هوش مصنوعی بهار 1405 **تمرین کامپیوتری چهارم**

مدرس: دکتر فدائی

طراحان: پارسا دقیق، یاسمن عموجعفری و مصطفی کرمانی‌نیا

مهلت تحویل: پایان
یک‌شنبه، 24 خرداد

مقدمه

در این پروژه، شما در قالب یک روایت داستانیِ پیوسته، مهم‌ترین الگوریتم‌های طبقه‌بندی در یادگیری ماشین را هم با استفاده از ابزارهای آماده و هم با پیاده‌سازی از پایه تجربه خواهید کرد. داستان ما از دل چالش‌های کنونی همه‌ی دانشجوها آغاز می‌شود:

بخش ۱ – پیام‌رسان محلی (دانش‌چت)

با مجازی شدن دانشگاه و قطعی اینترنت، مصطفی که از باگ‌های سوپر اپلیکیشن‌های داخلی خسته شده است، تصمیم گرفته تا برای فرار از افسردگی و انزوا، خودش یک پیام‌رسان جدید به نام دانش‌چت روی سرورهای دانشگاه راه‌اندازی کند تا دانشجویان بتوانند از آن طریق با یکدیگر در ارتباط باشند. اما این سیستم نوپا با دو چالش اساسی مواجه است:

۱ .کامپیوترهای واسطِ دانشگاه که وظیفه انتقال پیام‌ها را دارند، بسیار ضعیف هستند. به همین دلیل در زمان‌های شلوغی، خیلی از پیام‌ها در میانه راه گم می‌شوند و هرگز به دست گیرنده نمی‌رسند.

۲ .عده‌ای از دانشجویان با ساخت ربات‌های خودکار، در حال ارسال پیام‌های تبلیغاتی و اسپم هستند که این موضوع سیستم را بی‌دلیل کندتر کرده است.

شما به عنوان تیم یادگیری ماشین این سامانه، باید دو مدل توسعه دهید: مدلی برای پیش‌بینی گم شدن پیام و مدلی برای تشخیص پیام‌های اسپم.

آشنایی با مجموعه داده

داده‌های ثبت شده از پیام‌ها در فایلی با نام daneshchat_logs.csv قرار دارد. برای سادگی کار، متن پیام‌ها پیشاپیش تحلیل شده و ویژگی‌های کلیدی آن استخراج شده است.

نام ستون	توضیح
message_size_kb	حجم پیام (بر حسب کیلوبایت).
transfer_steps	تعداد کامپیوترهایی که پیام باید از آن‌ها عبور کند تا به گیرنده برسد.
connection_quality	کیفیت اتصال سیستم در آن لحظه (عدد اعشاری بین ۰ و ۱).
has_urgent_keyword	آیا کلماتی مثل "ددلاین"، "کلاس" یا "امتحان" در پیام بوده است؟ (۱: بله، ۰: خیر).
has_link	آیا پیام حاوی لینک است؟ (۱: بله، ۰: خیر).
is_spam	(متغیر هدف اول): آیا پیام اسپم است؟ (۱: اسپم، ۰: سالم).
is_lost	(متغیر هدف دوم): آیا پیام در مسیر گم شد؟ (۱: گم شد، ۰: رسید).

توسعه هسته هوش مصنوعی از صفر

با توجه به عدم دسترسی مصطفی به اینترنت برای دانلود پکیج‌های آماده، شما باید موتور اصلی پیش‌بینی را خودتان بنویسید. در این بخش انتظار می‌رود کلاس درخت تصمیم (Decision Tree) را بدون استفاده از کتابخانه‌هایی مثل Sklearn و صرفا با Numpy و Pandas پیاده‌سازی کنید. این کلاس باید شامل متدهای زیر باشد:

متد __init__: تعریف آرگومان‌های اولیه. حتماً باید آرگومان max_depth (حداکثر عمق درخت) در آن تعبیه شود تا بتوانیم از پیچیدگی بیش از حد درخت جلوگیری کنیم.

