علیرضا جابری راد
610397174

داده کاوی
تمرین سوم

داده های مذکور در صورت سوال ابتدا در متغیر ppl ذخیره میشوند و سپس داده های اولیه (ماتریس مربوط به عکس ها) در x و کلاس هر داده اولیه در y ذخیره میشوند.

سپس تعدادی از عکس های موجود در مجموعه داده با استفاده از تابع show_original_image() نمایش داده میشوند.

در ادامه با استفاده از تابع train_test_split (که در کد با نام split از آن استفاده میکنم) مجموعه داده را به صورت رندوم به داده تست (سی درصد داده ها) و داده آموزش (هفتاد درصد داده ها) تقسیم میشوند.

در بلاک ششم برنامه، الگوریتم PCA را روی داده آموزش فیت کرده و نمودار تعداد کامپوننت های حاصل از اعمال الگوریتم PCA را از کم تا بیشترین مقدار ممکن زیاد میکنیم و تغییرات واریانس داده ها را با آن در نمودار اول میتوانید مشاهده کنید.

در ادامه با توجه به نمودار بخش قبل، انتخاب تعداد کامپوننت ها بین 100 تا 200 میتواند با تقریب خوبی مدل کننده ی داده های اصلی باشد و به دلیل کاهش ابعاد داده های اولیه، میتواند هزینه پردازشی کمتری را بر روی پردازنده تحمیل کند. لذا من تعداد کامپوننت 150 را برای این منظور در کد لحاظ کردم.

پس از انجام مرحله پیش پردازش روی داده های اولیه، وقت آن رسیده است که مدل های پیش بینی کننده را روی آن ها فیت کنیم. در گام اول ماشین بردار پشتیبان را روی داده های آموزش فیت میکنیم. طبق تغییراتی که در پارامترهای این classifier دادم، مقادیر پیش فرض آن، منجر به بهترین نتایج ممکن میشدند. نتیجه بدست آمده از این classifier حدود نود درصد است که با توجه به اینکه الگوریتم ماشین بردار پشتیبان (معمولا) بر اساس محاسبات جبری درونیابی لاگرانژ بدست می آید، اگر هربار داده یکسانی به آن داده شود منجر به پاسخ یکسانی خواهد شد.

در مرحله بعد، داده های آموزش را با multi layer perceptron فیت میکنیم. بهترین نتایج طبق آزمایش و خطاهایی که داشتم، استفاده از دو لایه مخفی با تعداد گره بین 100 تا 150 و تابع فعال سازی relu منجر به بهترین نتیجه میشود. نتیجه حاصل از این classifier با توجه به مقادیر رندوم اولیه ای که به وزن ها اختصاص پیدا میکند میتواند در هر بار آموزش آن، حتی با داده یکسان، متفاوت باشد. طبق تست هایی که داشتم نتیجه حاصل از این بخش هم حدود 85 تا 90 درصد دقت دارد.

به طور کلی SVM کمی نتیجه بهتری نسبت به MLP داشته که شاید علت آن کوچک بودن دیتاست بوده و شاید اگر دیتاست بزرگ تری داشتیم، MLP میتوانست نتیجه ی بهتری را کسب کند. ولی نکته ی جالب اینجاست که درباره کلاس 1 که داده های بیشتری را هم در دیتاست داشته، معمولا MLP بهتر عمل کرده است.