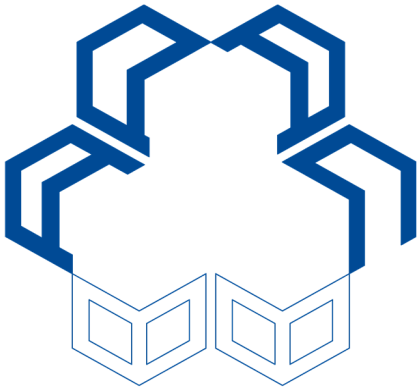


یادگیری ماشین

مینی پروژه اول

نام و نام خانوادگی	امیر جهانگرد | علیرضا امیری
شمارهٔ دانشجویی	۴۰۳۱۵۳۸۴ | ۴۰۲۰۲۴۱۴
تاریخ	فروردین ماه ۱۴۰۴

فهرست مطالب

۱ پیش بینی آب و هوا مبتنی بر یادگیری ماشین

لینک درایو گوگل کولب حاوی کدهای این تمرین

۱.۱ دادگان

۱.۱.۱

پیش‌بینی وضعیت آب‌وهوا یکی از کاربردهای مهم یادگیری ماشین است که نیاز به مجموعه داده‌های دقیق برای آموزش مدل‌های پیش‌بینی دارد. این گزارش، مروری بر مجموعه داده‌ای که در مقاله مورد بررسی برای پیش‌بینی آب‌وهوا استفاده شده است، ارائه می‌دهد.

مجموعه داده مورد استفاده در مقاله از پایگاه داده ارزیابی آب‌وهوای اروپا (ECA&D) استخراج شده است. این مجموعه شامل مشاهدات روزانه هواشناسی از شهرهای مختلف اروپایی در بازه زمانی ۲۰۰۰ تا ۲۰۱۰ می‌باشد. اطلاعات موجود در این مجموعه شامل موارد زیر است:

- دمای سطحی هوا
- میزان بارش روزانه
- سایر متغیرهای هواشناسی

این داده‌ها از ۱۸ شهر مختلف اروپایی شامل شهرهایی مانند بازل (سوئیس)، بوداپست (مجارستان)، درسدن، دوسلدورف، کاسل و مونیخ (آلمان)، و همچنین شهرهایی از هلند و بریتانیا جمع‌آوری شده‌اند.

این مجموعه داده برای آموزش مدل‌های یادگیری ماشین جهت پیش‌بینی شرایط آب‌وهوا بر اساس داده‌های تاریخی بسیار مهم است. استفاده از اطلاعات آب‌وهوایی از چندین منطقه مختلف، امکان توسعه مدل‌های پیش‌بینی دقیق‌تر و قابل‌اعتمادتر را فراهم می‌کند.

۲.۱.۱

در این بخش از تمرین، هدف این است که داده‌های مربوط به شهرهای فرانسه از مجموعه داده‌ی اصلی استخراج شده و داده‌های سایر شهرها حذف شوند. این کار برای متمرکز کردن تحلیل‌ها بر روی وضعیت آب‌وهوا در شهرهای خاص انجام می‌شود.

برای این بخش، ابتدا فایل اصلی داده‌ها که شامل اطلاعات هواشناسی از چندین شهر مختلف است، بارگذاری شد. این داده‌ها شامل ۱۶۵ ستون و ۳۶۵۴ ردیف است که هر ستون به اطلاعات مربوط به یک شهر خاص اختصاص دارد. برای استخراج داده‌های مربوط به شهرهای فرانسه، از نام ستون‌ها که شامل نام شهرها بودند، استفاده شد.

از آنجایی که در مجموعه داده اطلاعات مربوط به چندین شهر اروپایی وجود دارد، ابتدا شهرهای فرانسه از جمله تور (TOURS) شناسایی شدند. سایر شهرهای موجود در مجموعه داده که متعلق به فرانسه نبودند، از داده‌ها حذف شدند.

با اجرای کد پایتون، تنها داده‌های مربوط به شهرهای فرانسه استخراج و در یک فایل جدید ذخیره شدند. این فایل شامل داده‌های مربوط به متغیرهای جوی مختلف همچون سرعت باد، رطوبت، فشار هوا، دما و بارش است.

این فرآیند امکان تحلیل دقیق‌تری از وضعیت آب‌وهوا در شهرهای فرانسه را فراهم می‌آورد و داده‌های مربوط به سایر شهرها که برای این تحلیل ضروری نبودند، حذف شدند. این کار کمک می‌کند که مدل‌های پیش‌بینی آب‌وهوا به طور دقیق‌تر برای مناطق خاص تنظیم شوند.

۳.۱.۱

در این بخش به بررسی پیش‌پردازش‌هایی که بر روی داده‌های مقاله انجام شده است و همچنین اقداماتی که ما برای آماده‌سازی داده‌ها انجام داده‌ایم، پرداخته می‌شود.

پیش‌پردازش‌های انجام‌شده در مقاله:

طبق توضیحات مقاله، مراحل زیر برای آماده‌سازی داده‌ها انجام شده است:

- حذف داده‌های نامعتبر و گمشده: داده‌هایی که شامل مقدار Null یا مقادیر غیرمنطقی بودند حذف شده‌اند.
- نرمال‌سازی داده‌ها: متغیرهای عددی در یک بازه مشخص (مثلاً [۰،۱]) مقیاس‌بندی شده‌اند.
- ایجاد پنجره‌های زمانی: برای مدل‌های یادگیری ماشین، از داده‌های گذشته برای پیش‌بینی آینده استفاده شده است.
- تقسیم داده‌ها: مجموعه داده به دو بخش آموزش و آزمون تقسیم شده است.

بازه زمانی و تعداد نمونه‌ها:

مجموعه داده شامل اطلاعات از سال ۲۰۰۰ تا ۲۰۱۰ می‌باشد. تعداد کل نمونه‌های موجود در دیتاست برابر با ۳۶۵۴ رکورد است.

پیش‌پردازش‌های اعمال‌شده توسط ما:

برای بهبود کیفیت داده‌ها و آماده‌سازی آن‌ها برای مدل‌های یادگیری ماشین، مراحل زیر انجام شده است:

۱. تبدیل فرمت تاریخ: ستون DATE به فرمت datetime تغییر داده شد.

۲. حذف داده‌های نامعتبر: ردیف‌های دارای مقدار Null حذف شدند.

۳. نرمال‌سازی داده‌ها: تمام متغیرهای عددی با استفاده از روش Min-Max Scaling بین ۰ و ۱ مقیاس‌بندی شدند.

۴. تقسیم داده‌ها: داده‌ها به دو مجموعه‌ی آموزش (۸۰٪) و آزمون (۲۰٪) تقسیم شدند.

نتایج: پس از انجام این فرآیند:

- داده‌های نامعتبر حذف شدند.
- داده‌ها از نظر مقیاس نرمال شدند.
- فرمت تاریخ اصلاح شد.
- داده‌ها به مجموعه‌های آموزش و آزمون تقسیم شدند.

با اجرای این پیش‌پردازش‌ها، مجموعه داده برای استفاده در مدل‌های یادگیری ماشین آماده شده و می‌توان از آن برای پیش‌بینی دقیق وضعیت آب‌وهوا استفاده کرد.

۴.۱.۱

در این بخش، هدف استفاده از پنجره‌های زمانی برای پیش‌بینی مقادیر زمانی است. به‌ویژه، از پنجره‌های زمانی به‌صورت $[x_{t-1}, x_{t-2}, ..., x_{t-n}]$ برای پیش‌بینی مقدار x_t در زمان t استفاده می‌شود. این تکنیک به مدل‌ها این امکان را می‌دهد که از داده‌های گذشته برای پیش‌بینی مقادیر آینده استفاده کنند.

در این گزارش، دو حالت مختلف برای انتخاب پنجره‌ها و ارزیابی عملکرد مدل‌های پیش‌بینی بررسی خواهد شد. داده‌های مربوط به سال ۲۰۰۹ به‌عنوان داده‌های آزمون جدا می‌شوند و داده‌های دیگر سال‌ها به‌عنوان مجموعه داده‌های آموزش استفاده خواهند شد.

تعریف پنجره‌های زمانی:

برای پیش‌بینی مقادیر زمان، از پنجره‌های زمانی به‌صورت $[x_{t-1}, x_{t-2}, ..., x_{t-n}]$ استفاده می‌کنیم. این پنجره‌ها به مدل‌ها این امکان را می‌دهند که برای پیش‌بینی مقدار x_t در زمان t، از n لحظه قبلی اطلاعات استفاده کنند.

در این بخش، دو حالت مختلف برای انتخاب پنجره‌ها مورد بررسی قرار می‌گیرد:

- پنجره‌های بدون همپوشانی: در این حالت، هر پنجره به‌طور مستقل برای پیش‌بینی استفاده می‌شود. به عبارت دیگر، هر پنجره فقط برای پیش‌بینی مقدار مربوط به خود استفاده می‌شود.
- پنجره‌های با همپوشانی: در این حالت، پنجره‌ها می‌توانند با یکدیگر همپوشانی داشته باشند. این بدین معناست که اطلاعات از پنجره‌های مختلف می‌توانند به‌طور همزمان برای پیش‌بینی استفاده شوند.

تقسیم داده‌ها به مجموعه‌های آموزش و آزمون:

برای ارزیابی مدل‌ها، ابتدا باید داده‌ها به دو بخش آموزش و آزمون تقسیم شوند. داده‌های مربوط به سال ۲۰۰۹ باید به‌عنوان مجموعه داده‌های آزمون جدا شوند. داده‌های آموزش باید از داده‌های دیگر سال‌ها تهیه شوند.

بسته به اندازه پنجره و مقدار همپوشانی، داده‌ها به دو مجموعه آموزش و آزمون تقسیم می‌شوند. به عنوان مثال، اگر اندازه پنجره ۵ باشد و همپوشانی ۴ باشد، حجم داده‌های آموزش و آزمون به شرح زیر خواهد بود:

- اندازه پنجره: ۵
- همپوشانی: ۴
- شکل داده‌های آموزش: $365 \times n$
- شکل داده‌های آزمون: $361 \times n$

در اینجا، از داده‌های قبل از سال ۲۰۰۹ به‌عنوان مجموعه داده‌های آموزشی استفاده می‌کنیم و داده‌های مربوط به سال ۲۰۰۹ را به‌عنوان مجموعه داده‌های آزمون اختصاص می‌دهیم.

آموزش مدل‌ها و ارزیابی عملکرد:

برای پیش‌بینی از مدل‌های مختلف یادگیری ماشین مانند رگرسیون خطی یا مدل‌های پیچیده‌تر استفاده خواهیم کرد. ابتدا مدل‌های مجزایی برای هر یک از پنجره‌ها آموزش داده می‌شود و سپس عملکرد آن‌ها ارزیابی خواهد شد.

مدل‌های بدون همپوشانی:

در این حالت، هر پنجره به‌طور مجزا برای پیش‌بینی استفاده می‌شود. به عبارت دیگر، اطلاعات موجود در هر پنجره فقط برای پیش‌بینی لحظه بعدی استفاده خواهد شد.

مدل‌های با همپوشانی:

در این حالت، پنجره‌ها می‌توانند با یکدیگر همپوشانی داشته باشند. به این معنا که یک نمونه ممکن است در بیش از یک پنجره ظاهر شود. این می‌تواند باعث بهبود پیش‌بینی‌ها شود چرا که مدل می‌تواند از اطلاعات بیشتری استفاده کند.

مقایسه عملکرد مدل‌ها:

پس از آموزش مدل‌ها، باید عملکرد کلی آن‌ها مقایسه شود. این مقایسه بر اساس معیارهای مختلف ارزیابی مانند میانگین مربع خطا (MSE) یا ریشه میانگین مربع خطا (RMSE) انجام می‌شود. برای این کار باید داده‌های آزمون را که از سال ۲۰۰۹ استخراج شده‌اند، به‌عنوان داده‌های واقعی استفاده کنیم و آن‌ها را با پیش‌بینی‌های مدل‌های مختلف مقایسه کنیم.

معیارهای ارزیابی:

- Error Squared Mean (MSE): این معیار خطای مدل را به‌صورت مربع تفاوت بین پیش‌بینی‌ها و مقادیر واقعی اندازه‌گیری می‌کند.
- Error Squared Mean Root (RMSE): این معیار ریشه مربعی از MSE است و نشان‌دهنده میزان خطا در واحدهای اصلی داده‌ها است.

انواع پنجره‌های زمانی مورد استفاده:

در مقاله از دو رویکرد برای انتخاب داده‌های ورودی در مدل‌های یادگیری ماشین استفاده شده است:

۱. مدل تک‌مرحله‌ای (Single-Step): تنها مقدار لحظه قبل (x_{t-1}) برای پیش‌بینی مقدار آینده (x_t) استفاده می‌شود.

۲. مدل چندمرحله‌ای (Multi-Step) Sliding (Window: چند مقدار گذشته ($x_{t-1}, x_{t-2}, ..., x_{t-n}$) برای پیش‌بینی مقدار آینده استفاده می‌شوند.

روش‌های مختلف برای داده‌های ورودی:

دو مدل اصلی مورد بررسی به‌صورت زیر هستند:

- مدل تک‌مرحله‌ای:

$$x_t = f(x_{t-1}) \quad (۱)$$

- مدل چندمرحله‌ای (پنجره‌ی متحرک):

$$x_t = f(x_{t-1}, x_{t-2}, ..., x_{t-n}) \quad (۲)$$

در مدل چندمرحله‌ای، مقدار n تعیین می‌کند که از چند مقدار گذشته برای پیش‌بینی مقدار آینده استفاده شود.

نحوه آماده‌سازی داده‌ها برای مقایسه این دو روش:

برای مقایسه عملکرد این دو مدل، مراحل زیر انجام می‌شود:

۱. ایجاد یک پنجره‌ی متحرک (Sliding Window) در داده‌ها.

۲. ساخت دو مجموعه داده برای:
   - مدل تک‌مرحله‌ای (یک مقدار گذشته برای پیش‌بینی).
   - مدل چندمرحله‌ای (چند مقدار گذشته برای پیش‌بینی).

۳. تفکیک داده‌ها به مجموعه‌های آموزش و آزمون.

۴. مقایسه عملکرد مدل‌ها در هر دو روش.

در ادامه کدی در پایتون برای آماده سازی داده ها و مقایسه دو روش ارائه شد که نتایج آن را می بینیم:

- داده‌ها برای دو حالت تک‌مرحله‌ای و چندمرحله‌ای آماده شدند.
- مجموعه داده‌های جدید ساخته و ذخیره شدند.
- داده‌ها به مجموعه‌های آموزش و آزمون تقسیم شدند.
- امکان مقایسه عملکرد دو روش در پیش‌بینی فراهم شد.

این مقایسه نشان می‌دهد که روش چندمرحله‌ای معمولاً عملکرد بهتری دارد اما نیازمند داده‌های بیشتری است.