متد fit: دریافت داده‌های آموزش (X_train و y_train) و ساختاردهی درخت بر اساس آن‌ها. در هر گره، بهترین ویژگی برای تقسیم داده‌ها را با محاسبه معیار آنتروپی (Entropy) پیدا کنید.

متد predict: دریافت داده‌های جدید، پیمایش درخت تا رسیدن به برگ‌ها، و برگرداندن لیبل پیش‌بینی شده.

پیش‌بینی وضعیت رسیدن پیام‌ها (با استفاده از درخت دست‌ساز)

در این بخش می‌خواهیم گم شدن یا رسیدن پیام‌ها (is_lost) را پیش‌بینی کنیم.

1. متغیر هدف را is_lost در نظر بگیرید (دقت کنید ستون is_spam را از فیچرهای ورودی حذف کنید تا مدل گمراه نشود). داده‌ها را به نسبت مناسب به Train و Validation تقسیم کنید .
2. مدل درخت دست‌ساز خود را روی داده‌های آموزش فیت کرده و پیش‌بینی را روی Validation انجام دهید.
3. معیارهای Accuracy، Precision، Recall و F1-Score را محاسبه کرده و ماتریس درهم‌ریختگی (Confusion Matrix) را رسم و به اختصار تحلیل کنید.

فیلترینگ پیام‌های مزاحم با ابزارهای آماده

سرعت تشخیص برای پیام‌های اسپم (is_spam) حیاتی است؛ بنابراین از الگوریتم سریع Naive Bayes استفاده می‌کنیم. (در این بخش مجاز به استفاده از کتابخانه Sklearn هستید).

1. متغیر هدف را is_spam در نظر بگیرید (ستون is_lost را از فیچرها حذف کنید.)
2. یک مدل Naive Bayes آموزش دهید و متریک‌های ارزیابی (Accuracy و F1-Score) را روی Validation گزارش کنید.

3. برای مقایسه، یک مدل RandomForestClassifier از Sklearn فراخوانی کنید. با استفاده از GridSearchCV، بهینه‌ترین عمق (max_depth) را پیدا کرده و دقت آن را با Naive Bayes مقایسه کنید.

سوالات تحلیلی

1. الگوریتم Naive Bayes فرض می‌کند تمام ویژگی‌ها از هم مستقل‌اند. آیا در داده‌های ما، فرض استقلال برای متغیرهایی مثل has_urgent_keyword و has_link منطقی است؟ این فرض چه تاثیری روی عملکرد مدل ما می‌گذارد؟

2. اگر درخت تصمیم خود را اجباراً با عمق بسیار کم (مثلاً max_depth = 1) آموزش دهید، چه اتفاق برای متریک‌ها می‌افتد؟ این پدیده در یادگیری ماشین چه نام دارد و نشان‌دهنده غلبه Bias است یا Variance؟

3. در مسئله پیش‌بینی گم شدن پیام‌ها (is_lost)، از بین دو متریک Precision و Recall، کدام یک برای ما اهمیت بیشتری دارد؟ (اشتباه گرفتن یک پیام سالم به عنوان گم‌شده آسیب بیشتری دارد یا اشتباه گرفتن یک پیام واقعاً گم‌شده به عنوان سالم؟) تحلیل خود را بنویسید.

