هوش مصنوعی

تمرین شماره‌ی 3

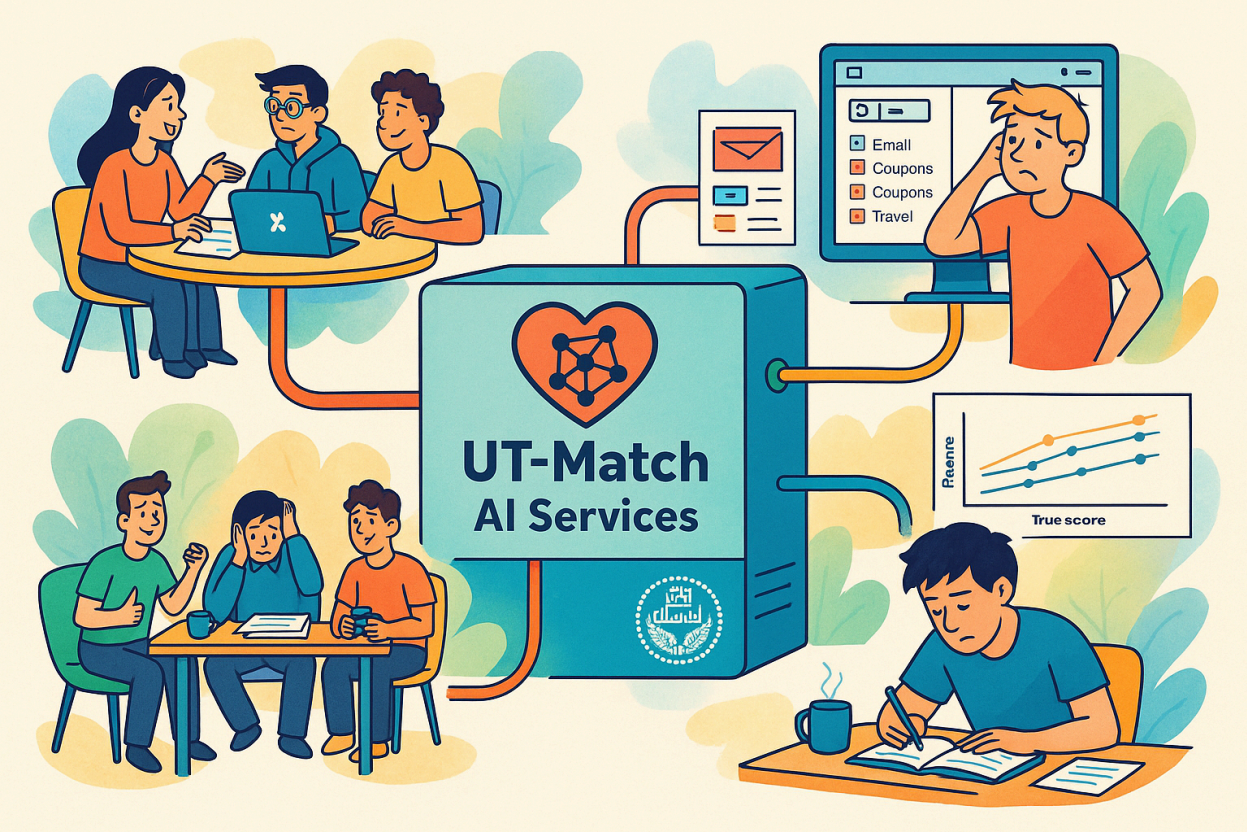
مدرسین: **دکتر فدایی و دکتر یعقوب‌زاده**

طراحان: محمد امانلو، امین آقاکثیری، مهدی نائینی

مهلت تحویل: یکشنبه ۱۶ آذر ۱۴۰۴، ساعت ۲۳:۵۹

مقدمه

دانشکده برق و کامپیوتر تصمیم گرفته یک سامانه‌ی هوشمند داخلی به اسم UT-Match بسازد که با استفاده از روش‌های یادگیری ماشین، رفتار و وضعیت تحصیلی/روانی دانشجویان را پیش‌بینی کند.
در این پروژه، شما نقش تیم ML این سامانه را دارید و باید چند زیرسیستم مختلف را طراحی و پیاده‌سازی کنید.



برای انجام هر چه بهتر این تمرین نیز می‌توانید از این دفترچه به عنوان راهنما در پیاده‌سازی بخش‌های اول و دوم تمرین استفاده کنید.

