به نام ایزد منان

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

پروژه پایانی
درس داده‌کاوی

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 ژوپیتر نوتبوک (و یا گوگل کلب) ارسال کنید.
- کد و گزارش پروژه را در قالب یک فایل فشرده (.zip) با الگوی زیر در صحفه‌ی 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 با تدریس‌یاران درس در ارتباط باشید.
- مهلت ارسال پروژه تا ساعت ۱۱:۵۵ روز چهارشنبه مورخ ۸ تیر ۱۴۰۱ است.

نیم‌سال دوم ۱۴۰۰-۱۴۰۱

فهرست مطالب

صفحه

بخش پیاده‌سازی

هدف کلی از انجام این پروژه، انجام یک طبقه‌بند[1] 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[2] 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

[1] Classifier

[2] 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 : آ یا برای فرد مورد نظر رخ داده که به علت هزینه ها 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 : درامد

در این پروژه هدف بر این ا ست که با ا ستفاده از کتابخانه XGBoost، با طبقه بندی داده ها 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 پیاده سازی کرد. جدای بهبودهای عملکرد مانند سرعت بالا، این کتابخانه مزایای دیگری من جمله موازی سازی در ساخت درخت ها، استفاده از حافظه نهان برای افزایش سرعت، رگولاریزاسیون برای مقابله با بیش برازش[۳]، استفاده از اعتبارسنجی متقابل[۴] 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 ستون ها و یا کلماتی از مجموعه داده که دارای فاصله[۵] 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.

برای این قسمت، اسامی ستون ها را تغییر ندادم و فقط اسامی داخل داده هایی را که اسپیس داشتند، اسپیس را با آندرلاین جایگزین کردم

[۳] Overfitting

[۴] Cross validation

[۵] White space

- **نرمالیزه کردن**[۶] برخی از ویژگی‌ها، به طوری که بین ویژگی‌ها تفاوت مشـــهودی از بابت طول بازه نباشد. برای این قسمت، جدا از تغییر مقیاس[۷] 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 دســـته‌بندی شـــده**[۸]. به طور کلی داده‌های ما به دو دســـته عددی[۹] و یا دسته‌بندی شده[۱۰] 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 مهم ترین ویژگی های داده‌های عددی به نســبت دســته‌بندی شــده، امکان مقایسه عددی است. مثلا اگر ۴ دسته رنگ (آبی، قرمز، زرد، سبز) داشته باشیم نمی‌توانیم مقایســه عددی بین این دســته‌ها داشــته باشــیم 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 شــما باید ســتون Diabetes_binary را به عنوان برچســب از مجموعه داده جدا کنید و در این صورت دادگان و برچسب‌ها به صورت مجزا برای عملیات یادگیری و ساخت مدل آماده هستند.

مقادیری که بر اساس
one hot encoding
به ویژگی
general health
داده شد:
very low = 10000
low = 01000
medium = 00100
Good = 00010
high = 00001

۶ Normalization

۷ scale

۸ Categorical Features

۹ Numerical

۱۰ Categorical

ساخت مدل طبقه‌بند (۲۰ نمره):

در این مرحله شـــما باید یک XGBoost Classifier تعریف کنید و مراحل یادگیری را بر روی مجموعه دادگان 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 می شود که early_stopping_rounds را برابر ۱۰ قرار دهید تا یادگیری زمانبر نباشد.

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

تنظیم هایپرپارامترها (۳۰ نمره):

همانگونه که در قسمت قبل مشاهده کردید، پارامتر های 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 ســـزایی 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 قســـمت نیاز اســـت تا مدل را بر روی ترکیب‌های متفاوت از پارامترهای زیر تســـت 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 اعتبار سنجی متقابل[۱۱] با ۳ نقطه جدا سازی ا ستفاده کنید. برای تابع امتیازدهی می‌توانید از تابع زیر ا ستفاده کنید (لازم به ذکر ا ست 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.pred
ict_proba(X)[:,1])
```

برای ساخت مدل از پارامتر های زیر استفاده کنید:

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

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

تصویر سازی تغییر هایپرپارامترها (۱۰ نمره):

در انتها تغییرات مدل بر اســاس چهار پارامتر گفته شــده در قســمت قبل بر روی نمودار ببرید و نشان دهید تغییر این پارامترها چه تاثیری بر دقت و یا عملکرد مدل ایجاد می‌کنند.

موفق باشید

[۱۱] Cross validation