بخش ۲ – رادار سلامت روان

با وجود راه‌اندازی دانش‌چت، یاسمن که یکی دیگر از دانشجویان درگیر با وضعیت کنونی است، متوجه می‌شود که دانشجویان همچنان بی‌انگیزه و خسته‌اند. او به این نتیجه می‌رسد که نبود ارتباطات تنها یک عامل کوچک بوده و عوامل دیگری مثل دنبال کردن اخبار منفی، تمرین‌های تلنبار شده و تلاش‌های پیاپی و ناموفق برای اتصال فیلترشکن نیز فشار روانی زیادی به بچه‌ها وارد می‌کند. بنابراین، یاسمن تصمیم می‌گیرد سامانه‌ای هوشمند طراحی کند تا وضعیت سلامت روان دانشجویان را مانیتور کرده و در صورت نیاز، هشدارهای لازم برای استراحت را صادر کند. شما در این بخش باید به او کمک کنید تا مدلی توسعه دهد که با گرفتن اطلاعات روزمره دانشجو، وضعیت سلامت روان او را در سه سطح (سالم، خسته، داغون) طبقه‌بندی کند.

آشنایی با مجموعه داده

داده‌های جمع‌آوری شده از وضعیت روزمره دانشجویان در فایلی با نام student_burnout.csv قرار دارد.

نام ستون	توضیح
sleep_hours_avg	میانگین ساعات خواب دانشجو در روز.
news_read_count	تعداد اخبار منفی خوانده شده در روز.
vpn_connection_attempts	تعداد دفعات تلاش ناموفق برای وصل شدن به فیلترشکن.
pending_hw	تعداد تمرین‌های دانشگاهی تلنبار شده و انجام نشده.
screen_time_hours	کل ساعات خیره شدن به صفحه نمایشگر لپ‌تاپ.
mental_state	متغیر هدف: وضعیت خستگی ذهنی دانشجو (۰: سالم، ۱: خسته، ۲: داغون).

توسعه مدل طبقه‌بند از صفر

برای درک عمیق‌تر الگوریتم‌های ریاضی، در این بخش باید مدل رگرسیون لجستیک چندکلاسه را کاملاً از پایه و تنها با استفاده از کتابخانه Numpy پیاده‌سازی کنید.

کلاس شما باید دارای متدهای زیر باشد:

متد __init__: تعریف learning_rate و تعداد epochها

متد fit: پیاده‌سازی حلقه آموزش. در این بخش باید تابع هزینه (Cross-Entropy Loss) را محاسبه کرده و ماتریس وزن‌ها را با الگوریتم گرادیان کاهشی (Gradient Descent) آپدیت کنید.

متد predict: محاسبه احتمالات خروجی با استفاده از تابع Softmax و برگرداندن کلاسی که بیشترین احتمال را به خود اختصاص داده است.

آماده‌سازی داده‌ها و آموزش مدل

۱. داده‌ها را به دو بخش آموزش (Train) و اعتبارسنجی (Validation) تقسیم کنید.

۲. داده‌های ویژگی (X) را اسکیل کنید (مثلاً با StandardScaler).

۳. مدل دست‌ساز خود را آموزش دهید.

۴. جهت مقایسه، از کتابخانه Sklearn مدل LogisticRegression را فراخوانی کرده و روی داده‌ها آموزش دهید. زمان اجرای آموزش (Training Time) و دقتِ مدل دست‌ساز خود را با مدل کتابخانه‌ای مقایسه کنید.

ارزیابی داده‌های نامتوازن

از آنجا که خوشبختانه تعداد دانشجویانِ داغون نسبت به دانشجویان سالم در این مجموعه داده کمتر است (Imbalanced Data)، استفاده از معیار Accuracy به تنهایی گول‌زننده است.

1. متریک Macro F1-Score را برای مدل‌های خود محاسبه کنید و تفاوت آن را با Accuracy نشان دهید.
2. ماتریس درهم‌ریختگی را رسم کنید و بررسی کنید مدل بیشتر در تشخیص کدام کلاس‌ها دچار خطا می‌شود.

