

به نام خدا



دانشگاه تهران
دانشکده فنی
دانشکده مهندسی برق و کامپیوتر



درس بازیابی هوشمند اطلاعات

تمرین ۴

استاد درس: خانم دکتر آزاده شاکری

سرپرست دستیاران آموزشی: سمانه پیمانی‌راد

طراح تمرین: امیرحسین صفدریان

آذر ماه ۱۴۰۴

۴.....	مقدمه.....
۵.....	سوالات عملی (۶۰ نمره).....
۶.....	سوال ۱: استخراج word association.....
۶.....	مجموعه داده.....
۶.....	پیش پردازش و آماده سازی پیکره.....
۶.....	استخراج روابط Paradigmatic با pseudo-document.....
۷.....	استخراج روابط Syntagmatic با Mutual Information.....
۸.....	سوال ۲: کار با Pretrained Word Embedding.....
۸.....	بارگذاری مدل GloVe.....
۸.....	همسایه های معنایی در فضای Embedding و تحلیل با PCA.....
۹.....	ساخت Query Embedding و Document Embedding برای Cranfield و تحلیل.....
۱۰.....	سوال ۳: پیاده سازی مدل های Learning to Rank.....
۱۰.....	مجموعه داده.....
۱۰.....	ساخت Feature Vector برای هر زوج (Query, Document).....
۱۰.....	پیاده سازی معیارهای DCG, NDCG, Precision و MAP.....
۱۱.....	آموزش و تحلیل مدل Pointwise LTR.....
۱۱.....	آموزش و تحلیل مدل Pairwise LTR.....
۱۱.....	آموزش و تحلیل مدل Listwise LTR.....
۱۳.....	سوالات تئوری (۴۰ نمره).....
۱۴.....	سؤال ۱ - روابط Word Association.....
۱۴.....	سؤال ۲ - تحلیل Mutual Information.....
۱۴.....	سؤال ۳ - ماهیت فضای Word Embedding.....
۱۴.....	سؤال ۴ - مقایسه CBOW و Skip-gram.....
۱۵.....	سؤال ۵ - رویکردهای Pointwise, Pairwise و Listwise در LTR.....
۱۵.....	سؤال ۶ - سازگاری LambdaRank با معیارهای رتبه بندی.....
۱۶.....	سؤال ۷ - شباهت واژگان با ویژگی های Interpretable.....
۱۷.....	ملاحظات (حتما مطالعه شود).....
۱۸.....	استفاده مسئولانه از هوش مصنوعی.....

۱. هدف و اصول کلی.....	۱۸
۲. استفاده مجاز از LLMها.....	۱۸
۳. استفاده غیرمجاز از LLMها.....	۱۹
۴. مستندسازی.....	۱۹
۵. آمادگی ارائه شفاهی.....	۱۹
۶. پیامدهای تخلفات.....	۱۹
۷. موارد تکمیلی.....	۱۹
۸. اظهارنامه.....	۲۰

اهداف تمرین

- درک عملی از انواع روابط واژگانی و مشاهده اینکه چگونه می‌توان آن‌ها را از داده واقعی استخراج کرد.
- آشنایی با ایده نمایش برداری واژه‌ها (word embeddings) و بررسی تفاوت آن با ویژگی‌های معنایی قابل تفسیر.
- ساخت نمایش برداری برای کوئری و سند و استفاده از شباهت‌های برداری برای تحلیل و بازیابی اسناد.
- آشنایی با مفهوم ویژگی‌سازی در IR و اینکه چگونه ویژگی‌های مختلف (آماري، معنایی، مبتنی بر embedding) بر کیفیت رتبه‌بندی تأثیر می‌گذارند.
- درک رویکردهای مختلف Learning to Rank و تفاوت نگاه Pairwise، Pointwise و Listwise به مسئله Ranking.
- توانایی ارزیابی کیفیت سیستم‌های بازیابی با استفاده از معیارهای استاندارد مثل MAP و NDCG.
- پیوند دادن مباحث نظری درس با تجربه عملی بر روی یک مجموعه داده واقعی.

