Ensemble Classifiers

مقدمه

در این پروژه قصد داریم با پیاده سازی Ensemble Classifiers که در کلاس تدریس شد آشنا شده، خروجی آنها، مراحل و Classifierهایشان را بر روی یک دیتاست ساده visual کنیم، در قدم بعد نتایج به دست آمده را بررسی و تحلیل کنیم. در مرحله آخر نیز الگوریتم‌ها را بر روی یک دیتاست واقعی اجرا کنیم و سعی کنیم نمونه‌ای از عملکرد این الگوریتم‌ها برروی یک دیتاست پزشکی را ببینیم.

۱ فاز اول: آشنایی و بررسی الگوریتم‌ها

برای این فاز یک دیتاست با عنوان Dataset1 داده شده است بعد از لود کردن و ویژوال کردن آن الگوریتم‌ها و خواسته‌هایی که در ادامه عنوان شده است را پیاده‌سازی کنید. به یاد داشته باشید که لازم است نمودار دقت بر اساس تعداد Estimatorها را برای هر کدام از الگوریتم‌های Bagging، Random Forest و Adaboost در بخش‌های ۱ تا ۳ این فاز رسم کنید. فراموش نکنید در انتهای هر کدام از بخش‌های ۱ تا ۴، معیارهای ارزیابی Precision، Recall و F1-Score را برای‌داده‌های Train و Test به دست بیاورید.(برای این معیارها می‌توانید از توابع آماده استفاده کنید.)

۱.۱ بخش اول: **Bootstrap Aggregating(Bagging)**

در این بخش ابتدا الگوریتم Bagging را پیاده‌سازی کنید برای اینکار می‌توانید از کلاس‌های DecisionTreeClassifier و BaggingClassifier کتابخانه sklearn استفاده کنید مقادیر هایپرپارامترها را خودتان قرار دهید سعی کنید با تغییر آنها عملکرد الگوریتم را بهبود ببخشید. در ادامه خروجی پیش‌بینی شده الگوریتم Bagging و ۵ کلسیفایر Decision Tree را برروی دیتاست Visual کرده و برداشت خود از عملکرد هر کدام از کلسیفایرهای ۶ تصویر را در فایل گزارش خود بنویسید (سعی کنید درختانی را انتخاب کنید که تفاوت عملکردشان در شکل‌ها مشخص باشد.)

۲.۱ بخش دوم: **Random Forest**

در این بخش الگوریتم Random Forest را پیاده‌سازی کنید برای اینکار می‌توانید از کلاس RandomForestClassifier کتابخانه sklearn استفاده کنید. مقادیر هایپرپارامتر ها را تعیین کنید و بهبود ببخشید سپس خروجی الگوریتم و ۵ مورد از درختان Randomizedشده را برروی دیتاست ویژوال کرده و بررسی کنید.

۳.۱ بخش سوم: **AdaBoost**

در این بخش الگوریتم AdaBoost را برروی دیتاست اجرا کنید برای اینکار میتوانید از کلاس AdaBoostClassifier استفاده کنید بعد از رسیدن به نتیجه‌ای خوب و امتحان کردن مقادیر مختلف هایپرپارامترها میخواهیم عملکرد الگوریتم را در مراحل اجرای آن ترسیم کنید. برای اینکار ۸ مرحله از اجرای آن را برروی دیتاست ترسیم کرده و بررسی کنید که در آن مرحله کلسیفایرهای ضعیف چه عملکردی داشته‌اند چه بخش‌هایی را به درستی کلسیفای کرده و این بخش‌ها در طول انجام الگوریتم و در مراحلی که ترسیم کرده‌اید چگونه بهبود یافته و تغییر کرده‌اند.(بهتر است این ۸ مرحله با فاصله از یکدیگر انتخاب شوند تا تفاوت بین آنها بهتر مشخص شود.) کلسیفایرهای ضعیف این بخش را به دلخواه خودتان تعیین کنید.