۲.۱

۱.۲.۱

مفهوم Learning: Machine Collaborative

یادگیری ماشین مشترک به یک روش پیش‌بینی اشاره دارد که در آن داده‌های جمع‌آوری‌شده از چندین مکان یا منبع مختلف به‌طور همزمان برای آموزش مدل‌ها استفاده می‌شود. این مدل‌ها می‌توانند برای پیش‌بینی متغیرهایی مانند دما، رطوبت و فشار جو در مناطق مختلف استفاده شوند.

در روش‌های سنتی یادگیری ماشین، هر مدل معمولاً برای یک منطقه خاص آموزش داده می‌شود و داده‌های مربوط به همان منطقه برای پیش‌بینی استفاده می‌شود. اما در یادگیری ماشین مشترک، داده‌ها از چندین منطقه مختلف جمع‌آوری می‌شوند و مدل‌ها با استفاده از این داده‌های ترکیبی آموزش داده می‌شوند. این رویکرد می‌تواند به مدل‌ها کمک کند که پیش‌بینی دقیق‌تری برای مناطق خاص انجام دهند، به‌ویژه زمانی که داده‌ها از مناطق مختلف همبستگی دارند.

ویژگی‌های یادگیری ماشین مشترک:

یادگیری ماشین مشترک دارای ویژگی‌های زیر است:

- ترکیب داده‌ها از چندین مکان: داده‌ها از چندین منطقه یا مکان مختلف برای آموزش مدل‌ها استفاده می‌شود.
- مدل‌های مشترک: به‌جای آموزش مدل‌های مجزا برای هر مکان، از یک مدل مشترک استفاده می‌شود که به‌طور همزمان از داده‌های مختلف بهره می‌برد.
- بهبود دقت پیش‌بینی: این رویکرد می‌تواند به بهبود دقت پیش‌بینی‌ها کمک کند، به‌ویژه در شرایطی که داده‌ها در مکان‌های مختلف با یکدیگر همبستگی دارند.

استفاده از Learning Machine Collaborative در مقاله:

در مقاله‌ای که مورد بررسی قرار گرفته، **یادگیری ماشین مشترک** برای پیش‌بینی وضعیت آب و هوا در مناطق مختلف استفاده شده است. داده‌های جمع‌آوری‌شده از چندین مکان مختلف، مانند **Basel**، **Stockholm** و **Tours** برای آموزش مدل‌ها استفاده شده‌اند. این مدل‌ها به‌طور خاص برای پیش‌بینی پارامترهای مختلف مانند دما، رطوبت، فشار و تابش خورشیدی استفاده می‌شوند.

استفاده از داده‌های مشترک:

در این مقاله، داده‌های جمع‌آوری‌شده از مکان‌های مختلف (شهرهای مختلف) به‌طور همزمان به مدل‌های یادگیری ماشین وارد می‌شوند. این داده‌ها شامل پارامترهای مختلف آب و هوا مانند دما، رطوبت و فشار هستند. این داده‌ها سپس برای پیش‌بینی وضعیت آب و هوا در یک منطقه خاص مورد استفاده قرار می‌گیرند.

این روش می‌تواند به پیش‌بینی دقیق‌تری منجر شود زیرا اطلاعات بیشتری از مناطق مختلف در دسترس است و مدل‌ها می‌توانند از این اطلاعات برای آموزش استفاده کنند. مقاله نشان می‌دهد که پیش‌بینی‌هایی که از داده‌های مشترک استفاده می‌کنند، دارای دقت بالاتری نسبت به پیش‌بینی‌هایی هستند که فقط از داده‌های یک منطقه خاص استفاده می‌کنند.

مدل‌های یادگیری ماشین استفاده‌شده:

در این مقاله، از پنج مدل مختلف یادگیری ماشین برای پیش‌بینی وضعیت آب و هوا استفاده شده است. این مدل‌ها عبارتند از:

- رگرسیون خطی چندگانه (MLR)
- رگرسیون چندجمله‌ای (MPR)
- (KNN) Neighbors K-Nearest
- پرسپترون چند لایه (MLP)
- شبکه عصبی کانولوشنی (CNN)

این مدل‌ها به‌طور خاص برای پیش‌بینی ویژگی‌های مختلف آب و هوا آموزش داده شدند. مقاله نشان داد که استفاده از داده‌های مشترک برای آموزش این مدل‌ها باعث بهبود قابل توجهی در دقت پیش‌بینی‌ها شد.

نتایج استفاده از Learning: Machine Collaborative

در این مقاله، نتایج نشان داد که پیش‌بینی‌های مشترک با استفاده از داده‌های چندین مکان به‌طور قابل توجهی دقت پیش‌بینی‌ها را بهبود بخشیده است. در آزمایش‌های انجام‌شده، مدل‌هایی مانند رگرسیون چندجمله‌ای (MPR) و رگرسیون خطی چندگانه (MLR) که از داده‌های مشترک استفاده کرده بودند، عملکرد بهتری نسبت به مدل‌های پیش‌بینی غیر مشترک داشتند. مقاله همچنین نشان داد که پیش‌بینی‌های مشترک می‌توانند میانگین خطای پیش‌بینی را تا حدود ۵% کاهش دهند.

استفاده از پیش‌بینی‌های مشترک برای بهبود دقت:

نتایج این مقاله نشان داد که استفاده از پیش‌بینی‌های مشترک به‌ویژه در مناطقی که داده‌های آن‌ها به‌طور طبیعی همبسته هستند، می‌تواند دقت پیش‌بینی‌ها را بهبود بخشد. این روش از طریق ترکیب داده‌های مختلف و استفاده از یک مدل مشترک به‌جای آموزش مدل‌های جداگانه برای هر منطقه، به دقت بالاتر در پیش‌بینی‌ها کمک می‌کند.

۳.۱

در این بخش از تمرین، هدف پیاده‌سازی رگرسیون چندجمله‌ای درجه ۱ بدون استفاده از مدل‌های آماده است. به‌طور خاص، از حداقل مربعات (Least Squares) برای محاسبه وزن‌ها استفاده شده است. همچنین، مدل به‌طور دستی آموزش داده شده و خطا (با استفاده از معیار میانگین مربع خطا ((MSE) برای داده‌های آموزش و آزمون محاسبه شد. علاوه بر این، نوار پیشرفت با استفاده از کتابخانه `tqdm` برای نمایش روند آموزش استفاده شده است.

هدف این است که روند آموزش مدل شامل پیش‌بینی ورودی‌ها، محاسبه خطا، محاسبه گرادیان و بروزرسانی وزن‌ها باشد. در این گزارش، نتایج آموزش مدل، وزن‌های نهایی، و همچنین مقایسه خطاهای مدل در طول آموزش برای داده‌های آموزش و آزمون آورده شده است.

روش‌شناسی:

در این بخش، نحوه پیاده‌سازی رگرسیون چندجمله‌ای درجه ۱ شرح داده شده است. مدل رگرسیون چندجمله‌ای درجه ۱ در واقع همان رگرسیون خطی است که در آن رابطه بین ویژگی‌ها و هدف به‌صورت خطی فرض می‌شود.

معیار ارزیابی:

برای ارزیابی عملکرد مدل، از معیار میانگین مربع خطا (MSE) استفاده شده است. این معیار به‌طور معمول برای ارزیابی دقت مدل‌های رگرسیونی به کار می‌رود و نشان‌دهنده تفاوت بین مقادیر پیش‌بینی‌شده و مقادیر واقعی است.

:فرمول رگرسیون چندجمله‌ای درجه ۱

رگرسیون چندجمله‌ای درجه ۱ به‌صورت زیر تعریف می‌شود:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

که در آن:

- y متغیر وابسته است.
- $x_1, x_2, ..., x_n$ ویژگی‌های ورودی هستند.
- $\beta_0, \beta_1, ..., \beta_n$ پارامترهای مدل هستند که باید با استفاده از داده‌های آموزشی به‌دست آیند.

پیاده‌سازی مدل:

در این بخش، رگرسیون چندجمله‌ای درجه ۱ به‌طور دستی پیاده‌سازی شده است. در این پیاده‌سازی از روش حداقل مربعات برای محاسبه وزن‌ها استفاده شده است.

توضیحات کد:

۱. بارگذاری داده‌ها: داده‌ها از فایل CSV بارگذاری می‌شوند. ویژگی‌ها به‌عنوان ورودی و دمای شهر Tours به‌عنوان هدف انتخاب می‌شوند. ۲. آموزش مدل: در این بخش، از رگرسیون خطی درجه ۱ برای مدل‌سازی استفاده شده است. برای آموزش مدل، از حداقل مربعات برای محاسبه وزن‌ها و رادیان برای به‌روزرسانی آن‌ها استفاده می‌شود. ۳. محاسبه خطا: در هر اپوک، خطا (MSE) برای داده‌های آموزش و آزمون محاسبه و نمایش داده می‌شود. ۴. نوار پیشرفت: نوار پیشرفت با استفاده از tqdm نمایش داده می‌شود تا روند آموزش به‌طور زنده مشاهده شود.

نتایج:

۱. وزن‌های نهایی: پس از اتمام آموزش مدل، وزن‌های نهایی به‌صورت زیر به دست آمدند:

$$\beta_0 = 0.4794 \quad \beta_1 = 11.4253$$

این مقادیر نشان‌دهنده این است که رابطه بین ویژگی‌های ورودی (دمای شهرهای مختلف) و هدف (دمای شهر Tours) به‌صورت خطی است و وزن‌ها بر اساس داده‌های آموزشی به این مقادیر رسیدند.

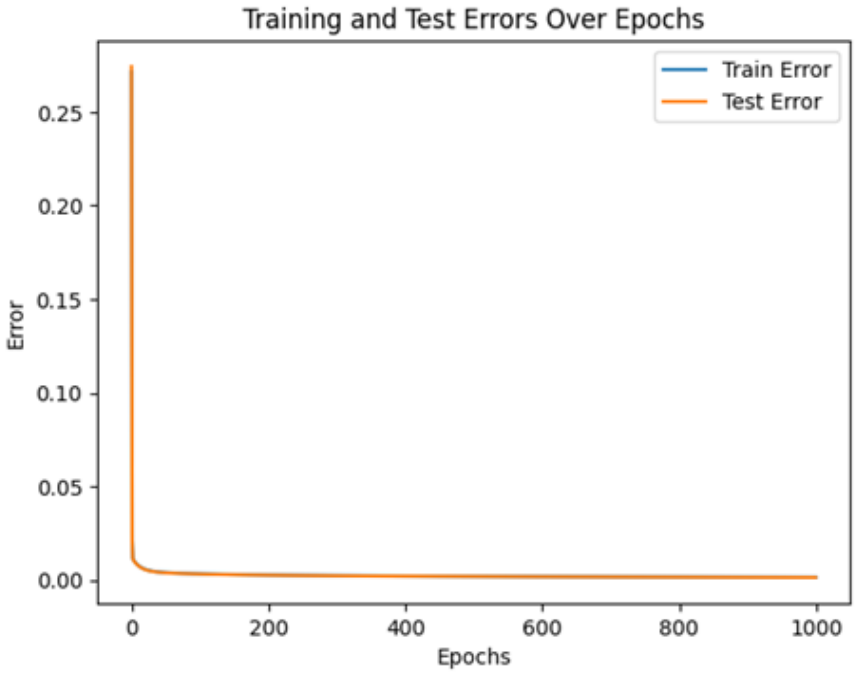
۲. نمودار خطا: در نمودار شکل ۱، خطای آموزش و خطای آزمون در طول حلقه‌های آموزش (اپوک‌ها) نشان داده شده است. مشاهده می‌شود که خطای مدل در ابتدا به‌طور چشمگیری کاهش می‌یابد و سپس کاهش آن کند می‌شود.

۳. نتایج نهایی: - خطای آموزش در ابتدا به‌سرعت کاهش یافت و به حد مطلوب رسید. - خطای آزمون نیز کاهش قابل توجهی داشت و به حد قابل قبولی رسید. - در نهایت، مدل توانست روابط خطی میان ویژگی‌ها و هدف را مدل‌سازی کند و به دقت مناسبی رسید.

۱.۳.۱

در این بخش از تمرین، هدف پیاده‌سازی و مقایسه سه مدل رگرسیونی مختلف از کتابخانه `scikit-learn` است. مدل‌های انتخاب‌شده عبارتند از:

- رگرسیون خطی (Linear Regression)



شکل ۱: نمودار خطاهای آموزش و آزمون در طول اپوک‌ها

- رگرسیون لاسو Regression) (Lasso
- رگرسیون ریدج Regression) (Ridge

این مدل‌ها برای پیش‌بینی دمای شهر Tours بر اساس ویژگی‌های دمایی دیگر شهرها مانند Basel و Stockholm استفاده شدند. در این گزارش، ابتدا تئوری و فرمول‌های ریاضی مرتبط با هر یک از این مدل‌ها توضیح داده شده و سپس عملکرد آن‌ها با استفاده از داده‌های آموزشی و آزمون مقایسه می‌شود.

تئوری مدل‌ها و فرمول‌های ریاضی مرتبط:

در این بخش، توضیحاتی در مورد سه مدل رگرسیونی مختلف و فرمول‌های ریاضی آن‌ها آورده شده است.

رگرسیون خطی Regression): (Linear

رگرسیون خطی ساده‌ترین و ابتدایی‌ترین مدل در یادگیری ماشین است که سعی می‌کند رابطه‌ای خطی میان ویژگی‌های ورودی و هدف برقرار کند.

فرمول رگرسیون خطی به‌صورت زیر است:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

که در آن:

- y متغیر هدف است.
- $x_1, x_2, \ldots, x_n$ ویژگی‌های ورودی هستند.
- $\beta_0, \beta_1, \ldots, \beta_n$ ضرایب مدل هستند که باید از داده‌ها به‌دست آیند.

رگرسیون لاسو (Lasso Regression):

رگرسیون لاسو نسخه‌ای از رگرسیون خطی است که در آن از تنظیم‌گر L۱ برای محدود کردن ضرایب مدل استفاده می‌شود. هدف این تنظیم‌گر کاهش پیچیدگی مدل و جلوگیری از اورفیتینگ است.

فرمول رگرسیون لاسو به‌صورت زیر است:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left[\|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

که در آن:

- $\|y - X\beta\|_2^2$ خطای مدل است.
- λ پارامتر تنظیم است که شدت جریمه L۱ را کنترل می‌کند.
- $\sum_{j=1}^{p} |\beta_j|$ جریمه L۱ است که از ضرایب مدل استفاده می‌کند.

رگرسیون ریدج (Ridge Regression):

رگرسیون ریدج هم مشابه رگرسیون لاسو است، اما به‌جای تنظیم‌گر L۱ از تنظیم‌گر L۲ استفاده می‌کند. در این روش، جریمه به مربع مقادیر ضرایب اضافه می‌شود.

فرمول رگرسیون ریدج به‌صورت زیر است:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left[\|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right]$$

که در آن:

- $\|y - X\beta\|_2^2$ خطای مدل است.
- λ پارامتر تنظیم است که شدت جریمه L۲ را کنترل می‌کند.
- $\sum_{j=1}^{p} \beta_j^2$ جریمه L۲ است که از مربع ضرایب استفاده می‌کند.