نکات قابل توجه در هنگام پاسخ به سؤالات:

- تمامی کدهای نوشته شده باید قابلیت اجرای مجدد داشته باشند. به تفسیرهایی که بدون آزمایش و صرفاً به صورت فرضی بیان کردند نمره‌ای تعلق نمی‌گیرد.
- نمره اصلی تمرین مربوط به کیفیت و درستی تفسیرهای ارائه شده است.
- بهتر است از **نمودارها** و کشف نمونه‌های مرتبط از اسناد و پرس‌وجوها برای افزایش کیفیت تفسیرها استفاده گردد.
- بدیهی است که حجم تمرین معیار نمره‌ی شما نیست، بلکه صحت انجام آزمایش‌ها و کیفیت تفسیرهای شما مهم است.
- توصیه می‌شود انجام تمرین را به روزهای آخر موکول نکنید! **اجرای آزمایش‌ها و پیدا کردن پارامتر بهینه زمان‌بر بوده** و تفسیر هر کدام نیازمند تحلیل و بررسی است.

سوالات عملی (۶۰ نمره)

این تمرین شامل سه بخش است:

در بخش اول، روی پیکره newsgroups20 روابط واژگانی از نوع paradigmatic و syntagmatic را استخراج می کنید.

در بخش دوم، از یک pretrained word embedding برای تحلیل همسایه های معنایی و همچنین ساخت ویژگی های embedding برای query و document استفاده می کنید.

در بخش سوم، از مجموعه cranfield1400 برای پیاده سازی مدل های Learning to Rank در سه رویکرد Pointwise، Pairwise و Listwise استفاده می کنید و عملکرد آن ها را با معیارهای رتبه بندی مانند NDCG و MAP مقایسه و تحلیل می کنید.

پاسخ سوالات این بخش را در همان نوت بوک تمرین خود (*IIR-CA4-Code.ipynb*) بنویسید.

سوال ۱: استخراج WORD ASSOCIATION ها

مجموعه داده

در این بخش از پیکره *newsgroups20* استفاده می‌کنید. این مجموعه شامل حدود ۲۰ هزار متن از گروه‌های خبری مختلف (*sci*, *rec*, *comp* و ...) است. نسخه آماده این مجموعه به صورت خام یا از طریق تابع *fetch_20newsgroups* در اختیار شما قرار می‌گیرد.

پیش پردازش و آماده سازی پیکره

ابتدا باید داده‌های *newsgroups20* را بارگذاری و پیش‌پردازش کنید. همه متن‌ها را *lowercase* کنید، *punctuation* های ساده (مثل نقطه، ویرگول، علامت سؤال و ...) را حذف کنید و متن را به کلمات *token* کنید. سپس یا هر متن کامل، یا هر جمله را یک *segment* در نظر بگیرید.

در پایان این زیربخش، در گزارش بنویسید:

- چه تصمیم‌هایی برای پیش‌پردازش گرفتید (حذف *stopword*، حداقل فراوانی واژه، ...)
- تعداد *segment* ها و اندازه نهایی *vocabulary* چقدر شده است.
- و در یک پاراگراف کوتاه تحلیل کنید که این نوع پیش‌پردازش چه مزایا و محدودیت‌هایی برای کارهای بعدی شما دارد.

استخراج روابط Paradigmatic با pseudo-document

در این زیربخش باید روابط *paradigmatic* را برای تعدادی واژه محاسبه کنید. ابتدا یک مجموعه نسبتاً بزرگ‌تر از کلمات هدف انتخاب کنید (مثلاً ۱۵-۲۰ واژه نسبتاً پرتکرار و معنادار). برای هر واژه، همه بافت‌هایی را که آن واژه در آن ظاهر می‌شود جمع‌آوری کنید: هر بار که واژه هدف را در یک *segment*

می‌بینید، کلمات موجود در پنجره‌ای با اندازه k (مثلاً ۲ یا ۵) در اطراف آن واژه را در نظر می‌گیرید. تمام این کلمات با هم یک pseudo-document context برای آن واژه می‌سازند.

