دانشگاه صنعتی شریف
دانشکده مهندسی برق

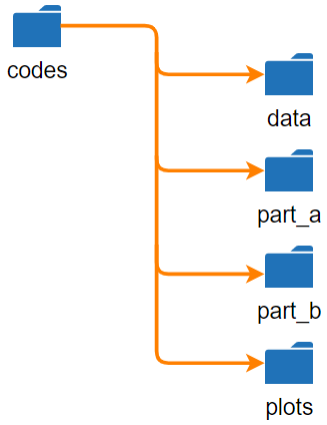
پروژه پایانی درس یادگیری آماری

نیمسال دوم تحصیلی ۰۲-۰۱

❖ نکات مهم

لطفا به هنگام انجام پروژه و آماده سازی نتایج به موارد زیر توجه نمایید.

- گزارش پروژه باید به صورت کامل و با تمام جزئیات نوشته شود. در گزارش خود، بخش‌ها و زیربخش‌های مربوط به هر بخش را به صورت جداگانه بیاورید.
- کدها خوانا و مرتب نوشته شده و تا حد امکان کامنت‌گذاری شوند.
- کدها بدون ایراد اجرا شده و خروجی‌های مطلوب را تولید نمایند. بدیهی است در صورتی که کد دارای ایراد بوده و اجرا نشود، نمره‌ی آن بخش به دانشجو تعلق نمی‌گیرد.
- در انجام پروژه مشورت مجاز است ولی بدیهی است در صورت مشاهدهٔ هرگونه تشابه غیر معمول بین کدها و یا نتایج، طرفین نمرهٔ صفر از پروژه دریافت خواهند کرد.
- لطفا هرگونه ابهام و یا سؤال را در سامانه CW مطرح نمایید تا سایر دانشجویانی که سوالی مشابه دارند نیز به پاسخ‌ها دسترسی داشته باشند.
- در پایان تمامی مستندات لازم را در یک فایل zip قرار دهید. نام فایل باید به صورت `YourName_YourStudentID` باشد. در داخل فایل zip باید یک پوشه به همین نام وجود داشته باشد.
- ساختار پوشه `starter_code` که به همراه صورت پروژه ارائه شده است، به صورت زیر می‌باشد.



شکل ۱- ساختار فایل‌ها

در مورد این ساختار نکات زیر را در نظر بگیرید:

- با توجه به اینکه نتایج کدها به صورت اتوماتیک تولید می‌شوند، از هر گونه تغییر در ساختار پوشه‌ها و فایل‌ها خودداری نمایید.
- اطلاعات خود را در تابع `student_information()` در فایل `part_a/__init__.py` وارد نمایید.
- محتویات فایل `test.py` را تغییر ندهید! با اجرای این فایل، فایل‌های مربوط به هر بخش از فاز اول به صورت خودکار اجرا شده و نتایج تولید شده در یک فایل `.pkl` ذخیره خواهد شد. از آنجاییکه از این فایل برای بررسی

کدهایتان استفاده خواهد شد، قبل از ارسال نهایی، با اجرای این فایل از دریافت خروجی صحیح اطمینان حاصل نمایید.

- دادهای مورد نیاز در پوشه **data** قرار داده شده‌اند. در مورد جزییات داده‌ها در ادامه توضیح داده خواهد شد.
- در پوشه **part_a**، کدهای اولیه و راهنما برای هر بخش داده شده است. کدهای خود را تنها در بخش‌های مشخص شده وارد نمایید و با توجه به خروجی‌هایی که از توابع موردانتظار است، متغیرهای لازم را تعریف و یا مقداردهی کنید. برای مثال تابع زیر مقدار دقت (acc) را برمی‌گرداند. برای همین لازم است تا مقدار دقت را پس از محاسبه (به طور مثال بوسیله کدهای موجود در فایل utils.py) در متغیر acc ذخیره کنید تا توسط تابع قابل بازگرداندن باشد.

```
def knn_impute_by_item(matrix, valid_data, k):
    """ Fill in the missing values using k-Nearest Neighbors based on
    question similarity. Return the accuracy on valid_data.

    :param matrix: 2D sparse matrix
    :param valid_data: A dictionary {user_id: list, question_id: list,
    is_correct: list}
    :param k: int
    :return: float
    """
    ###########################################################################
    # TODO:                                                                   #
    # Implement the function as described in the docstring.                   #
    ###########################################################################
    acc = None
    ###########################################################################
    #                       END OF YOUR CODE                                  #
    ###########################################################################
    return acc
```

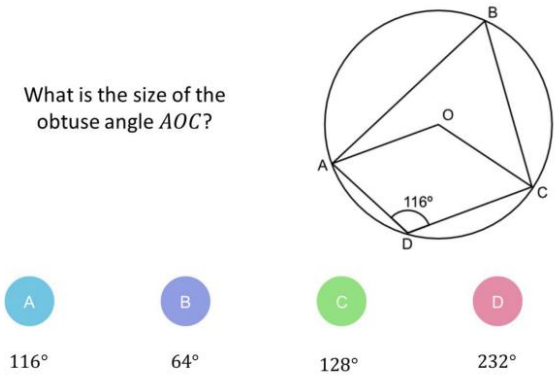
شکل ۲- نمونه از ساختار توابع داده شده

- در انتهای هر فایل یک دیکشنری **results = {}** وجود دارد. لازم است تا مواردی که در این دیکشنری خواسته شده است را فراهم نمایید.
- در مورد رسم پلات‌ها، در کد نهایی که تحویل می‌دهید، همه دستورهای نمایش نمودارها مانند **plt.show()** را با دستورهای ذخیره تصویر مانند **plt.save()** جایگزین نمایید و پلات های هر بخش را در پوشه مربوط به آن در مسیر **/plots** ذخیره نمایید.
- در تمامی فایل‌ها، برای آنکه نتایج بدست آمده قابلیت باز تولید داشته باشند، حتما seed ها با مقدار **42** ست نمایید.
- جهت اجرای صحیح کدها، یک فایل requirements.txt در پوشه اصلی قرار داده شده است. در ابتدا با استفاده از دستور زیر نسخه‌های ذکر شده در این فایل را نصب نمایید.

```
pip install -r requirements.txt
```

تعریف مسئله

سرویس‌های آموزش آنلاین مانند Khan Academy[1] و Coursera[2] امکان دسترسی به آموزش با کیفیت برای یک جمعیت گسترده را فراهم کرده‌اند. در این پلتفرم‌ها، دانش‌آموزان با تماشای سخنرانی، مطالعه مواد درسی و گفتگو با مربیان در یک انجمن، می‌توانند مفاهیم جدید را یاد بگیرند. با این حال، یک معضل در این پلتفرم‌های آنلاین این است که سنجش درک دانش‌آموزان از مواد درسی سخت است. برای مقابله با این مشکل، بسیاری از پلتفرم‌های آموزش آنلاین شامل یک بخش ارزیابی هستند تا اطمینان حاصل شود که دانش‌آموزان موضوعات اصلی را به خوبی فرا گرفته‌اند. بخش ارزیابی معمولاً شامل سوالات تشخیصی است، که هر کدام یک سوال چندگزینه با یک پاسخ صحیح است. سوال تشخیصی به گونه‌ای طراحی شده است که هر یک از پاسخ‌های نادرست، یک درک اشتباه از موضوع را برجسته می‌کند. نمونه‌ای از یک سوال تشخیصی در شکل ۳ نشان داده شده است. زمانی که دانش‌آموزان به درستی به سوال تشخیصی پاسخ نمی‌دهند، این نشان می‌دهد که ماهیت اشتباهشان چیست و با درک این اشتباه‌ها، پلتفرم می‌تواند راهنمایی‌های لازم را به دانش‌آموزان ارائه دهد و به حل آنها کمک کند.



شکل ۳- نمونه‌ای از یک سوال تشخیصی

در این پروژه، الگوریتم‌های یادگیری ماشین برای پیش‌بینی آنکه یک دانش‌آموز به سوال تشخیصی خاصی پاسخ درست خواهد داد یا خیر، استفاده خواهند شد. این پیش‌بینی بر اساس پاسخ‌های قبلی آن دانش‌آموز به سوالات دیگر و پاسخ دیگر دانش‌آموزان خواهد بود. پیش‌بینی درستی پاسخ هر دانش‌آموز به سوالات تشخیصی‌ای که هنوز ندیده است، به تخمین سطح توانایی دانش‌آموز برای ساخت یک پلتفرم آموزشی شخصی‌سازی شده کمک می‌کند. علاوه بر این، این پیش‌بینی‌ها پایه‌ای برای بسیاری از وظایف سفارشی پیشرفته را فراهم می‌کنند. به عنوان مثال، با استفاده از پیش‌بینی صحت، پلتفرم‌های آنلاین می‌توانند به طور خودکار مجموعه‌ای از سوالات تشخیصی مناسب با سطح دشواری مناسبی که با پس‌زمینه و وضعیت یادگیری دانش‌آموز سازگار است، به او پیشنهاد دهند.

شما در ابتدا به پیاده‌سازی الگوریتم‌های یادگیری ماشینی که در این درس آموخته‌اید، خواهید پرداخت. سپس عملکرد الگوریتم‌های مختلف را مقایسه و مزایا و معایب آن‌ها را تحلیل خواهید کرد. در مرحله بعدی، الگوریتم‌های موجود را باید به گونه‌ای تغییر دهید که

[1] Khanacademy.org
[2] Coursera.org

پیش‌بینی پاسخ دانش‌آموزان را با دقت بیشتری انجام دهد. در نهایت، تغییرات اعمالی خود را با بررسی نتایج دقت بر روی داده‌های بررسی و گزارش خواهید نمود.

برای بررسی عملکرد سیستم یادگیری لازم است تا از معیار دقت پیش‌بینی که در رابطه زیر آورده شده است، استفاده نمایید. اگرچه علاوه بر معیار ذکر شده، می‌توانید از معیارهای دیگری که فکر می‌کنید دیدگاه دیگری به تحلیل سیستم یادگیری اضافه می‌کنند، استفاده نمایید.

$$\text{Prediction Accuracy} = \frac{\text{the number of correct predictions}}{\text{the number of total predictions}}$$

توضیح داده‌ها

ما پاسخ‌های ۵۴۲ دانش‌آموز به ۱۷۷۴ سوال تشخیصی از مجموعه داده‌های ارائه شده توسط Eedi[3] را نمونه‌برداری کرده‌ایم. Eedi یک پلتفرم آموزشی آنلاین است که در حال حاضر در بسیاری از مدارس استفاده می‌شود. این پلتفرم سوالات تشخیصی ریاضی را به دانش‌آموزان ابتدایی شتا دبیرستان (بین ۷ و ۱۸ سال) ارائه می‌دهد. مجموعه داده‌های انتخاب شده در پوشه `/data` قرار داده شده است.

داده‌های اصلی

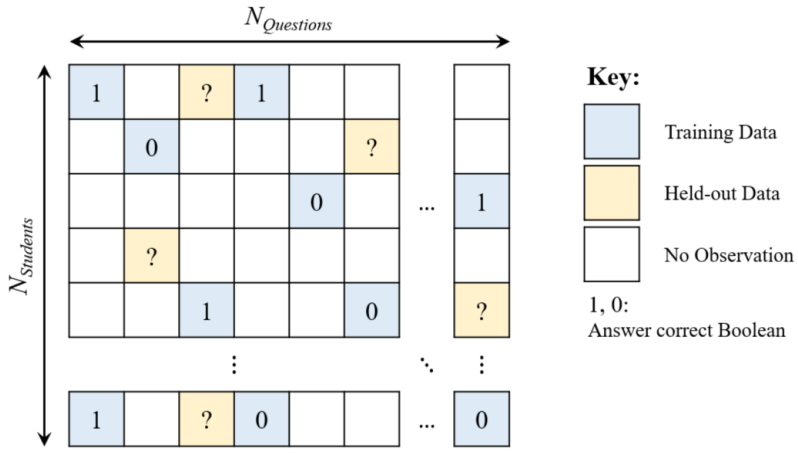
شما برای آموزش مدل‌های خود از داده `train_data.csv` استفاده خواهید نمود. همچنین لازم است تا برای انتخاب مدل و ارزیابی آن‌ها از داده `valid_data.csv` استفاده نمایید. این دادگان دارای سه ستون اصلی می‌باشند:

- `question_id`: شماره سوال پاسخ داده شده (شروع از صفر)
- `user_id`: شماره مربوط به دانش‌آموزی که سوال را پاسخ داده است (شروع از صفر)
- `is_correct`: یک شاخص باینری که مشخص می‌کند آیا پاسخ دانش‌آموز به سوال درست بوده است یا خیر (صفر: پاسخ اشتباه، یک: پاسخ درست)

همچنین در پوشه داده‌ها، یک ماتریس تنک[4]، `train_sparse.npz` قرار داده شده است که هر سطر آن متناظر با یک `user_id` و هر ستون آن متناظر با یک `question_id` می‌باشد. شکل ۴ نمونه‌ای از ماتریس تنک را نشان می‌دهد. برچسب‌های هر خانه متناظر با آدرس (`user_id,question_id`) به صورت زیر تعریف می‌شوند:

- `1`: پاسخ دانش‌آموز user_id به سوال question_id صحیح بوده است.
- `0`: پاسخ دانش‌آموز user_id به سوال question_id اشتباه بوده است.
- `NaN`: دانش‌آموز user_id به سوال question_id پاسخی نداده است و یا به عنوان داده ارزیابی، از دادگان آموزشی کنار گذاشته شده است.

[3] https://eedi.com/
[4] Sparse matrix



شکل ۴- کلیت ماتریس تنک داده شده

Question Metadata

فایل `question_meta.csv` شامل اطلاعات مربوط به سوالات می‌باشد. این فایل دارای ستون‌های اصلی زیر است:

- `question_id`: شماره سوال پاسخ داده شده (شروع از صفر)
- `subject_id`: شماره مباحثی که سوال، آن‌ها را کاور می‌کند. این مباحث از بین موضوعات موجود در علم ریاضیات است که در اطلاعات آن در فایل `subject_meta.csv` آمده است.

Student Metadata

اطلاعات مربوط به دانش‌آموزان در فایل `student_meta.csv` آمده است. این فایل دارای ستون‌های اصلی زیر است:

- `user_id`: شماره مربوط به دانش‌آموزی که سوال را پاسخ داده است (شروع از صفر)
- `gender`: جنسیت دانش‌آموز (1: خانم، 2: آقا، 0: نامشخص)
- `date_of_birth`: تاریخ تولد
- `premium_pupil`: اطلاعات مربوط به بورسیه تحصیلی دانش‌آموز

فاز اول (۷۰ امتیاز)

در این بخش به پیاده‌سازی چندین الگوریتم یادگیری ماشین بر روی داده‌ها برای پیش‌بینی اینکه آیا دانش‌آموز به سوال تشخیصی پاسخ درستی خواهد داد یا خیر، می‌پردازیم. در این قسمت تنها از داده‌های اصلی (`train_data.csv`، `valid_data.csv`) استفاده خواهید نمود. همچنین شما می‌توانید از توابع کمکی که در فایل `utils.py` آورده شده است، برای بارگذاری داده‌ها و ارزیابی مدل خود استفاده نمایید. در هر بخش در ابتدا فایل پایتون مربوطه معرفی شده است که لازم است تا کدهای مربوط به پیاده‌سازی آن الگوریتم در آن نوشته شوند.

الگوریتم k-Nearest Neighbor (۱۰ امتیاز)

فایل پایتون مربوطه: `codes/knn.py`

دراین قسمت می‌خواهیم با استفاده از الگوریتم k-Nearest Neighbor، سوالاتی که دانش‌آموزان به آن پاسخ نداده‌اند (در ماتریس تنک، مقدار NaN دارند) را پیش‌بینی کنیم. برای این منظور می‌خواهیم از پاسخ‌های دیگر دانش‌آموزان برای پیش‌بینی اینکه آیا دانش‌آموز خاص می‌تواند به برخی سوالات تشخیصی پاسخ درست بدهد، استفاده نماییم. روند کار به این صورت است که با دادن یک دانش‌آموز، الگوریتم kNN، نزدیک‌ترین دانش‌آموز که به سوالات دیگر پاسخ‌های مشابهی داده باشد، را پیدا می‌کند و با توجه به درستی پاسخ نزدیک‌ترین دانش‌آموز، صحت پاسخ دانش‌آموز موردنظر را پیش‌بینی می‌کند.

فرضیه اصلی‌ای که در اینجا در نظر گرفتیم آن است که اگر دانش‌آموز A در سوالات دیگری پاسخ‌های درست و نادرستی شبیه به دانش‌آموز B داشته باشد، درست بودن یا غلط بودن پاسخ دانش آموز A در سوالات تشخیصی خاص با دانش آموز B همخوانی دارد.

در ادامه بخش‌های زیر را انجام دهید.

الف/a) تابع knn_impute_by_user() را تکمیل نمایید. این تابع از یک فیلتر پالایش گروهی[5] برای پیش‌بینی درستی پاسخ برای یک دانش‌آموز بر اساس پاسخ‌های دیگر دانش‌آموزان، استفاده می‌کند[6]. برای پیاده سازی می‌توانید از تابع KNNImputer[7] استفاده نمایید. ورودی‌های تابع به همراه توضیحات هر کدام در docstring تابع آورده شده است.

ب/b) تابع main() را کامل نمایید و تابع knn_impute_by_user() را به ازای $k \in \{1,6,11,16,21,26\}$ اجرا نموده و دقت خروجی بر روی داده‌های validation را برحسب مقدار k گزارش و پلات نمایید.

پ/c) بهترین مقدار k^* که بیشترین دقت را بر روی داده‌های validation دارد را انتخاب کرده و به همراه دقت آن گزارش نمایید.

ت/d) این بار تابع knn_impute_by_item() را پیاده‌سازی نمایید. این تابع به جای اعمال فیلتر پالایش گروهی بر اساس دانش‌آموزان، یک فیلتر پالایش گروهی را بر اساس سوالات استفاده می‌کند. در واقع بر اساس یک سوال، الگوریتم kNN نزدیک‌ترین سوالی که مشابه با سوال موردنظر پاسخ داده شده است را پیدا کرده و درستی پاسخ به آن سوال را بر اساس درستی پاسخ به نزدیکترین سوال همسایه آن پیش‌بینی می‌کند.[8] همانند توضیحات داده شده در ابتدای روش، فرضیه اصلی‌ای که در اینجا وجود دارد را بیان نموده و مراحل (ب) و (پ) را تکرار نمایید.

ث/e) نتایج بدست آمده از دو مدل پیاده‌سازی شده بر روی داده‌های validation را با یکدیگر مقایسه نمایید. کدام مدل بهتر عمل می‌کند؟

ج/f) حداقل دو مورد از محدودیت‌های بالقوه‌ای که روش kNN برای انجام این تسک دارد را بیان نمایید.

روش Item Response Theory (IRT) (۲۰ امتیاز)

فایل پایتون مربوطه: `codes/item_response.py`

[5] Collaborative filtering - Collaborative filtering - Wikipedia
[6] User-based collaborative filtering
[7] sklearn.impute.KNNImputer — scikit-learn 1.2.2 documentation
[8] Question-based collaborative filtering

روش IRT به هر دانش‌آموز یک درجه توانایی[9] و به هر سوال یک درجه سختی[10] برای فرمول‌بندی یک توزیع احتمال نسبت می‌دهد. در مدل IRT تک متغیره، β_j نمایانگر درجه سختی سوال jام و θ_i نمایانگر درجه توانایی دانش‌آموز iام می‌باشند. سپس احتمال آنکه سوال jام توسط دانش‌آموز iام به درستی پاسخ داده شده باشد، به صورت زیر فرمول‌بندی می‌شود:

$$p(c_{ij} = 1 | \theta_i, \beta_j) = \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)}$$

الف/a) رابطه log-likelihood ($\log p(\boldsymbol{C}|\boldsymbol{\theta}, \boldsymbol{\beta})$) را برای همه دانش‌آموزان و همه سوالات بدست آورید. در اینجا $\boldsymbol{C}$ همان ماتریس تنک می‌باشد. همچنین رابطه مشتق log-likelihood نسبت به پارامترهای θ_i و β_j را محاسبه و گزارش نمایید.

ب/b) توابعی که در فایل پایتون مربوطه قرار دارند را کامل نمایید تا با استفاده از روش گرادیان کاهشی[11] بر روی متغیرهای $\boldsymbol{\theta}$ و $\boldsymbol{\beta}$ منجر به ماکزیمم شدن log-likelihood شوند. هایپرپارامترهایی که استفاده کردید را گزارش نموده و با استفاده از هایپرپارامترهای انتخابی، نمودارهای آموزش شامل آموزش و ارزیابی log-likelihood بر حسب تکرارها[12] را رسم نمایید.

پ/c) با استفاده از کدهای پیاده‌سازی شده، مقدار بهترین دقت و log-likelihood را بر روی داده‌های validation گزارش نمایید.

ت/d) پنج سوال را به دلخواه انتخاب نمایید (j_1, j_2, j_3, j_4, j_5). با استفاده از مقادیر آموزش دیده شده متغیرهای $\boldsymbol{\theta}$ و $\boldsymbol{\beta}$، پنج نمودار را در یک پلات رسم کنید به گونه‌ای که هر نمودار نشان‌دهنده احتمال درست بودن پاسخ $p(c_{ij})$ بر حسب تابعی از $\boldsymbol{\theta}$ برای آن سوال مشخص می‌باشد. در مورد شکل نمودارها توضیح داده و به صورت مختصر آن‌ها را تحلیل نمایید.

روش Matrix Factorization (۲۰ امتیاز)

فایل پایتون مربوطه: `part_a/matrix_factorization.py`

الف/a) در این تمرین می‌خواهیم روش matrix factorization را پیاده سازی می‌کنیم. با استفاده از تابع `svd_reconstruct()`, ماتریس تنک موردنظر را با استفاده از روش تجزیه مقادیر منفرد[13] تجزیه کنید. اینکار را حداقل با ۵ مقدار مختلف k (تعداد ویژگی‌ها) انجام دهید و بهترین مقدار k را با استفاده از دادگان اعتبارسنجی انتخاب کنید. دقت نهایی روی دادگان اعتبارسنجی را گزارش نمایید.

ب/b) یک از محدودیت SVD را در این تمرین بیان کنید. (مقادیر خالی ماتریس تنک را چگونه پر می‌کنید؟)

پ/c) توابع `als()` و `update_u_z()` روش alternating least square (ALS) را پیاده‌سازی می‌کنند. این روش را به صورت مختصر توضیح داده و پیاده‌سازی توابع را کامل کنید. تابع هدف در روش ALS به‌صورت زیر می‌باشد:

$$\min_{\mathbf{U},\mathbf{Z}} \frac{1}{2} \sum_{(n,m) \in O} \left(C_{nm} - \mathbf{u}_n^\top \mathbf{z}_m \right)^2$$

[9] Ability value
[10] Difficulty value
[11] Gradient descent
[12] Iterations
[13] Singular Value decomposition

که در آن $\boldsymbol{C}$ ماتریس تنک و O مجموعه‌ای از اندیس المان‌های ماتریس $\boldsymbol{C}$ است.

ت/d) با استفاده از ALS ماتریس‌های **U** و **Z** را آموزش دهید. نرخ آموزش[۱۴] و تعداد تکرار[۱۵] را تنظیم کرده و هایپرپارامترهای انتخابی را گزارش نمایید. حداقل ۵ مقدار مختلف را برای k امتحان کرده و بهترین مقدار k که کمترین خطا روی داده‌های اعتبارسنجی دارد را انتخاب و به همراه ماتریس‌های **U** و **Z** و ماتریس نهایی بازتولید شده گزارش کنید.

ث/e) با مقدار k انتخاب شده نمودار مجموع مربعات خطای در آموزش و اعتبارسنجی را برحسب تعداد تکرار(iteration) رسم کنید. همچنین دقت نهایی بر روی دادگان اعتبارسنجی را گزارش نمایید.

ج/f) برای ALS مدل را مانند مسائل رگرسیون آموزش دادیم و تابع هزینه مجموع مربعات خطا بود. اگر بخواهیم از این مدل در یک مساله‌ی طبقه‌بندی باینری استفاده کنیم, چه اصلاحی باید روی تابع هزینه صورت بگیرد؟ اصلاحاتی را که روی تابع هزینه اعمال می‌کنید توضیح دهید.

روش Ensemble (۲۰ امتیاز)

فایل پایتون مربوطه: `part_a/ensemble.py`

در این تمرین, bagging ensemble را پیاده‌سازی خواهیم کرد که باعث بهبود پایداری و دقت مدل‌های پایه خواهد شد. سه مدل پایه را انتخاب کنید و با بوت استراپینگ[۱۶] دادگان آموزش, مدل‌ها را آموزش دهید. این مدل‌ها می‌توانند یکسان و یا متفاوت باشند. پیاده‌سازی خود را در فایل پایتون مربوطه انجام دهید. ماتریس باز تولید شده نهایی را به کمک خروجی هر سه مدل بدست بیاورید. برای اینکار می‌توانید از میانگین‌گیری، vo دقت نهایی روی دادگان اعتبارسنجی را گزارش کنید. پیاده‌سازی که انجام داده‌اید را در گزارش توضیح دهید. آیا با این کار به نتایج بهتری دست یافته‌اید؟ نتایج را تحلیل نمایید.

فاز دوم (۳۰ امتیاز)

در قسمت دوم می‌خواهیم یکی از الگوریتم‌هایی که در فاز اول پیاده‌سازی کرده‌ایم را بهبود دهیم تا پیش‌بینی بهتری از جواب دانش‌آموز به سوالات تشخیصی داشته باشیم. نتایج بدست آمده از مدل‌های پیاده‌سازی شده در فاز اول را در نظر بگیرید. عواملی که باعث محدودیت عملکرد آن‌ها شده است را شرح دهید (به عنوان مثال سختی بهینه سازی؟ بیش برازش[۱۷]؟ کم برازش[۱۸]؟) و یک روش برای اصلاح آن پیشنهاد دهید. عملکرد الگوریتم اصلاح شده خود را آزمایش کنید.

این قسمت براساس عملکرد شما در تحلیل نتایج نمره‌دهی می‌شود. در صورت نیاز می‌توانید از دادگان `question_meta.csv` و `student_meta.csv` برای بهبود دقت مدل استفاده کنید. همچنین در صورت نیاز می‌توانید از کدها و ایده‌های موجود در منابع دیگر استفاده نمایید. توجه داشته باشید که لازم است به هر منبعی که استفاده می‌کنید، در گزارش خود ارجاع دهید.

---

[14] Learning Rate
[15] Iteration
[16] Bootstrapping
[17] Overfitting
[18] Underfitting

امتیازی: دقت‌های بدست آمده در این فاز به صورت رقابتی بین دانشجویان مقایسه خواهد شد و به بالاترین دقت نمره کامل و به سایر دقت‌ها به نسبت جایگاه، درصد کمتری از نمره کامل داده خواهد شد.

طول گزارش شما در قسمت دوم نهایت پنج صفحه باید باشد. دستورعمل قسمت دوم به شرح زیر است:

۱- توضیحات: روش خود برای توسعه و بهبود الگوریتم انتخابی را به دقت توضیح دهید. همچنین درصورت امکان معادلات و بلوک دیاگرام الگوریتم را در گزارش بیاورید. الگوریتم پیشنهادی شما قرار است عملکرد کدام بخش را بهبود دهد؟ توضیح دهید. به‌عنوان مثال, این الگوریتم عملکرد بهینه سازی را بهبود می‌دهد و یا از بیش برازش جلوگیری می‌کند و یا؟

۲- شکل‌ها و نمودارها: تمامی نمودارها و شکل‌هایی که به درک بیشتر گزارش شما کمک می‌کند را در گزارش خود بیاورید.

۳- مقایسه شامل:

a. مقایسه دقت‌های به‌دست آمده توسط مدل شما با دقت‌های مدل‌های پایه در فاز اول. از جدول و نمودارهای مناسب برای مقایسه استفاده نمایید.
b. آزمایشی را طراحی نمایید که نشان دهد فرضیه شما در توضیح اینکه چرا مدل توسعه داده شده، فلان بخش را بهبود می‌دهد، درست می‌باشد.

۴- محدودیت‌های مدل پیشنهادی:

a. برخی از شرایط خاص و حالاتی را شرح دهید که انتظار دارید رویکرد شما در آن‌ها عملکرد خوبی ندارد یا همه مدل‌ها بسا شکست مواجه می‌شوند.
b. سعی کنید حدس بزنید یا توضیح دهید چرا این محدودیت‌ها به این صورت هستند.
c. چند نمونه از راه‌هایی برای رفع این محدودیت‌ها و توسعه مدل ارائه کنید.