پیاده‌سازی مدل‌ها:

در این بخش، از سه مدل رگرسیونی رگرسیون خطی، رگرسیون لاسو و رگرسیون ریدج استفاده شد. داده‌ها از فایل CSV بارگذاری شده و به مجموعه‌های آموزش و آزمون تقسیم شدند. سپس، هر سه مدل آموزش داده شدند و عملکرد آن‌ها با استفاده از معیار میانگین مربع خطا (MSE) ارزیابی شد.

نتایج مدل‌ها:

پس از آموزش هر مدل، MSE برای داده‌های آموزش و آزمون محاسبه شد. نتایج به‌صورت زیر به‌دست آمد:

- رگرسیون خطی:

  ۰ MSE: Train, ۰ MSE: Test

- رگرسیون لاسو:

  ۰۳۰۰.۰ MSE: Train, ۰۲۹۶.۰ MSE: Test

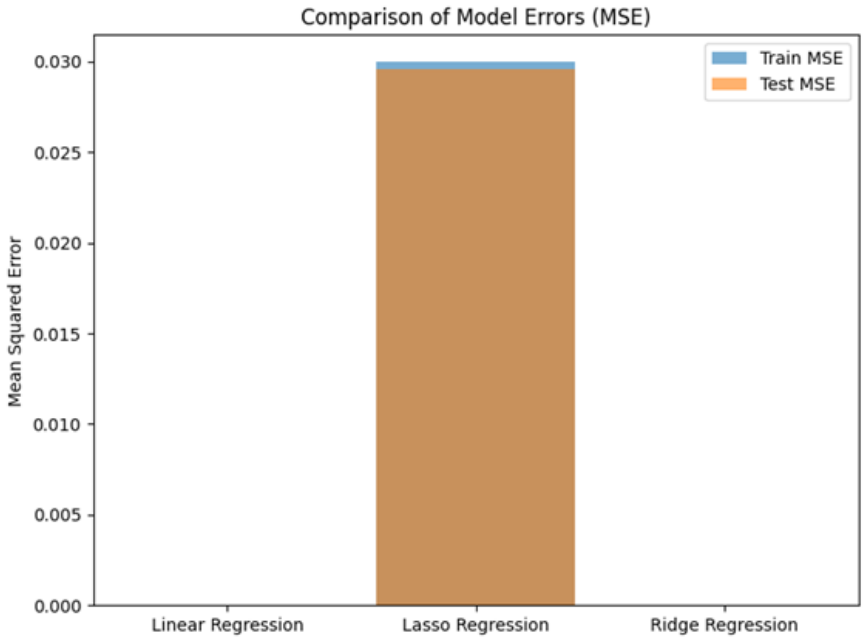
- رگرسیون ریدج:

  ◦ MSE: Train, ◦ MSE: Test

بر اساس نتایج بالا، مشاهده می‌شود که رگرسیون خطی کمترین MSE را در داده‌های آزمون به‌دست آورد.

نمودار مقایسه MSE:

نمودار شکل ۲ MSE برای داده‌های آموزش و آزمون را برای هر سه مدل نمایش می‌دهد. همانطور که مشاهده می‌کنید، رگرسیون خطی بهترین عملکرد را در داده‌های آزمون داشت و کمترین خطا را به‌دست آورد.



شکل ۲: مقایسه خطای MSE برای مدل‌های رگرسیونی

انتخاب بهترین مدل:

با توجه به نتایج MSE، مدل رگرسیون خطی بهترین عملکرد را در داده‌های آزمون داشت. بنابراین، رگرسیون خطی به‌عنوان بهترین مدل انتخاب شد.

۴.۱ امتیازی

در ادامه، هدف مقایسه عملکرد چهار مدل رگرسیونی مختلف برای پیش‌بینی دمای دو شهر Basel و Budapest است. این مدل‌ها عبارتند از:

- رگرسیون خطی (Linear Regression)
- رگرسیون لاسو (Lasso Regression)
- رگرسیون ریدج (Ridge Regression)
- رگرسیون چندجمله‌ای (Polynomial Regression)

نتایج و تجزیه و تحلیل برای شهر :Basel

مدل‌ها روی داده‌های مربوط به شهر Basel آموزش داده شدند. نتایج MSE برای مدل‌های مختلف به شرح زیر است:

- رگرسیون خطی (Linear Regression):

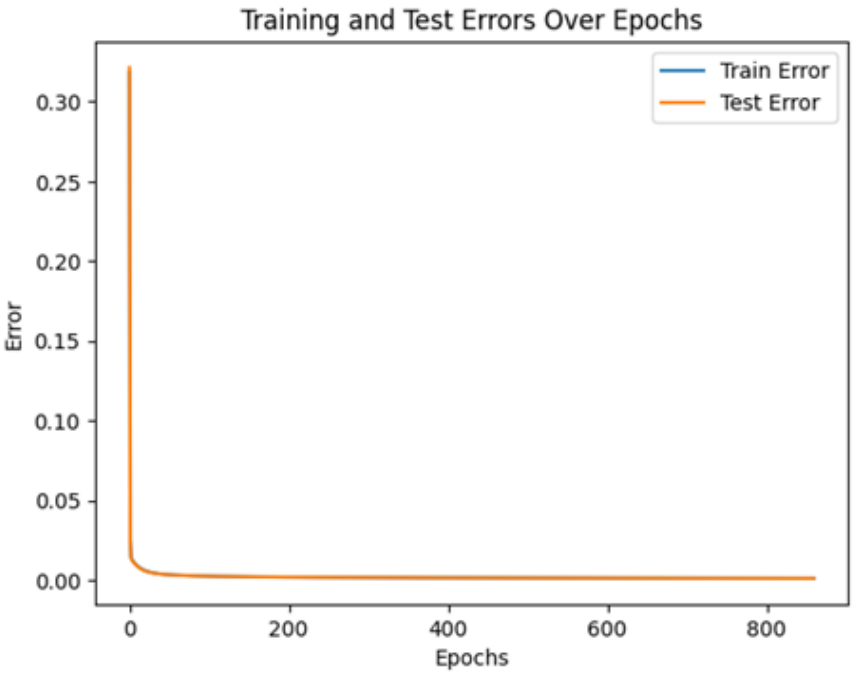
$$\text{MSE Train} = 0.0001, \quad \text{MSE Test} = 0.0001$$

- رگرسیون لاسو (Lasso Regression):

$$\text{MSE Train} = 0.0376, \quad \text{MSE Test} = 0.0368$$

- رگرسیون ریدج (Ridge Regression):

$$\text{MSE Train} = 0.0001, \quad \text{MSE Test} = 0.0001$$



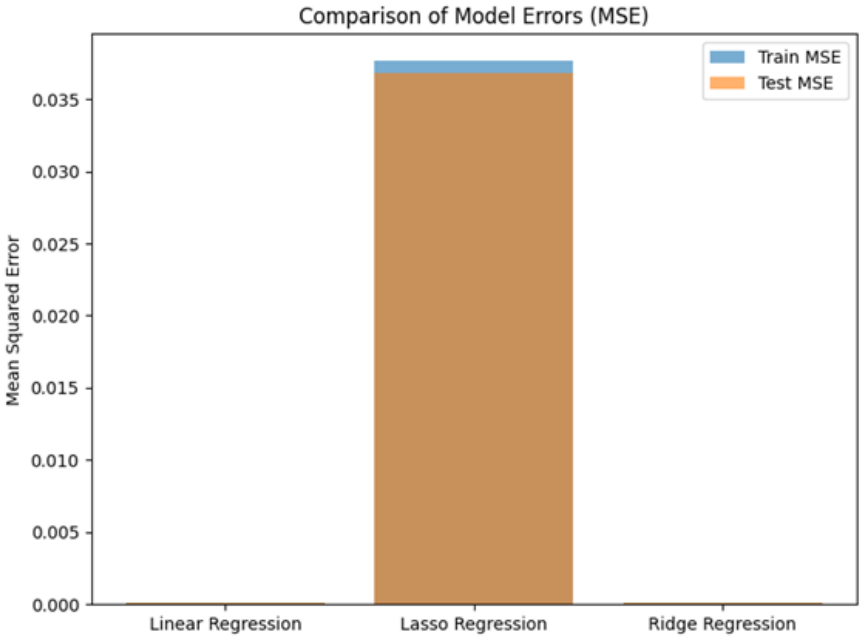
شکل ۳: نمودار خطاهای آموزش و آزمون در طول اپوک‌ها

نتیجه‌گیری:

- رگرسیون خطی و رگرسیون ریدج کمترین MSE را در داده‌های آزمون و آموزش داشتند و عملکرد مشابهی ارائه دادند.
- رگرسیون لاسو عملکرد ضعیف‌تری داشت.
- مدل ریدج بهترین عملکرد را در داده‌های آزمون نشان داد.

نتایج و تجزیه و تحلیل برای شهر :Budapest

مدل‌ها روی داده‌های مربوط به شهر Budapest آموزش داده شدند. نتایج MSE برای مدل‌های مختلف به شرح زیر است:



شکل ۴: مقایسه خطای MSE برای مدل‌های رگرسیونی

- رگرسیون خطی (Linear Regression):

$$\text{MSE Train} = 0.0004, \quad \text{MSE Test} = 0.0004$$

- رگرسیون لاسو (Lasso Regression):

$$\text{MSE Train} = 0.0418, \quad \text{MSE Test} = 0.0406$$

- رگرسیون ریدج (Ridge Regression):

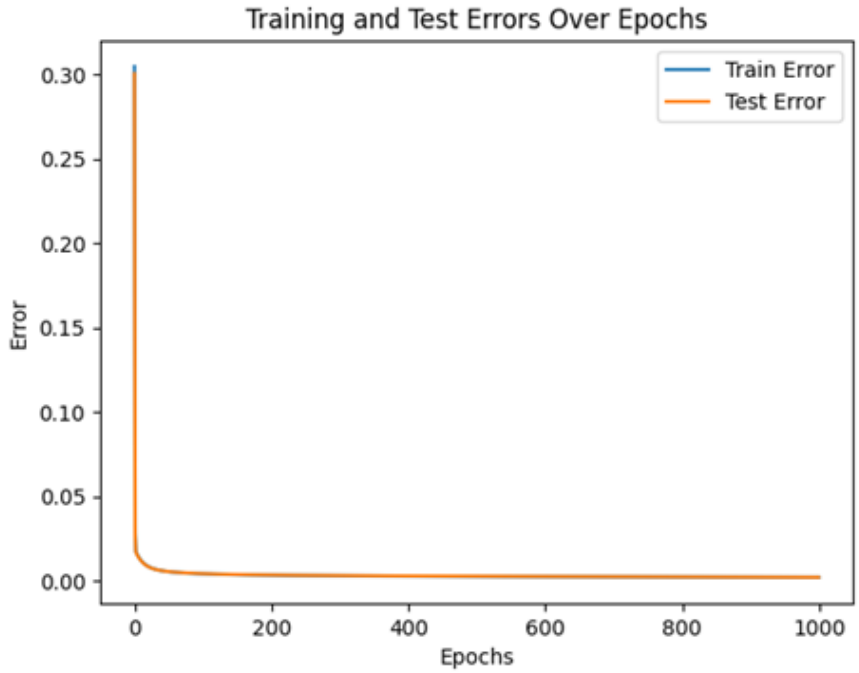
$$\text{MSE Train} = 0.0004, \quad \text{MSE Test} = 0.0004$$

نتیجه‌گیری:

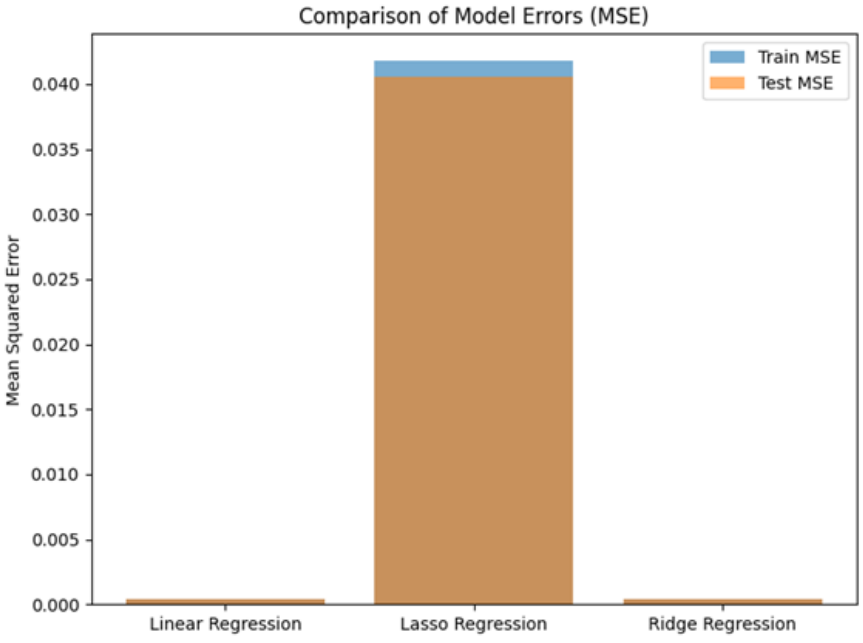
- رگرسیون خطی کمترین MSE را در داده‌های آزمون و آموزش داشت و بهترین عملکرد را ارائه داد.
- رگرسیون لاسو عملکرد ضعیف‌تری نسبت به سایر مدل‌ها نشان داد.
- مدل رگرسیون خطی بهترین مدل برای پیش‌بینی دمای شهر Budapest انتخاب شد.

مقایسه مدل‌ها بین دو شهر:
در مقایسه عملکرد مدل‌ها برای Basel و Budapest:

- مدل رگرسیون خطی در هر دو شهر کمترین MSE را در داده‌های آزمون نشان داد و به‌عنوان بهترین مدل انتخاب شد.
- رگرسیون لاسو عملکرد ضعیف‌تری در هر دو شهر نشان داد.



شکل ۵: نمودار خطاهای آموزش و آزمون در طول اپوک‌ها



شکل ۶: مقایسه خطای MSE برای مدل‌های رگرسیونی

- رگرسیون ریدج مشابه رگرسیون خطی عمل کرد، اما در نهایت رگرسیون خطی به‌عنوان بهترین مدل انتخاب شد.

نتیجه‌گیری نهایی:

پس از ارزیابی و مقایسه مدل‌های مختلف، رگرسیون خطی بهترین مدل برای پیش‌بینی دمای هر دو شهر Basel و Budapest بود. این مدل توانست با کمترین خطا در داده‌های آزمون پیش‌بینی‌های دقیقی ارائه دهد.