برای هر pseudo-document یک بردار TF-IDF (یا بردار با وزن‌دهی شبیه BM25) بسازید و سپس cosine similarity بین بردارهای pseudo-document واژه‌های مختلف را محاسبه کنید. به این ترتیب notion ای از شباهت paradigmatic بین واژگان به دست می‌آورد.

برای تحلیل، از میان واژه‌های هدف خودتان ۵ واژه دلخواه انتخاب کنید. برای هر یک از این ۵ واژه، نزدیک‌ترین واژه‌ها از نظر cosine similarity در فضای pseudo-document را (مثلاً ۱۰ همسایه نزدیک) پیدا کنید و در گزارش، برای هر واژه یک جدول حاوی همسایه‌ها بیاورید. سپس برای هر یک از این ۵ واژه در چند جمله توضیح دهید که این همسایه‌ها چه رابطه‌ای با واژه اصلی دارند و آیا از نظر شما واقعاً «هم‌نوع» یا «قابل جایگزینی در context» هستند یا نه.

در ادامه، روی بردارهای pseudo-document همان ۵ واژه و تعدادی از همسایه‌های آن‌ها، PCA را به ۲ بعد اعمال کنید و نقاط را روی یک نمودار دوبعدی رسم کنید (هر نقطه یک واژه، همراه با label آن). در گزارش توضیح دهید که آیا کلماتی که از نظر معنایی به هم نزدیک‌اند در این نمودار نیز نزدیک به هم قرار گرفته‌اند یا خیر، و اگر خوشه‌هایی دیده می‌شود آن‌ها را توصیف کنید.

استخراج روابط Syntagmatic با Mutual Information

در این زیربخش باید روابط syntagmatic را اندازه‌گیری کنید. برای هر واژه مورد نظر، یک متغیر دودویی روی segment‌ها تعریف کنید: مقدار این متغیر ۱ است اگر واژه در آن segment ظاهر شود و ۰ در غیر این صورت. برای جفت‌هایی از واژه‌ها که هر دو حداقل در N segment حضور دارند (مثلاً $N = ۲۰$)، فراوانی حضور هر واژه و حضور هم‌زمان هر دو را بشمارید و با یک smoothing کوچک، احتمال‌ها را تخمین بزنید و مقدار **Mutual Information** را محاسبه کنید.

پس از محاسبه MI برای تعداد زیادی جفت‌واژه، دوباره بر همان ۵ واژه انتخابی تمرکز کنید. برای هر یک از این ۵ واژه، واژگانی را که بالاترین MI را با آن دارند (مثلاً ۱۰ همسایه syntagmatic برتر) استخراج کنید و در گزارش جدول آن‌ها را بیاورید. سپس برای هر واژه، تفاوت بین لیست همسایه‌های paradigmatic (زیربخش قبل) و syntagmatic را در چند جمله توضیح دهید.

سوال ۲: کار با PRETRAINED WORD EMBEDDING

بارگذاری مدل GloVe

در این زیربخش باید فایل GloVe 6B, 100d را بارگذاری کنید و با خواندن آن، یک map از واژه به بردار ۱۰۰ بعدی بسازید. در گزارش بنویسید که اندازه واژگان موجود در فایل چقدر است و تقریباً چه نسبتی از واژه‌های پیکره newsgroups20 شما در این واژگان پوشش داده می‌شوند.

همسایه‌های معنایی در فضای Embedding و تحلیل با PCA

با استفاده از همان ۵ واژه انتخابی بخش اول (به شرط حضور در واژگان GloVe)، همسایه‌های آن‌ها را در فضای embedding پیدا کنید. برای هر واژه، cosine similarity آن را با سایر واژگان محاسبه کرده و نزدیک‌ترین کلمات (مثلاً ۱۰ همسایه) را استخراج کنید. در گزارش، برای هر واژه یک جدول از این همسایه‌ها نشان دهید و درباره نوع رابطه آن‌ها (مترادف، مرتبط موضوعی، هم‌آیند و...) و تفاوت‌ها با همسایه‌های paradigmatic و syntagmatic بخش ۱ صحبت کنید.

