهت سنجش قدرت تفکیک کلاس‌ها.
- **تحلیل نهایی (۳ نمره)**
  - رسم ماتریس آشفتگی (Confusion Matrix) برای تحلیل دقیق‌تر خطاهای مدل.
  - معرفی مدل برنده و استخراج مهم‌ترین ویژگی‌های اثرگذار بر خروجی.
  - ارائه پیشنهاد تجاری مشخص بر اساس تحلیل نتایج به دست آمده.

بخش مسابقه (Bonus Challenge)
- بیشینه‌سازی Recall برای کلاس ریزش (Churn = 1) مشروط به اینکه F1-Score کمتر از ۵۰% نباشد.

پروژه شماره ۸ (۱۱۰ نمره)

Anomaly Detection on MVTec AD

هدف کلی پروژه
در این پروژه، شما یک سیستم Anomaly Detection و Anomaly Localization برای تصاویر بازرسی صنعتی پیاده‌سازی می‌کنید. دیتاست پروژه MVTec AD است. شما ابتدا baseline پروژه یعنی مدل PaDiM را تکمیل و اجرا می‌کنید و سپس روش خودتان را پیاده‌سازی کرده و با baseline مقایسه می‌کنید.

آنچه تحویل می‌دهید

- یک نوت‌بوک شامل تمام TODOهای پیاده‌سازی شده که بدون خطا اجرا شود.
- خروجی‌های گزارش‌محور: جدول‌های نتایج، نمودارها، و visualization کیفی (overlay + GT contour).

قوانین و محدودیت‌ها

- آموزش (fit) **فقط** مجاز است از تصاویر train/good استفاده کند.
- خروجی predict باید شامل **نمره‌ی سطح تصویر** و **نقشه‌ی ناهنجاری سطح پیکسل** باشد.
- کد باید قابلیت تکرار داشته باشد (ثبت seed).

فازبندی و نمره‌دهی

فاز ۱ (۶ نمره): آماده‌سازی داده و Sanity Check

- پیاده‌سازی MVTecDataset و DataLoaderها شامل:
  - بارگذاری صحیح تصویر، برچسب و ماسک GT.
  - هم‌تراز بودن ترتیب داده‌ها و بازگشت دیکشنری استاندارد در __getitem__.

فاز ۲ (۳۰ نمره): خط مبنا PaDiM

- **استخراج ویژگی (۶ نمره):** پیاده‌سازی ResNetFeatureExtractor.
- **استخراج** embeddings **(۶ نمره):** پیاده‌سازی تابع extract_embeddings.
- **برازش مدل آماری (۸ نمره):** پیاده‌سازی تابع fit_padim_gaussian (میانگین/کوواریانس مکانی).
- **محاسبه‌ی** score map **(۶ نمره):** استفاده از فاصلهٔ Mahalanobis.
- **اتصال مراحل (۴ نمره):** تولید خروجی استاندارد در تابع run_padim_category.

فاز ۳ (۶۰ نمره): روش دانشجو (StudentMethod**)**

- fit **(۴۰ نمره):**
  - رعایت قوانین آموزش و عدم نشت اطلاعات (۱۰ نمره).
  - آماده‌سازی بخش سطح تصویر (۱۰ نمره).
  - آماده‌سازی بخش سطح پیکسل (۲۰ نمره).

- predict **(۲۰ نمره):**
  - تولید خروجی‌های صحیح و معنادار برای سطح تصویر (۱۰ نمره).
  - تولید خروجی‌های سطح پیکسل با ابعاد صحیح و قابل ارزیابی (۱۰ نمره).

فاز ۴ (۱۴ نمره): متریک‌ها، گزارش و مقایسه

- **پیاده‌سازی متریک‌ها (۱۰ نمره):**
  - سطح تصویر: AUROC, AP, best-F1, balanced accuracy.
  - سطح پیکسل: Pixel AUROC, best Dice.
- **تولید گزارش‌ها (۴ نمره):**
  - جدول نتایج per-category و خلاصهٔ کلی (macro mean).