۵.۱ امتیازی

در این قسمت از ویژگی رطوبت که برای همه شهر ها موجود است را استفاده کردیم

مدل‌های مختلف رگرسیونی بر روی داده‌های مربوط به شهر Budapest آموزش داده شدند. نتایج MSE برای مدل‌های مختلف به شرح زیر است:

- رگرسیون خطی (Linear Regression):

$$\text{MSE Train} = 0.0218, \quad \text{MSE Test} = 0.0211$$

- رگرسیون لاسو (Lasso Regression):

$$\text{MSE Train} = 0.0418, \quad \text{MSE Test} = 0.0406$$

- رگرسیون ریدج (Ridge Regression):

$$\text{MSE Train} = 0.0218, \quad \text{MSE Test} = 0.0211$$

تجزیه و تحلیل نتایج:

رگرسیون خطی (Linear Regression):

مدل رگرسیون خطی نتایج خوبی برای داده‌های آموزش و آزمون نشان داد. با MSE کم‌تر از ۰۲۲.۰، این مدل به‌خوبی توانست پیش‌بینی‌های دقیقی انجام دهد. عملکرد مشابهی در داده‌های آموزش و آزمون داشت که نشان‌دهنده عدم اورفیتینگ مدل است.

رگرسیون لاسو (Lasso Regression):

مدل رگرسیون لاسو در مقایسه با مدل‌های دیگر نتایج ضعیف‌تری نشان داد. MSE بالاتر در هر دو مجموعه آموزش و آزمون نشان‌دهنده این است که جریمه L۱ اعمال‌شده در این مدل باعث کاهش دقت پیش‌بینی‌ها شده است.

رگرسیون ریدج (Ridge Regression):

مدل رگرسیون ریدج مشابه رگرسیون خطی عمل کرد و نتایج بسیار خوبی در داده‌های آموزش و آزمون داشت. MSE بسیار نزدیک به رگرسیون خطی در هر دو مجموعه نشان‌دهنده عملکرد مطلوب این مدل است.

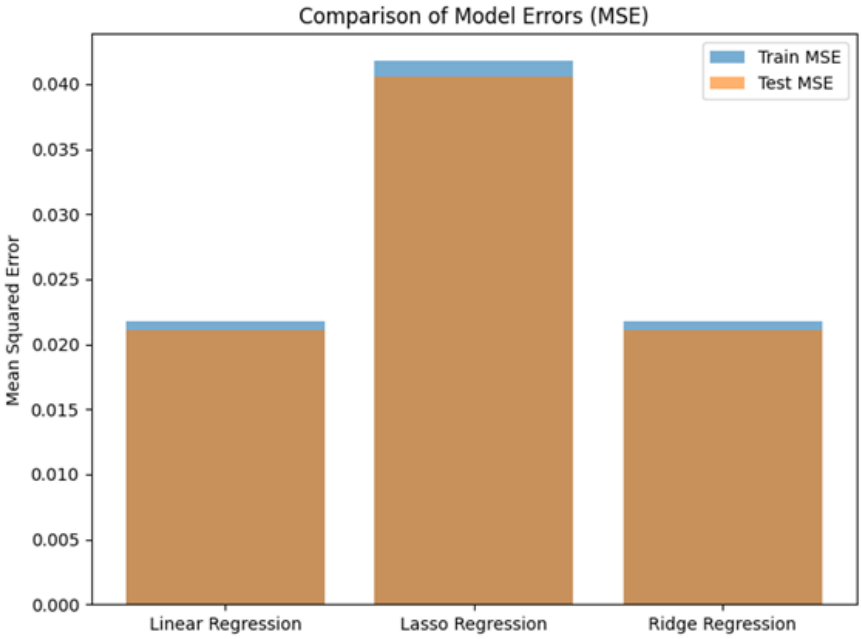
مقایسه نتایج:

در نمودار شکل ۷، مقایسه MSE برای مدل‌های مختلف آورده شده است. همانطور که مشاهده می‌شود، مدل رگرسیون خطی و رگرسیون ریدج مشابه‌ترین نتایج را دارند و بهترین عملکرد را ارائه می‌دهند.

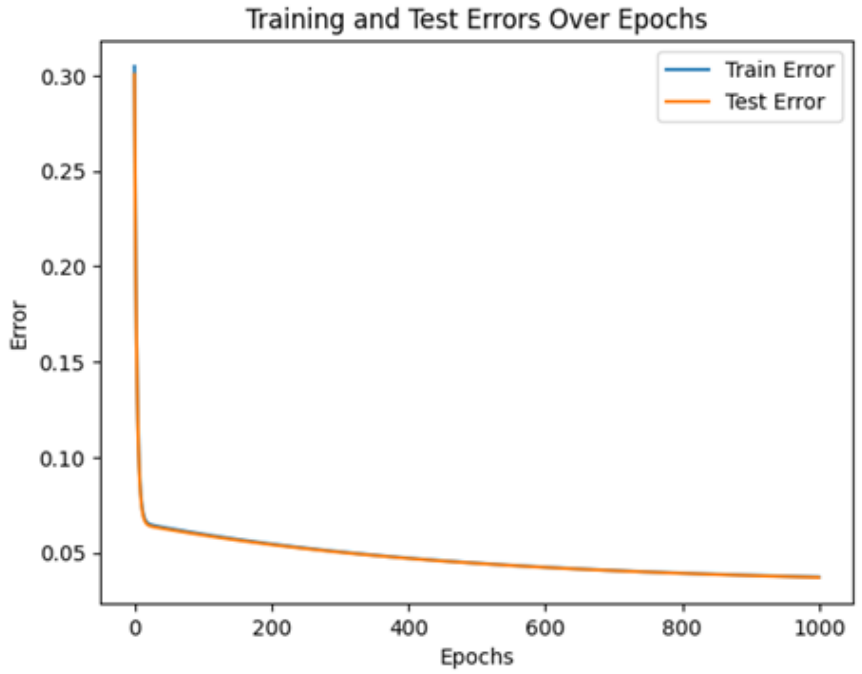
نتیجه‌گیری:

مدل رگرسیون خطی و رگرسیون ریدج بهترین عملکرد را برای پیش‌بینی دمای شهر Budapest داشتند. این دو مدل با کمترین MSE در داده‌های آزمون به‌عنوان بهترین مدل‌ها برای این پیش‌بینی‌ها شناخته شدند. مدل رگرسیون لاسو عملکرد ضعیف‌تری در مقایسه با سایر مدل‌ها داشت.

بر اساس نتایج به‌دست‌آمده، **رگرسیون ریدج** به‌عنوان بهترین مدل برای پیش‌بینی دمای شهر Budapest انتخاب می‌شود.



شکل ۷: مقایسه خطای مدل‌ها (MSE)



شکل ۸: نمودار خطاهای آموزش و آزمون در طول اپوک‌ها

۲ تشخیص عیب یاتاقان غلتشی بر مبنای دسته بندی های سلسله مراتبی

لینک درایو گوگل کولب حاوی کدهای این تمرین

لینک Github

۱.۲ دادگان

۱.۱.۲

دیتاست MalFaulDa که در این پژوهش مورد بررسی قرار می گیرد، منبعی غنی برای طراحی مدل هایی برای شناسایی عیب ماشین های دوار است. در این دیتاست، با استفاده از نرم افزار شبیه عیب ماشین آلات، داده های دستگاه آزمایشی تراز – تعادل – لرزش از شرکت درSpectraQuest شرایط مختلف از جمله حالت نرمال و حالت های مختلف پیش آمد عیب با شدت های گوناگون ایجاد و ذخیره شده است. در طی این شبیه سازی ها، داده های ۸ سنسور به شرح زیر ذخیره شده و هر یک در ستون های مجزای دیتاست ذخیره شده اند.

۱. سیگنال تاکومتر
۲. سیگنال ارتعاش محوری ۱
۳. سیگنال ارتعاش شعاعی ۱
۴. سیگنال ارتعاش مماسی ۱
۵. سیگنال ارتعاش محوری ۲
۶. سیگنال ارتعاش شعاعی ۲
۷. سیگنال ارتعاش مماسی ۲
۸. سیگنال میکروفون

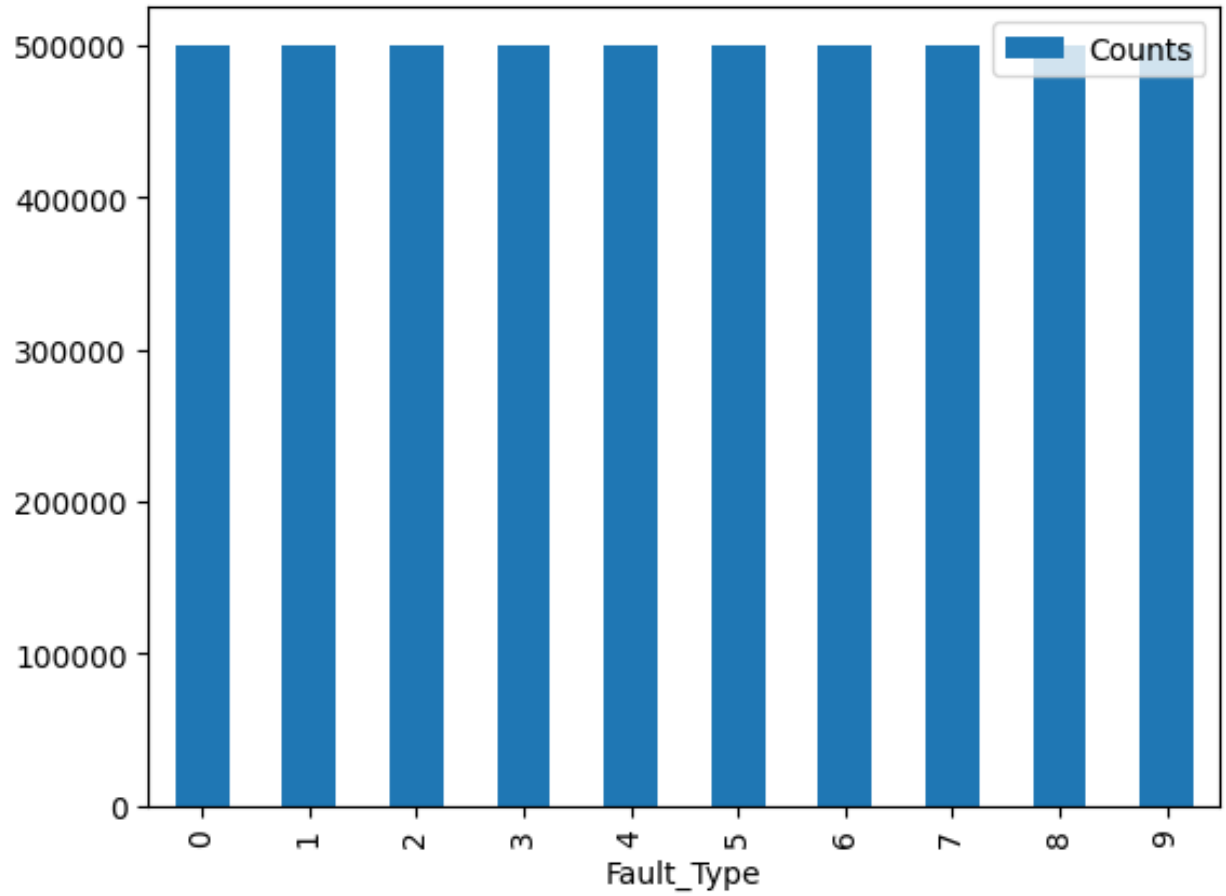
در هر مرحله از شبیه سازی، ۵ ثانیه نمونه برداری با فرکانس ۵Khz انجام شده است که منجر به تولید ۲۵۰۰۰۰ نمونه برای هر آزمایش می شود.

۲.۲

عیب های ذکر شده برای این سیستم به شرح زیر می باشند.

۱. انحراف محور افقی از 0.5 تا ۲ میلیمتر که در این گزارش از دو داده ی ۱ و ۲ میلی متر استفاده شده است.
۲. انحراف محور عمودی از 0.51 تا 1.9 میلی متر که در این گزارش از دو داده ی 0.51 و 1.9 میلی متر استفاده شده است
۳. خطاهای یاتاقان که خود شامل موارد زیر می شوند.
   (آ) خطای قفس
   (ب) عیب بیرونی
   (ج) عیب توپ

در هر یک از موارد بالا، از دو داده در دسته های ۰g و ۲۰g استفاده شده است.



شکل ۹: توزیع داده ها در دسته ها

۴. عدم تعادل در بازه ی ۶ تا ۳۵ گرم که در این گزارش از داده های ۱۰ و ۲۰ گرم استفاده شده است.

بنابراین، داده های مورد استفاده از این دیتاست در این پژوهش به صورت نمایش داده شده در شکل ۹ خواهد بود. لازم است توجه شود که در این نمودار، دسته بندی ها با مقادیر عددی به صورت زیر جایگزین شده اند.

۰. حالت عادی

۱. انحراف افقی

۲. انحراف عمودی

۳. خرابی توپ اورهانگ

۴. خرابی قفس اورهانگ

۵. خرابی رینگ خارجی اورهانگ

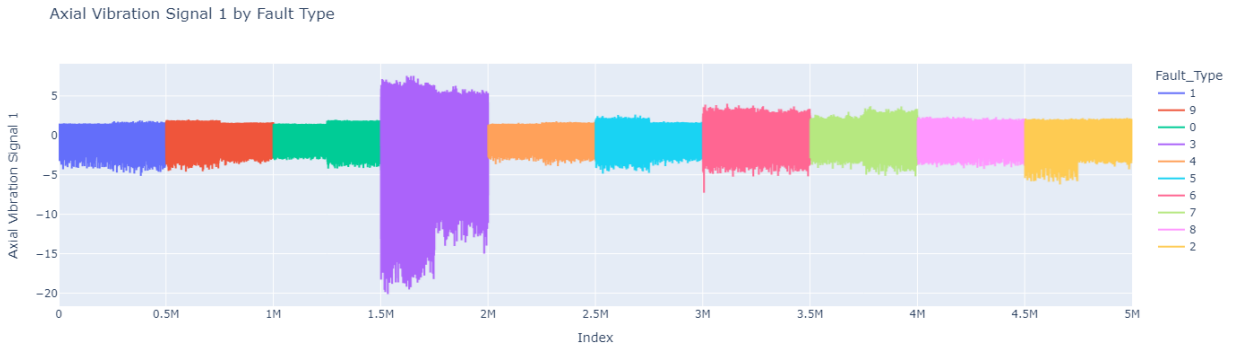
۶. خرابی توپ آندرهانگ

۷. خرابی قفس آندرهانگ

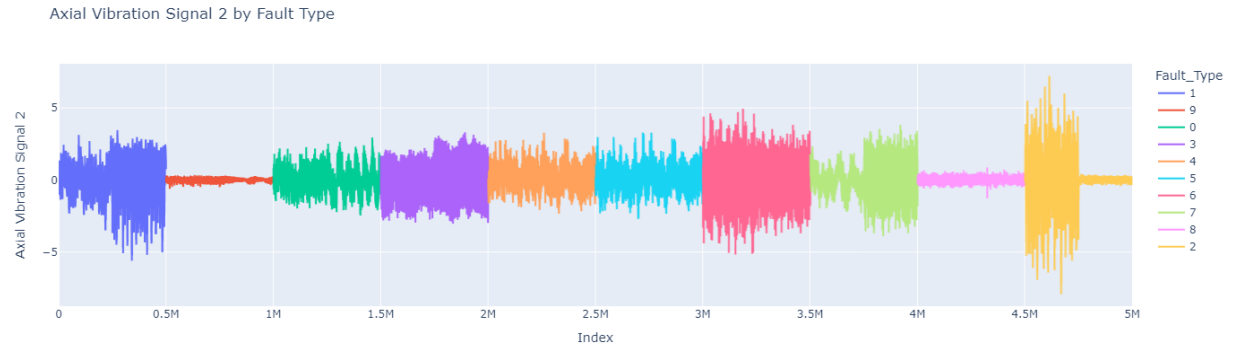
۸. خرابی رینگ خارجی آندرهانگ

۹. ناهماهنگی

از رسم نمودار این سیگنال ها به ازای انواع عیب های ممکن، نمودار های زیر به دست می آیند.



