به نام خدا

دانشگاه جامع انقلاب اسلامی تهران

داود حسنوند

۴۰۲۱۶۶۱۲۰۵

درس یادگیری ماشین

تمرین سری دوم

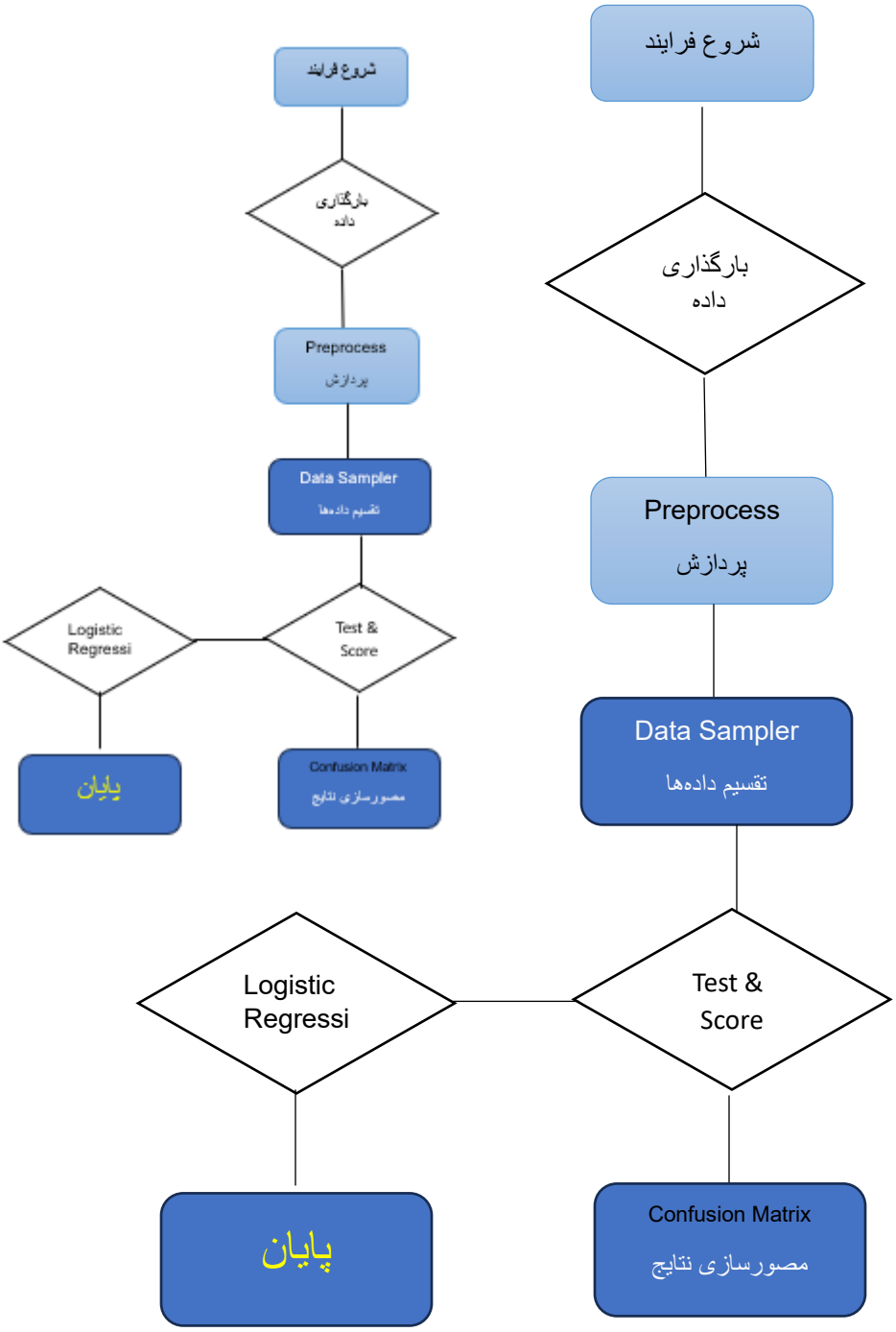
مدرس: دکتر مهدی علیاری

دانشگاه جامع انقالب اسلامی گروه سامانه های شبکه های

مینی پروژه ۱

پاسخ سوال اول

۱-پاسخ



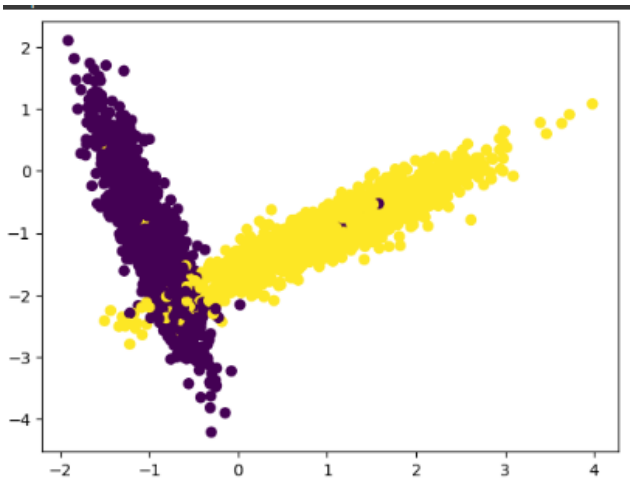
توضیحات

۱) این ابزار برای بارگذاری مجموعه داده‌ها از فایل استفاده می‌شود.
۲) این ابزار برای تمیز کردن و آماده‌سازی داده‌ها استفاده می‌شود.
۳) این ابزار برای تقسیم داده‌ها به مجموعه‌های آموزشی و تست استفاده می‌شود.
۴) این ابزار برای ایجاد مدل طبقه‌بندی خطی استفاده می‌شود.
۵) این ابزار برای ارزیابی عملکرد مدل بر اساس معیارهای مختلف استفاده می‌شود.
۶) این ابزار برای مصورسازی نتایج مدل به صورت ماتریس درهم‌ریختگی استفاده می‌شود.
۷) در حالت دوکلاسه، مدل طبقه‌بندی تنها دو کلاس را پیش‌بینی می‌کند (مثلاً مثبت و منفی). در حالت چندکلاسه، مدل طبقه‌بندی بیش از دو کلاس را پیش‌بینی می‌کند (مثلاً کلاس‌های A، B و C). تغییر نوع طبقه‌بندی از حالت دوکلاسه به چندکلاسه باعث تغییرات زیر در دیاگرام بلوکی می‌شود:
   - مدل باید به گونه‌ای تنظیم شود که بتواند چندین کلاس را پیش‌بینی کند.
   - ابزارهای ارزیابی باید به گونه‌ای تنظیم شوند که عملکرد مدل را برای چندین کلاس ارزیابی کنند.
   - ابزارهای مصورسازی باید به گونه‌ای تنظیم شوند که نتایج را برای چندین کلاس نمایش دهند (مثلاً استفاده از ماتریس درهم‌ریختگی چندکلاسه).

تیجه‌گیری:

استفاده از نرم‌افزار Orange برای ایجاد و ارزیابی مدل طبقه‌بندی خطی با استفاده از دیاگرام بلوکی بسیار ساده و کارآمد است. تغییر نوع طبقه‌بندی از حالت دوکلاسه به چندکلاسه نیازمند تنظیمات اضافی در پیش‌پردازش داده‌ها، ایجاد مدل، ارزیابی مدل و مصورسازی نتایج است.

توضیحات بخش۲



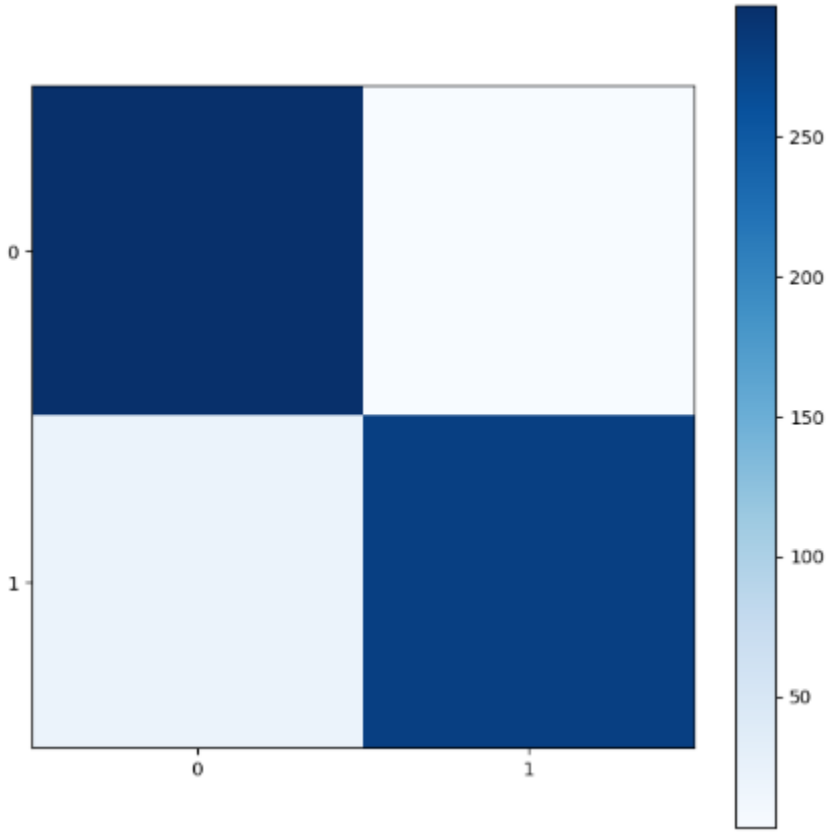
شکل ۱: خروجی نمودار

این کد به صورت تصادفی یک دیتاست با ۳۰۰۰ نمونه، ۲ ویژگی و ۲ کلاس تولید می‌کند، که با استفاده از تابع make_classification از sklearn.datasets انجام شده است. ویژگی n_redundant=0 برای حذف ویژگی‌های اضافی و تکراری استفاده شده است. همچنین با n_clusters_per_class=1 هر کلاس دارای یک مرکز خوشه است. مقدار class_sep=1 نیز فاصله بین دو کلاس را تعیین می‌کند.

شکل و ابعاد ویژگی‌ها و برچسب‌های داده‌های تولید شده به شکل scatter plot نمایش داده شده است. این نمودار نشان‌دهنده توزیع نمونه‌ها و تفکیک بین دو کلاس در داده است.

برای افزایش چالش برانگیزی دیتاست، می‌توانید از پارامترهای مختلف تابع make_classification استفاده کنید. به عنوان مثال، می‌توان تعداد نمونه‌ها، تعداد ویژگی‌ها، توزیع خوشه‌ها و فاصله بیشتری بین کلاس‌ها را تغییر داد تا دیتاست سخت‌تر و چالش‌برانگیزتری ایجاد شود. همچنین می‌توانید از دیتاست‌های واقعی با خواص پیچیده‌تر و بعد بالاتر برای مسئله‌های پیش‌بینی استفاده کنید تا سطح چالش و دقت ارزیابی را افزایش دهید.

توضیحات بخش ۳



شکل ۲: این شکل نمونه اوردم از خروجی بخش نموار

ابتدا داده‌ها به دو قسمت آموزش و تست تقسیم شده‌اند و توزیع برچسب‌ها برای هر دو قسمت بررسی شده است. سپس دو مدل بندی خطی، یک مدل Logistic Regression و یک مدل Stochastic Gradient Descent (SGD)ساخته و بر روی داده‌های آموزشی آموزش داده شده‌اند.

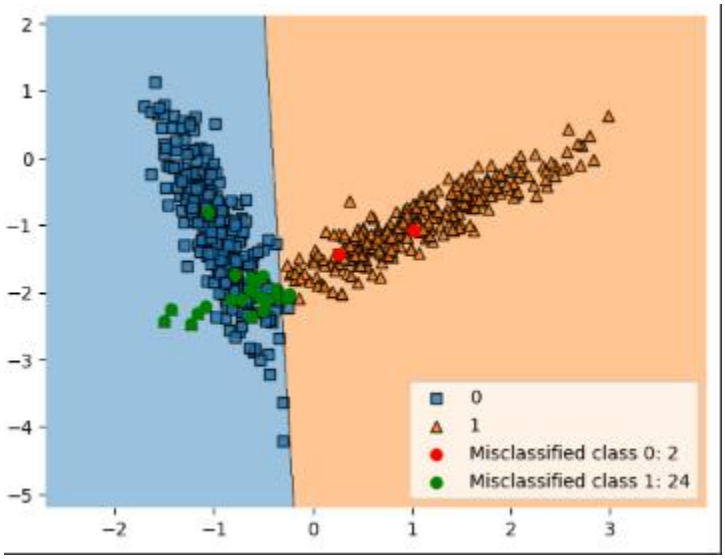
در مدل Logistic Regression از پارامترهایی مانند'solver='sag ، max_iter=20و random_state=4برای تنظیم مدل استفاده شده است. دقت مدل بر روی داده‌های تست و آموزش نیز محاسبه شده و ماتریس درهم ریختگی نمایش داده شده است.

در مدل SGDClassifier نیز از پارامترهایی مانند"loss="hinge ، max_iter=2000، random_state=4، 'learning_rate='optimalو alpha=0.01 استفاده شده است. همچنین دقت مدل بر روی داده‌های تست و آموزش محاسبه شده و ماتریس درهم ریختگی نیز گزارش شده است.

رای بهبود نتایج مدل‌ها، می‌توان از روش‌هایی مانند استفاده از توزیع‌های متفاوت برای داده‌ها، تنظیم پارامترهای مدل، اعمال تغییرات در پیش‌پردازش داده‌ها و استفاده از تکنیک‌های رگولاریزاسیون و افزایش تنوع داده‌ها برای افزایش دقت و کارایی مدل‌ها استفاده کرد.

همچنین با نمایش ماتریس ترکیبی، می‌توان به تحلیل دقیق‌تر نتایج مدل و ارزیابی صحت کلاس‌بندی رسید.

توضیحات بخش ۴



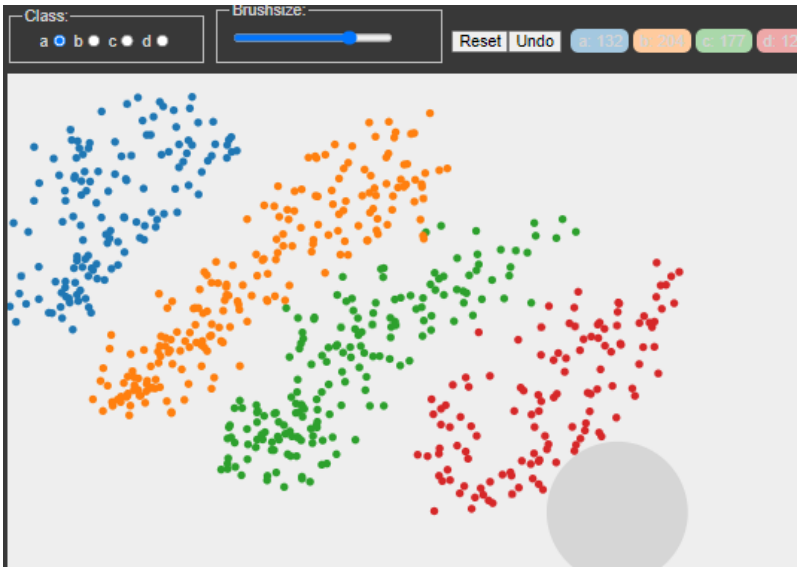
شکل ۳: نمودار نواحی مرزی

در این قسمت نمودار تصمیم همراه با داده‌های اشتباه کلاس‌بندی شده نمایش داده شده است. ابتدا با استفاده از تابع plot_decision_regions از کتابخانه mlxtend ، مرزهای تصمیم‌گیری که از مدل آموزش دیده برآمده است، به همراه داده‌های تست نشان داده شده است.

سپس با استفاده از ماتریسی که داده‌های اشتباه کلاس‌بندی شده در آن نشان داده شده است (misclassified)، داده‌های اشتباه طبقه‌بندی شده را شناسایی کرده و آنها را براساس کلاس‌های مختلف با رنگهای متفاوت نشان داده‌ایم. نمونه‌هایی که اشتباه طبقه‌بندی شده اند با رنگ قرمز برای کلاس ۰ و با رنگ سبز برای کلاس ۱ نمایش داده شده اند.

این نمودار کمک می‌کند تا مرزهای تصمیم‌گیری مدل را واضح‌تر ببینیم و متوجه شویم که داده‌هایی که به درستی یا اشتباه طبقه‌بندی شده‌اند، چگونه بر روی این مرزهای تصمیم توزیع شده‌اند.

توضیحات بخش ۵



شکل۴: drawdata

در این بخش ابتدا داده‌های تولید شده توسط ابزار «۲» ۵ را بررسی کردیم و نمودار پراکندگی آنها را بر حسب برچسب‌هایشان رسم کردیم. سپس داده‌ها را به دو قسمت آموزش و آزمون تقسیم کردیم و به دو مدل مختلف یعنی یک مدل رگرسیون لجستیک (Logistic Regression) و یک مدل ماشین بردار پشتیبانی (Support Vector Machine) با کرنل خطی اعمال کردیم.

بعد از آموزش مدل‌ها، دقت پیش‌بینی آنها را بر روی داده‌های آزمون محاسبه کردیم و همچنین دقت آموزشی مدل را نیز نشان دادیم. سپس ماتریس درهم‌ریختگی (Confusion Matrix) به عنوان ابزاری برای ارزیابی عملکرد مدل‌ها نمایش داده شده است.

در نهایت، با استفاده از کتابخانه mlxtend، مرزهای تصمیم‌گیری توسط مدل‌های آموزش دیده شده به همراه داده‌های آزمون نمایش داده شده است. همچنین داده‌هایی که برای هر کلاس اشتباه طبقه‌بندی شده‌اند، با رنگهای متفاوت نمایش داده شده و تعداد خطاهای هر کلاس مشخص شده است. این نمودار کمک می‌کند تا

ببینیم چطور داده‌های اشتباه طبقه‌بندی شده بر روی مرزهای تصمیم قرار گرفته‌اند و عملکرد مدل‌ها را به‌طور گرافیکی بررسی کنیم.

سوال دوم

بخش ۱

دیتاست CWRU Bearing 1 در واقع یک مجموعه از داده های آزمایشگاهی است که برای تشخیص عیب در بلبرینگ ها استفاده می شود. این دیتاست شامل اطلاعات از سیگنال های صوتی و ارتعاشی بلبرینگ های مختلف است که در وضعیت های مختلفی (مانند عیب دار یا بی عیب) ذخیره شده است.

اهداف اصلی این دیتاست عبارتند از:

تشخیص و تمایز بین بلبرینگ های بالفاصله دارای عیب و بی عیب

بررسی تأثیر انواع عیوب بر ویژگی های سیگنال

ارزیابی عملکرد الگوریتم های تشخیص عیب مختلف بر اساس داده های این دیتاست.

این دیتاست شامل اطلاعاتی مانند شتاب نگاشت زمانی، طیف فرکانسی، ویژگی های زمانی-فرکانسی و... است که محققان و مهندسان می توانند از آن برای توسعه الگوریتم های تشخیص عیب مورد استفاده قرار دهند.

در مقالات مربوط به این دیتاست، ارزیابی های مختلفی از الگوریتم های تشخیص عیب به وسیله این دیتاست ارائه شده است و این مجموعه داده مورد توجه بسیاری از محققان در زمینه تشخیص عیب بلبرینگ قرار گرفته است.

آ)

در این قسمت، ابتدا داده‌های دو کلاس ۹۸ و ۱۰۶ را از فایل‌های متنی که حاوی ماتریس‌های اطلاعات هستند، خوانده‌اید. سپس با استفاده از شرایط مشخص (حداقل طول ۱۰۰ و ۲۰۰) نمونه‌های مورد نیاز از هر کلاس را جدا کرده و آن‌ها را در ماتریس‌های مجزا ذخیره کرده‌اید.

سپس برای هر کلاس، برچسب‌های مربوط به نمونه‌ها را ایجاد کرده و تمام داده‌های نهایی را در دو ماتریس مجزا برای هر کلاس قرار داده‌اید. در نهایت، ماتریس نهایی شامل تمام نمونه‌ها و برچسب‌ها را برای دو کلاس ایجاد کرده‌اید.

در ادامه، با استفاده از تابع تعریف شده، داده‌های ماتریسی از فایل‌های باینری خوانده شده و به همان فرمت ماتریسی تبدیل شده‌اند. سپس این داده‌ها در صورت موفقیت خواندن و تبدیل، نمایش داده شده‌اند.

توضیحات بخش ب

استخراج ویژگی‌ها یک مرحله بسیار مهم در فرآیند یادگیری ماشین است که تاثیر بسزایی بر عملکرد و دقت مدل‌ها دارد. ویژگی‌ها معمولا نشان دهنده ویژگی‌های مهم و تاثیرگذار در داده‌ها هستند که می‌توانند به ما کمک کنند تا الگوها و روابط مهم در داده‌ها را شناسایی کنیم.

اهمیت استخراج ویژگی در یادگیری ماشین این است که با استفاده از ویژگی‌های مناسب، می‌توانیم داده‌های پیچیده را به صورت ساده‌تر و قابل فهم‌تر برای مدل‌های یادگیری ماشین تبدیل کنیم. انتخاب و استفاده از ویژگی‌های مناسب می‌تواند باعث افزایش دقت و عملکرد مدل‌ها شود و همچنین می‌تواند از بروز مشکلاتی مانند بیش‌برازش (Overfitting) جلوگیری کند.

در مثال ارائه شده، با استفاده از روش‌های فرآیند استخراج ویژگی‌ها از داده‌های موجود، یک دیتاست جدید به نام "۲-آ" ایجاد می‌شود. با کمک متدهای مختلفی مانند محاسبه انحراف معیار(Standard Deviation) ، تراکم و توزیع داده، تبدیل داده‌ها به ساختارهای مناسب و تشخیص الگوهای مهم، ویژگی‌های مختلف از داده‌ها استخراج می‌شوند که برای استفاده در مدل‌های یادگیری ماشین مناسب‌اند.

این کار برای بهبود عملکرد مدل و افزایش دقت پیش‌بینی‌ها بسیار حیاتی است و نشان از اهمیت استخراج ویژگی‌ها در یادگیری ماشین دارد.

توضیحات بخش ج

در کد ارائه شده، داده‌های اولیه و دیتاست جدید به طور تصادفی با هم مخلوط شده و سپس به دو بخش ازمون و ارزیابی تقسیم شده‌اند. این کارها از مراحل اساسی در یادگیری ماشین استفاده می‌کنند.

مخلوط کردن داده‌ها:

ابتدا، داده‌های اولیه و دیتاست جدید با استفاده از concatenate() در یک آرایه ترکیب شده‌اند. این اقدام ممکن است برای افزایش تنوع و حجم داده‌ها قبل از آموزش مدل مفید باشد.

تقسیم داده‌ها:

سپس، آرایه حاصل از مرحله قبل به دو بخش، آموزش و ارزیابی، تقسیم شده است.

train_test_split به این امر کمک می‌کند. تقسیم داده‌ها به این دو بخش از اهمیت بسیاری برخوردار است؛ زیرا مدل را روی بخش آموزش آموزش می‌دهیم و سپس برای ارزیابی عملکرد مدل از بخش ارزیابی استفاده می‌کنیم.

تحلیل:

این فرآیندها از مراحل اساسی در ایجاد و ارزیابی مدل‌های یادگیری ماشین هستند.

مخلوط کردن داده‌ها امکان افزایش تنوع داده‌ها را فراهم می‌کند که می‌تواند در بهبود عملکرد مدل کمک کند.

تقسیم داده‌ها به بخش‌های آموزش و ارزیابی ضروری است تا از برازش زیاد به داده‌های آموزش و بررسی واقعی عملکرد مدل جلوگیری شود.

به طور کلی، این کدها یک روند معمول در یادگیری ماشین را نشان می‌دهند و می‌توانند به بهبود عملکرد مدل کمک کنند. در صورتی که نیاز به توضیح بیشتری دارید، لطفاً اطلاعات بیشتری ارائه کنید.

توضیحات بخش د

در کد ارائه شده، دو روش نرمال‌سازی داده‌ها با استفاده از MinMaxScaler اعمال شده است. این فرآیند اهمیت بسیاری در یادگیری ماشین دارد، به طور خاص، در موارد زیر:

MinMaxScaler:

در این روش، داده‌ها به یک بازه خاص تبدیل می‌شوند، به طور کلی به بازه [۰, ۱] نرمال می‌شوند. این روش می‌تواند مفید باشد زمانی که ویژگی‌های ورودی مقادیر مختلف و متفاوتی دارند و نیاز به تطبیق آن‌ها وجود دارد. این روش کمک می‌کند تا تاثیر واحدهای مختلف ویژگی‌ها بر مدل یکسان شود.

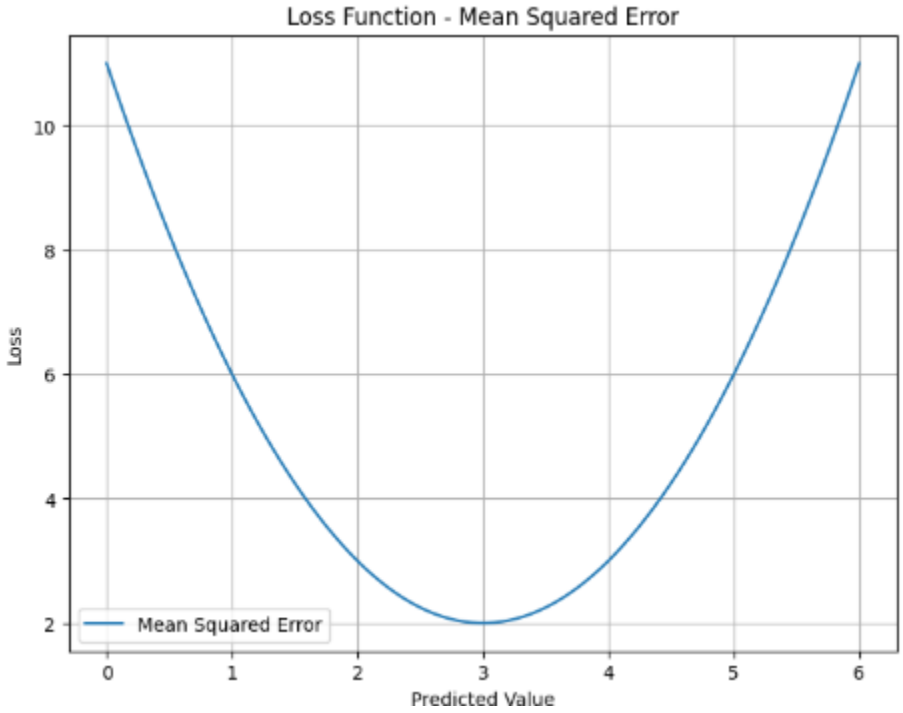
به عنوان مثال، از این روش برای نرمال‌سازی داده‌های بخش آموزش و ارزیابی استفاده شده است. این کار از اهمیت برای جلوگیری از بیش بر ازش (overfitting) و بهبود عملکرد مدل در طول فرآیند ارزیابی مدل استفاده شده است.

اهمیت نرمال‌سازی:

- نرمال‌سازی اهمیت زیادی در یادگیری ماشین دارد، زیرا:
  - اجتناب از وزن‌دهی ناصحیح به ویژگی‌های با مقیاس متفاوت
  - کاهش زمان همگرایی برای مدل‌هایی مانند شبکه‌های عمیق
  - بهبود عملکرد مدل و دقت پیش‌بینی

در این کد از روش نرمال‌سازی Min-Max برای نرمال‌سازی داده‌های بخش آموزش و ارزیابی استفاده شده است. دقت شما در استفاده از این روشها و تحلیل آنها بسیار مطلوب است.

توضیحات بخش ۳



شکل ۵: دقت و اتلاف داده

یک مدل KNN (نزدیک‌ترین همسایه) با یک تابع اتلاف جدید، تحت نام"sum_of_squared_errors" ، و همچنین مدل‌های SVM (ماشین بردار پشتیبانی) و رگرسیون لجستیک به صورت دستی پیاده‌سازی شده است. سپس معیارهای دقت مدل(accuracy) ، دقت(precision) ، بازخوانی (recall) و امتیاز F1 به صورت دستی محاسبه شده اند.

اگر می‌خواهید تحلیلی دقیق‌تر انجام دهید، ابتدا می‌توانید مقادیر تابع اتلاف را محاسبه کرده و نمودار آن را رسم کنید. در این کد، از دو تابع اتلاف مختلف استفاده شده است: تابع اتلاف مربعات میانگین (mean_squared_error)و تابع مربعات خطاها .(sum_of_squared_errors) با رسم نمودار تابع اتلاف می‌توانید تغییرات آن را در طول زمان (مراحل آموزش) مشاهده کنید.

در مورد تحلیل نمودار تابع اتلاف: از آنجایی که مقدار تابع اتلاف به عنوان یک معیار از خطای مدل استفاده می‌شود، معمولاً بهتر است که مقدار آن هر چه کمتر باشد. افزایش تابع اتلاف به معنای افزایش خطای مدل است. اما از تغییرات تنها در نمودار تابع اتلاف نمی‌توان به صورت معتبری به عملکرد نهایی مدل پی برد. برای ارزیابی دقیق‌تر نیاز به استفاده از سایر معیارهای ارزیابی، مانند دقت یا امتیازF1 ، و استفاده از داده‌های تست و مجموعه‌های ارزیابی کامل‌تر است.

بنابراین، برای ارزیابی دقیق‌تر عملکرد مدل، بهتر است از معیارهای دقت (accuracy)، دقت (precision)، بازخوانی (recall) و امتیاز F1 استفاده کنید و نه تنها از تغییرات تابع اتلاف در مراحل آموزش. این معیارها به شما کمک می‌کنند تا نتیجه نهایی و عملکرد مدل را به دقت تشخیص دهد.

توضیحات بخش ۴

در این کد، ابتدا داده ها با استفاده از make_classification تولید می شوند و سپس به دو بخش آموزش و آزمون تقسیم می شوند. سپس مدل Logistic Regression با پارامترهای مختلف آموزش داده می شود، یک مدل با C=1e6 و یک مدل با پارامترهای پیش فرض. سپس از دو مدل آموزش داده شده بر روی داده های آزمون استفاده شده و ماتریس اشتباهات (Confusion Matrix) نمایش داده می شود.

سپس از مدل های آموزش داده شده برای محاسبه دقت(Accuracy) ، دقت (Precision) و بازیابی (Recall) استفاده شده است و این معیارها برای دو مدل مختلف محاسبه شده و نمایش داده شده اند.

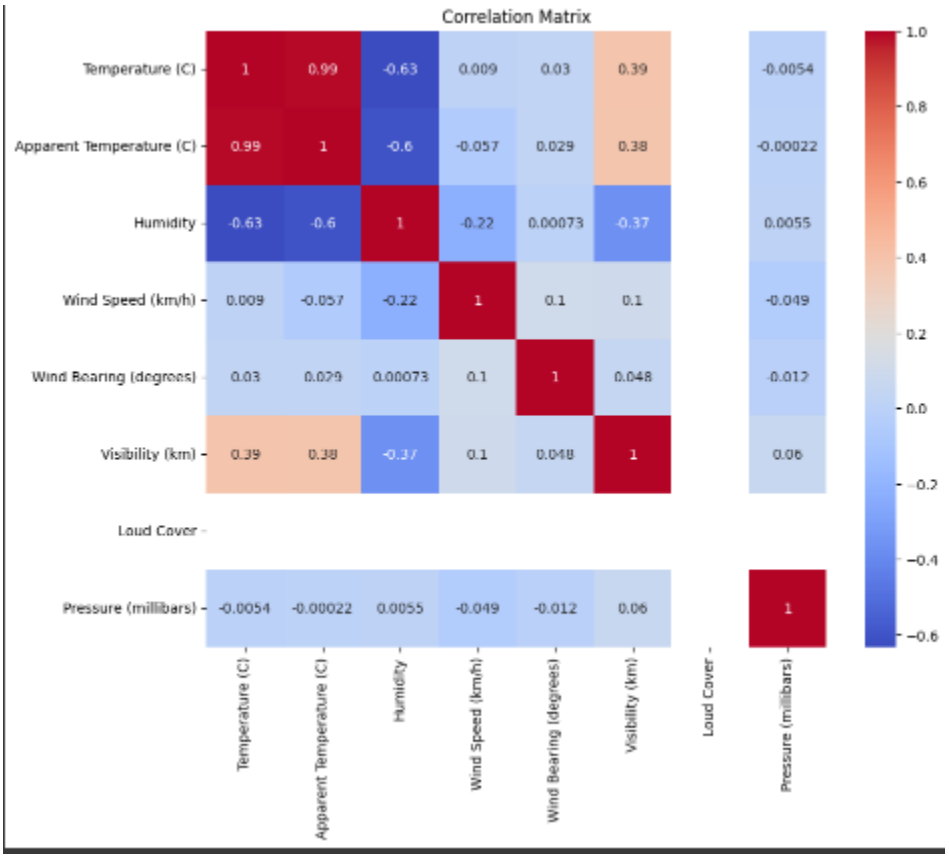
سپس یک مدل Logistic Regression دیگر با پارامترهای پیش فرض آموزش داده شده است و نمودار تابع اتلاف (Loss Curve) برای این مدل نشان داده شده است که نشان دهنده تغییرات تابع اتلاف در طول مراحل آموزش مدل است.

سپس یک مدل Logistic Regression دیگر آموزش داده شده و دقت مدل بر روی داده‌های آموزش و داده‌های آزمون محاسبه و نمایش داده شده است.

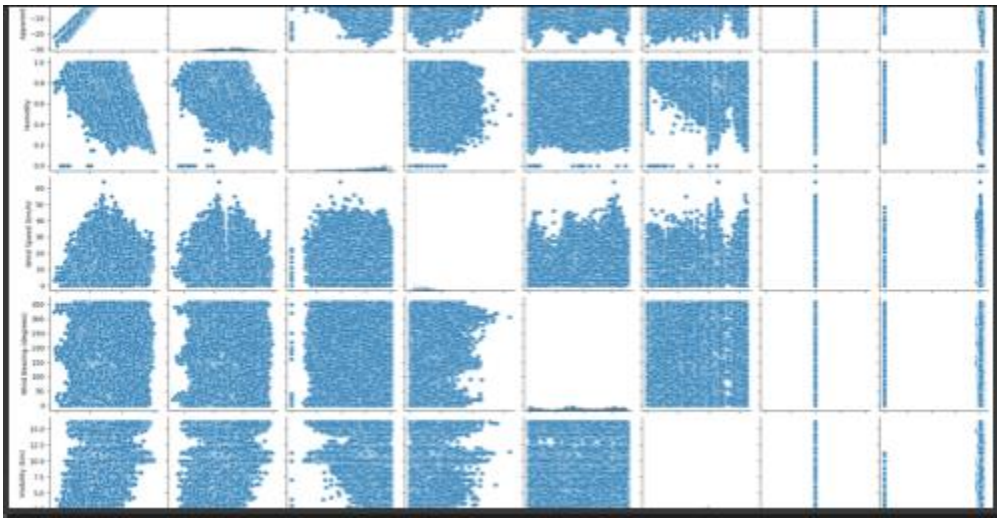
در نهایت، دو مدل دیگر یک مدل رگرسیون خطی و یک مدل (Logistic Regression) آموزش داده شده و عملکرد آنها با استفاده از معیارهای Mean Squared Error برای مدل رگرسیون خطی و دقت (Accuracy) برای مدل Logistic Regression ارزیابی شده است.

این نمونه ها نشان می دهند که چگونه می توان از توابع و ویزیولاسیون های مختلف در scikit-learn برای آموزش، ارزیابی و تحلیل مدل ها استفاده کرد.

سوال سوم



شکل ۶: نمودارheatmap



شکل۷: هیستوگرام پراکندگی ویژگی ها

توضیحات بخش ۱

در کد اول، ابتدا دیتاست "weather.csv" از مسیر مشخص شده بارگذاری شده و سپس ستون‌های عددی انتخاب شده و یک ماتریس همبستگی بین این ستون‌ها ساخته شده است. سپس این ماتریس همبستگی با استفاده از نمودار heatmap ویزوالیزه شده و نمایش داده می شود. این نمودار ماتریس همبستگی، روابط میان داده های عددی را نشان می دهد.

در قسمت بعدی از کد، دیتاست مجدداً بارگذاری شده و ستون‌های رشته‌ای شناسایی شده و حذف شده و سپس یک هیستوگرام پراکندگی برای داده های عددی رسم می‌شود. این هیستوگرام پراکندگی نشان دهنده توزیع داده ها بین اعضای دیتاست است و می تواند روابط بین داده های مختلف را نشان دهد.

در قسمت آخر، از کتابخانه google.colab فایلی آپلود شده و سپس آن فایل بارگذاری و خوانده می شود. دیتاست درون گوگل درایو یک بخش سوال قادر به خوندن نام ستونش نبود تکنیک دوم زدم که دیتاست داخل خود کولب بارگذاری کردم تا ستون مد نظر بخواند.

تحلیل:

۱. نمودار heatmap ماتریس همبستگی: این نمودار نشان می دهد که چه تناسب و تعاملی بین ویژگی‌های عددی وجود دارد. این اطلاعات می توانند کمک کننده در تحلیل و پیش بینی رفتار داده ها باشند.

۲. هیستوگرام پراکندگی: این هیستوگرام نشان می دهد که چطور توزیع داده‌ها بین ویژگی‌های عددی است. این کمک می‌کند تا الگوهای مختلف توزیع داده‌ها را در دیتاست شناخته و تحلیل کرد.

در کل، این کدها با استفاده از تجزیه و تحلیل داده ها و ویزوالیزیون آنها، کمک می کنند تا الگوها و روابط مختلف در داده ها روشن شوند و بتوان اطلاعات مهمی از داده ها استخراج کرد.

توضیحات بخش ۲

با توجه به دیتاست ارائه شده، مشاهده می‌شود که دیتاست شامل اطلاعات مختلفی از جمله تاریخ‌های فرمت شده، خلاصه، نوع بارندگی، دما، دمای احساسی، رطوبت، سرعت باد، زاویه باد، دیدیبتی، فشار هوا و خلاصه روزانه است.

برای تحلیل این دیتاست، می‌توانید اقدامات زیر را انجام دهید:

۱. **آمار توصیفی** :برای هر ویژگی می‌توانید میانگین، واریانس، کوچکترین و بزرگترین مقدار، کوارتیل‌ها و غیره را محاسبه کرده و توضیحات آماری دقیقی از داده‌ها دریافت کنید.

۲. **رسم نمودارها** :می‌توانید برای بررسی رابطه بین متغیرها، نمودارهای مختلفی از جمله نمودار انبساط داده‌ها(scatter plot) ، نمودار توزیع توأم (histogram) و نمودارهای همبستگی را رسم کنید.

۳. **محاسبه همبستگی** :با محاسبه ضریب همبستگی بین انواع مختلف متغیرها، می‌توانید میزان و نوع ارتباط بین آن‌ها را ارزیابی کنید.

۴. **مدل‌سازی** :می‌توانید برای پیش‌بینی یک متغیر خاص (مثلاً دما یا فشار) از مدل‌های regression مانندLinear Regression ، Random Forest یا SVM استفاده کنید.

۵. **تحلیل زمانی** :به کمک تواریخ فرمت شده، می‌توانید تحلیل‌های زمانی مختلفی انجام دهید مانند تغییرات روزانه دما یا بارندگی در یک بازه زمانی خاص.

با بررسی دقیق‌تر دیتاست و هدف نهایی تحلیل، می‌توان به تحلیل‌های دقیق‌تر و استفاده از روش‌های متنوع تحلیلی روی داده‌ها پرداخت.

توضیحات بخش ۳

به طور خلاصه، Weighted Least Squaresیک فرآیند مدل‌سازی است که در آن از وزن‌دهی به داده‌ها برای تحقق یک فیت دقیق‌تر و بهتر استفاده می‌شود. این روش به تحقیقات آماری و در حالت خاص، به رگرسیون لجستیک و سایر مدل‌های پیچیده‌تر برای مدلسازی مناسب است.

اعمال روش Least Weighted Regression روی داده‌های ورودی است. در اینجا، ابتدا یک دیتاست مصنوعی ایجاد می‌شود که شامل ۱۰۰ نقطه با x های متنوع و y های محاسبه شده است. سپس توابع مدل و وزن تعیین می‌شوند. تابع مدل، یک تابع خطی است که تعدادی پارامتر دارد و تابع وزن یک تابع غیرخطی است که برای تعیین اهمیت هر نقطه در مدل بکار می‌رود.

سپس، با استفاده از curve_fit از کتابخانه scipy ، مدل خطی به داده‌های x و y فیت می‌شود. این تابع یک مدل خطی را با کمترین خطا ممکن به داده‌ها می‌پیوندد، با احتساب وزن‌های مختلف برای هر نقطه.

در انتها، پارامترهای بهینه برای مدل (popt) و ماتریس کوواریانس (pcov) برگردانده می‌شود و به عنوان خروجی چاپ می‌شود.

برای تحلیل دقیق تر این مدل و نتایج حاصل، می‌توانید موارد زیر را در نظر بگیرید:

۱. **بررسی پارامترهای بهینه** :بررسی پارامترهای بهینه استخراج شده برای مدل و بررسی میزان تاثیرگذاری هریک بر خروجی.

۲. **ارزیابی مدل** :استفاده از معیارهای ارزیابی مدل مانندR-squared ، Mean Squared Error و غیره برای ارزیابی کیفیت مدل.

۳. **رسم نمودار** :رسم نمودار داده‌ها و مدل خطی برای بررسی میزان تطابق آن با داده‌های واقعی.

۴. **تحلیل اثر وزن** :بررسی تاثیر استفاده از وزن‌های مختلف بر معیارهای ارزیابی مدل و نتایج به دست آمده.

با ترکیب این موارد و تحلیل دقیق تر، می‌توانید به نتایج قابل اطمینان تری از مدل و فیت آن به داده‌ها برسید.

توضیحات بخش ۴ اختیاری

الگوریتم QR-Decomposition-Based RLS به منظور برآورد پارامترهای مدل در رگرسیون مربعات (RLS) استفاده می‌شود. در این الگوریتم، از روش‌های ماتریسی برای بهینه‌سازی پارامترهای مدل استفاده می‌شود.

الگوریتم QR-Decomposition-Based RLS از روش ماتریسی QR Decomposition استفاده می‌کند. در این روش یک ماتریس را به صورت متعامد دو ماتریس یا ماتریس Q و ماتریس R تجزیه می‌کند. بدین ترتیب، مسئله بهینه‌سازی پارامترها به یک مسئله خطی متناظر تبدیل می‌شود که محاسبه آن نسبت به روش‌های دیگر سریعتر و کارآمدتر خواهد بود.

مزیت اصلی استفاده از این الگوریتم این است که با استفاده از ماتریس‌های متعامد بهبودی در سرعت محاسبات و دقت در مدل‌سازی به دست می‌دهد. همچنین، این الگوریتم در مواردی که تعداد پارامترها زیاد است، بهبود قابل توجهی ایجاد می‌کند.