سپس embedding‌های همان ۵ واژه و تعدادی از همسایه‌هایشان را با PCA به ۲ بعد کاهش دهید و آن‌ها را روی نمودار دوبعدی رسم کنید. نام هر واژه را کنار نقطه مربوط به آن قرار دهید. در گزارش، به صورت کیفی تحلیل کنید که آیا کلماتی که از نظر شما از نظر معنا به هم نزدیک‌اند در این فضای embedding دوبعدی نیز به هم نزدیک شده‌اند، آیا خوشه‌های قابل توجهی می‌بینید و این خوشه‌ها چه تفاوتی با نمودارهای به دست آمده در بخش ۱ دارند.

تحلیل ساخت Query Embedding و Document Embedding برای Cranfield و

در این زیربخش embedding ها را وارد دنیای Cranfield می‌کنید. ابتدا اسناد و کوئری‌های Cranfield را مشابه بخش ۱ پیش‌پردازش کنید (lowercase، حذف punctuation و tokenization) برای هر document، واژه‌هایی را که در واژگان GloVe هستند انتخاب کنید و document embedding را به صورت میانگین بردارهای آن‌ها بسازید. برای هر query نیز همین کار را انجام دهید و query embedding بسازید.

برای هر زوج (query, document) یک ویژگی embedding تعریف کنید که برابر است با cosine similarity بین query embedding و document embedding. این مقدار را برای همه زوج‌های دارای برچسب ربط در train/val/test محاسبه و ذخیره کنید.

در گزارش، چند query نمونه انتخاب کنید و برای هر query یکی دو document مرتبط و یکی دو document نامرتب پیدا کنید. برای این اسناد مقدار embedding similarity را در جدول بنویسید و در چند جمله تحلیل کنید که آیا این similarity مطابق انتظار شماسست یا خیر؛ یعنی آیا معمولاً برای اسناد مرتبط بالاتر است و اگر موارد خلاف مشاهده کردید، حدس بزنید چرا embedding ممکن است این‌طور رفتار کرده باشد.

سوال ۳: پیاده‌سازی مدل‌های LEARNING TO RANK

مجموعه داده

در این بخش از مجموعه کلاسیک Cranfield 1400 استفاده می‌کنید. این مجموعه شامل حدود ۱۴۰۰ چکیده مقاله در حوزه آیرودینامیک، بیش از ۲۰۰ query و relevance judgmentهای مشخص بین queryها و اسناد است. نسخه این داده همراه با یک تقسیم‌بندی مشخص به train، validation و test (براساس Query ID) در اختیارتان قرار داده می‌شود.

می‌توانید از این آدرس به این مجموعه داده دسترسی پیدا کنید:

https://ir.dcs.gla.ac.uk/resources/test_collections/cran/cran.tar.gz

ساخت Feature Vector برای هر زوج (Query, Document)

در این زیربخش باید برای هر زوج (query, document) در مجموعه‌های train، validation و test یک بردار ویژگی بسازید. حداقل ویژگی‌هایی که باید استفاده کنید عبارت‌اند از: TF/IDF cosine similarity، تعداد یا نسبت کلمات مشترک، طول document (یا log آن)، طول query و embedding-based similarity که در بخش ۲.۴ تعریف کرده‌اید.

اگر مایل بودید می‌توانید ویژگی‌های دیگری (مثلاً میانگین IDF واژه‌های query، یا BM25 یا ویژگی‌های مبتنی بر Word Association) نیز اضافه کنید. در گزارش، یک جدول کوچک تهیه کنید که نام هر feature را بیاورد و در یک جمله توضیح دهد آن feature چه چیزی را اندازه می‌گیرد و چرا ممکن است برای رتبه‌بندی مفید باشد.

پیاده‌سازی معیارهای DCG، NDCG، Precision و MAP

برای محاسبه معیارهای ارزیابی نظیر $DCG@k$ ، $NDCG@k$ ، $Precision@k$ و MAP می‌توانید از فایل evaluation.py که در تمرین‌های قبلی در اختیار شما قرار داده شده است کمک بگیرید.