شکل ۱۰: Signal Vibration Axial ۱

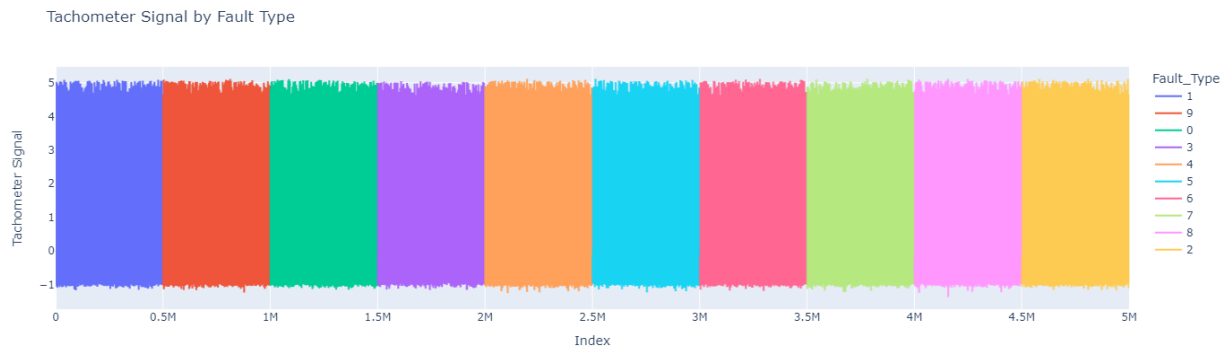


شکل ۱۱: Signal Vibration Axial ۲

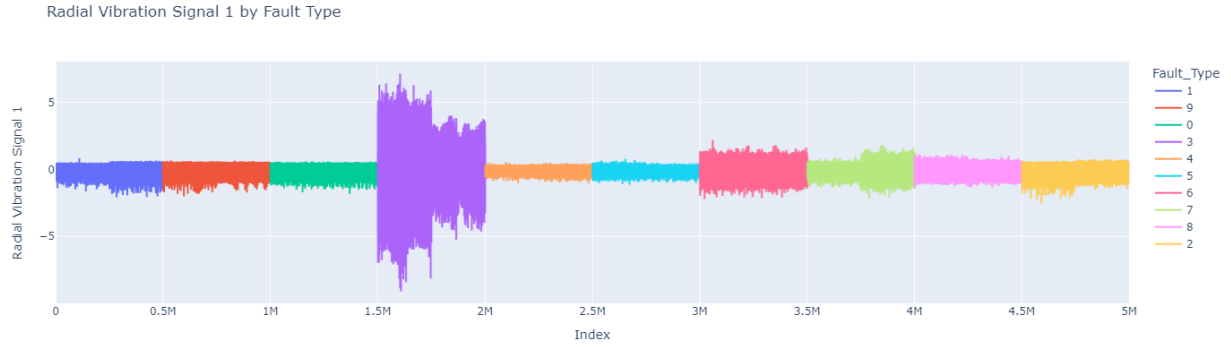
۳.۲ پیش پردازش و استخراج ویژگی

۱.۳.۲

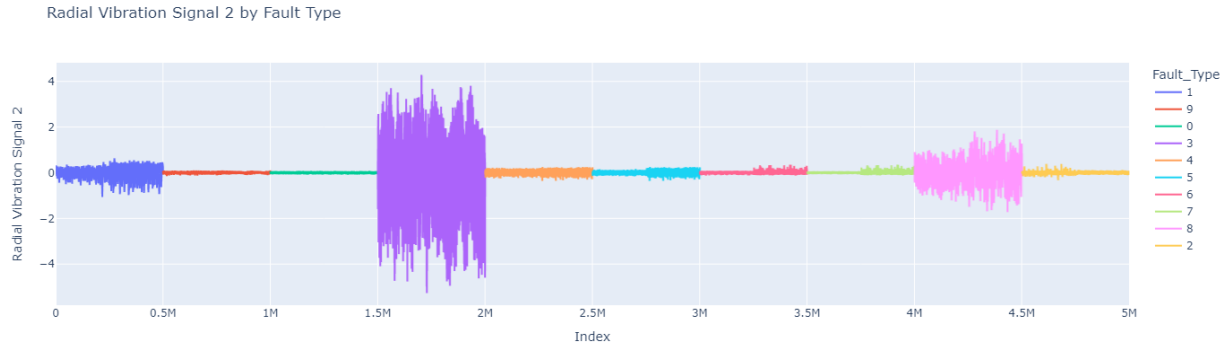
در این بخش، به پیش پردازش دیتاست حاصل می پردازیم. مطابق با فایل پایان نامه، اولین گام بررسی نمودار های ستون های موجود در دیتاست و توزیع و دامنه ی هر یک است تا بتوانیم داده هایی را که اثر کمتری در تحلیل ها دارند را برای سادگی حذف کنیم. مطابق با آنچه که در این پایان نامه ذکر شده است، در راستای بررسی سیگنال های اغتشاش، می توانیم دو سیگنال سرعت چرخش که توسط تاکومتر اندازه گیری شده و سیگنال صوت



شکل ۱۲: Signal Tachometer



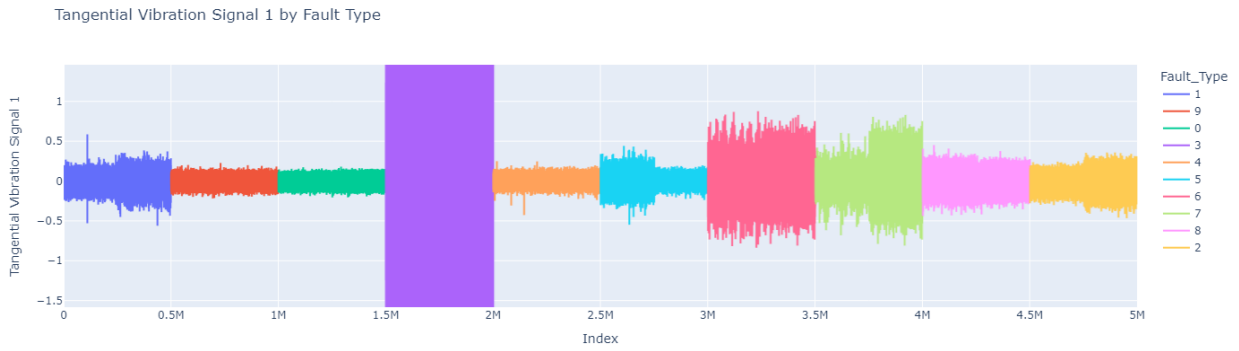
شکل ۱۳: Signal Vibration Radial ۱



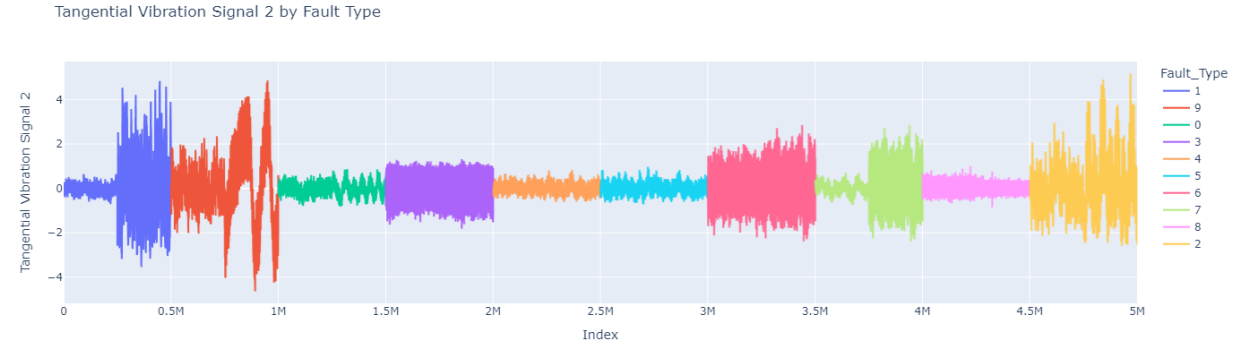
شکل ۱۴: Signal Vibration Radial ۲

را ابتدا از دیتاست حذف کنیم. در همین حال، در شکل ۱۲ مشاهده می کنیم که دامنه ی این سیگنال برای تمامی حالت ها مشابه است و تنها ممکن است با استفاده از فیچر های حوزه ی فرکانس بتونا تمایزی میان آنها قائل شد. بنابراین، می توانیم این ستون را در مدل خود در نظر نگیریم.

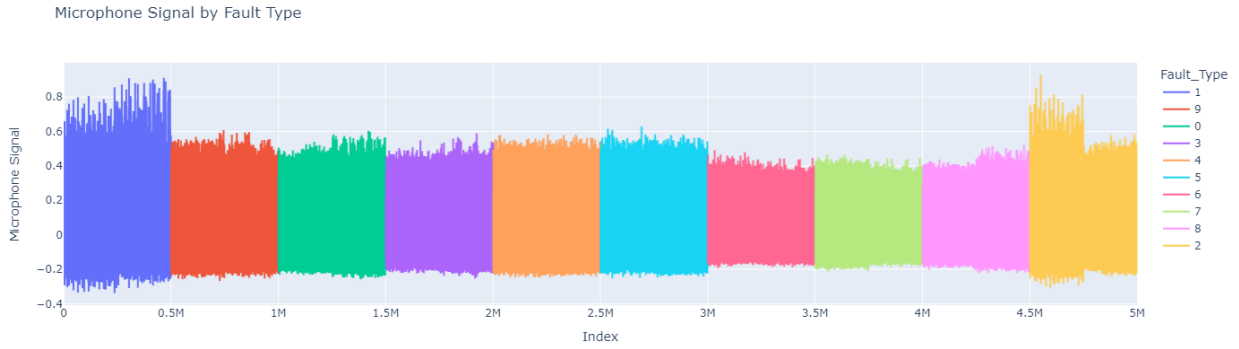
همچنین، با توجه به نمودار ۱۴ و مطابق با فایل پایان نامه، مشاهده می کنیم که دامنه ی اکثر داده ها به جز حالت ۱ که با رنگ بنفش نمایش



شکل ۱۵: Signal Vibration Tangential ۱



شکل ۱۶: Signal Vibration Tangential ۲



شکل ۱۷: Signal Microphone

داده شده است و به معنای انحراف افقی است، دارای دامنه های کمتری در مقایسه با سایر ستون ها هستند. با این حال، به دلیل آنکه استفاده از داده های این ستون می تواند کمک خوبی برای تشخیص حالت ۱ از بقیه حالت ها باشد، در این گزارش آن را حذف نمی کنیم. در این بخش، دیتاست به صورت نمایش داده شده در شکل ۱۸ به دست می آید.

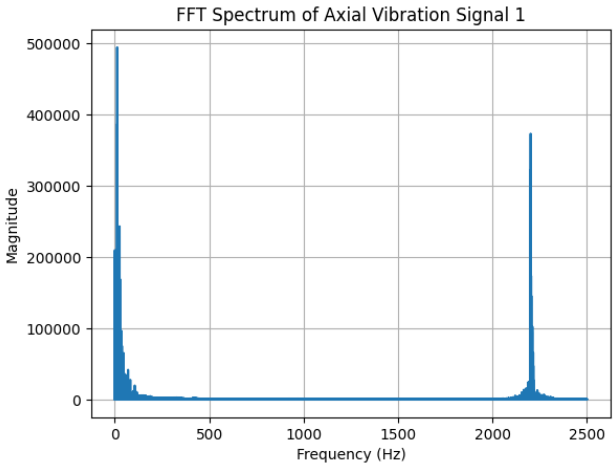
	Axial Vibration Signal 1	Radial Vibration Signal 1	Tangential Vibration Signal 1	Axial Vibration Signal 2	Radial Vibration Signal 2	Tangential Vibration Signal 2	Fault_Type
0	1.29000	0.246300	0.017209	-0.64557	0.077080	-0.10002	1
1	-2.02510	-0.902780	-0.125530	-0.71259	0.061588	-0.16430	1
2	0.61405	0.335120	0.000702	-0.65959	0.066658	-0.11457	1
3	-1.08860	-0.760080	-0.068845	-0.69720	0.065130	-0.15646	1
4	-0.64317	0.062457	0.034943	-0.68747	0.067832	-0.14811	1

شکل ۱۸: $removedcolumnsdataset$

۲.۳.۲ انتخاب پنجره زمانی و تقسیم دادگان

در بخش قبل، سیگنال های سنسورهای مورد نیاز انتخاب شدند و در شش ستون در دیتافریم ذخیره شدند. با این حال، برای آموزش مدل های یادگیری ماشین، لازم است تا از این سری های زمانی در فواصل مشخص نمونه برداری شده و سپس عملیات استخراج ویژگی بر آنها پیاده سازی شود. بنابراین، اولین گام برای پردازش این داده ها، انتخاب طول پنجره زمانی صحیح برای نمونه برداری و تقسیم دادگان است.

بهتر آن است که پنجره های زمانی به گونه ای انتخاب شوند که یک سیکل کامل از عملکرد موتور را در بر بگیرند. بنابراین، برای تشخیص این دوره ی تناوب، به بررسی محتوای فرکانسی هر یک از سیگنال ها خواهیم پرداخت تا بتوانیم فرکانسی را که در آن سیگنال ها شدت بیشتری دارند و متعاقبا با آن فرکانس متناوب هستند را به دست بیاوریم. در ادامه، با استفاده از دستور fft، محتوای فرکانسی سیگنال ها نمایش داده می شود.



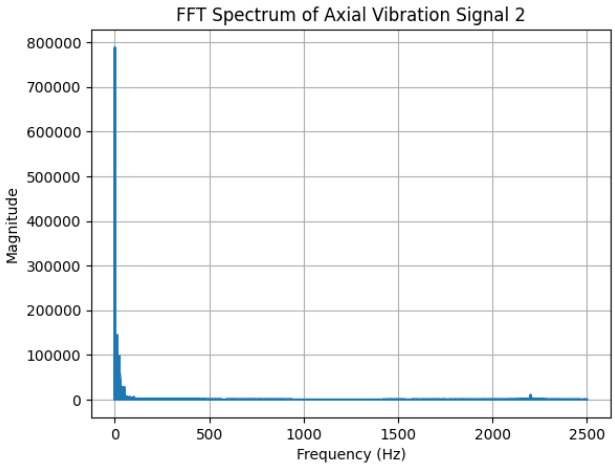
شکل ۱۹: FFT for Signal Vibration Axial ۱

از مشاهده ی محتوای فرکانسی این سیگنال ها در میابیم که در فرکانس ۲۲۰۰، مقدار زیادی انرژی نمایش داده شده است که این فرکانس، می تواند نشان دهنده ی دوره ی تناوب موتورها باشد. بنابراین، پنجره زمانی ای با طول ۲۲۰۰ نمونه انتخاب می شود که پس از اعمال بر دادگان موجود، به تولید ۴۹۹۸ نمونه داده هر یک به طول ۲۲۰۰ منجر می شود.

