

به نام ایزد منان



دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

## پروژه پایانی درس داده کاوی

### توضیحات:

- حل این پروژه باید به صورت انفرادی صورت گیرد. حداقل برخورد با پاسخ‌های مشابه، تخصیص نمره کامل منفی به طرفین خواهد بود.
- نوشتن گزارش برای این پروژه الزامی بوده و تمیزی و خوانایی گزارش پروژه از اهمیت بالایی برخوردار است. در صورت امکان می‌توانید گزارش و کد را در یک قابل ژوپیتِر نوتبوک (و یا گوگل کلب) ارسال کنید.
- کد و گزارش پروژه را در قالب یک فایل فشرده (.zip) با الگوی زیر در صفحه‌ی درس بارگذاری کنید:  
  
DM\_FP\_[Student\_number].zip
- در صورتی که درمورد این تمرین سوال یا ابهامی داشتید با ایمیل [dm.1401.spring@gmail.com](mailto:dm.1401.spring@gmail.com) با تدریس‌یاران درس در ارتباط باشید.
- مهلت ارسال پروژه تا ساعت ۱۱:۵۵ روز چهارشنبه مورخ ۸ تیر ۱۴۰۱ است.

نیم‌سال دوم ۱۴۰۰-۱۴۰۱

صفحه

فهرست مطالب

بخش پیاده‌سازی ..... ۳

## بخش پیاده‌سازی

هدف کلی از انجام این پروژه، انجام یک طبقه‌بند<sup>۱</sup> برای یافتن علائم دیابت و یا پیش‌دیابت برای دادگان داده شده است. در این مجموعه داده، ما اطلاعات بیش از ۷۰ هزار بیمار را از طریق پرسشنامه‌ای که برای سازمان کنترل و جلوگیری از ابتلا بیماری‌ها<sup>۲</sup> پر کرده اند داریم. این مجموعه داده شامل ۲۲ ستون به حالت زیر است:

- Diabetes\_binary : ستون هدف که مشخص می‌کند فرد مبتلا به دیابت و یا پیش‌دیابت است یا خیر
- HighBP : ابتلا به فشار خون
- High Cholesterol : کلسترول بالا
- Cholesterol Check : آیا فرد مورد نظر چک آپ برای کلسترول داشته و یا خیر
- BMI
- Smoker : استفاده از مواد مخدر
- Stroke : رخداد سکته
- HeartDiseaseorAttack : حمله قلبی
- Physical Activity : فعالیت فیزیکی
- Fruits : مصرف میوه‌جات

---

<sup>۱</sup> Classifier

<sup>۲</sup> Centers for Disease Control and Prevention (CDC)

- Veggies : مصرف سبزیجات
- Heavy Alcohol Consumption : مصرف الکل بالا
- Any Health Care : فرد مورد نظر بیمه درمانی دارد و یا خیر
- No Doctor because of Cost : آیا برای فرد مورد نظر رخ داده که به علت هزینه‌ها در موقعیت لازم به دکتر رجوع نکند؟
- General Health : سلامت کلی
- Mental Health : سلامت روان
- Physical Health : سلامت فیزیکی
- Difficulty Walking : راه رفتن برای فرد مورد نظر مشکل است و یا خیر
- Sex : جنسیت
- Age : سن
- Education : تحصیلات
- Income : درآمد

در این پروژه هدف بر این است که با استفاده از کتابخانه XGBoost، با طبقه‌بندی داده‌ها بتوانید وجود دیابت و یا عدم وجود آن را تشخیص دهید.

## معرفی XGBoost:

XGBoost و یا Extreme Gradient Boost یک کتابخانه متن باز برای رگولاریزاسیون Gradient Boosting در دسترس در زبان های ++C، جاوا، پایتون، R، اسکالا، جولیا و پرل است. با استفاده از این کتابخانه، می‌توان بسیاری از الگوریتم‌های یادگیری ماشین را پیاده‌سازی کرد. جدای بهبودهای عملکرد مانند سرعت بالا، این کتابخانه مزایای دیگری من جمله موازی سازی در ساخت درخت‌ها، استفاده از حافظه نهان برای افزایش سرعت، رگولاریزاسیون برای مقابله با بیش برازش<sup>۳</sup>، استفاده از اعتبارسنجی متقابل<sup>۴</sup> و ... دارد.

## مرحله پیش‌پردازش (۴۰ نمره):

در این مرحله شما بایستی مجموعه دادگان را برای عملیات آموزش مدل طبقه‌بند آماده کنید. دقت کنید که با گذشتن از این مرحله ممکن است عملکرد مدل شما به طرز قابل توجهی کاهش یابد. اعمال لازم برای این مرحله:

- **حذف داده‌های پوچ:** در مجموعه داده ممکن است برخی از نقاط جدول با Null و یا دیگر کلمات خارج از معنی پر شده باشند. وظیفه شما این است که این موارد را پیدا کنید و طبق موقعیت یکی از تصمیمات لازم را اتخاذ کنید. از تصمیمات ممکن می‌توان به جایگذاری این موارد با میانگین و یا یکی از داده‌های ستون انجام شود. حتی می‌توانید یک سطر و یا ستون را در صورت وجود داده‌های پوچ بیش از حد، حذف کنید.
- **تغییر نام ستون‌ها و یا کلماتی از مجموعه داده که دارای فاصله<sup>۵</sup> هستند.** این فواصل در ساخت درخت ممکن است باعث خطای مدل شوند. به همین خاطر توصیه می‌شود این فواصل را حذف کنید و یا با خط تیره " \_ " جایگذاری کنید.

<sup>۳</sup> Overfitting<sup>۴</sup> Cross validation<sup>۵</sup> White space

- **نرمالیزه کردن** <sup>۶</sup> برخی از ویژگی‌ها، به طوری که بین ویژگی‌ها تفاوت مشهودی از بابت طول بازه نباشد. برای این قسمت، جدا از تغییر مقیاس <sup>۷</sup> ویژگی، می‌توانید یک رنج بزرگ را به تعدادی کوچکتر دسته تقسیم کنید. به طور مثال برای بازه ۱ تا ۱۰۰، اعداد ۱ تا ۱۰ را به دسته ۱ تبدیل کنید، اعداد ۱۱ تا ۲۰ را به دسته ۲ و ...
- **یافتن ویژگی‌های دسته‌بندی شده** <sup>۸</sup>. به طور کلی داده‌های ما به دو دسته عددی <sup>۹</sup> و یا دسته‌بندی شده <sup>۱۰</sup> تقسیم می‌شوند. در ساخت درخت تصمیم ترجیح بر این است که از داده‌های دسته‌بندی شده استفاده نکنیم. در این قسمت شما باید پس از شناسایی ویژگی‌های دسته‌بندی شده، آنها را به روش one-hot-encoding به ویژگی عددی تبدیل کنید.
- \*\*\* دقت کنید که یکی از مهم‌ترین ویژگی‌های داده‌های عددی به نسبت دسته‌بندی شده، امکان مقایسه عددی است. مثلاً اگر ۴ دسته رنگ (آبی، قرمز، زرد، سبز) داشته باشیم نمی‌توانیم مقایسه عددی بین این دسته‌ها داشته باشیم ولی در عوض اگر اعداد بازه ۱ تا ۱۰۰ را به ۱۰ دسته ۱ تا ۱۰، ۱۰ تا ۲۰، ۲۰ تا ۳۰ و ... تقسیم کنیم، می‌توانیم برای هر دسته مقایسه عددی داشته باشیم چراکه به طور مثال اعداد دسته اول کمتر از دسته دوم هستند! پس مثال ۱۰ گروه اعداد ۱ تا ۱۰ ویژگی دسته‌بندی شده حساب نمی‌شوند!
- در مرحله آخر شما باید ستون Diabetes\_binary را به عنوان برچسب از مجموعه داده جدا کنید و در این صورت دادگان و برچسب‌ها به صورت مجزا برای عملیات یادگیری و ساخت مدل آماده هستند.

---

<sup>۶</sup> Normalization
<sup>۷</sup> scale<sup>۸</sup> Categorical Features<sup>۹</sup> Numerical<sup>۱۰</sup> Categorical

## ساخت مدل طبقه‌بند (۲۰ نمره):

در این مرحله شما باید یک XGBoost Classifier تعریف کنید و مراحل یادگیری را بر روی مجموعه داده‌گان انجام دهد. برای ساخت مدل می‌توانید از پارامترهای زیر استفاده کنید:

```
Learning_rate=0.1
Max_depth=4
N_estimator=200
Subsample=0.5
Colsample_bytree=1
Random_seed=123
Eval_metric='auc'
Verbosity=1
```

برای انجام عملیات یادگیری توصیه می‌شود که early\_stopping\_rounds را برابر ۱۰ قرار دهید تا یادگیری زمانبر نباشد.

در انتها باید دقت مدل بر روی داده‌گان آموزش و تست را گزارش دهید. همچنین ماتریس درهم‌ریختگی و precision و recall را نیز محاسبه کنید.

## تنظیم هایپر پارامترها (۳۰ نمره):

همانگونه که در قسمت قبل مشاهده کردید، پارامترهای زیادی در ساخت مدل دخیل هستند. برای مشاهده تمامی پارامترها می‌توانید از خط کد زیر استفاده کنید.

```
xgboost.XGBClassifier().get_params()
```

تمامی این پارامترها در عملکرد این مدل تاثیر به سزایی می‌گذارند. برای پیدا کردن بهترین پارامترها، می‌توانیم مدل را بر روی ترکیب‌های متفاوتی از این پارامترها بسازیم و بهترین مدل را خروجی دهیم. در این قسمت نیاز است تا مدل را بر روی ترکیب‌های متفاوت از پارامترهای زیر تست کنید و بهترین مدل را خروجی دهید.

```
learning_rate_list = [0.02, 0.05, 0.1, 0.3]
max_depth_list = [2, 3, 4]
n_estimators_list = [100, 200, 300]
colsample_bytree = [0.8, 1]
```

دقت کنید که نیازی به ساخت دستی مدل‌ها نیست و می‌توانید از GridSearchCV بهره ببرید. در این مورد از اعتبارسنجی متقابل<sup>۱۱</sup> با ۳ نقطه جدا سازی استفاده کنید. برای تابع امتیازدهی می‌توانید از تابع زیر استفاده کنید (لازم به ذکر است که roc\_auc\_score را از پکیج sklearn.metrics باید import کنید):

```
def my_roc_auc_score(model, X, y): return roc_auc_score(y, model.predict_proba(X)[:,1])
```

برای ساخت مدل از پارامترهای زیر استفاده کنید:

```
Eval_metric='auc'  
Subsample=0.5
```

پس از ساخت مدل و انجام اعمال یادگیری بر روی تمامی ترکیبات پارامترها، بهترین پارامترها را گزارش کنید. بهترین مدل (که توسط بهترین پارامترها تنظیم شده) را بدست آورید و دقت در داده‌های تست و آزمون، ماتریس درهم‌ریختگی و precision و recall را برای این مدل محاسبه کنید.

تصویر سازی تغییر هایپر پارامترها (۱۰ نمره):

در انتها تغییرات مدل بر اساس چهار پارامتر گفته شده در قسمت قبل بر روی نمودار ببرید و نشان دهید تغییر این پارامترها چه تاثیری بر دقت و یا عملکرد مدل ایجاد می‌کند.

موفق باشید