آموزش و تحلیل مدل Pointwise LTR

در این مرحله باید یک مدل Pointwise Learning to Rank بسازید. هر زوج (query, document) یک نمونه آموزشی مستقل است؛ feature vector آن را در زیربخش ۳.۲ ساخته‌اید و label آن همان درجه ربط است. می‌توانید برای این بخش از مدل‌ها یا کتابخانه‌های آماده یادگیری ماشین (مانند پیاده‌سازی‌های موجود در scikit-learn) استفاده کنید.

از یک مدل مناسب (مثلاً LinearRegression یا Decision Tree) استفاده کنید و آن را روی train آموزش دهید. سپس روی test، برای هر query نمره predicated برای اسناد را محاسبه کرده و براساس این نمره آن‌ها را رتبه‌بندی کنید. با استفاده از کد زیربخش ۳.۳، NDCG@10 و P@10 و MAP را روی query‌های test محاسبه کنید.

در گزارش، علاوه بر نوشتن مقادیر کلی NDCG@10 و P@10 و MAP، چند query را انتخاب کنید که مدل روی آن‌ها خوب یا بد عمل کرده است. برای هر مورد، بخشی از لیست رتبه‌بندی‌شده اسناد را نشان دهید و در چند جمله تحلیل کنید که چرا مدل ممکن است روی آن query خاص موفق یا ناموفق بوده باشد (مثلاً به خاطر کوتاهی/بلندی query، فقر یا غنای featureها، یا رفتار embedding)

آموزش و تحلیل مدل Pairwise LTR

در این زیربخش یک مدل Pairwise Learning to Rank آموزش می‌دهید. برای هر query در مجموعه train، جفت‌سندهایی را که یکی از دیگری مرتبط‌تر است استخراج کرده و از آن‌ها برای آموزش یک مدل pairwise (مانند Ranking SVM) استفاده کنید.

در این بخش نیز می‌توانید از پیاده‌سازی‌های آماده موجود در کتابخانه‌های یادگیری ماشین استفاده کنید. پس از آموزش، روی test مانند مدل pointwise برای هر سند یک score تعریف کنید (مثلاً با استفاده از $w^T x$) و اسناد را بر اساس این score رتبه‌بندی کنید و NDCG@10 و P@10 و MAP را محاسبه کنید. در گزارش، نتایج مدل pairwise را در کنار مدل pointwise در یک جدول قرار دهید و تحلیل کنید که کدام model بهتر عمل کرده است.

آموزش و تحلیل مدل Listwise LTR

در این بخش باید با استفاده از یک کتابخانه آماده (مانند LightGBM یا XGBoost) یک مدل Listwise Learning to Rank بسازید. از فرمت مناسب کتابخانه برای group کردن داده بر اساس Query ID استفاده کنید و از objective رتبه‌بندی (نظیر LambdaMART) بهره بگیرید. می‌توانید با استفاده از داده validation چند پارامتر ساده مثل تعداد درخت‌ها، حداکثر عمق و learning rate را تنظیم کنید.

پس از آموزش، روی test، برای هر query score اسناد را به‌دست بیاورید و NDCG@10 و P@10 و MAP را محاسبه کنید. در گزارش،

(۱) توضیح دهید چه کتابخانه و چه objective استفاده کرده‌اید،

(۲) خروجی نهایی NDCG@10 و P@10 و MAP مدل listwise را کنار pointwise و pairwise قرار دهید،

(۳) برای چند query نمونه لیست‌های رتبه‌بندی سه مدل را مقایسه کنید و در چند جمله تحلیل کنید که کدام روش و به چه دلیل عملکرد بهتری داشته است.

سوالات تئوری (۴۰ نمره)

باتوجه به مباحث تدریس شده سوالات زیر را حل کنید. پاسخ سوالات این بخش باید در قالب مربوط به

سوالات تئوری (HIR-CA4-Theory) بنویسید.

نکات سوالات تئوری:

- صرفاً به جواب نهایی نمره تعلق نمی‌گیرد و راه‌حل نوشته شده بخش اصلی نمره‌ی شماست.
- می‌توانید جهت اطمینان از محاسبات خود از کدنویسی استفاده کنید، اما باید راه‌حل به طور کلی به صورت ریاضی در گزارش تمرین آمده باشد و صرفاً قراردادن کد راه‌حل باعث کسر نمره شما می‌شود.