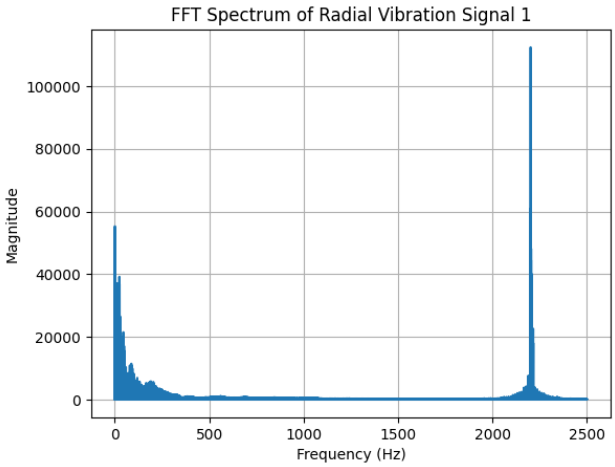
با در اختیار داشتن این نمونه ها و ذخیره کردن برچسب های آنها به عنوان متغیر labels، می توانیم به مرحله ی بعد که استخارج ویژگی است بپردازیم.

۳.۳.۲ استخراج ویژگی

ویژگی هایی که در این بخش استخراج می شوند، مطابق با مطالب نوشته شده در پایان نامه، شامل ویژگی های حوزه زمان و فرکانس برای سری زمانی های به دست آمده خواهند بود. رویکرد مورد استفاده در این بخش، محاسبه ی تمامی ویژگی های ذکر شده برای این داده ها بوده که پس از این در



شکل ۲۰: FFT for Axial Vibration Signal 2



شکل ۲۱: FFT for Radial Vibration Signal 1

فرایندهای انتخاب ویژگی، موثر ترین آ«نها انتخاب می شود.

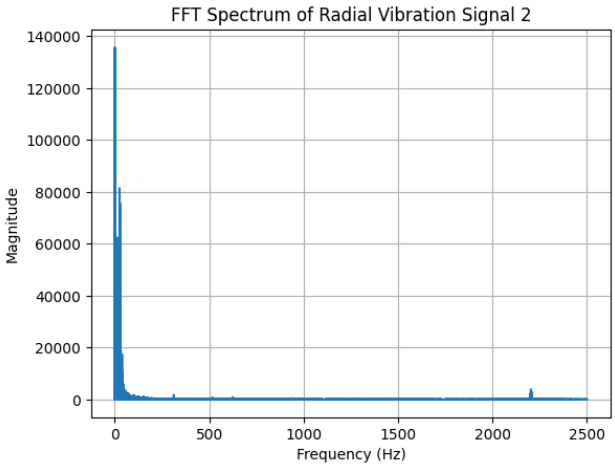
- ویژگی های حوزه زمان

با در اختیار داشتن سیگنال های سری زمانی، ویژگی های آماری زیر مطابق با فرمول های قرار داده شده استخراج شده اند.

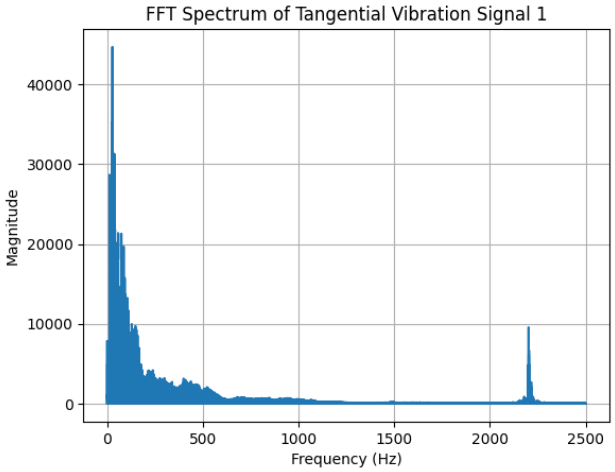
$$\text{Mean:} \quad \mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\text{Deviation: Standard} \quad \sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2}$$

$$\text{Square: Mean Root} \quad \text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$



شکل ۲۲: Signal Vibration Radial for FFT ۲



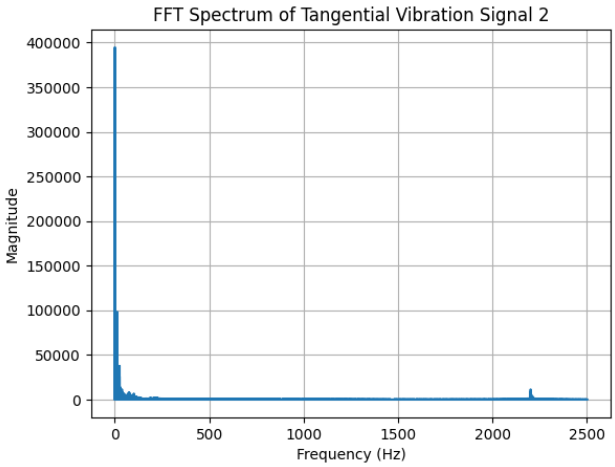
شکل ۲۳: Signal Vibration Tangential for FFT ۱

$$\text{Kurtosis:} \quad K = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^4}{\sigma^4}$$

$$\text{Skewness:} \quad S = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^3}{\sigma^3}$$

$$\text{Factor: Crest} \quad \text{CF} = \frac{\max(|x_i|)}{\text{RMS}}$$

$$\text{Factor: Shape} \quad \text{SF} = \frac{\text{RMS}}{|\mu|}$$



شکل ۲۴: Signal Vibration Tangential for FFT ۲

```
Total segments created: 4998
First segment preview:
   Axial Vibration Signal 1  Radial Vibration Signal 1  \
0                   1.29000                  0.246300
1                  -2.02510                 -0.902780
2                   0.61405                  0.335120
3                  -1.08860                 -0.760080
4                  -0.64317                  0.062457

   Tangential Vibration Signal 1  Axial Vibration Signal 2  \
0                       0.017209                  -0.64557
1                      -0.125530                  -0.71259
2                       0.000702                  -0.65959
3                      -0.068845                  -0.69720
4                       0.034943                  -0.68747

   Radial Vibration Signal 2  Tangential Vibration Signal 2 Fault_Type
0                   0.077080                      -0.10002          1
1                   0.061588                      -0.16430          1
2                   0.066658                      -0.11457          1
3                   0.065130                      -0.15646          1
4                   0.067832                      -0.14811          1
Label for the first segment: 1
```

شکل ۲۵: dataframe Segmented

$$\text{Rate: Crossing Zero} \quad \text{ZCR} = \frac{1}{2N}\sum_{i=1}^{N-1}|\text{sign}(x_{i+1}) - \text{sign}(x_i)|$$

$$\text{Deviation: Absolute Mean} \quad \text{MAD} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \mu|$$

$$\text{Range: Interquartile} \quad \text{IQR} = Q_3 - Q_1$$

$$\text{Variation: of Coefficient} \quad \text{CV} = \frac{\sigma}{\mu}$$

با محاسبه ی این ویژگی ها برای هر یک از سیگنال ها، در نهایت به دیتافریمی با ۶۶ ستون ویژگی دست خواهیم یافت.

- ویژگی های حوزه فرکانس

مشابه بخش قبل، ویژگی های حوزه فرکانس طبق روابط زیر محاسبه می شوند. در اینجا نیز تلاش شده است تا تمامی ویژگی های ممکن در این گام ابتدایی استخراج شوند.

$$\text{Values: FFT} \quad X_k = \left| \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N} \right|$$

$$\text{Frequencies:} \quad f_k = \frac{k}{N \cdot \Delta t}, \quad \Delta t = \frac{1}{\text{rate sampling}}$$

$$\text{Spectrum: Power} \quad P_k = |X_k|^2$$

$$\text{Spectrum: Power Normalized} \quad \hat{P}_k = \frac{P_k}{\sum_{j=0}^{N-1} P_j}$$

$$\text{Coefficients: FFT Five First of Amplitude} \quad \{X_0, X_1, X_2, X_3, X_4\}$$

$$\text{Amplitude: Maximum} \quad \max_k(|X_k|)$$

$$\text{Variance: Frequency} \quad \text{Var}_f = \frac{1}{N} \sum_{k=0}^{N-1} (|X_k| - \bar{X})^2$$

$$\text{Entropy: Frequency} \quad H_f = -\sum_{k=0}^{N-1} \hat{P}_k \log(\hat{P}_k + \varepsilon)$$

$$\text{Frequency: Dominant} \quad f_d = f_{\arg\max_k |X_k|}$$

$$\text{Skewness: Spectral} \quad \text{Skew}_f = \frac{1}{N} \sum_{k=0}^{N/2-1} \left(\frac{|X_k| - \bar{X}}{\sigma} \right)^3$$

$$\text{Kurtosis: Spectral} \quad \text{Kurt}_f = \frac{1}{N} \sum_{k=0}^{N/2-1} \left(\frac{|X_k| - \bar{X}}{\sigma} \right)^4$$

$$\text{Flux: Spectral} \quad \Phi = \sum_{k=1}^{N/2-1} (|X_k| - |X_{k-1}|)^2$$

$$\text{Slope: Spectral} \quad \text{Slope} = \text{polyfit}(f_k, \log(1 + |X_k|), 1)_{[0]}$$

$$\text{Crest: Spectral} \quad \text{Crest}_f = \frac{\max_k |X_k|}{\sum_{k=0}^{N/2-1} |X_k|}$$

$$\text{Centroid: Spectral} \quad C = \frac{\sum_{k=0}^{N/2-1} f_k \cdot |X_k|}{\sum_{k=0}^{N/2-1} |X_k|}$$

$$\text{Bandwidth: Spectral} \quad B = \sqrt{\frac{\sum_{k=0}^{N/2-1} (f_k - C)^2 \cdot |X_k|}{\sum_{k=0}^{N/2-1} |X_k|}}$$

بنابراین، در این فرایند در مجموع ۹۶ ویژگی از سیگنال های موجود در دادگان استخراج می شود.

- تشکیل دیتافریم ویژگی ها

از به هم پیوستن و Concat کردن این ویژگی ها، در نهایت دیتافریم ویژگی ها محاسبه می شود. با این حال، هنوز دانش کافی مبنی بر میزان مفید بودن هر یک از این دیتافریم ها در دست نیست، به این دلیل، در بخش بعد با استفاده از روش های مربوطه، میزان اهمیت ویژگی ها محاسبه می شود.

۴.۳.۲ انتخاب ویژگی

با در اختیار داشتن تعداد زیادی از ویژگی ها، انتخاب ویژگی های موثر از اهمیت زیادی برخوردار خواهد بود. برای ارزیابی میزان اهمیت هر یک از این ویژگی ها، از کتابخانه ی LightGBM و دستور های موجود در آن برای امتیازدهی به هر یک از ویژگی ها بر اساس معیار های مورد استفاده در این کتابخانه استفاده شده است.

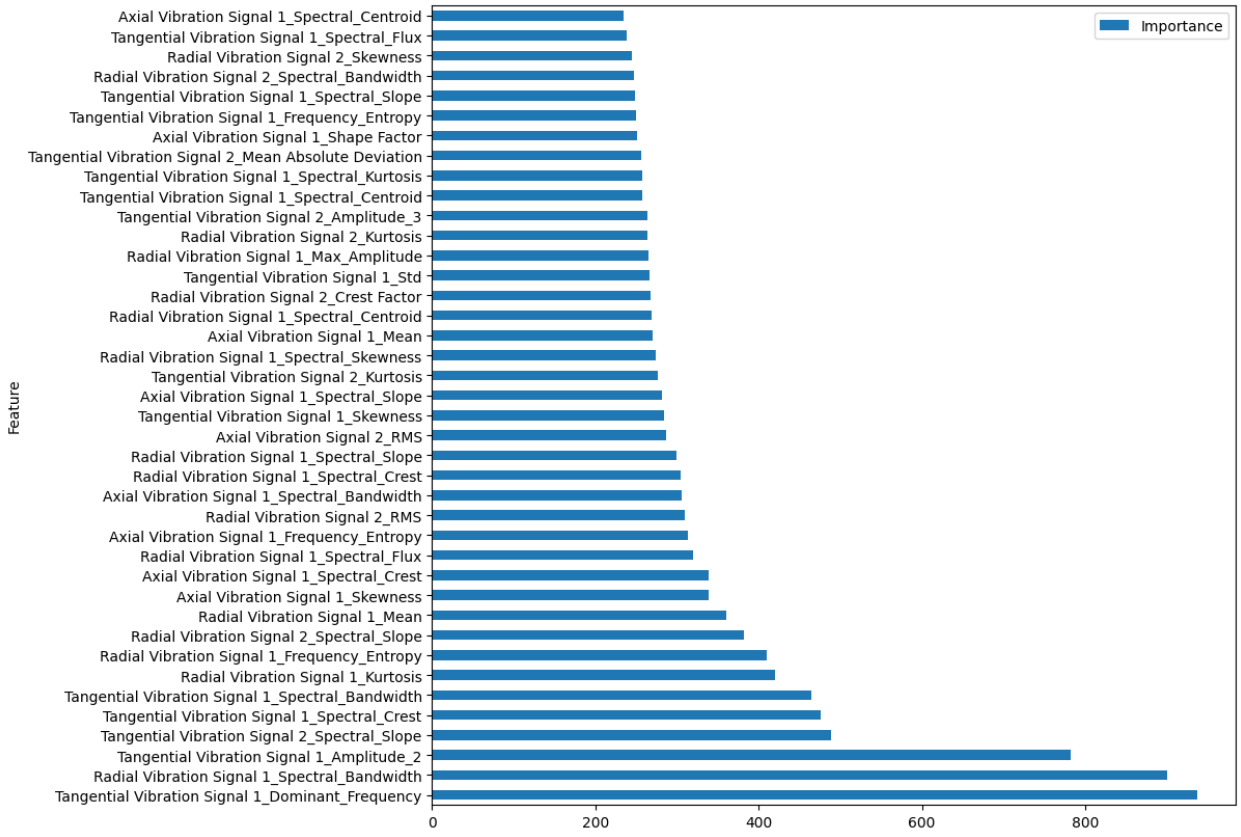
پس از فیت کردن این مدل بر داده های موجود در هر یک از ویژگی ها، در پایان مقدار اهمیت هر یک از ویژگی ها مشخص شده است که با مرتب کردن آن و نمایش ویژگی ها، می توانیم تصمیم درستی برای نگه داشتن ویژگی های موثر و حذف ویژگی های دیگر داشته باشیم. در شکل ۲۶ ۴۰ ویژگی اول که بیشترین امتیاز را از نظر اهمیت داشته اند نمایش داده شده اند. با استفاده از نتایج این بررسی و مشاهده ی امتیازات هر یک از ویژگی ها، تصمیم بر آن داریم تا تنها ویژگی هایی را که به طور مشخص بر عملکرد سیستم تاثیر گذار هستند و یا عملکرد آن را به خوبی بیان می کنند نمایش دهیم. پس از تکرار مراحل اموزش مدل و بررسی دقت آنها، به این نتیجه رسیده ایم که در صورت نگه داشتن ویژگی هایی با امتیاز اهمیت بالاتر از ۴۰۰، مدل می تواند دقت و وعملکرد مناسبی داشته باشد. بنابراین، ویژگی های مورد استفاده از این پس در این پژوهش به شرح نمایش داده شده در شکل ۲۷ خواهد بود:

۴.۲ آموزش مدل یادگیری ماشین

پس از تکمیل مراحل پیش پردازش داده ها و استخراج ویژگی ها، مدل های یادگیری ماشین برای تشخیص خطاهای ممکن در این آزمایش آموزش داده می شوند. در ادامه ی این گزارش، به شرح دو مدل یادگیری ماشین مجزا خواهیم پرداخت.

