دانشگاه صنعتی خواجه نصیرالدین طوسی
۱۳۰۷
دانشکده مهندسی برق

مینی پروژه اول

نام و نام خانوادگی:

فاطمه پاکدامن

شماره دانشجویی:

۹۹۲۴۷۵۳

استاد درس:

دکتر مهدی علیاری

درس:

مبانی سیستم‌های هوشمند

آذر ۱۴۰۲


بسم الله الرحمن الرحيم

فهرست مطالب

چکیده

در این پروژه، علاوه بر آشنایی با کتابخانه‌های آموزش ماشین، سعی شده است تا به بررسی مبانی ریاضی مرتبط با این حوزه پرداخته شود. این مبانی شامل اصول و تئوری‌های مهمی می‌شوند که اساس یادگیری ماشین را تشکیل می‌دهند. علاوه بر این، پیاده‌سازی دستی تعدادی از توابع مرتبط با کلاس‌بندی داده‌ها به منظور درک بهتر از این مفاهیم ریاضی انجام شده است.

از بخش‌های مهم این پروژه، مباحث مربوط به تولید داده و پیش‌پردازش است. پیش‌پردازش در فرآیند یادگیری ماشین برای بهبود کیفیت داده و افزایش دقت مدل‌ها از اهمیت خاصی برخوردار است. علاوه بر این، مفهوم مرز تصمیم‌گیری نیز بررسی شده است. نهایتاً، شاخصهای ارزیابی متنوعی که به تخمین عملکرد مدل‌ها می‌پردازند، مورد بررسی قرار گرفته و اهمیت این ابزارها در ارتقاء کیفیت ماشین بررسی شده است.

در کل هدف از این پرژه آشنایی با کتابخانه‌های آموزش ماشین برای کلاس‌بندی داده‌ها و همچنین برسی مبانی ریاضی و پیاده سازی تعدادی از این توابع به صورت دستی به هدف یادگیری مفهوم پشت این مباحث است. در این میان به موادی از جمله تولید داده، پیش پردازش، مرز تصمیم گیری و انواع شاخصه‌های ارزیابی نیز اشاره شده است.

بخش اول

مقدمه

در بخش اول با استفاده از کتابخانه sklearn مجموعه داده‌ای ایجاد شده و سپس با طبقه بندی‌های آماده این مجموعه سعی بر جدا سازی ۲ طبقه از هم شده است. تکنیک‌های استفاده شده شامل LogisticRegression، perceptron و sgd است. پس از جدا سازی دقت این جدا سازی برسی شده و مرز تصمیم‌گیری رسم شده است. پس از آن یک مسئله دارای سه کلاس برسی شده و این تکنیک‌ها بر آن اعمال شده است.

۱. تولید داده

خواسته سوال یه مجموعه داده با ۱۰۰۰ نمونه، ۲ کلاس و ۲ ویژگی است. به این منظور کلاس مربوط به تولید داده‌ها با استفاده از دستور زیر بارگزاری شده است. پس از آن با استفاده از تابع make_classification نمونه‌ها تولید شده است. در این تابع n_samples مربوط به تعداد نمونه‌ها، n_features مربوط به تعداد ویژگی‌ها، n_clusters_per_class مربوط به تعداد خوشه‌ها در هر کلاس، class_sep مربوط به میزان جدا پذیری دو کلاس است.

```
from sklearn import datasets
X, y = datasets.make_classification(n_samples=1000, n_features=2,
n_redundant=0,n_clusters_per_class=1, class_sep=1, random_state=53)
```

پس از آن دو ویژگی مربوطه جدا شده و با توجه به کلاس هر داده، رنگ آن تعیین شده است.

۲. طبقه بندی

با استفاده از حداقل سه طبقه بند آمادهٔ پایتون و در نظر گرفتن فراپارامترهای مناسب، دو کلاس موجود در دیتاست قسمت قبلی را از هم تفکیک شده‌اند. این سه طبقه بند شامل LogisticRegression، perceptron و sgd است. در هر بخش ابتدا مدل مربوطه ساخته شده، سپس به داده‌های آموزشی برازش شده است. پس از آن دقت آن بر روی داده‌های آموزش و تست برسی شده است. منظور از دقت دصد تعداد داده‌های درست تشخیص داده شده به تعداد کل داده‌ها است که با فراخوانی تابع score بر روی مدل محاسبه شده است. نمونه‌ای از کد این فرایند و نتایج آن در زیر آمده است.

```
#2ed method for classification - perceptron
model2 = linear_model.Perceptron(max_iter=100 , random_state=53)
model2.fit(X_train, y_train)
train2_acc = model2.score(X_train, y_train)
test2_acc = model2.score(X_test, y_test)
print(f"accuracy on train set :{train2_acc*100:.2f}%")
print(f"accuracy on test set : {test2_acc*100:.2f}%")
accuracy on train set :96.75%
accuracy on test set : 95.50%
```