سؤال ۱ – روابط WORD ASSOCIATION

الف) تفاوت دو نوع رابطه *syntagmatic* و *paradigmatic* را توضیح دهید.

ب) هر کدام چه نوع شباهتی را در زبان مدل می کنند؟

پ) برای هر کدام یک مثال واقعی بزنید.

سؤال ۲ – تحلیل MUTUAL INFORMATION

الف) توضیح دهید چرا *Mutual Information* برای تشخیص واژه هایی که در یک موضوع مشترک مشارکت می کنند مناسب است، حتی اگر شباهت معنایی نداشته باشند.

ب) یک مثال از واژه هایی بدهید که *MI* بالایی دارند ولی ارتباط معنایی مستقیمی ندارند.

سؤال ۳ – ماهیت فضای WORD EMBEDDING

توضیح دهید که *word embedding* ها چه نوع اطلاعاتی را در بردارهای خود رمزگذاری می کنند. سپس مفهوم «خطی بودن» در فضای *embedding* را توضیح دهید و بیان کنید چرا رابطه هایی مثل:

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

در فضای *embedding* مشاهده می شود.

سؤال ۴ – مقایسه CBOW و SKIP-GRAM

در مدل های *Word2Vec*، تفاوت های دو روش *CBOW* و *Skip-gram* را توضیح دهید. در پاسخ خود مشخص کنید:

- ورودی هر مدل چیست؟
- خروجی هر مدل چیست؟
- کدام مدل برای کلمات پرتکرار بهتر است و چرا؟
- کدام مدل برای کلمات کم تکرار بهتر عمل می کند و چرا؟

سؤال ۵ – رویکردهای POINTWISE، PAIRWISE و LISTWISE در LTR

در یادگیری به‌روش رتبه‌بندی (*Learning to Rank*)، سه رویکرد زیر را مقایسه کنید:

- *Pointwise*
- *Pairwise*
- *Listwise*

برای هر کدام بیان کنید:

- چه چیزی را مدل می‌کند (*label* چیست)
- تابع هدف چه فرمی دارد؟
- چه مزایا و محدودیت‌هایی دارد؟
- کدام رویکرد با معیارهایی مثل *NDCG* سازگارتر است و چرا؟

سؤال ۶ – سازگاری LAMBDARANK با معیارهای رتبه‌بندی

در الگوریتم *LambdaRank* توضیح دهید:

- مفهوم *λ -gradient* چیست؟
- چرا تغییر رتبه سندهای بالای لیست نسبت به سندهای پایین‌تر *gradient* بزرگ‌تری ایجاد می‌کند؟
- چگونه این روش به‌طور مؤثر با معیارهایی مانند *NDCG* سازگار می‌شود؟

سؤال ۷ - شباهت واژگان با ویژگی‌های INTERPRETABLE

در جدول زیر تعدادی واژه و چند ویژگی معناساختی *interpretable* آورده شده‌اند (۰ و ۱ نشان‌دهنده نبودن/بودن ویژگی است).

Word	is_animal	has_tail	can_fly	is_fruit	is_vehicle	is_food
<i>dog</i>	۱	۱	۰	۰	۰	۰
<i>cat</i>	۱	۱	۰	۰	۰	۰
<i>bird</i>	۱	۰	۱	۰	۰	۰
<i>apple</i>	۰	۰	۰	۱	۰	۱
<i>banana</i>	۰	۰	۰	۱	۰	۱
<i>car</i>	۰	۰	۰	۰	۱	۰

الف) برای هر واژه، بردار ویژگی آن را یادداشت کنید.

ب) *cosine similarity* را برای جفت‌های زیر محاسبه کنید:

۱. *cat, dog*

۲. *bird, dog*

۳. *banana, apple*

۴. *apple, dog*

۵. *car, apple*

پ) نتایج به‌دست‌آمده را تحلیل کنید:

- کدام جفت‌ها بیشترین شباهت را دارند؟
- چه نوع شباهتی در این فضای *interpretable* اندازه‌گیری شده است؟
- این شباهت‌ها چگونه با شباهتی که از *word embedding* انتظار داریم تفاوت دارد؟
- به نظر شما کدام نوع نمایش برای مدل کردن *paradigmatic* مناسب‌تر است و کدام برای *syntagmatic*؟

تمامی نتایج شما باید در یک فایل فشرده با عنوان IIR-CA4-StudentID تحویل داده شود.

- خوانایی و دقت بررسی‌ها در گزارش نهایی از اهمیت ویژه‌ای برخوردار است. به تمرین‌هایی که به صورت کاغذی تحویل داده شوند یا به صورت عکس در سایت بارگذاری شوند، ترتیب اثری داده نخواهد شد. **دقت** کنید که **حتما گزارشات خود را در قالب ارائه شده برای تحویل تکالیف که در سامانه برای شما بارگذاری شده است ارسال بفرمایید.**
- کدهای نوشته شده برای هر بخش را با نام مناسب مشخص کرده و به همراه گزارش تکلیف ارسال کنید. همه‌ی کدهای پیوست گزارش بایستی قابلیت اجرای مجدد داشته باشند. در صورتی که برای اجرا مجدد آن‌ها نیاز به تنظیمات خاصی می‌باشد بایستی تنظیمات مورد نیاز را نیز در گزارش خود ذکر کنید. **دقت** کنید که **تمامی کدها باید توسط شما اجرا شده باشند و نتایج اجرا در فایل کدهای ارسالی مشخص باشد.** **به کدهایی که نتایج اجرای آن‌ها در فایل ارسالی مشخص نباشد نمره‌ای تعلق نمی‌گیرد.**
- تمرین تا یک هفته بعد از مهلت تعیین شده با تاخیر تحویل گرفته می‌شود. دقت کنید که شما جمعا برای تمام تکالیف، ۱۴ روز زمان تحویل بدون جریمه دارید که تنها از ۷ روز آن برای هر تمرین می‌توانید استفاده کنید، در صورتی که این ۱۴ روز به اتمام رسیده باشد، به ازای هر روز تأخیر در ارسال تمرین، ده درصد جریمه می‌شوید.
- **توجه کنید این تمرین باید به صورت تک نفره انجام شود و پاسخ‌های ارائه شده باید نتیجه فعالیت فرد نویسنده باشد (همفکری و به اتفاق هم نوشتن تمرین نیز ممنوع است).** در صورت مشاهده تشابه به همه افراد مشارکت کننده، نمره ۵۰- تعلق می‌گیرد و به استاد نیز گزارش می‌گردد.
- در صورت بروز هرگونه مشکل با ایمیل زیر در ارتباط باشید:

<mailto:safdarian2000@gmail.com>

استفاده مسئولانه از هوش مصنوعی

۱. هدف و اصول کلی

هدف

- ترویج استفاده اخلاقی و مسئولانه از LLMها (مانند ChatGPT، Deepseek) به عنوان ابزار کمکی
- اطمینان از مشارکت فعال دانشجویان در تکالیف و درک راه‌حل‌های آن‌ها
- حفظ صداقت علمی در عین بهره‌گیری از ابزارهای مدرن هوش مصنوعی

اصول کلی

- تمرین باید نتیجه تلاش و زحمت شخصی شما باشد.
- باید به تمام بخش‌های تمرین، اعم از پیاده‌سازی و تحلیل نتایج مسلط باشید.
- تمامی کدها باید توسط خود شما اجرا شده و نتایج قابل مشاهده باشند.
- تمام مراحل انجام تمرین باید مستند و قابل پیگیری باشد.
- هرگونه نتیجه‌گیری و تحلیل باید بر اساس درک شخصی شما باشد.
- LLMها ممکن است پاسخ‌های نادرست یا قدیمی تولید کنند، اولویت با مطالب و کارگاه‌های درس است.