۱.۴.۲ مدل کلاس بندی کامل

در این بخش، بدون نیاز به تغییر دیتافریم های ویژگی ها و برچسب ها، مراحل آموزش مدل را آغاز می کنیم.



شکل ۲۶: importance LightGBM

	Tangential Vibration Signal 1_Dominant_Frequency	Radial Vibration Signal 1_Spectral_Bandwidth	Tangential Vibration Signal 1_Amplitude_2	Tangential Vibration Signal 2_Spectral_Slope	Tangential Vibration Signal 1_Spectral_Crest	Tangential Vibration Signal 1_Spectral_Bandwidth	Radial Vibration Signal 1_Kurtosis	Radial Vibration Signal 1_Frequency_Entropy
0	0.440909	0.185919	10.747357	-2.131289	0.031606	0.139841	1.901589	1.973746
1	-0.440909	0.184653	10.923249	-2.112860	0.029237	0.132172	1.907050	1.925343
2	0.440909	0.182893	4.438001	-1.947116	0.030443	0.132012	1.979936	2.013758
3	0.440909	0.185929	4.998141	-1.856708	0.030838	0.131750	2.230070	2.144462
4	0.440909	0.186133	4.702931	-1.940394	0.025934	0.130369	2.158870	2.158046
...	...	...	...	...	...	...	...	...
4993	0.440909	0.184119	7.411056	-3.270246	0.021311	0.145555	2.013406	3.022726
4994	0.440909	0.182951	9.177093	-2.863172	0.022012	0.141427	2.072137	2.928927
4995	-0.440909	0.186577	4.029003	-3.021927	0.018477	0.137852	2.199736	2.997065
4996	-0.440909	0.185274	4.137302	-2.363640	0.017775	0.135062	2.216920	3.055064
4997	0.440909	0.187527	2.802090	-3.670943	0.021725	0.140777	2.034238	2.957217

4998 rows × 8 columns

شکل ۲۷: features lgbm

۱. تقسیم بندی دادگان به داده های آموزش و تست

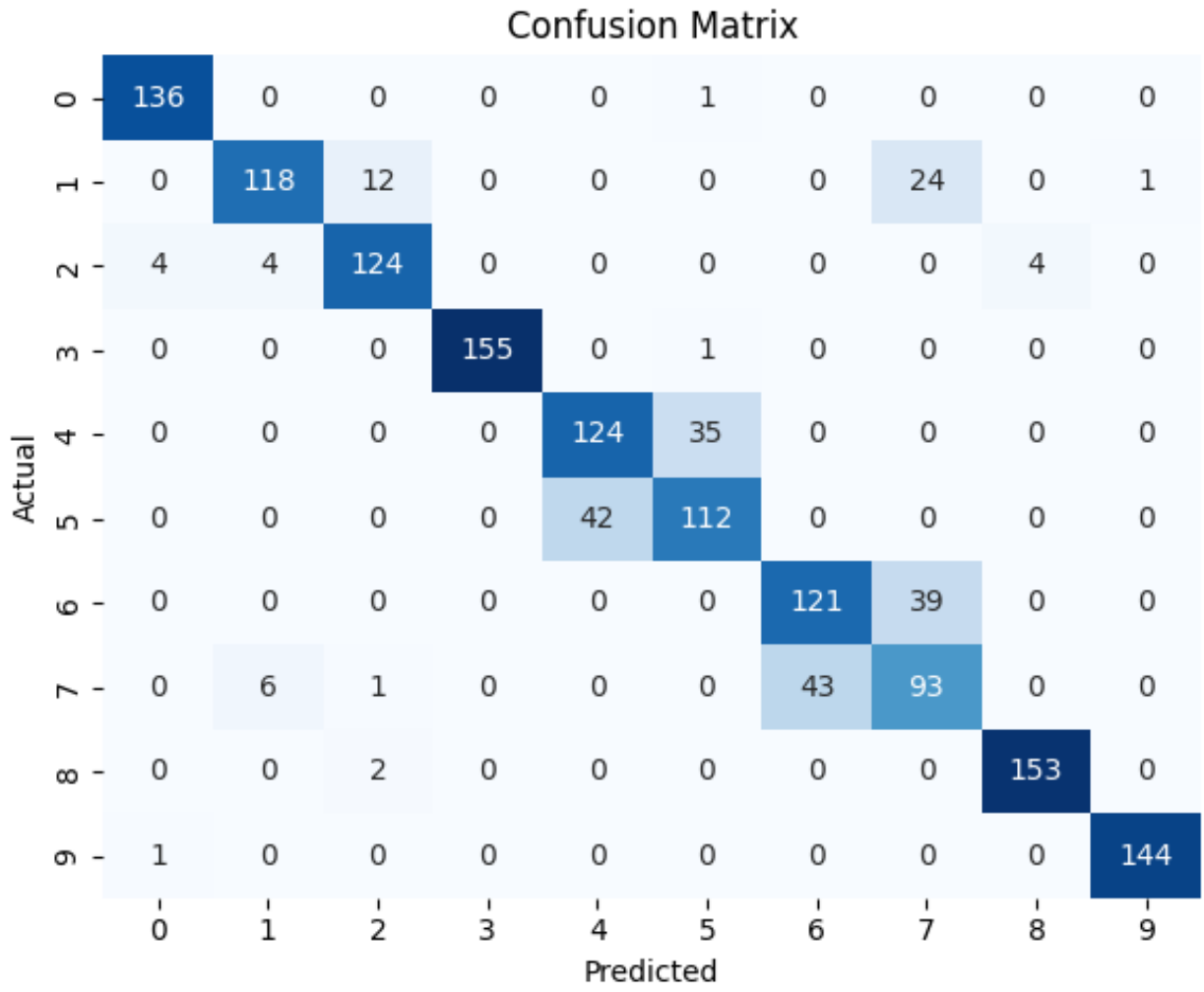
در این بخش، برای آنکه بتوانیم در پایان آموزش مدل، عملکرد آن را ارزیابی و بعد تر با عملکرد سایر مدل ها مقایسه کنیم، بخشی از دادگان را به عنوان داده ی تست جدا کرده و آموزش مدل را تنها بر داده های آموزشی انجام می دهیم. این فرایند با استفاده از روش Train-Test-Split از پکیج sklearn انجام می شود. در این پژوهش، ۳۰ درصد دادگان به داده های تست اختصاص یافته است.

۲. نرمال سازی دادگان

با توجه به ماهیت های متفاوت ویژگی ها، دامنه ی تغییرات هر یک از انها با دیگری متفاوت بوده و علاوه بر این، در صورتی که مقدارهای بسیار کوچک و یا بزرگی داشته باشند، می توانند بر عملکرد مدل نهایی اثر منفی داشته باشند. بنابراین، در این قسمت با استفاده از روش StandardScaler از کتابخانه ی sklearn، داده های آموزشی را نرمالایز کرده و سپس همین مقادیر محاسبه شده را بر داده های تست نیز اعمال می کنیم.

۳. آموزش مدل کلاس بند خطی Regression Logestic

در این بخش، به آموزش مدل Regression Logistic بر اساس داده های آماده شده در بخش های قبل خواهیم پرداخت. پیاده سازی این مدل به راحتی توسط مدل های قرار داده شده در پکیج sklearn نوشته شده است. پس از تعریف و آموزش مدل، عملکرد آن را ارزیابی می کنیم. مشاهده می شود که این مدل توانسته است دقت ۸۵ درصد را به دست بیاورد. گزارش های دقیق تر از عملکرد این مدل در شکل های ۲۹ و ۲۸ نمایش داده شده اند.



شکل ۲۸: confusion class all

۴. آموزش مدل کلاس بند درخت تصمیم در تلاشی دیگر، برای دسته بندی کلاس های موجود در این دادگان؛ از یک درخت تصمیم استفاده می شود. عمق تعیین شده برای این درخت، با استفاده چند مرحله آزمون و خطا برابر با ۶ در نظر گرفته شده است. در ادامه، نمای کلی این

```
Accuracy: 0.8533333333333334
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       137
           1       0.92      0.76      0.83       155
           2       0.89      0.91      0.90       136
           3       1.00      0.99      1.00       156
           4       0.75      0.78      0.76       159
           5       0.75      0.73      0.74       154
           6       0.74      0.76      0.75       160
           7       0.60      0.65      0.62       143
           8       0.97      0.99      0.98       155
           9       0.99      0.99      0.99       145

    accuracy                           0.85      1500
   macro avg       0.86      0.86      0.86      1500
weighted avg       0.86      0.85      0.85      1500
```

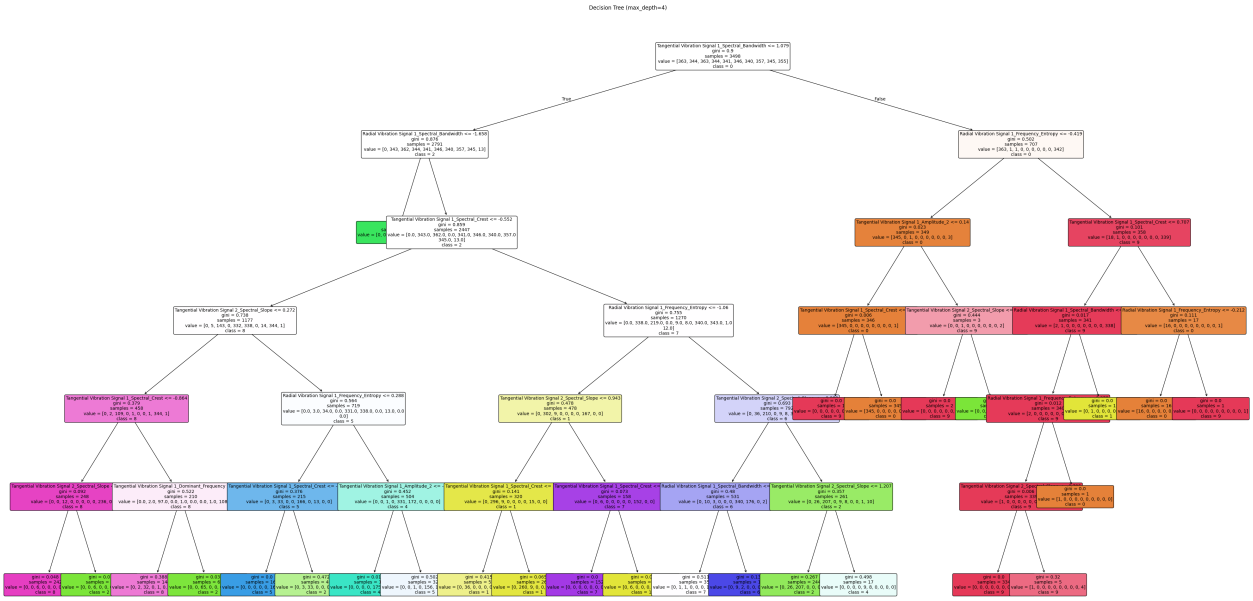
شکل ۲۹: report class all

درخت و گزارش های عملکرد آن را خواهیم دید. دقت این مدل نیز برابر با ۸۵ درصد بمحاسبه شده است.

۵.۲ طبقه‌بند سلسله‌مراتبی پیشنهادی

برای آموزش یک طبقه‌بند سلسله‌مراتبی، با فرض در اختیار داشتن داده‌ها، لازم است ابتدا مجموعه داده و برچسب‌های آن به‌گونه‌ای اصلاح شوند که امکان آموزش مدل‌های مختلف فراهم گردد. در گام ابتدایی، نوع خطا به عنوان یک ستون جدید به چارچوب ویژگی‌ها (DataFrame) افزوده می‌شود تا بتوان در مراحل بعدی از این برچسب برای پردازش‌های سلسله‌مراتبی بهره گرفت.

در زیربخش‌های آتی، به نحوه نگاشت برچسب‌ها و فرآیند آموزش مدل‌ها خواهیم پرداخت. فرآیند آموزش شامل دو مرحله اصلی است: در مرحله نخست، هدف مدل شناسایی نوع کلی خطا یا حالت نرمال سیستم است. در مرحله دوم، مدل‌هایی آموزش داده می‌شوند که مسئول تشخیص دقیق نوع هر خطا در صورت وجود هستند.