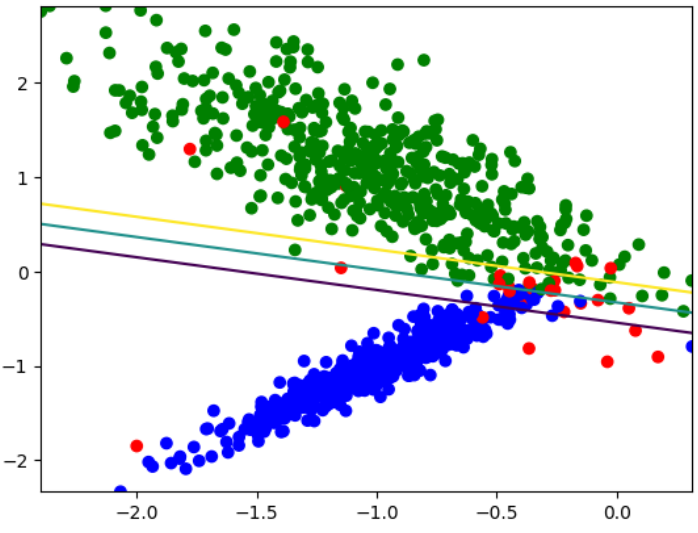
لازم به ذکر است پیش از انجام فرایند آموزش داده‌ها به دو قسمت آموزش و تست تقسیم شده و پس از آن در یکی از روندهای آموزش از داده نرمالایز شده استفاده شده است به طوری که داده‌ها بین صفر و یک قرار گرفته‌اند. این پیش پردازش سبب بهبود عملکرد مدل شده است. در بخش دوم در این رابطه توضحات بیشتری داده شده است.

قابل توجه است که میزان دورهای آموزش و نوع تلفات در مدل‌ها قابل تعیین است.

۳. مرز تصمیم گیری

برای رسم مرز تصمیم گیری مدل‌ها، ابتدا حداقل و حداکثر مقدار داده‌ها محاسبه شده است. پس از آن تعداد ۲۰۰ نقطه بین فاصله بین حداقل و حداکثر تولید شده است. سپس برای آن که به صورت زوج نقطه در یک صفحه باشند از meshgrid استفاده شده است. در نهایت برای استفاده از تابع decision_function ابتدا این نقاط فلت شده و سپس به هم چسبانده شده‌اند. در نهایت پیش بینی مدل بر روی داده‌ها محاسبه شده و بر اساس آن که داده در کدام کلاس است و درست تشخیص داده شده است، رنگ آن برای نمایش تعیین شده است. داده‌هایی که در کلاس اشتباه هستند به رنگ قرمز نمایش داده شده‌اند. نتایج حاصل از این عملیات در شکل ۱ آمده است.

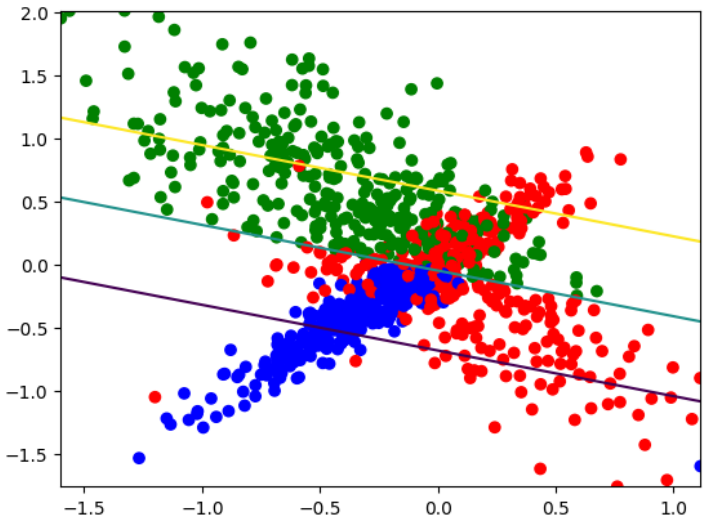
میزان دقت مدل‌های مختلف برای داده‌های آموزش حدود ۹۷ درصد و بر روی داده‌های تست بین ۹۵ تا ۹۷ درصد گزارش شده است.



شکل ۱) داده‌ها و مرز تصمیم‌گیری

۴. ایجاد چالش در طبقه بندی

برای آنکه داده‌تولید شده میزان چالش بیشتری برای طبقه بندی داشته باشد پارامتر class_sep مربوط به میزان جدا پذیری دو کلاس است در تابع را کاهش داده و سپس مجددا طبقه بندی را بر داده‌ها اعمال شده است. مشاهده می‌شود که با در هم رفتن کلاس‌ها در هم، میزان جدا پذیری آن‌ها کم شده و یک خط به تنهایی قادر به تفکیک آن‌ها به نخواهد بود. نتایج حاصل در شکل ۲ قابل مشاهده است.