سوالات تحلیلی

1. در مدل‌هایی که با گرادیان کاهشی (Gradient Descent) بهینه می‌شوند، اسکیل کردن داده‌ها (Scaling) چه تاثیر ریاضی روی شکلِ تابع هزینه و سرعتِ همگرایی مدل دارد؟ اگر داده‌ها را اسکیل نکنیم چه خطری مدل دست‌ساز شما را تهدید می‌کند؟
2. پس از آموزش مدل دست‌ساز، ماتریس وزن‌های (W) مدل خود را چاپ کنید. با بررسی علامت و بزرگیِ وزن‌های مربوط به ویژگیِ pending_hw، تحلیل کنید که آیا تمرین‌های تلنبار شده باعث افزایش احتمال قرار گرفتن دانشجو در کلاس‌های شدیدتر، مخصوصاً کلاس «داغون»، می‌شود یا خیر.
3. فرض کنید می‌خواهیم این مسئله ۳ کلاسه را به جای تابع Softmax، با روش One-vs-Rest (ساخت ۳ مدل باینری جداگانه) حل کنیم. تفاوت ساختاری این روش با مدل چندکلاسه فعلی چیست و در چه شرایطی ممکن است یکی بر دیگری برتری داشته باشد؟

بخش ۳ – پیام‌نوریاب

پس از استفاده از سامانه یاسمن، مصطفی متوجه شد که وضعیت او واقعا داغون است و دلیل اصلی این وضعیت هم دسترسی نداشتن به اینترنت آزاد برای انجام پروژه‌هایش است. طبق اخبار، او می‌فهمد که با رفتن به دانشگاه پیام‌نور کرج، که در کوهستان‌های اطراف کرج قرار دارد، می‌تواند به اینترنت دسترسی پیدا کند. برای همین بلافاصله تصمیم می‌گیرد راهی این مسیر کوهستانی شود.

اما مسیر ناشناخته و خطرناک است. مصطفی برای اینکه بی‌گدار به آب نزند، به یک مدل هوشمند نیاز دارد تا بر اساس شرایط محیطی و وضعیت دانشجویان، پیش‌بینی کند که آیا مسیر صعود امن است یا نه.

آشنایی با مجموعه داده

داده‌های جغرافیایی مسیر و شرایط دانشجویان داوطلب در فایلی با نام trekking_expedition.csv جمع‌آوری شده است.

نام ستون	توضیح
slope_angle	زاویه شیب مسیر صعود (درجه).
wolf_prob	تخمین احتمال حضور گرگ در مسیر بر اساس گزارشات محلی (۰ تا ۱).
rain_mm	میزان بارش در ۲۴ ساعت گذشته (میلی‌متر).
cold_resistance	مقاومت دانشجو در برابر سرما (Low, Medium, High).
is_safe	متغیر هدف: آیا این مسیر برای صعود امن است؟ (۱: امن، ۰: خطرناک).

مهندسی ویژگی

پارسا که از دوستان مصطفی و متخصص دیتا است، متوجه می‌شود که مدل اولیه دقت کافی ندارد. به نظر او، خطرناک بودن مسیر فقط به یک عامل وابسته نیست، بلکه ترکیبی از شیب مسیر، میزان بارش و احتمال حضور گرگ می‌تواند وضعیت مسیر را بهتر توضیح دهد.

برای همین از شما می‌خواهد یک ویژگی جدید به نام environmental_danger بسازید. این ویژگی باید با استفاده از فرمول زیر محاسبه شود:

$$environmental\ danger\ =\ slope\ angle\ \times\ rain\ mm\ \times\ (1\ +\ wolf\ probability)$$

پیش‌پردازش و مدل‌سازی

در این بخش برخلاف بخش‌های قبلی، با داده‌هایی طرف هستیم که هم ستون عددی دارند و هم ستون دسته‌ای. می‌خواهیم عملیات پیش‌پردازش را به شکلی انجام دهیم که مدل نهایی روی داده‌های ندیده نیز درست عمل کند و دچار Data Leakage نشود.

1. داده‌ها را ابتدا به Train و Validation تقسیم کنید.
2. یک OneHotEncoder را **فقط** روی ستون cold_resistance داده‌های Train فیت کرده و از آن برای تبدیل این ستون در هر دو مجموعه Train و Validation استفاده کنید.
3. به‌طور مشابه، یک StandardScaler را **فقط** روی ستون‌های عددی Train فیت کرده و همان را به کل داده‌های Train و Validation اعمال کنید.
4. پس از آماده‌سازی، یک مدل DecisionTreeClassifier از sklearn را روی Train آموزش دهید و روی Validation معیارهای Accuracy و F1-Score را گزارش دهید.

سوالات تحلیلی

1. اگر پیش ازSplit ، همه داده‌ها را یکجا با StandardScaler اسکیل می‌کردیم، اطلاعات آماری Validation (میانگین و واریانس) وارد فرآیند آموزش می‌شد. این خطا که Data Leakage نام دارد، باعث می‌شود مدل در مرحله ارزیابی عملکرد بهتری از آنچه در واقعیت دارد نشان دهد. توضیح دهید که انجام فیت روی Train و سپس Transform روی Validation چگونه از این مشکل جلوگیری می‌کند.
2. با استفاده از تابع plot_tree، ساختار درخت تصمیم آموزش‌دیده را رسم کنید. دو مورد از قوانین (Rules) جالبی که درخت برای تشخیص «امن بودن مسیر» کشف کرده است را از روی تصویر استخراج کرده و بنویسید.
3. ویژگی cold_resistance در ذات خود دارای یک ترتیب منطقی است ($Low < Medium < High$). ما در این بخش از OneHotEncoder استفاده کردیم که این مقادیر را به ستون‌های صفر و یک و کاملاً مستقل از هم تبدیل می‌کند. اگر به جای آن از OrdinalEncoder استفاده می‌کردیم (و به آن‌ها مقادیر ۱، ۲ و ۳ می‌دادیم)، چه تفاوتی در نحوه تصمیم‌گیری (Split) گره‌های درخت تصمیم ایجاد می‌شد؟ کدام روش برای این ستون خاص منطقی‌تر است؟

بخش ۴ – سمفونی کوهستان

دو ماه بعد مصطفی درحالی که در کوهستان به سوی دانشگاه پیام‌نور می‌رفت، پارسا را دید که کل این مدت در جنگل گم شده بود و به علت تغذیه از گیاهان کوهی، حال و هوای عجیبی پیدا کرده بود و در میان درختان مه‌گرفته‌ی کوهستان، گیتار دست‌سازش را می‌نواخت. مصطفی که از نوای گیتار او به وجد آمده بود، پیشنهاد داد که انجمن موسیقی کوهی دانشگاه را راه بیندازند و برای جشن عیدانه‌ی مجازی/کوهستانی دانشگاه یک کنسرت آنلاین برگزار کنند. برای اینکه پلی‌لیست عیدانه برای تمام دانشجویان جذاب باشد، باید به پارسا کمک کنید که سلیقه‌ی موسیقی دانشجویان را پیش‌بینی کند. در این بخش، با استفاده از الگوریتم نزدیک‌ترین همسایه‌ها (KNN)، مدلی می‌سازید که با دریافت ویژگی‌های جمعیت‌شناختی و رفتاری دانشجو، ژانر موسیقی مورد علاقه‌ی او را تشخیص دهد.

آشنایی با مجموعه داده

اطلاعات ثبت‌شده در فایل student_music_taste.csv جمع‌آوری شده است.

نام ستون	توضیح
age	سن دانشجو (۱۸ تا ۳۰ سال)
daily_music_hours	ساعت گوش‌دادن به موسیقی در روز (۰.۰ تا ۱۰.۰)
favorite_instrument	ساز مورد علاقه (Guitar, Piano, Violin, None)
personality_trait	تیپ شخصیتی (Introvert, Extrovert, Ambivert)
music_genre	متغیر هدف: ژانر محبوب (Rock, Pop, Classical, Traditional)

آماده‌سازی داده‌ها و آموزش مدل

در این بخش از کلاس KNeighborsClassifier در کتابخانه‌ی sklearn استفاده کنید.مراحل کار به صورت زیر است:

1. داده‌ها را به دو بخش Train و Validation تقسیم کنید.

2. ستون‌های دسته‌ای(favorite_instrument و personality_trait) را با استفاده از OneHotEncoder به داده‌های عددی تبدیل کنید.

3. ستون‌های age و daily_music_hours را با استفاده از StandardScaler اسکیل کنید.

4. یک مدل KNeighborsClassifier با مقدار k=5 بسازید و آن را روی داده‌های Train آموزش دهید.

5. عملکرد مدل را روی داده‌های Validation بررسی کرده و معیارهای Accuracy و Macro F1-Score را گزارش کنید.

سؤالات تحلیلی

1. از نظر ریاضی و فرمول فاصله اقلیدسی، توضیح دهید که چرا اسکیل کردن داده‌ها برای الگوریتم KNN حیاتی است، اما برای درخت تصمیم (در بخش‌های قبلی) هیچ ضرورتی نداشت؟

2. فرض کنید به جای OneHotEncoder، از LabelEncoder برای ستون favorite_instrument استفاده می‌کردیم (مثلاً Guitar=1, Piano=2, Violin=3, None=4). چه مشکل ریاضی و منطقی در محاسبه فاصله نمونه‌ها توسط الگوریتم KNN به وجود می‌آمد؟

3. یک حلقه for بنویسید و مدل خود را برای مقادیر $k \in \{1,3,5,7,11,21,51\}$ آموزش دهید. نموداری رسم کنید که محور افقی آن مقادیر k و محور عمودی آن مقدار Accuracy باشد. روی این نمودار، دو خط رسم کنید (یکی برای دیتای Train و یکی برای Validation). بر اساس این نمودار توضیح دهید که در الگوریتم KNN، افزایش مقدار k چه تاثیری بر Bias و Variance مدل دارد؟ (کدام مناطق نمودار Overfit و کدام Underfit هستند؟)

4. از نظر تئوری، زمان لازم برای پیش‌بینی (Predict) یک نمونه جدید در الگوریتم درخت تصمیم بسیار کمتر از الگوریتم KNN است. دلیل این تفاوت فاحش در فاز پیش‌بینی چیست؟

نکات پایانی

- در شرایط فعلی که دسترسی به اینترنت بین‌الملل نداریم، می‌توانید پکیج‌های پایتون موردنیاز خود را از این mirror دریافت کنید. برای مثال کافی‌ست دستور زیر را اجرا کنید:

```
pip install -i https://mirror-pypi.runflare.com/simple {package_name}
```

  در صورت ابهام این لینک راهنما را مطالعه کنید.

- کد شما باید به نحوی زده شود که نتایج با پارامترهای مختلف قابلیت بازتولید داشته باشند.
- توضیحات مربوط به هر بخش از تمرین را خلاصه و مفید در گزارش ذکر کنید. حجم توضیحات هیچ تاثیری در نمره نخواهد داشت و تحلیل شماست که ارزشمند است.
- فایل‌های خود که شامل گزارش و کد می‌باشند را در قالب یك فایل فشرده با فرمت Al_CA4_[stdNum].zip در سامانه‌ی elearn بارگذاری کنید. برای مثال:

Al_CA4_810103127.zip

- توجه کنید این تمرین باید به صورت تک‌نفره انجام شود و پاسخهای ارائه شده باید نتیجه‌ی فعالیت فرد نویسنده باشد. در صورت مشاهده تقلب به همه‌ی افراد مشارکت‌کننده، نمره‌ی 100- داده شده و به استاد نیز گزارش می‌گردد. همچنین نوشته شدن کدها توسط هوش مصنوعی نیز بررسی میشود!
- سوالات خود را از طراحان تمرین از طریق ایمیل یا گروه درس بپرسید.

موفق باشید.