توضیح مسئله

با توجه به دستورالعمل امنیتی دانشگاه شما در این پروژه امکان استفاده از مدل های آماده کتابخانه‌ای را مگر در صورت کسب مجوز کتبی از تیم امنیت دانشگاه ندارید. برای انجام پیش‌بینی‌ها، ضروری است در ابتدا مدل درخت تصمیم توسط تیم ML این سامانه از صفر تا صد پیاده سازی شود. در این بخش، از شما انتظار می رود که یک کلاس درخت تصمیم را به صورت from scratch پیاده سازی کنید. پیاده سازی کلاس این مدل، باید بدین صورت باشد که حتما دارای سه متد زیر باشد:

1. متد init : در اینجا آرگومان هایی که فکر می کنید ممکن است مدل نیاز داشته باشد را جهت instanciate کردن مدل، تعریف کنید. یکی از آرگومان هایی که حتما نیاز است تعریف شود، آرگومان max_depth است؛ این آرگومان مشخص می کند که درخت تا چه عمقی ساخته شود؛ به عنوان مثال اگر عمق درخت 4 باشد، در یک پیمایش از ریشه درخت، با طی کردن 4 گره از درخت، به برگ می رسیم که در آن، برای خروجی دادن، از عملیات majority vote استفاده می کنیم؛ منظور از این عملیات، این است که در آن زیر درخت خاص(که در این مثال با پیمایش 4 ویژگی به آن رسیدیم)، به ازای هر برچسب تعداد نمونه ها را می شماریم؛ به عنوان مثال اگر در این زیردرخت 5 نمونه از برچسب 0 و 7 نمونه از برچسب 1 داشته باشیم، در اینجا برچسب داده تست، 1 خواهد بود.
2. متد fit : ورودی این متد، X_train و y_train یا به عبارتی داده ها و برچسب آن ها است. این متد باید درخت تصمیم را بر اساس داده های آموزشی بسازد؛ منظور از ساخت درخت تصمیم، مشخص کردن این است که در هر سطح از ارتفاع درخت، کدام ویژگی ها وجود داشته باشند و همچنین چگونه هر ویژگی، درخت را به زیر درخت های کوچک تر تقسیم می کند. این تصمیم گیری را همانطور که در درس با آن آشنا شدید، با استفاده از معیار آنتروپی (Entropy) یا جینی (Gini) انجام دهید.
3. متد predict : این متد بعد از اینکه درخت ساخته شد، وظیفه این را دارد که درخت را پیمایش کند تا به برگ برسد و برچسب داده را خروجی دهد.

توجه کنید که شما کاملا آزاد هستید که به هر نحو و با استفاده از هر ساختمان داده ای این درخت را پیاده سازی کنید؛ اما یک ایده، می تواند استفاده از ساختار بازگشتی باشد؛ به عبارتی درخت تصمیم، می تواند چند مشخصه (attribute) به صورت درخت داشته باشد و آن ها را با استفاده از معیار آنتروپی بسازد؛ تاکید می شود که هیچ الزامی به استفاده از این ساختار نیست. قالب کلی این کلاس در نوتبوکی که در اختیارتان قرار گرفته، وجود دارد.

بخش ۱ – سرویس پیش‌بینی امکان اتمام به موقع پروژه‌های گروهی

هدف اصلی ما در این بخش این است که مدلی داشته باشیم که پیش‌بینی کند آیا یک پروژه گروهی در درس هوش مصنوعی به موقع به اتمام می‌رسد یا خیر. شما در این بخش مجاز به استفاده از مدل‌های آماده در کتابخانه‌های مربوطه نیستید.

دیتاست

دیتاست این مرحله از بخش 1 را می‌توانید از این لینک دریافت کنید.

- `group_size` : اندازه گروه (۲، ۳ یا ۴)
- `meetings_per_week` : تعداد جلسات هفتگی گروه (۰ تا ۵)
- `prog_skill` : مهارت برنامه‌نویسی گروه که به صورت دسته‌ای (Low, Medium, High) است و به صورت One-Hot یا عددی می توانید آن را کد کنید
- `Is_finished`: مقدار آن ۰ یا ۱ است (متغیر هدف)

آموزش و مقایسه درخت‌ها

1. شما تنها دادگان آموزش از این دیتاست را دارید و دادگان تست در اختیار تیم تست (دستیاران آموزشی) قرار دارد، شما می‌بایست مدل را روی این دادگان آموزش داده و روی داده validation فراپارامترهای آن را تنظیم کنید و تست نهایی توسط تیم تست در جلسه تحویل پروژه انجام خواهد شد. در خصوص نحوه تحویل پیش‌بینی‌ها روی دادگان تست به بخش توضیحات درباره دیتاست‌ها و نحوه تست مدل‌ها در انتهای توضیحات تمرین مراجعه کنید.
2. داده را به train/validation تقسیم کنید (مثلا ۷۰٪ / ۳۰٪).
3. از مدل درخت تصمیمی که پیش از شروع این بخش به صورت from scratch از صفر تا صد خودتان طراحی کردید استفاده کنید. آن را روی دادگان آموزشی آموزش داده و روی دادگان validation فراپارامتر های آن را تنظیم کرده و ارزیابی کنید. برای جلوگیری از overfit شدن، سعی کنید تا جای ممکن عمق درخت را (با کسب دقت بالا) کاهش دهید.
4. روی داده‌ی validation:
    a. Accuracy, Precision, Recall, F1 را حساب کنید.
    b. Confusion Matrix را رسم و تفسیر کنید.
5. با استفاده از ابزارهایی که دارید، ساختار درخت خود را استخراج کنید. (شکل نهایی درخت خود را رسم کنید). کدام ویژگی مهم‌ترین نقش را دارد؟

6. بار دیگر درخت تصمیم خود را این بار با محدود کردن عمق به عدد ۱ آموزش دهید و مراحل قبل را تکرار کنید. چه چیزی مشاهده می‌کنید.

پس از بررسی نتایج قسمت قبل، تیم دیتابیس UT Match یک ویژگی جدید را معرفی کرد. و پیشنهاد داد تا با سنجش عملکرد مدل در حضور یک فیچر جدید دقت مدل را افزایش دهیم. فیچری استخراج شده از قرار زیر است:

`avg_previous_grade` : میانگین نمرات قبلی اعضای گروه (بین ۱۰ تا ۲۰)

دیتاست جدید این مرحله را می‌توانید از این لینک دریافت کنید.

حال بار دیگر درخت تصمیم را یکبار با کمینه عمق (بیشتر از ۱) ممکن و یکبار با عمق ۱ روی این دیتاست در حضور این متغیر مستقل آموزش داده، متریک‌ها را محاسبه کرده و هر دو نتیجه به دست آمده را با هر دو نتیجه بدست آمده در بخش قبل مقایسه کنید.

- چگونه اضافه شدن یک فیچر جدید باعث شد یک مدل بسیار ساده (مثلا درخت عمق ۱) بتواند مسئله را تقریبا حل کند؟
- این تجربه در مورد اهمیت Feature Engineering و رابطه‌ی آن با پیچیدگی مدل چه چیزی به شما یاد می‌دهد؟
- اگر به جای اضافه کردن این فیچر، فقط عمق درخت را خیلی زیاد می‌کردید، چه اتفاقی ممکن بود بیفتد؟

بخش 2 – سرویس تشخیص ایمیل آموزشی مهم در مقابل ایمیل اسپم

با توجه به تعدد ایمیل‌های دریافتی، می‌خواهیم آن ها را به دو دسته تقسیم کنیم: (`Important` : ایمیل‌های استاد، آموزش، تمرین، امتحان و Spam : ایمیل‌های تبلیغاتی، تور شمال، تخفیف فست‌فود، کلاس کنکور بی‌ربط و ...)

1. دادگان مربوط به این بخش را می‌توانید از این لینک دریافت کنید.
2. شما تنها دادگان آموزش از این دیتاست را دارید و دادگان تست در اختیار تیم تست (دستیاران آموزشی) قرار دارد، شما می‌بایست مدل را روی این دادگان آموزش داده و روی داده validation فراپارامترهای آن را تنظیم کنید و تست نهایی توسط تیم تست در جلسه تحویل پروژه انجام خواهد شد. در خصوص نحوه تحویل پیش‌بینی‌ها روی دادگان تست به بخش توضیحات درباره دیتاست‌ها و نحوه تست مدل‌ها در انتهای توضیحات تمرین مراجعه کنید.
3. با پیگیری‌های انجام شده توسط تیم دستیاران آموزشی درس هوش مصنوعی، بالاخره تیم امنیت UT Match مجوز استفاده از مدل‌های آماده را صادر کرده و شما بعنوان تیم ML در تمامی مراحل این بخش امکان استفاده از مدل‌های موجود در Sklearn را دارید.
4. هر ایمیل را به یک بردار ویژگی تبدیل کنید. برای این کار به نکات زیر توجه کنید:
   a. ابتدا stopword ها را حذف می کنیم. همچنین متن‌ها را lowercase می کنیم و نشانه‌ها و کاراکترهای اضافی (مانند علائم نگارشی) را حذف می کنیم. این کارها باعث می‌شود داده‌ها تمیزتر شوند و مدل بتواند الگوهای واقعی را بهتر تشخیص دهد. همچنین از lemmatization برای یکنواخت سازی متن‌ها استفاده می کنیم (می توانید درباره آن مطالعه کنید). برای سادگی، این پیش پردازش ها در نوتبوک اولیه به شما داده شده است (برای این بخش لازم است کتابخانه nltk را نصب کنید).
   b. یک vocab از تمامی کلمات متمایزی که در ایمیل ها آمده اند بسازید.
   c. برای هر ایمیل، باید یک وکتور بسازید. این وکتور می تواند به صورت باینری باشد، به این صورت که نشان دهد کدام واژه‌ها در آن حضور دارند.مثلا `exam = 1` اگر کلمه exam در ایمیل هست وگرنه ۰. همچنین می تواند به جای تعیین حضور هر کلمه با ۰ یا ۱، تعداد تکرار هر کلمه را در وکتور بگذارید (مثلا اگر کلمه exam دو بار در یک ایمیل آمده باشد، مقدار آن برابر ۲ می شود). نتایج (توضیح داده شده در بخش ۴) مربوط به هر دوی این روش ها را گزارش کنید و علت تفاوت را ذکر کنید. برای ساختن وکتور می توانید از CountVectorizer از کتابخانه sklearn استفاده کنید.
   d. برچسب هدف به صورت $is\_important \in \{0,1\}$ تعریف شده است.
5. داده را به train/validation تقسیم کنید (مثلا ٪۷۰ / ٪۳۰).
6. دو مدل آموزش دهید (مجاز به استفاده از کتابخانه هستید):

a. Naive Bayes (با استفاده از BernoulliNB یا MultinomialNB از کتابخانه sklearn با توجه به اینکه وکتور ساخته شده باینری است یا شمارشی)
b. Decision Tree با عمق متوسط (مثلا `max_depth` برابر ۴ یا ۵)

7. روی داده‌ی validation:
   a. Accuracy, Precision, Recall, F1 را حساب کنید.
   b. Confusion Matrix را رسم و تفسیر کنید.
8. در داده‌ای که بر اساس کلمه‌ها (bag-of-words) ساخته‌اید، چرا فرض استقلال ویژگی‌ها برای Naive Bayes تا حدی معقول است؟
9. آیا در داده‌ی شما واژه‌هایی وجود دارند که کاملا مستقل نباشند (مثلا وجود exam و deadline)؟ این موضوع چه تاثیری روی Naive Bayes می‌گذارد؟
10. آیا درخت تصمیم توانسته قوانین واضحی مثل "اگر exam و homework و deadline هست آنگاه ایمیل مهم است" را یاد بگیرد؟ عملکردش در مقایسه با Naive Bayes چگونه است؟

بخش 3 – سرویس پیش‌بینی نوع بحران شب امتحان درس هوش مصنوعی

هدف ما در این بخش ساخت مدلی است که با توجه به ویژگی‌های جمع‌آوری‌شده از دانشجویان پیش‌بینی کند «نوع» بحران دانشجو در شب قبل از امتحان چه خواهد بود.

دیتاست

دیتاست این مرحله از بخش 3 را می‌توانید از این لینک دریافت کنید.

- `days_before_started_study` : چند روز قبل از امتحان شروع به خواندن کرده است؟ (بازه‌ی تقریبی ۰ تا ۱۰)
- `num_slides` : تعداد اسلایدهای درس (مثلا بین ۵۰ تا ۳۰۰)
- `num_assignments_done` : چند تمرین از تمرین‌های درس را واقعا حل کرده است؟
- `avg_sleep_last_week` : میانگین ساعت خواب در هر روز از هفته‌ی قبل (بین ۴ تا ۹ ساعت)
- `coffee_cups_per_day` : تعداد لیوان قهوه در روز (۰ تا مثلا ۷)
- `hours_studied_last_week` : تعداد ساعت مطالعه درس هوش مصنوعی در هفته‌ی قبل از امتحان (۰ تا ۴۰ ساعت)
- `avg_quiz_score` : میانگین نمرات کوییز ها (۰ تا ۲۰)
- `midterm_score` : نمره‌ی میان‌ترم دانشجو (۰ تا ۲۰)
- `assignments_done_ratio` : نسبت تمرین‌هایی که حل و تحویل داده شده‌اند (بین ۰ و ۱)
- `class_attendance_ratio` : نسبت حضور در جلسات کلاس (بین ۰ و ۱)
- `sleep_hours_before_exam` : میانگین ساعت خواب در سه شب قبل از امتحان (۳ تا ۹ ساعت)
- `stress_level` : سطح استرس (مثلا عدد صحیح بین ۱ و ۵)
- `ai_background` : پیش‌زمینه‌ی قبلی دانشجو نسبت به این حوزه
  - `'none'` (هیچ پیش‌زمینه‌ای ندارد)
  - `'basic'` (مثلا یک دوره مقدماتی گذرانده)
  - `'strong'` (قبلاً درس‌های مرتبط یا پروژه‌های جدی داشته)
- `mentality_type` : نوع تفکر
  - `'theory_heavy'` (تئوری‌محور)
  - `'project_heavy'` (پروژه‌محور)
  - `'memorization_heavy'` (حفظ‌محور)
- `exam_time` : ساعت امتحان
  - `'morning'`

- `'afternoon'`
- `'Evening'`

برچسب هدف را `crisis_type` تعریف می‌کنیم.

- `"No_issue"`: دانشجو اوضاعش اوکی است، نه بحران خاصی دارد، نه انکار.
- `"panic_mode"`: دانشجو به بحران شدید رسیده: شب‌بیداری، گریه، استرس شدید، ...
- `"Denial_mode"`: دانشجو در حالت انکار است: نه می‌خواند، نه نگران است! (مثلا می‌گوید: «فردا می‌بینیم چی می‌شه» و می‌رود FIFA بازی کند.)

آموزش و مقایسه درخت‌ها

1. شما تنها دادگان آموزش از این دیتاست را دارید و دادگان تست در اختیار تیم تست (دستیاران آموزشی) قرار دارد، شما می‌بایست مدل را روی این دادگان آموزش داده و روی داده validation فراپارامترهای آن را تنظیم کنید و تست نهایی توسط تیم تست در جلسه تحویل پروژه انجام خواهد شد. در خصوص نحوه تحویل پیش‌بینی‌ها روی دادگان تست به بخش توضیحات درباره دیتاست‌ها و نحوه تست مدل‌ها در انتهای توضیحات تمرین مراجعه کنید.
2. با پیگیری‌های انجام شده توسط تیم دستیاران آموزشی درس هوش مصنوعی، بالاخره تیم امنیت UT Match مجوز استفاده از مدل‌های آماده درختی را صادر کرده و شما بعنوان تیم ML در این بخش امکان استفاده از مدل‌های موجود در Sklearn را دارید.
3. قبل از هرگونه مدل‌سازی، روی داده‌ها مراحل Preprocessing زیر را انجام دهید:
   a. بررسی و مدیریت مقادیر گمشده (Imputation مناسب برای عددی ها و دسته‌ای ها).
   b. در صورت نیاز مقادیر ستون‌های مختلف را scale کنید. در صورتی که نیاز به این موضوع هست، علت نیازمندی را مشخص کرده و بهترین نوع scale کردن را انتخاب کنید، در غیر این صورت علت عدم نیاز به scale کردن را بیان کنید.
   c. مدیریت Outlier ها (مثلا Clip کردن یا حذف چند مورد خیلی غیرواقعی)
   d. ساخت فیچر جدید.
   e. تبدیل دادگان با مقادیر object به مقادیر عددی با در نظر گرفتن روش صحیح encode کردن آن‌ها و توضیح دلیل استفاده از هر روش.
   f. رسم ماتریس correlation، از این ماتریس چه نتیجه ای می‌گیرید؟
4. داده را به train/validation تقسیم کنید (مثلا ۷۰٪ / ۳۰٪).
5. یک مدل درخت تصمیم با عمق دلخواه و با استفاده از کتابخانه sklearn پیاده‌سازی کرده و مقدار برچسب هدف را برای تمامی دادگان پیش بینی کنید.
6. روی داده‌ی validation:

a. متریک‌های چندکلاسه روی validation را حساب کنید و درباره تفاوت هر یک با متریک‌های بخش قبل تحقیق کنید:
    i. Overall Accuracy
    ii. Macro-averaged Precision, Recall, F1
    iii. Confusion Matrix سه‌کلاسه
7. با استفاده از کتابخانه plot_tree ساختار درخت را بررسی کنید. کدام فیچر/فیچر ها نقش بیشتری در جداسازی داشته/داشته اند؟
8. توضیح دهید کدام نوع بحران بیشتر اشتباه تشخیص داده می‌شود (مثلا `panic_mode` با `denial_mode` قاطی می‌شود؟).

پس از برگزاری جلسات متعدد توسط تیم ارزیابی و با تخصیص حافظه بیشتر توانمندی مدل‌های فعلی و قدرت تعمیم آن‌ها مورد نقد قرار گرفت. و از شما خواسته شده است تا با استفاده از مدل‌های پیچیده‌تر و با حافظه بیشتر مصرفی، مدل درخت تصمیم فعلی را بهبود ببخشید. پیشنهاد دستیاران آموزشی درس هوش مصنوعی به شما استفاده از روش‌های Ensamble است تا شرایط فوق را برای شما فراهم آورد.

1. یک RandomForestClassifier یا BaggingClassifier با base_estimator=DecisionTreeClassifier بسازید.
2. مدل را روی همان داده‌ی preprocessed شده از مرحله قبل آموزش دهید و با استفاده از کتابخانه RandomizedSearchCV فراپارامتر های بهینه آن را پیدا کنید.
3. متریک‌های چندکلاسه را مانند بخش قبل گزارش کنید.
4. یک مدل Boosting مثل AdaBoostClassifier یا GradientBoostingClassifier یا XGBoost تعریف کنید.
5. مدل را train کنید و با استفاده از GridSearchCV فراپارامتر های بهینه را پیدا کنید.
6. همان متریک‌های بخش قبل را روی دادگان validation حساب کنید.
7. در تفسیر معنایی، چرا One-Hot در کل برای مدل‌های خطی/فاصله‌محور مناسب‌تر است؟

پس از پایان طراحی مدل‌ها در بخش قبل، یک مشکل مهم باعث نیاز مجدد به تغییر مدل‌های قبلی می‌شود، لیبل‌های هدف دارای هیچ‌گونه ترتیبی نبوده‌اند، اما در طراحی مدل‌های قبلی ترتیب 0 تا 2 برای آن‌ها در نظر گرفته شده است. در این بخش فرض کنید می‌خواهید برای هر کلاس (`no_issue, panic_mode,` `denial_mode`) یک مدل باینری جدا بسازید (One-vs-Rest).

1. یک ماتریس برچسب باینری بسازید با سه ستون:
    a. ستون ۱: `y_no_issue = 1` اگر crisis_type = no_issue، در غیر این صورت ۰
    b. ستون ۲: `y_panic = 1` اگر crisis_type = panic_mode

c. ستون ۳: `y_denial = 1` اگر crisis_type = denial_mode

2. برای هر ستون، یک DecisionTreeClassifier باینری جدا train کنید.
3. برای پیش‌بینی یک نمونه جدید، سه احتمال خروجی بگیرید و کلاس با بیشترین امتیاز را انتخاب کنید.
4. متریک‌های چندکلاسه را روی دادگان validation گزارش کرده و با نتایج مرحله قبل مقایسه کنید.
5. چرا LabelEncoding ممکن است به‌طور تصادفی یک رابطه‌ی ترتیبی کاذب بین مقادیر دسته‌ای ایجاد کند (هرچند در درخت‌ها معمولا آسیب جدی نمی‌زند).
6. توضیح دهید که LabelEncoding ای که برای مدل چندکلاسه استفاده شد و One-Hot Labels که برای one-vs-rest ساخته شد چه تفاوت مفهومی دارند.

بخش 4 – سرویس پیش‌بینی نمره کسب شده در امتحان درس هوش مصنوعی

پس از پیش بینی نوع بحران در شب امتحان میخواهیم میزان دقیق نمره دانشجویان را در این امتحان با پیش‌بینی کنیم. در این شرایط ویژگی `crisis_type` را به عنوان متغیر مستقل و مقدار ستون جدید final_exam_score را به عنوان متغیر وابسته لحاظ می‌کنیم.

دادگان این بخش با بخش قبلی یکسان است، اما حتما مجددا آن را از این لینک دانلود کرده و صرفا مراحل پیش‌پردازش قبلی را روی آن تکرار کنید.

1. روی دادگان پیش پردازش شده از بخش قبلی سه مدل مختلف را روی آن آموزش داده و مقایسه کنید:
    a. مدل `DecisionTreeRegressor` از `sklearn.tree`
    b. مدل `RandomForestRegressor` از `sklearn.ensemble`
    c. مدل `XGBRegressor` از `xgboost`
2. برای هر 3 مدل متریک های زیر را روی دادگان Validation محاسبه و گزارش کنید (دقت کنید که این متریک ها باید توسط خودتان پیاده سازی شوند و استفاده از کتابخانه به جز pandas و numpy مجاز نیست).
    a. MSE یا Mean Squared Error
    b. MAE یا Mean Absolute Error
    c. R^2 Score
3. هر یک از این متریک‌ها چه موضوعی را مشخص می‌کنند؟ چگونه؟ تفاوت هر یک از این متریک‌ها را با متریک‌های بخش‌های قبلی بررسی کنید و توضیح دهید.

حالا می‌خواهیم رفتار بایاس–واریانس را برای همین سه مدل به‌صورت تجربی ببینیم.

1. یک نقطه‌ی ورودی ثابت انتخاب کنید. (`x_star`)
2. یک حلقه (مثلا 100 بار) بنویسید که در هر تکرار:
    a. 30 درصد دادگان را به صورت اتفاقی انتخاب کند.
    b. هر سه مدل ذکر شده را روی `X_train, y_train` آموزش دهد.
    c. مقدار پیش‌بینی هر مدل روی `x_star` را محاسبه کند و در سه لیست جدا ذخیره کند. (به‌صورت مشابه می‌توانید برای هر مدل MSE تخمینی روی همان نقطه را هم حساب کنید.)
3. یک نمودار ساده طراحی کنید که:
    a. محور افقی: شماره مدل یا دسته؛
    b. محور عمودی: مقدار پیش‌بینی مدل روی `x_star`.
4. برای هر مدل، تمام مقادیر `pred_*` را به صورت نقاط رسم کنید:
    a. مثلا برای `DecisionTreeRegressor`، همه‌ی `pred_A` را در x=0 با scatter رسم کنید؛
    b. برای `RandomForestRegressor`، همه‌ی `pred_B` را در x=1؛
    c. برای `XGBRegressor`، همه‌ی `pred_C` را در x=2.

d. همچنین روی کل نمودار یک خط افقی در `true_y` رسم کنید که نقش bullseye را دارد.

5. با توجه به `true_y` و نقاط توزیع شده حول آن، میزان بایاس و واریانس را طبق نمودار توضیح دهید.
6. متغیر تصادفی‌ای که در این آزمایش با آن کار می‌کنید چیست؟ آیا از توزیع خاصی پیروی می‌کند؟

توضیحات درباره دیتاست ها و نحوه تست مدل ها

داده های تست، لیبل نهایی ندارند و شما باید نتایج تست مربوط به هر بخش را ذخیره کنید و در هنگام تحویل، دقت مدل های شما توسط دستیاران آموزشی سنجیده می شود. در واقع در خصوص دادگان تست شما می‌بایست یک ستون label به دیتاست اضافه کرده و مقادیر پیش‌بینی خود را در آن قرار داده و در کنار دفترچه پاسخ در فایل zip تحویلی قرار دهید.

نکات پایانی

- دقت کنید که کد شما باید به نحوی زده شده باشد که نتایج قابلیت بازتولید داشته باشند.
- توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید. حجم توضیحات گزارش شما هیچ گونه تاثیری در نمره نخواهد داشت و تحلیل شما بیشترین ارزش را دارد.
- سعی کنید از پاسخ‌های روشن در گزارش خود استفاده کنید و اگر پیش‌فرضی در حل سوال در ذهن خود دارید، حتما در گزارش خود آن را ذکر نمایید.
- فایل‌های خود را در قالب یک فایل فشرده با فرمت زیر در سامانه ایلرن بارگذاری کنید. به طور مثال AI_CA3_[stdNum].zip؟ AI_CA3_810102123.zip.
- محتویات پوشه باید شامل گزارش و کدهای شما باشد.

موفق باشید.