۴.۱ بخش چهارم: Stacked Learners

برای پیاده سازی الگوریتم Stacked Learners در مرحله اول از شما میخواهیم الگوریتم K-Nearest Neighbors(KNN) که در کلاس به صورت مختصر معرفی و اشاره شد را بعد از آشنایی با استفاده از کلاس KNeighborsClassifier برروی دیتاست پیاده‌سازی کنید سپس با استفاده از این کلسیفایر و چندین کلسیفایر دیگر به پیاده سازی الگوریتم Stacked Learners بپردازید(در صورت امکان میتوانید از کلسیفایرهای بخش‌های ۱ تا ۳ نیز استفاده کنید.) در نظر داشته باشید که برنامه نویسی این بخش به عهده شماست و نباید از کتابخانه‌ها یا کلاس‌های آماده استفاده کنید(صرفا برای ترکیب کلسیفایرها باید پیاده‌سازی خودتان را داشته باشید پس امکان استفاده از کلاس‌هایی که در این بخش و بخش‌های قبلی نام‌برده شد را دارید). اگر هر یک از کلسیفایرهای به کار برده شده در این بخش overfit شده بود می‌توانید راهکاری که در درس برای رفع این مشکل گفته شده بود را پیاده‌سازی کنید در نظر داشته باشید که پیاده‌سازی این راهکار و مدیریت این مشکل با آن دارای نمره اضافه می‌باشد.(در صورتی که کلسیفایری overfit نشده بود و قصد دریافت نمره اضافه دارید می‌توانید یک کلسیفایر که overfit شده است را به مجموعه کلسیفایرهای اولیه این الگوریتم اضافه کنید.)

۲ فاز دوم: استفاده از الگوریتم‌ها

دیتاستی با عنوان Dataset2 در اختیار شما قرار داده شده است پس از بررسی این دیتاست الگوریتم‌های فاز اول(Bagging, AdaBoost, Random Forest و Stacked Learners) را برروی این دیتاست نیز اجرا کنید. دقت کنید که این دیتاست نیاز به پیش‌پردازش دارد در صورتی که در یکی از مراحل پیش‌پردازش Correlation بین ویژگی‌ها را ترسیم و تحلیل کنید سپس ویژگی‌هایی که ارتباط نزدیکی با یکدیگر دارند را حذف کنید نمره اضافه دریافت خواهید کرد. پس از اجرا Precision، Recall و F1-Score را برای داده‌های Train و Test محاسبه کنید.(برای محاسبه این معیارها می‌توانید از توابع آماده استفاده کنید.)

نکات و توضیحات تکمیلی

- میتوانید الگوریتم‌های طبقه‌بندی دلخواه خود را در فازها برای پیاده‌سازی الگوریتم‌های Ensemble استفاده کنید.
- نیازی نیست در گزارش خود به توضیح مسیری که برای تعیین هایپرپارامترها طی کرده‌اید بپردازید.
- انجام پروژه می‌تواند در قالب گروه‌های دو نفره و یا به صورت انفرادی صورت گیرد. آپلود فایل‌ها همین که توسط یکی از اعضای گروه انجام شود کافی است.
- علاوه بر سورس کد پروژه، باید فایل مستندات نیز آپلود شود. در این فایل نام هر دو عضو گروه را ذکر کنید.
- هر گونه شباهت نامتعارف بین کد شما و کد سایر گروه‌ها و یا کدهای موجود بر روی اینترنت تقلب محسوب می‌شود و نمره‌ای برای این پروژه دریافت نخواهید کرد.
- در صورت نوشتن داکیومنت تمیز (برای مثال با LaTeX) نمره اضافه برای شما در نظر گرفته خواهد شد.
- فایل شامل سورس کد پروژه و مستندات را در قالب فایل zip و با نام شماره دانشجویی خود ذخیره و ارسال نمایید.
- در صورت داشتن هرگونه سوال می‌توانید با MohannaAnsari یا MohMollaei در ارتباط باشید اما بهتر است در گروه درسی مطرح نمایید.