معیار رقابت و امتیاز اضافی (Leaderboard**)**

تیم‌ها بر اساس عملکردشان در سه معیار زیر، رتبه‌بندی می‌شوند و یک امتیازی از هر معیار می‌گیرند. امتیاز نهایی هر تیم از جمع امتیازش در هر معیار به دست می‌آید و در نهایت تیم‌ها بر اساس جمع امتیازشان رتبه‌بندی نهایی می‌شوند.

رتبه	ImageScore	PixelScore	OverallScore
اول	۳+	۳+	۳+
دوم	۲+	۲+	۲+
سوم	۱+	۱+	۱+

-

پروژه شماره ۹ (۱۱۵ نمره)

۱. مقدمه و هدف پروژه

دانشجویان دانشگاه صنعتی شریف در طول دوران تحصیل خود با پرسش‌های متعددی دربارهٔ قوانین و مقررات آموزشی مواجه هستند. این مقررات به‌صورت رسمی در آیین‌نامه‌ها و دستورالعمل‌های منتشرشده توسط **مدیریت امور آموزشی دانشگاه صنعتی شریف** در دسترس قرار دارد، اما جست‌وجو و استخراج پاسخ دقیق از این متون معمولاً دشوار و زمان‌بر است.

هدف این پروژه، طراحی و پیاده‌سازی یک **چت‌بات هوشمند مبتنی بر** Retrieval-Augmented Generation (RAG) است که بتواند صرفاً بر اساس آیین‌نامه‌ها و مقررات رسمی دانشگاه صنعتی شریف به پرسش‌های آموزشی دانشجویان پاسخ دهد و از تولید اطلاعات نادرست یا خارج از متن آیین‌نامه جلوگیری کند.

۲. تعریف مسئله

در این پروژه، هر تیم موظف است یک چت‌بات راهنمای آموزشی طراحی کند که بتواند به پرسش‌هایی در حوزه‌های زیر پاسخ دهد:

- قوانین آموزشی دورهٔ کارشناسی؛
- آیین‌نامهٔ امتحانات و مقررات غیبت؛
- کارآموزی، پروژهٔ کارشناسی و کار و آموزش پایدار (کوآپ)؛
- مهمانی، انتقال و معرفی به استاد؛
- معافیت تحصیلی و مهلت فراغت از تحصیل؛
- و سایر موضوعات مندرج در آیین‌نامه‌های رسمی آموزشی.

چت‌بات باید **فقط** بر اساس آیین‌نامه‌ها و مقررات منتشرشده در وب‌سایت رسمی مدیریت امور آموزشی دانشگاه صنعتی شریف به آدرس زیر پاسخ دهد:

https://ac.sharif.edu/rules/

الزامات پاسخ‌دهی

پاسخ‌های چت‌بات باید دارای ویژگی‌های زیر باشند:

- عدم تولید قانون یا تفسیر خارج از متن آیین‌نامه؛
- ارجاع صریح به نام آیین‌نامه و در صورت امکان شمارهٔ ماده، بند یا تبصره؛
- اعلام شفاف عدم وجود پاسخ در آیین‌نامه با عباراتی مانند:
  - «در آیین‌نامهٔ موجود مطلبی در این‌باره یافت نشد»
  - «نیاز به استعلام از آموزش دارد»

۳. معماری مورد انتظار سامانه

معماری مورد انتظار یک **RAG تک‌مرحله‌ای (Single-hop)** است که شامل مراحل زیر می‌باشد:

۱. تبدیل سؤال کاربر به embedding؛

۲. بازیابی بخش‌های مرتبط از آیین‌نامه‌ها؛

۳. ارسال متن بازیابی‌شده به همراه سؤال به مدل زبانی؛

۴. تولید پاسخ کوتاه، دقیق و مستند.

۴. پایگاه دانش (Knowledge Base)

۱.۴. منبع داده

منبع دادهٔ رسمی و واحد برای تمام تیم‌ها، صفحهٔ آیین‌نامه‌ها و مقررات مدیریت امور آموزشی دانشگاه صنعتی شریف است. این صفحه شامل آیین‌نامه‌هایی از جمله موارد زیر می‌باشد:

- آداب‌نامه استفاده از ابزارهای هوش مصنوعی؛
- آیین‌نامه آموزشی دورهٔ کارشناسی (نسخهٔ جاری و نسخه‌های ورودی‌های قبل)؛
- آیین‌نامه برگزاری و غیبت در امتحانات؛
- آیین‌نامه روابط پیش‌نیازی و همنیازی؛
- آیین‌نامه کارآموزی؛
- دستورالعمل پروژهٔ کارشناسی؛
- شیوه‌نامه مهمانی و انتقال؛
- قوانین تغییر رشته؛
- مقررات معافیت تحصیلی؛
- نظام‌نامه آموزشی.

۲.۴. یکنواخت‌سازی داده

برای تضمین عدالت در ارزیابی:

- یک نسخهٔ آفلاین (HTML) از تمام آیین‌نامه‌ها توسط تیم تدریس تهیه و در اختیار همهٔ تیم‌ها قرار می‌گیرد؛
- استفاده از هر منبع دیگری خارج از این مجموعه مجاز نیست؛
- پایگاه دانش هر تیم باید صرفاً از این فایل‌ها ساخته شود.

۳.۴. ساخت پایگاه دانش

هر تیم باید مراحل زیر را انجام دهد:

۱. استخراج متن از فایل‌ها؛

۲. تمیزسازی حداقلی (حذف هدر، فوتر، شماره صفحه و تکرارها)؛

۳. تقسیم متن به بخش‌های کوچک (Chunking):

- ترجیحاً بر اساس ماده، بند یا تبصره؛
- یا پنجره‌های طول ثابت (حدود ۲۰۰ تا ۳۰۰ توکن فارسی).

هر chunk باید شامل متادیتای زیر باشد:

- متن chunk؛
- نام آیین‌نامه؛

- شمارهٔ ماده یا بخش (در صورت امکان)؛
- تاریخ آیین‌نامه (اختیاری)؛
- شناسهٔ یکتا.

۵. Embedding و Retrieval

۱.۵. Embedding

- استفاده از یک مدل embedding مناسب برای زبان فارسی یا چندزبانه؛
- محاسبهٔ embedding برای تمام هاchunk؛
- ذخیره در یک index (مانند FAISS یا ابزارهای مشابه).

۲.۵. Retrieval

برای هر سؤال کاربر:

۱. embedding سؤال محاسبه می‌شود؛

۲. k بخش مرتبط (مثلاً بین ۵ تا ۲۰) بازیابی می‌شود؛

۳. بخش‌های بازیابی‌شده برای تحلیل ذخیره و نمایش داده می‌شوند.

۶. مدل زبانی

- مدل: Qwen 2.5 – 7B (Instruct/Chat)؛
- اجرا روی Google Colab؛
- استفاده از quantization (۴ یا ۸ بیت)؛
- بدون fine-tuning و صرفاً با prompt engineering

۷. رابط کاربری

پیاده‌سازی رابط کاربری آزاد است و می‌تواند شامل موارد زیر باشد:

- Notebook تعاملی (Jupyter)؛
- رابط خط فرمان (CLI)؛
- رابط وب سبک (Gardio, Streamlit).

۸. ارزیابی

۱.۸. مجموعهٔ سؤالات مشترک

یک مجموعهٔ ثابت شامل حدود ۱۵ تا ۲۰ سؤال به تمام تیم‌ها داده می‌شود. هر تیم موظف است این سؤال‌ها را از طریق کد به سیستم داده و خروجی را در قالب JSON یا CSV تحویل دهد.

۲.۸. معیارهای نمره‌دهی پاسخ‌ها

برای هر سؤال:

- **درستی محتوایی (۰ تا ۲ نمره)**
- **وفاداری به منبع (۰ یا ۱ نمره)**

۹. نمره‌دهی کل پروژه

- ساخت پایگاه دانش: ۲۰ نمره
- پیاده‌سازی RAG: ۴۵ نمره
- کیفیت پاسخ‌ها روی مجموعهٔ ارزیابی: ۳۰ نمره
- گزارش نهایی و دمو: ۲۰ نمره

-