موارد ذکر شده در ادامه این سند، به عنوان راهنمایی بیشتر برای انجام تمرین آورده شده‌اند. با این حال، مسئولیت تطبیق کار با اصول کلی فوق بر عهده شماست. توجه داشته باشید که ممکن است مواردی در ادامه ذکر نشده باشند که با اصول کلی ذکر شده در تضاد باشند. در چنین مواردی به تشخیص دستیار آموزشی و دستیار مسئول، شما موظف به پاسخ‌گویی در قبال تمرین خود هستید. عدم رعایت هر یک از اصول فوق می‌تواند منجر به کسر نمره یا عدم پذیرش تمرین شود.

۲. استفاده مجاز از LLMها

شما می‌توانید از LLMها برای موارد زیر استفاده کنید:

- روشن‌سازی مفاهیم (مثال: "خوشه‌بندی DBSCAN چگونه کار می‌کند؟")
- کمک در اشکال‌زدایی (مثال: شناسایی خطاهای گرامری یا منطقی در کد)
- ایده‌پردازی رویکردها (مثال: "روش‌های مدیریت داده‌های missing را پیشنهاد دهید")

الزامات استفاده مجاز:

- ثبت تعاملات اصلی: (به بخش ۴ مراجعه کنید).
- درک راه‌حل: باید قادر به توضیح هر خط کد یا منطق استفاده شده باشید.

۳. استفاده غیرمجاز از LLMها

اقدامات ممنوع شامل:

- کپی-پیست مستقیم خروجی‌های LLM بدون تغییر
- استفاده از LLMها برای حل اصلی مسائل (مثال: "این سؤال تکلیف را برای من حل کن")
- گرفتن کد از سایر دانشجویان به هر شکل غیر مجاز است، تغییر و پارافریز کردن کد دیگران توسط LLM نیز قابل قبول نیست.
- هرگونه استفاده که منجر به عدم احاطه شما به موضوع تمرین شود.

۴. مستندسازی

ارجاع به مشارکت‌های LLM: افزودن پانویس یا توضیح (مثال: کد با رعایت قوانین به کمک ChatGPT نوشته شده است).

- نیازی به اشتراک گذاری پرامپت‌ها و سابقه چت نیست.
- مستندسازی تک تک تعاملات با هوش مصنوعی هدف این بخش نیست. اشاره کوتاه و کلی در بخش‌های مورد استفاده کافی است. در نظر داشته باشید که مستندسازی به معنای رفع مسئولیت نبوده و باید اصول کلی را رعایت کنید.

۵. آمادگی ارائه شفاهی

آماده دفاع از کار خود باشید: در صورت درخواست دستیار تمرین در بازه زمانی اعلام شده برای ارائه شفاهی، باید:

- رویکرد، کد یا نتایج خود را توضیح دهید.
- درک مفاهیم کلیدی را نشان دهید (مثلاً چرا یک الگوریتم خاص انتخاب شده است)
- عدم توضیح کافی کار شما ممکن است منجر به جریمه شود (بخش ۶)

۶. پیامدهای تخلفات

- تخلفات جزئی (مثل مستندسازی ناقص): کاهش نمره
- تخلفات عمده (مثل کپی-پیست بدون تغییر): نمره ۵۰- در تکلیف
- تخلفات مکرر: نمره ۵۰- در تکلیف و گزارش به استاد

۷. موارد تکمیلی

- از LLMها به عنوان معلم استفاده کنید، نه پاسخ‌نامه تمرین‌ها: اولویت را به مهارت‌های حل مسئله خود بدهید.

- خروجی‌ها را متقابلاً تأیید کنید: پیشنهادات LLM را با کتاب مرجع درس، اسلایدها و کارگاه‌ها مقایسه کنید.
- از دستیاران آموزشی کمک بگیرید: اگر پاسخ LLM یا نحوه استفاده شما را گیج می‌کند، در ساعات متعارف از دستیاران آموزشی کمک بگیرید.

۸. اظهارنامه

این عبارت را در تکلیف ارسالی خود قرار دهید:

"تأیید می‌کنم که از LLMها مطابق با دستورالعمل‌های بارگذاری شده در سامانه Elearn درس به طور مسئولانه استفاده کرده‌ام. تمام اجزای کار خود را درک می‌کنم و آماده بحث شفاهی درباره آنها هستم."