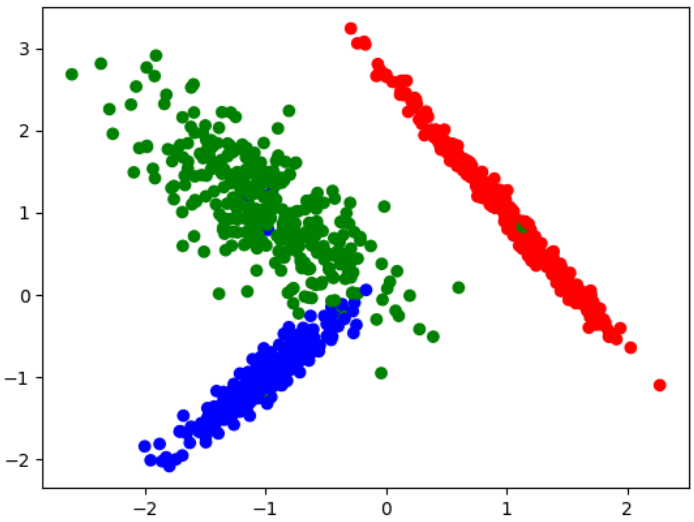


شکل ۲) داده‌ها و مرز تصمیم‌گیری با درهم رفتگی داده‌ها

میزان دقت مدل‌ها بر روی این داده‌ها طبیعتا کاهش پیدا کرده است و بین ۵۸ تا ۶۲ درصد بر روی داده‌های تست اعلام شده است. همان طور که قبلا اشاره شد با درهم رفتگی داده نمی‌توان از یک خط برای جدا سازی آن‌ها استفاده کرد. نمونه‌هایی که در شکل ۲ با رنگ قرمز نشان داده شده است و به طور چشمی نیز مشخص است که تعداد نمونه‌های اشتباه دسته‌بندی شده به طور قابل توجهی افزایش یافته است.

۵. افزودن یک کلاس

برای افزایش تعداد کلاس‌ها در بخش تولید داده کافی است تا پارامتر مربوطه را به ۳ افزایش دهیم. در این صورت ۳ کلاس داریم. با توجه به این که تعداد داده‌های تولید شده در هر ۳ کلاس برابراند در بخش تقسیم به آموزش و تست از هر نوع داده ۳۳ درصد در هر قسمت وجود دارد. نمایشی از داده‌ها در شکل ۳ آمده است.



شکل ۳) داده‌ها در سه کلاس

با توجه به استفاده از طبقه‌بندی های آماده نیازی به تغییر در فرایند آموزش نیست و فرایند آموزش به راحتی صورت گرفته است .نتایج حاصل بر روی داده‌های آموزش بین ۹۴ تا ۹۸ درصد و در داده‌های تست ۹۵ تا ۹۷ درصد گزارش شده است.

بخش دوم

مقدمه

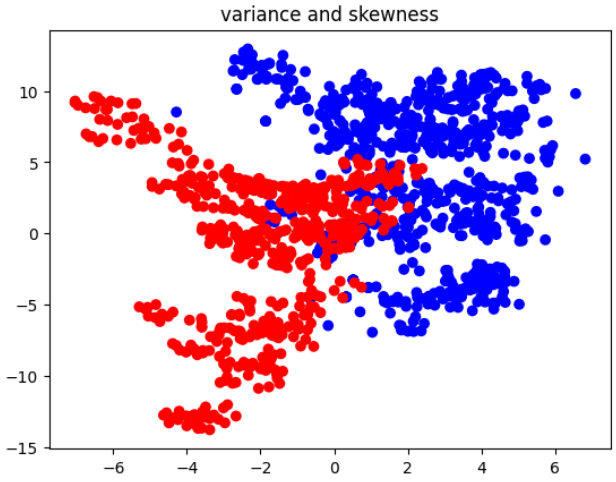
در این بخش تلاش بر آن است تا بدون استفاده از کتابخانه آمده، طبقه‌بندی صورت بگیرد. به این منظور مفاهیمی از جمله گرادیان نزولی، تابع اتلاف و دقت به صورت دستی نوشته شده و در نهایت یک تابه برای برازش داده‌ها نوشته شده است.

۱. داده‌های مورد بررسی

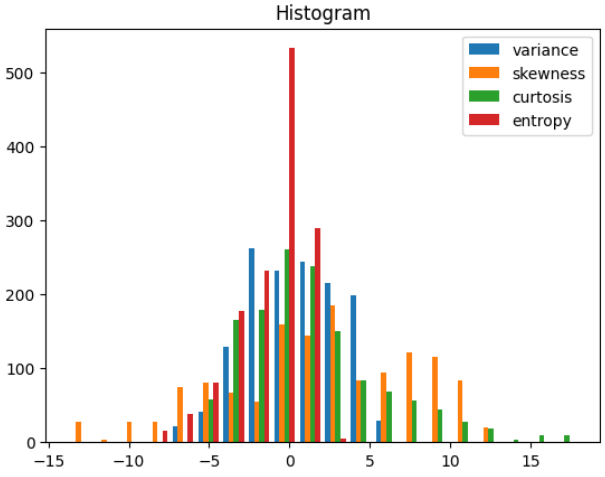
این داده‌ها به منظور احراز هویت اسکناس از روی تصاویر استخراج شده‌اند. دارای ۴ ویژگی است که شامل واریانس تصویر تبدیل شده موجک، چولگی تصویر تبدیل شده موجک، کشیدگی تصویر تبدیل شده موجک، و آنتروپی تصویر است که مقادیری پیوسته دارند. در انتها کلاس مربوط به هر داده مشخص شده است که شامل لیبل صفر و یک است که نشان می‌دهد اسکناس جعلی و یا اصلی است.

برای بارگزاری این داده‌ها از گوگل درایو از دستور gdown استفاده شده و سپس به پوشه‌ای منتقل شده است و در نهایت از حالت زیپ استخراج شده است. در انتها در حین خواندن فایل، برچسب مورد نظر هر سطون به آن نسبت داده شده است.

با توجه به وجود ۴ ویژگی نمایش داده‌ها بر حسب همه آن‌ها امکان پذیر نیست ولی برای نمایش حدودی آن بر حسب دو ویژگی و کلاس آن در شکل ۴ الف به تصویر کشیده شده است. علاوه بر آن هیستوگرام مربوط به این ویژگی‌ها رسم شده است تا دید بهتری نسبت به آن‌ها بدهد که در شکل ۴ ب آمده است.



شکل ۴) الف) نمایش داده‌ها برحسب دو ویژگی واریانس و چولگی



ب) هیستوگرام مربوط به ویژگی‌ها

۲. مخلوط و تقسیم داده‌ها

با توجه به این که ترتیب داده‌های دیده شده توسط ماشین مهم است، نیاز است تا داده‌ها به هم ریخته باشند پس به صورت رندم جای آن‌ها را عوض کرده و سپس برای آموزش شبکه استفاده می‌شود.

با توجه به این که مدل برازش شده داده‌های آموزشی را دیده است، برای تست این مدل باید داده‌هایی داشتته باشیم که ماشین آن‌ها را به عبارتی ندیده است و در فرایند آموزش مشارکت نداشته است. در این صورت اگر یک ویژگی یا ترتیب خاصی در داده‌های آموزش باشد و ماشین آن را یاد بگیرد با تست بر روی داده‌های نو می‌توان این موضوع را متوجه شد و با توجه به شرایط جلوی این کار را گرفت.

در این فرایند با توجه به اینکه ماشین به داده‌های زیادی برای آموزش نیاز دارد با توجه به تعداد داده‌ها ممکن است ۲۰، ۱۰ و حتی در مواردی که مجموعه داده بسیار بزرگ است ۱ با ۰.۱ دصد داده‌ها برای تست انتخاب شوند. لازم به ذکر است که انخاب این داده‌ها باید کاملا به صورت تصادفی صورت بگیرد تا ارتباط خاصی در آن‌ها نباشد. همچنین نیاز است که از هر نوع کلاس در داده تست و آموزش وجود داشته باشد چرا که ماشین نیاز به یادگیری آن و سپس ارزیابی آموزش دارد.

در این قسمت فرایند تقسیم و بر زدن با استفاده از تابع train_test_split صورت گرفته است که خود به صورت رندم تعدادی داده را برای تست و تعدادی برای آموزش انتخاب می‌کند. در این تابع درصد مجموعه تست از کل داده برابر با ۰.۲ قرار گرفته است تا ۲۰ درصد از داده‌ها به عنوان تست درنظر گرفته شوند. همچنین با تنظیم پارامتر stratify درصد هر کلاس از داده در تست و آموزش برابر خواهد بود.

۳. مدل، تابع اتلاف و الگوریتم یادگیری و ارزیابی

به طور کلی در روند آموزش نیاز است تا پیش‌بینی مدل بر روی داده‌ها، محاسبه خطا و دقت بر روی داده‌ها، محاسبه گرادیان خطای تابع هزینه، و در نهایت به‌روزرسانی وزن‌ها با استفاده از گرادیان کاهشی صورت بگیرد.

به این منظور تابع زیر پیاده سازی‌شده‌اند تا یک مدل رگرسیون پیاده‌سازی شود و داده‌های مورد نظر را به دو دسته تقسیم کند.

تابع sigmoid تابع فعال‌سازی را پیاده‌سازی می‌کند. ورودی آن یک مقدار است و خروجی آن مقدار تابع سیگموید بر اساس فرمول آن پیاده سازی شده است. پس از آن تابع مدل رگرسیون لجستیک را پیاده‌سازی می‌شود تا با دریافت ورودی و وزن‌ها، خروجی مدل را محاسبه کند. ورودی‌های آن شامل ویژگی‌های ورودی و بردار وزن و خروجی مدل را با استفاده از تابع سیگموید با ضرب داخلی این دو محاسبه می‌کند. تابع خطا، خطای (Binary Cross-Entropy) بین خروجی واقعی و خروجی مدل را محاسبه می‌کند. این خطا را به صورت یک بردار بر روی هر داده و همچنین و میانگین این خطا را برمی‌گرداند.

تابع گرادیان خطای تابع هزینه نسبت به وزن‌ها را محاسبه می‌کند. در نهایت برای بروزرسانی وزن ها تابع گرادیان نوزولی با توجه به گرادیان بدست آمده وزن‌ها را با یک ضریب یادگیری ثابت بروز می‌کند.

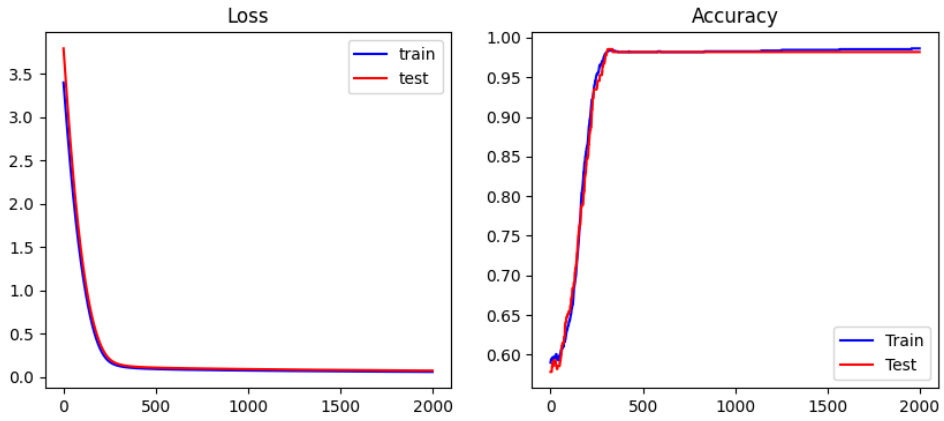
تابع دقت این مدل را با مقایسه خروجی واقعی با خروجی مدل محاسبه که چه درصدی از کل داده‌ها درست تشخیص داده شده‌اند. دقت به عنوان تعداد پیش‌بینی‌های صحیح تقسیم بر تعداد کل پیش‌بینی‌ها محاسبه می‌شود.

پس از پیاده سازی این توابع یک تابع کلی برای فرایند آموزش و تست نوشته شده است که این تابع یک مدل رگرسیون لجستیک را با استفاده از گرادیان کاهشی بر روی داده‌های آموزشی برازش می‌کند. ورودی های آن شامل ویژگی‌های ورودی آموزش، برچسب‌های واقعی داده‌های آموزش، بردار وزن‌ها (که در ابتدا به صورت رندم مقدار اولیه گرفته و به این تابع داده می‌شود)، تعداد دور یا تکرار آموزش، نرخ یادگیری، ورودی تست و برچب تست می‌شود. تعدادی از این موارد به صورت پیشفرض مقدار دارند یا می‌توان از داده تست استفاده نکرد. خروجی این تابع بردار وزن‌ها پس از آموزش است. تعدادی تست برای ارزیابی عملکرد درست این توابع نوشته شده است.

در هر حلقه محاسبه پیش‌بینی مدل بر روی داده‌های آموزش صورت می‌گیرد و اگر داده‌های تست موجود باشند، محاسبه پیش‌بینی مدل بر روی داده‌های تست نیز محاسبه خطا و دقت بر روی داده‌های آزمون محاسبه می‌شود. داده‌های تست تاثیری بر گرادیان‌ها ندارد و صرفا جهت ارزیابی عملکرد مدل اضافه شده‌اند. پس از آن خطا و دقت بر روی داده‌های آموزش، گرادیان خطای تابع هزینه محاسبه و به‌روزرسانی وزن‌ها با استفاده از گرادیان کاهشی انجام می‌شود. نمایش ویژگی‌هایی از جمله خطا و دقت ۱۰ بار در فرایند آموزش صورت می‌گیرد. علاوه بر آن این مقادیر در یک لیست زخیره و در انتهای آموزش برای ارزیابی فرایند آموزش نمودار آن‌ها رسم می‌شود.

برای بارگزاری این داده‌ها از گوگل درایو از دستور gdown استفاده شده و سپس به پوشه‌ای منتقل شده است و در نهایت از حالت زیپ استخراج شده است. در انتها در حین خواندن فایل، برچسب مورد نظر هر سطون به آن نسبت داده شده است.

ابتدا مقادیر اولیه وزن ها به صورت رندم، ضریب یادگیری و تعداد دور آموزش به صورت سعی و خطا تنظیم شده است. با اجرای این تابع بر روی داده‌ها با ضریب یادگیری ۰.۰۱ و ۲۰۰۰ دور آموزش، نتایج بدست آمده در شکل ۵ نمایش داده شده است. مقدار نهایی دقت بر روی داده تست برابر با ۹۸ درصد بدست آمده است.



شکل ۵) نتایج حاصل از آموزش

همان طور که انتظار می‌رفت خطا داده‌ها در هر دو گروه با پیشرفت آموزش کاهش یافته تا به جایی رسیده که دیگر قادر به کاهش آن نبوده و تقریبا تایت مانده است. به همین صورت دقت از ۶۰ درصد تا ۹۸ درصد افزایش یافته و ثابت شده است. بدون دیدن نتایج بر داده تست نمی‌توان مدل را ارزیابی کرد چرا که ممکن است در شرایطی مدل تنها به داده‌های آموزش عملکرد خوبی داشته باشد. با اضافه شدن نتایج بر داده تست مدل قابل ارزیابی است. در این جا مشاهده می‌شود که خطا و دقت تست و آموزش تقریبا با هم پیش رفته است که نشان می‌دهد آموزش به درستی صورت گرفته است.

۴. نرمال سازی

روش‌ها مختلفی برای نرمال‌سازی وجود دارد. نرمال‌سازی یک فرایند مهم در پردازش داده‌ها و آمار می‌باشد که به منظور تبدیل متغیرها به یک مقیاس مشترک و یا توزیع مشخص انجام می‌شود. این کار معمولاً برای بهبود عملکرد الگوریتم‌های یادگیری ماشین، استخراج ویژگی‌ها، و انجام تحلیل آماری استفاده می‌شود. در زیر دو روش معمول نرمال‌سازی روش Min-Max و روش نرمال‌سازی با میانگین صفر و واریانس یک است.

در این روش، مقادیر داده‌ها به یک بازه مشخص که بازه معمولاً از ۰ تا ۱ است تبدیل می‌شود. فرمول نرمال‌سازی بدین گونه است که کمترین داده از همه داده‌ها کم شده و سپس داده‌ها بر بازه وجود داده‌ها (حداکثر منهای حداقل) تقسیم می‌شوند.

در نرمال‌سازی میانگین صفر و واریانس یک، میانگین داده‌ها به ۰ تبدیل می‌شود و واریانس به ۱. این روش باعث می‌شود تا تمام متغیرها به یک توزیع نرمال با میانگین صفر و واریانس یک تبدیل شوند.

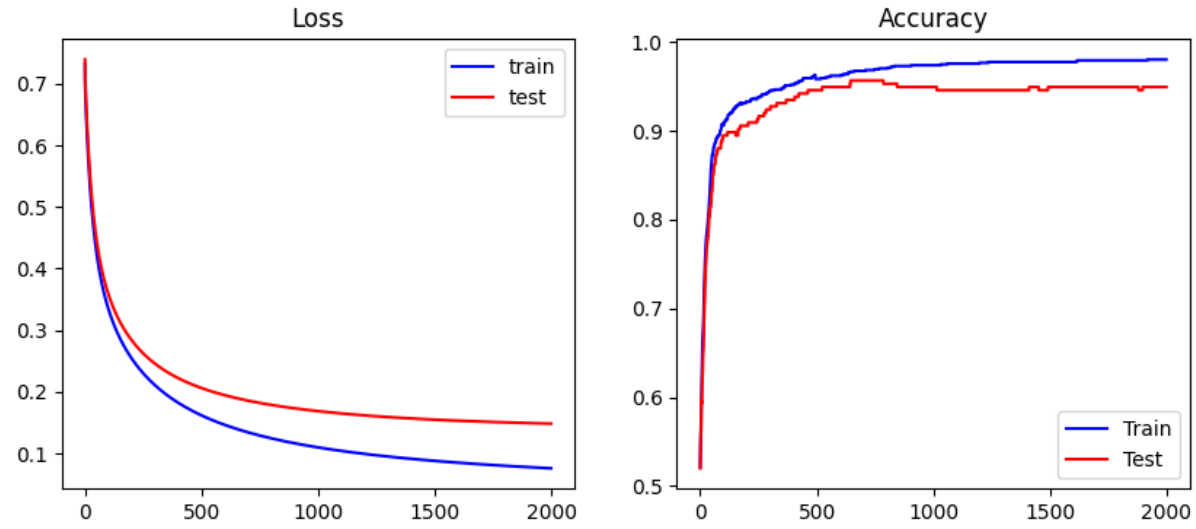
در این پروژه نرمال سازی به روش حداقل حداکثر با استفاده از کتابخانه و همچنین بدون استفاده از آن نیز پیاده سازی شده است.

نیاز است که فرایند نرمال سازی پس از تقسیم داده ها به آموزش و تست صورت بگیرد چرا که داده‌های تست نباید توسط مدل دیده شوند و با اعمال ویژگی آن‌ها در فرایند نرمال سازی، برخی ویژگی ها وارد مدل می‌شود و ارزیابی صورت گرفته دست نخواهد بود.

۵. نتایج پس از نرمال سازی

نتایج حاصل در شکل ۶ قابل مشاهده است. مقدار دقت نهایی آن بر داده تست حدود ۹۴ درصد بدست آمده است. پیش بینی برای ۵ نمونه رندم از تست نیز آمده است.

```
[['prediction' 'label']
 ['0' '0.0']
 ['1' '1.0']
 ['1' '0.0']
 ['1' '1.0']
 ['0' '0.0']]
```



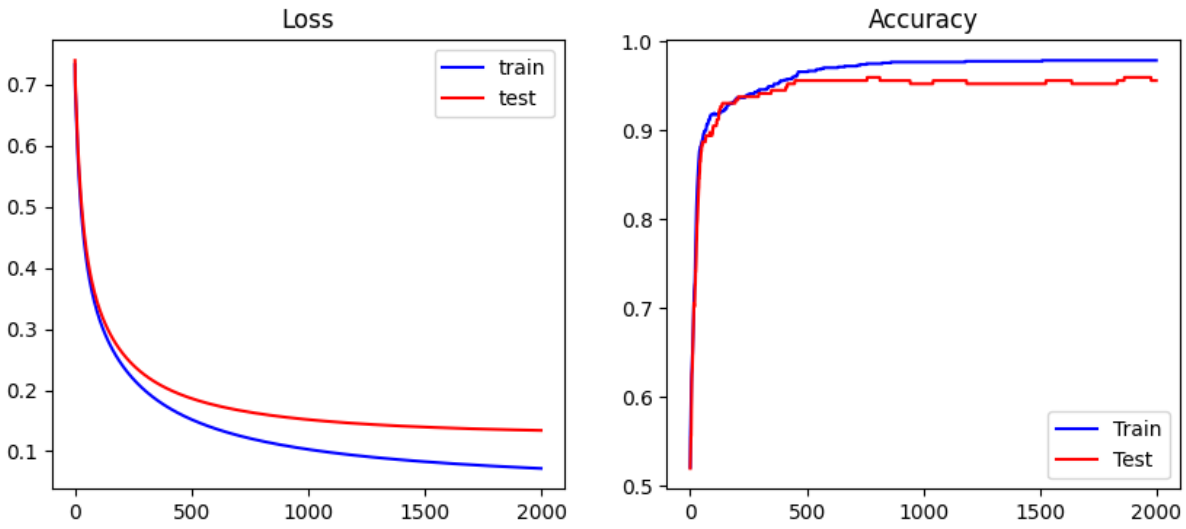
شکل ۶) نتایج پس از نرمال سازی

۶. تعادل کلاس‌ها

با گرفتن تعداد داده با برچسب یک و صفر نشان داده شد که تعداد داده در یک دسته برابر با ۷۶۲ و در دسته دیگر برابر با ۶۱۰ است که نشان می‌دهد تعادل کامل بین دو کلاس وجود ندارد. عدم تعدل می‌تواند به مواردی مانند کاهش دقت در کلاس با تعداد کمتر، بیش برازش به کلاس با تعداد بیشتر و غیر معتبر بودن دقت بدست آمده شود.

به منظور جلوگیری از این مورد می‌توان خطا را به صورت وزن‌دار اضافه کرد و یا در صورت وجود داده به مقدار کافی تعدادی داده از کلاس اکثریت را حذف و یا تعدادی داده در کلاس اقلیت را تکرار کرد. با افزودن وزن به داده‌های کلاس‌های کمتر تعداد، می‌توان تأثیر آنها را در فرآیند یادگیری بیشتر کرد. به این ترتیب، مدل بیشتر به داده‌های کمتر توجه می‌کند و این می‌تواند بهبود قابل توجهی در دقت و یادگیری مدل ایجاد کند. اقداماتی مانند حذف تعدادی از داده‌های کلاس اکثریت یا افزودن نمونه‌های تکمیلی به کلاس اقلیت باعث می‌شود تا تعادل بیشتری در دیتاست ایجاد شود و مدل بهتری بتواند بر روی آن آموزش ببیند.

در این پروژه با بازنویسی تابع گرادیان بر روی گرادیان حساب شده وزن اعلال شده است تا کلاس با داده کمتر خطای بیشتر تولید و در فرایند گرادیان نزولی تاثیری متناسب با کلاس دیگر داشته باشد. دقت نهایی در تست حدود ۹۵ درصد بدست آمده و نمودار های مربوطه در شکل ۷ آمده است.



۷) نتایج پس از ایجاد تعادل

۷. ایجاد تعادل با کتابخانه

مشابه با بخش یک از یک مدل آماده استفاده شده است با این تفاوت که با تعریف وزن‌ها برای هر کلاس تعادل بین دو کلاس ایجاد شده است. با توجه به تعداد داده ها در هر کلاس وزن کلاس اکثریت ۱ در نظر گرفته شده اشت. حال برای ایجاد تعدل، وزن گروه با تعداد اقلیت را بر تعداد آنها تقسیم کرده و سپس در تعداد کلاس دیگر ضرب شده است. دقت نهایی حال در آموزش و تست برابر با ۹۷.۷۵ درصد است.

بخش سوم

مقدمه

در این قسمت علاوه بر آموزش توسط کتابخانه به مواردی مثل خواندن و تقسیم داده‌ها و آماده سازی آن‌ها برای آموزش نیز اشاره شده است. سعی بر آن شده که نمودار اتلاف در هنگام استفاده از کتابخوانه رسم و همچنین تعدادی شاحص برای ارزیابی مدل معرفی شده‌اند.

۱. داده‌های مورد بررسی

مرکز کنترل و پیشگیری از بیماری‌ها (CDC) سه عامل اصلی را به عنوان عوامل خطر برای بیماری قلبی شناخته است این سه عامل فشار خون بالا، کلسترول خون بالا و سیگار کشیدن است. موسسه ملی قلب، ریه و خون، عوامل گسترده‌تری مانند سن، محیط و شغل، تاریخچه خانوادگی و ژنتیک، عادات زندگی، شرایط پزشکی دیگر، نژاد یا اقلیت‌ها و جنس را برای پزشکان برجسته کرده است تا در تشخیص بیماری عروق کرونر از آنها استفاده کنند. تشخیص معمولاً از طریق یک نظرسنجی اولیه از این عوامل خطر رایج آغاز شده و سپس با انجام آزمایش خون و سایر آزمون‌ها ادامه می‌یابد.

این مجموعه داده شامل ویژگی‌های گفته شده است که مجموعا ۲۱ ویژگی وجود دارد و برای پیش بینی بیماری با استفاده از مدل یادگیری ماشین استفاده شده است. سطون HeartDiseaseorAttack مربوط به برچسب داده‌ها است. این مجموعه داده شامل ۲۵۳،۶۸۰ پاسخ نظرسنجی است که هدف برای دسته‌بندی دو کلاسه بیماری قلب استفاده می‌شد. در این مجموعه داده تعادل قوی در کلاس‌ها وجود ندارد. موارد مورد بررسی آن است که در چه میزان می‌توان از پاسخ‌های نظرسنجی برای پیش‌بینی خطر بیماری قلب استفاده کرد و آیا می‌توان از زیرمجموعه‌ای از سوالات برای پیشگیری در برابر بیماری‌هایی مانند بیماری قلب استفاده کرد.

۲. دسته بندی داده‌ها

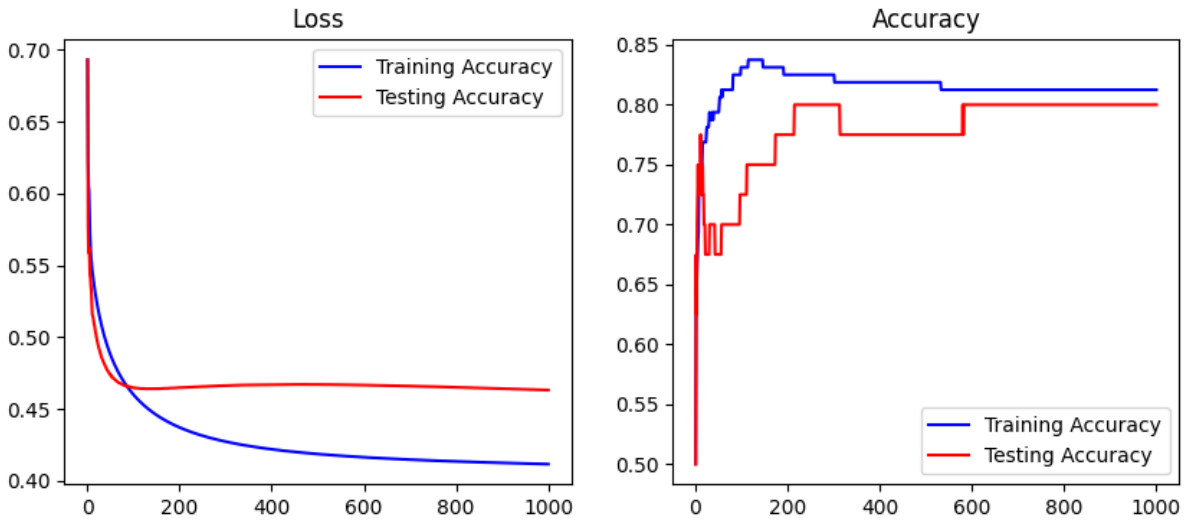
۱۰۰ نمونه از هر دسته به صورت رندم انتخاب شده و در یک دیتافریم جدید زخیره شده است. این کار در سه مرحله انجام شده است. ابتدا با فراخوانی تابع sample بر روی دایتافریم، ۱۰۰ نمونه از نمونه‌های سالم و ۱۰۰ نمونه بیمار به صورت رندوم انتخاب شده اند. سپس این دو به هم چسبانده شده‌اند و در نهایت داده‌ها درهم شده‌اند. در نهایت مجموعه داده‌بدست آمده را به داده آموزش و تست به نسبت ۸۰ و ۲۰ درصد تقسیم شده است.

۳. آموزش و ارزیابی

در این مرحله مشابه با بخش اول از مدل‌های آماده کتابخانه استفاده شده است. که نتایج حالصل نشان میدهد که بین ۶۷ تا ۸۰ درصد دقت در تست در این تقسیم بندی وجود دارد که عدد قابل قبولی است.

۴. نمایش تابع اتلاف

به طور کلی تابع اتلاف در زمان آموزش مدل را در هنگام آموزش با استفاده از کتابخانه آماده نمی‌توان بدست آورد و تنها محاسبه خطا پس از اتمام روند آموزش امکان پذیر است. اگر بخواهیم تابع اتلاف در طول آموزش را رسم کنیم، با توجه به ثابت بودن روند آموزش، می‌توان تعداد دورهای آموزش را از ۱ تا تعداد دور نهایی (برای مثال ۲۰۰۰) را رد یک حلقه تکرار کرد و اتلاف و دقت و مقادیر موررد نظر در آن را در یک لیست زخیره و در پایان آن را رسم کنیم. شکل ۸ نتایج حاصل از آموزش با رگرسیون لجستیک را نشان می‌دهد.



شکل ۸) نمودار اتلاف و دقت در هنگام آموزش در هنگام استفاده از کتابخانه آماده

۵. شاخص‌های دیگر ارزیابی

شاخص‌های ریکال (Recall) و پرسیژن (Precision) دو شاخص مهم در ارزیابی عملکرد مدل‌های کلاس‌بندی هستند. این شاخص‌ها بر مبنای ماتریس درهم‌سازی (Confusion Matrix) محاسبه می‌شوند.

Recall یا Sensitivity یا True Positive Rate در واقع نسبت تعداد نمونه‌های واقعی مثبتی که مدل صحیح تشخیص داده است به کل تعداد نمونه‌های واقعی مثبت در دیتاست و نشان می‌دهد که چه تعداد از نمونه‌های واقعی مثبت توسط مدل شناسایی شده‌اند و نسبت آن به تعداد نمونه‌های واقعی مثبت محاسبه می‌شود.

Precision یا Positive Predictive Value نیز نسبت تعداد نمونه‌های واقعی مثبتی که مدل صحیح تشخیص داده است به تعداد کل نمونه‌های مثبتی که مدل تشخیص داده است. این شاخص نشان می‌دهد که از نمونه‌هایی که مدل به عنوان مثبت شناسایی کرده، چه مقدار واقعاً مثبت بوده‌اند و نسبت به تمام نمونه‌هایی که مدل به عنوان مثبت اعلام کرده محاسبه می‌شود.

این دو در کنار هم می‌توانند ارزیابی کامل‌تری از عملکرد یک مدل ارائه دهند. با توجه به پیاده سازی این موارد توسط کتابخانه آماده به راحتی می‌توان این موارد را محاسبه کرد

```
logistic regression

        Train set:
        accuracy:81.25%
        precision:80.49%
        recall:82.50%

        Test set:
        accuracy : 80.00%
        precision : 80.00%
        recall : 80.00%

precision

        Train set:
        accuracy:76.88%
        precision:75.29%
        recall:80.00%

        Test set:
        accuracy : 70.00%
        precision : 66.67%
        recall : 80.00%
```

مراجع

[1]. https://github.com/MJAHMADEE/MachineLearning2023