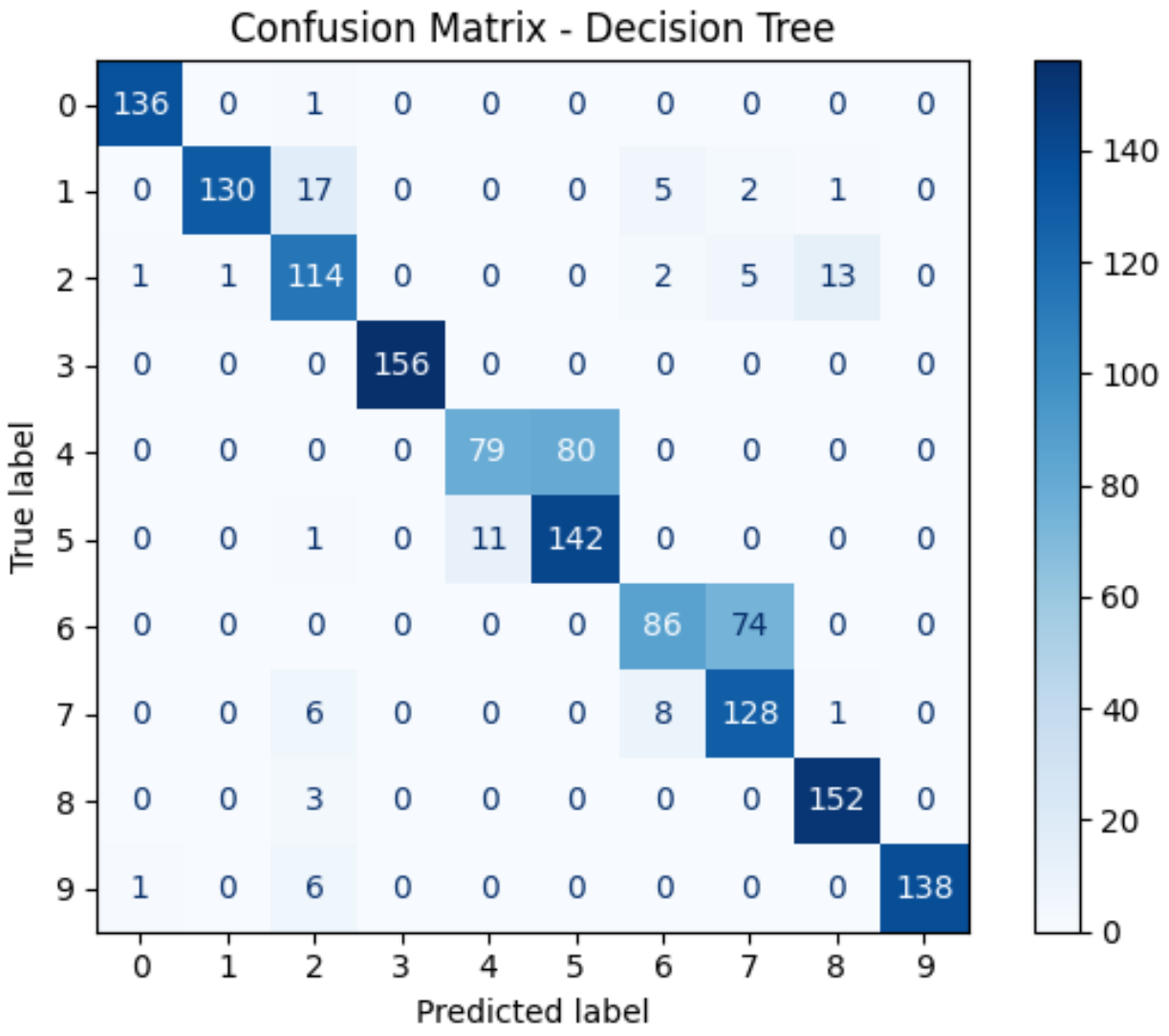
شکل ۳۰: Tree Decision

۱.۵.۲ آموزش مدل مرحله اول

برای آموزش مدل مرحله اول که نقش مدل اولیه را ایفا می‌کند، لازم است نگاشتی از انواع خطا به مجموعه‌ای کلی‌تر در نظر گرفته شود. در این راستا، دسته‌بندی‌های کلی شامل «نرمال»، «ناهم‌ترازی» (Misalignment)، «کوتاهی محور» (Underhang)، «بلندی محور» (Overhang) و «عدم تعادل» (Imbalance) در نظر گرفته می‌شوند. نگاشت انواع اصلی خطا به این دسته‌بندی‌ها به صورت زیر تعریف می‌گردد:

```
type_mapping = {
    0: 10,  # Normal -> 10
    1: 11,  # Misalignment -> 11
    2: 11,  # Misalignment -> 11
    3: 12,  # Underhang -> 12
    4: 12,  # Underhang -> 12
    5: 12,  # Underhang -> 12
    6: 13,  # Overhang -> 13
    7: 13,  # Overhang -> 13
    8: 13,  # Overhang -> 13
    9: 14   # Imbalance -> 14
}
```

بر این اساس، ستونی جدید در داده‌ها ایجاد می‌شود که شامل نوع خطای نگاشته‌شده است. نکته‌ای که باید در این مرحله مدنظر قرار گیرد، ایجاد تعادل میان تعداد نمونه‌ها در هر دسته است. با توجه به آن‌که برخی از دسته‌های خطا شامل زیرگروه‌های بیشتری هستند و تعداد نمونه‌های آن‌ها بیشتر است، برای متعادل‌سازی داده‌ها از روش resample استفاده می‌شود.



شکل ۳۱: Confusion Tree Decision

در این فرآیند، با تعیین یک مقدار random_state مشخص، اطمینان حاصل می‌شود که تمامی زیرگروه‌های خطا در نمونه‌های نهایی حضور دارند. در نهایت، داده نهایی شامل ستونی با مقادیر بین ۱۰ تا ۱۴ خواهد بود که نشان‌دهنده نوع کلی خطا در هر نمونه است.

آموزش مدل مرحله اول

فرآیند آموزش مدل در این پروژه به رویکردی مشابه تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی و سپس نرمال‌سازی داده‌ها و در نهایت آموزش مدل شباهت دارد. در اینجا، به تشریح هر مرحله از آموزش مدل مرحله اول خواهیم پرداخت.

گام اول: تقسیم داده‌ها

در ابتدا، چارچوب داده‌ها با استفاده از روش `train_test_split` از کتابخانه‌ی پایتون به دو بخش آموزشی و آزمایشی تقسیم می‌شود. اندازه مجموعه آزمایشی ۳۰ درصد از کل داده‌ها در نظر گرفته می‌شود. علاوه بر این، از پارامتر `stratify` برای حفظ توزیع متوازن داده‌ها بر اساس برچسب‌ها (Labels) استفاده می‌شود تا داده‌ها به صورت متوازن در مجموعه‌های آموزشی و آزمایشی توزیع شوند.

گام دوم: نرمال‌سازی داده‌ها

پس از تقسیم داده‌ها، از `StandardScaler` از کتابخانه‌ی `sklearn` برای نرمال‌سازی داده‌های آموزشی استفاده می‌شود. سپس، مقیاس‌گذار (Scaler) به داده‌های آزمایشی اعمال می‌شود تا از نشت اطلاعات از داده‌های آموزشی به داده‌های آزمایشی جلوگیری گردد.

گام سوم: آموزش مدل

برای آموزش مدل از `LogisticRegression` استفاده می‌شود. در این مرحله، بیشینه تعداد تکرارها (`max_iter`) برای مدل ۱۰۰۰ در نظر گرفته می‌شود تا فرآیند آموزش به طور کامل انجام گیرد.

گام چهارم: ارزیابی مدل

پس از آموزش مدل، گزارش مربوط به دسته‌بندی در تصویر زیر نشان داده شده است. همچنین، ماتریس اشتباهات (Confusion Matrix) در تصویر دیگری نمایش داده شده است. طبق این داده‌ها، نتیجه می‌گیریم که مدل قادر است به‌طور کامل دسته‌بندی‌های «نرمال» و «عدم تعادل» را شناسایی کند. اما برای دیگر انواع خطا، اگرچه مدل دقت قابل قبولی دارد، در برخی موارد دچار اشتباهاتی می‌شود. به‌ویژه در کلاس‌های یک و سه (که به ترتیب مربوط به «ناهم‌ترازی» و «کوتاهی محور» هستند)، اشتباهاتی در طبقه‌بندی مشاهده می‌شود. این موضوع همچنین در گزارش دسته‌بندی و دقت هر نوع خطا قابل مشاهده است. (شکل های ۳۲ و ۳۳)

```
Accuracy: 0.9494736842105264
Classification Report:
              precision    recall  f1-score   support

          10       1.00      0.99      1.00       150
          11       0.82      0.87      0.84        75
          12       1.00      1.00      1.00        50
          13       0.80      0.74      0.77        50
          14       0.99      1.00      1.00       150

    accuracy                           0.95       475
   macro avg       0.92      0.92      0.92       475
weighted avg       0.95      0.95      0.95       475
```

شکل ۳۲: Report classification model Stage۱

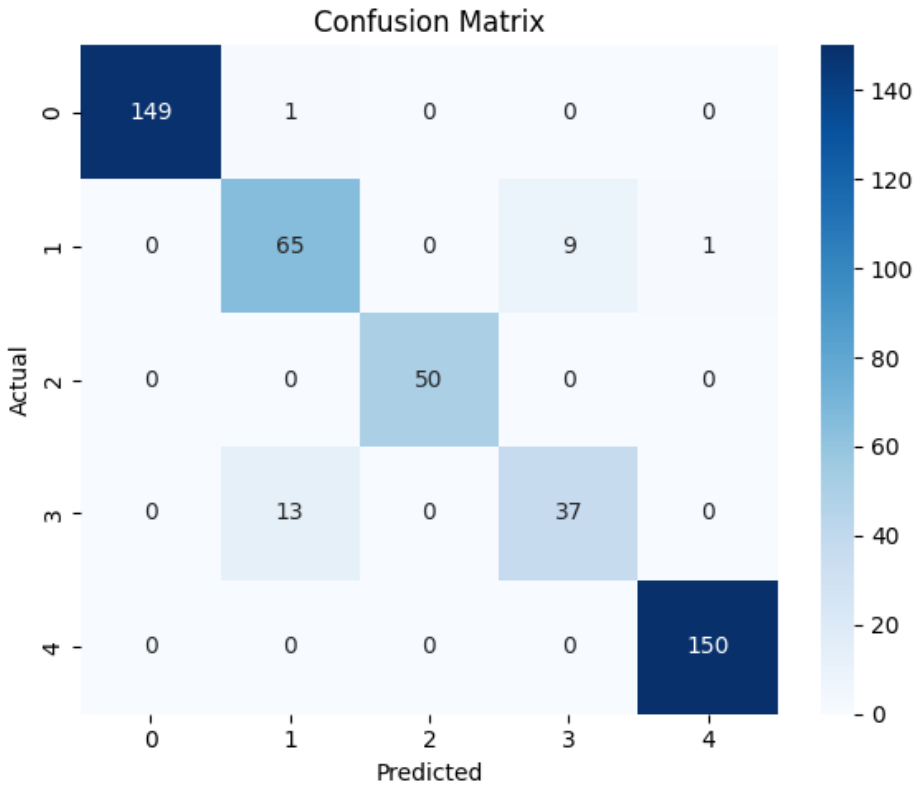
۲.۵.۲ مدل‌های مرحله دوم

در این مرحله، ما سه مدل اصلی داریم که مسئول شناسایی و آموزش انواع خطاهای دقیق «ناهم‌ترازی»، «کوتاهی محور» و «بلندی محور» هستند.

۱. ناهم‌ترازی (Misalignment)

برای ساخت مجموعه داده جهت آموزش مدل برای پیش‌بینی فقط نوع دقیق «ناهم‌ترازی»، که شامل دو نوع مختلف است، ابتدا مجموعه داده فیلتر می‌شود تا فقط ویژگی‌های مربوط به برچسب‌های یک و دو در آن باقی بماند.

پس از اطمینان از این که فقط این دو نوع خطا در ستون‌ها وجود دارند، مراحل مشابه مراحل قبلی برای آموزش مدل انجام می‌شود. به این صورت که ابتدا داده‌ها با استفاده از روش `train_test_split` به دو بخش آموزشی و آزمایشی تقسیم می‌شوند، با اندازه ۳۰ درصد برای داده‌های آزمایشی. سپس، داده‌ها با استفاده از `StandardScaler` نرمال‌سازی می‌شوند و مقیاس‌گذار (Scaler) به داده‌های آزمایشی نیز اعمال می‌شود.



شکل ۳۳: Confusion model Stage۱

س از انجام این مراحل، مدل `LogisticRegression` بر روی این مجموعه داده‌ها آموزش داده می‌شود. این مدل را `linear_model_Stage2_Misalignment` نام‌گذاری می‌کنیم.

گزارش‌های دسته‌بندی و ماتریس اشتباهات (Confusion Matrix) این مدل در تصاویر زیر ارائه شده است.

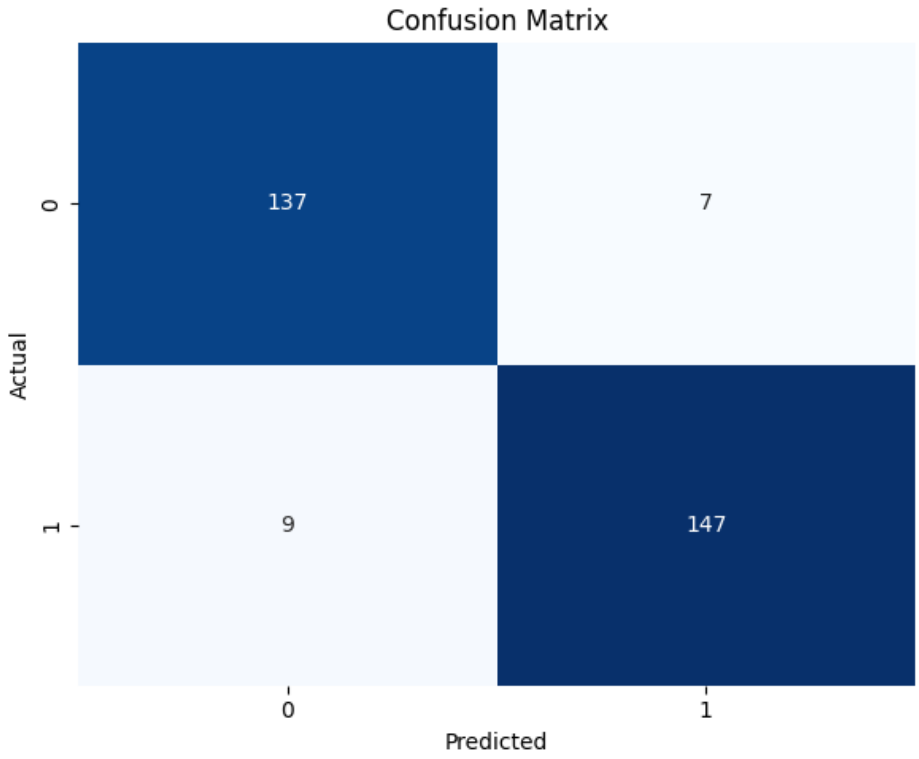
```
Accuracy: 0.9466666666666667
Classification Report:
              precision    recall  f1-score   support

           1       0.94      0.95      0.94       144
           2       0.95      0.94      0.95       156

    accuracy                           0.95       300
   macro avg       0.95      0.95      0.95       300
weighted avg       0.95      0.95      0.95       300
```

شکل ۳۴: Report classification misalignment

۲. بلندی محور (Overhang)



شکل ۳۵: Matrix Confusion misalignment

برای ساخت مجموعه داده جهت آموزش مدل برای پیش‌بینی نوع دقیق «بلندی محور»، از برچسب‌های ۳، ۴ و ۵ استفاده می‌شود. مشابه مدل‌های قبلی، داده‌ها ابتدا با استفاده از روش `train_test_split` به دو بخش آموزشی و آزمایشی تقسیم می‌شوند و سپس داده‌ها نرمال‌سازی می‌شوند. پس از این مراحل، مدل `LogisticRegression` بر روی داده‌های آموزشی آموزش داده می‌شود. این مدل را `linear_model_Stage2_Overhang` نام‌گذاری می‌کنیم.

گزارش‌های دسته‌بندی و ماتریس اشتباهات این مدل در تصاویر زیر نمایش داده می‌شود.

۳. کوتاهی محور (Underhang)

برای ساخت مجموعه داده جهت آموزش مدل برای پیش‌بینی نوع دقیق «کوتاهی محور»، از برچسب‌های ