سلام بر دانشجویان عزیز، چند نکته مهم:

۱. حجم گزارش به هیچ عنوان معیار نمره دهی نیست، در حد نیاز توضیح دهید.

۲. نکته‌ی مهم در گزارش نویسی روشن بودن پاسخ‌ها می‌باشد، اگر فرضی برای حل سوال استفاده می‌کنید حتما آن را ذکر کنید، اگر جواب نهایی عددی است به صورت واضح آن را بیان کنید.

۳. کد های نوشته شده برای هر سوال شبیه سازی را در فایل ipynb متناظر آن سوال بنویسید.

٤. کدهای ارسال شده بدون گزارش و یا کامنت گذاری دقیق در کد فاقد نمره می‌باشند.

٥. برای سوالات شبیه سازی، فقط از دیتاست داده شده استفاده کنید.

٦. نمره تمرین از ۱۰۰ نمره می‌باشد

۷. هرگونه شباهت در گزارش و کد مربوط به شبیه سازی، به منزله تقلب می‌باشد و کل تمرین برای طرفین ۱۰۰-خواهد شد.

۸. در صورتی که تشخیص داده شود از چت بات ها به صورت مستقیم برای پاسخ سوال های تئوری و شبیه سازی استفاده شده است، نمره ۱۰۰-در نظر گرفته خواهد شد.

۹. فایل نهایی خود را در یک فایل زیپ شامل PDF گزارش و فایل کدها آپلود کنید. نام فایل زیپ ارسالی الگوی ML_HW#_StudentNumber داشته باشد.

در صورت داشتن سوال، از طریق گروه درس یا ایمیل‌های زیر با تدریسیار مربوطه سوال‌های خود را مطرح کنید.

سوال ۱ و ۲ و ۵: javadkavian8@gmail.com
سوال ۳ و ۴ و ۶ : ta.mobin.roohi@gmail.com

سوال ۱: (۱۵ نمره)

فرض کنید در فرآیند انتخاب ویژگی(feature) در یک پروژه یادگیری ماشین، ویژگی های زیر به ترتیب از چپ به راست انتخاب شده اند:

$$X_2, X_8, X_7, X_1, X_9, X_5, X_3, X_6, X_4, X_{10}$$

همچنین فرض کنید معیار انتخاب(criterion)، مجموعه ویژگی هایی باشند که بیشترین Information Gain را داشته باشند. مقدار Information Gain روی ویژگی های $X_i, X_j, .., X_k$ را به صورت $F(X_i, X_j, \ldots, X_k)$ نمایش می دهیم.

الف) اگر الگوریتم انتخاب ویژگی(Sequential Forward Selection(SFS باشد، ثابت کنید:

$$F(X_2, X_8, X_5, X_{10}) \geq F(X_1, X_2)$$

ب) آیا این الگوریتم، لزوما بهترین مجموعه از ویژگی ها را با توجه به criterion تعریف شده به دست می آورد؟ برای پاسخ خود دلیل بیاورید.

سوال ۲: (۱۵ نمره)

همانطور که میدانید، یکی از روش های انتخاب ویژگی، استفاده از روش های مبتنی بر الگوریتم ژنتیک است.

الف) این روش را مختصرا توضیح دهید.

ب) اگر در این الگوریتم، fitness function را به صورت زیر قرار دهیم، توضیح دهید چه مشکلی ممکن است پیش بیاید؟

$$Fitness = accuracy$$

ج) برای حل مشکل fitness function فوق، آن را چگونه تغییر می دهید؟

سوال ۳: (۱۵ نمره)

در این سؤال، دو مرحله از الگوریتم **Adaboost**را روی مجموعه‌داده‌ای دوبعدی به صورت دستی اجرا می‌کنید. یادگیرنده‌های ضعیف مورد استفاده، **Decision Stump**هایی با مرزهای عمود بر محورهای مختصات هستند؛ یعنی هر decision stump فقط بر اساس یک ویژگی (x_1 یا x_2) تصمیم‌گیری می‌کند.

داده‌های مورد نظر بدین شکل هستند:

Sample	x_1	x_2	**Label** y
A	12	10	+
B	2	4	-
C	10	5	+
D	8	8	-
E	6	5	+

برای شفافیت، decision stump را تعریف میکنیم:

$$h(x) = \begin{cases} +1, & x_j \leq \theta, \\ -1, & x_j > \theta, \end{cases}$$

که در اینجا θ همان ویژگی تصمیم‌گیری است.

برای هر ویژگی، تنها میانگین بین مقادیر مرتب‌شده نمونه‌ها بررسی می‌شود:

- یعنی برای $x_1 \in \{2,6,8,10,12\}$، تنها مرزهای $\theta \in \{4,7,9,11\}$ بررسی شوند.
- یعنی برای $x_1 \in \{4,5,8,10\}$، تنها مرزهای $\theta \in \{4.5,5.5,6.5,9\}$ بررسی شوند.

یک نکته مهم:

برای پیدا کردن درختچه تصمیم بهینه، نیازی نیست تمام حالت‌های ممکن را بررسی کنید.

کافی است با نگاه کردن به نمودار نقاط، آن‌هایی را که به‌صورت شهودی احتمال دارد مرز تصمیم خوبی باشند انتخاب کرده و برای آن‌ها خطا را محاسبه کنید.

این کار باعث صرفه‌جویی در وقت و جلوگیری از محاسبات طولانی و تکراری می‌شود.

توجه کنید! در این مسئله ε_t همان weighted error اسلایدهای درس هست که برای گذر t ام Adaboost تعریف شده است. همچنین $D_t(i)$ همان α_i برای گذر t ام است.

(الف) گذر اول: فرض کنید که تمامی داده‌ها در ابتدا وزن‌های برابری دارند و $D_1(i)$ یک توزیع یکنواخت است. حال اولین گذر (iteration) الگوریتم Adaboost را با استفاده از decision stump تعریف شده انجام دهید. برای این کار، decision stump بهینه را بدست آورید و مرز بهینه را گزارش کنید. سپس، مقادیر خطای وزن‌دار ε_1 ، $\widehat{w_1}$ و وزن‌های آپدیت شده داده‌ها یعنی $D_2(i)$ را محاسبه کرده و اعلام کنید.

(ب) گذر دوم: حال گذر دوم را انجام بدهید. برای این کار، با استفاده از وزن های $D_2(i)$decision stump و مرز بهینه را محاسبه و گزارش کنید. سپس، مقادیر ε_2 و $\widehat{w_2}$ را محاسبه کنید.

(ج) طبقه‌بند نهایی: طبقه‌بند نهایی Adaboost را با استفاده از $\widehat{w_1}$، $\widehat{w_2}$ و دو decision stump بهینه هر گذر بدست آورید و سپس کلاس داده‌های (۱۱,۹) و (۵,۵) را با استفاده از آن پیش‌بینی کنید.

توضیح دهید طبقه‌بند Adaboost بدست آمده کدام داده را با اطمینان بیشتری طبقه‌بندی می‌کند؟ دلیل خود را بیاورید.

نکات بیشتر:

- در طول محاسبه اگر به بیش از یک decision stump بهینه دست پیدا کردید، یک مورد را با ذکر آن به تصمیم خود انتخاب کنید.
- مقادیر وزن‌های آپدیت شده، D_t، را قبل گزارش کردن نرمال‌سازی کنید که جمع مقادیر آن ۱ باشد.

سوال ۴: (۱۰ نمره)

(الف) نشان دهید که کران بالای زیر برای خطای آموزشی طبقه‌بند نهایی F حاصل از T گذر اجرای Adaboost وجود دارد.

$$\mathrm{err}_{\mathrm{train}}(H) \leq \prod_{t=1}^{T} Z_t$$

به طوری که:

$$F(x) = \sum_{t=1}^{T} \hat{w}_t f_t(x)$$

$$H(x) = \mathrm{sign}\big(F(x)\big) \in \{-1, +1\}$$

$$\mathrm{err}_{\mathrm{train}}(H) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big(H(x_i) \neq y_i\big)$$

$$Z_t = \sum_{i=1}^{N} D_t(i) \exp\left(-\hat{\mathrm{w}}_t y_i f_t\left(x_i\right)\right) \qquad \text{and} \qquad D_t(i) = \alpha_i \quad \text{at iteration } t$$

توجه کنید که خطای آموزش خطای 0-1 است و خطا در نظر گرفته است برای دسته بند ها در هر مرحله از روش Adaboost خطای exponential است.

سوال ۵: (شبیه سازی، ۲۰ نمره)

دیتاست داده شده در این سوال، برای تشخیص phishing email می باشد. دیتاست را دانلود کرده و در قالب pandas dataframe لود کنید:

۱. داده‌ها را به دو بخش ۸۰ درصد آموزش و ۲۰ درصد آزمون تقسیم کنید. تقسیم بندی برای داده های آموزش و آزمون بین دو کلاس باید همگن باشد به این معنا که نسب داده های دو کلاس در داده های آموزش و تست با هم برابر باشد و مقدار random state را برابر با ۴۲ قرار دهید.

۲. ابتدا با استفاده از تمام ویژگی های دیتاست، یک طبقه بند Logistic Regression به دیتاست آموزش دهید و دقت مدل را گزارش کنید.

۳. حال با استفاده از الگوریتم SFS موجود در کتابخانه mlxtend، به ترتیب به تعداد ۲، ۳، ۴ و ۵ ویژگی انتخاب کنید. معیار انتخاب accuracy و cv را ۲ قرار دهید. ویژگی هایی را که الگوریتم در هر مرحله انتخاب می کند، گزارش کنید.

٤. حال شما باید کلاس CustomSFS را به صورت FromScratch پیاده سازی کنید؛ برای این کار کافی است متد fit از این کلاس را پیاده سازی کنید؛ خروجی متد get_k_features را برای ۲، ۳، ۴ و ۵ ویژگی، با خروجی کتابخانه از قسمت قبل مقایسه کنید و در گزارش خود ذکر کنید.

۵. توجه کنید که برای پیاده سازی این کلاس، مجاز به استفاده از کد های mlxtend و sklearn.feature_selection نیستید.

سوال ۶: (شبیه سازی، ۲۵ نمره)

در این تمرین با دو روش یادگیری Ensemble یعنی Bagging و Adaboost آشنا می‌شوید. پیاده‌سازی‌های این دو الگوریتم باید به‌صورت دستی انجام شوند و نمی‌توانید از تابع‌ها و کلاس‌های آماده مربوط به این دو الگوریتم که در کتابخانه‌های آماده وجود دارند استفاده کنید. همچنین نحوه استفاده از مدل XGBoost را یاد خواهید گرفت و نتایج این روش‌ها را با هم مقایسه می‌کنید.

درخت تصمیم با عمق یک (Decision Stump) به‌عنوان مدل پایه در تمامی پیاده‌سازی‌ها استفاده شود که برای آن میتوانید از sklearn.tree.DecisionTreeClassifier با عمق یک استفاده کنید.

```
sklearn.tree.DecisionTreeClassifier(max_depth=1)
```

مراحل:

(الف) آماده‌سازی داده‌ها: در این مرحله، ابتدا مجموعه‌داده Pima Indians Diabetes را از کتابخانه Scikit-Learn بارگذاری نمایید. برای این کار می‌توانید از این کد استفاده کنید:

```
pima = sklearn.datasets.fetch_openml(
       name='diabetes',version=1, as_frame=True)

X = pima.data

# Map 'tested_negative' → 0, 'tested_positive' → 1

y = (pima.target == 'tested_positive').astype(int)
```

داده‌ها را به دو بخش ۸۰ درصد آموزش و ۲۰ درصد آزمون تقسیم کنید. تقسیم بندی برای داده های آموزش و آزمون بین دو کلاس باید همگن باشد به این معنا که نسب داده های دو کلاس در داده های آموزش و تست با هم برابر باشد.

در نهایت، یک مدل پایه اولیه از نوع decision stump تعریف نمایید که در ادامه مورد استفاده قرار خواهد گرفت.

(ب) تحلیل اولیه داده‌ها **(EDA):** در این بخش، شکل کلی و ابعاد داده‌ها را بررسی کنید و خلاصه‌ای از ویژگی‌ها و برچسب‌ها را مشاهده نمایید. همچنین برای درک بهتر توزیع داده یا ارتباط بین برخی ویژگی‌ها، نمودارهای ساده‌ای را ترسیم کنید. انتخاب اینکه نرمال‌سازی انجام بدهید یا ندهید به عهده خودتان است.

(پ) پیاده‌سازی **Bagging :** در این مرحله، الگوریتم Bagging را خودتان باید پیاده‌سازی کنید. برای این کار می‌توانید از decision stumpها استفاده کنید همانطور که قبل تر توضیح داده شد و تعداد مدل‌هارا ۵۰ در نظر بگیرید. حق استفاده از کلاس آماده Bagging از کتابخانه از پیش آماده‌ای را ندارید. پس از آموزش، پیش‌بینی نهایی باید با استفاده از رأی‌گیری بین پیش‌بینی‌های مدل‌ها انجام گیرد.

(ت) پیاده‌سازی**Adaboost :** در این مرحله، الگوریتم Adaboost را باید خودتان پیاده‌سازی کنید. از decsion stump برای این کار استفاده کنید و تعداد مدل‌های را ۵۰ در نظر بگیرید. استفاده از کلاس Adaboost از پیش آماده از کتابخانه‌ها ممکن نیست.

(ث) استفاده از **XGBoost :** در این مرحله یک مدل XGBoost را بر روی داده های آموزشی بدهید و پیش‌بینی انجام دهید. برای اینکار می‌توانید از کتابخانه xgboost استفاده کنید.

(ج) محاسبه دقت: در این بخش، دقت و امتیاز F1 را برای هر یک از مدل‌ها برای داده‌های آموزشی و آزمایشی را محاسبه کرده و به صورت یک جدول گزارش کنید. مدل‌های مورد نظر عبارتند از:

Base Classifier (Decision Stump), Bagging, Adaboost, XGBoost

(چ) گزارش نهایی: در نهایت، یک گزارش کلی از نتایج به‌دست‌آمده تهیه نمایید. در این گزارش به موارد زیر بپردازید:

- عملکرد پیاده‌سازی‌های Bagging و AdaBoost خود را با مدل تصمیم پایه (Decision Stump) مقایسه کنید. آیا بهبود مشاهده می‌شود؟ دلایل احتمالی آن را بر اساس سازوکار این الگوریتم‌ها توضیح دهید.
- مقایسه Bagging و AdaBoost: عملکرد دو پیاده‌سازی خود را با یکدیگر مقایسه کنید. کدام یک بهتر عمل کرد؟ چرا؟
- مقایسه با XGBoost: عملکرد پیاده‌سازی‌های خود را با XGBoost مقایسه کنید.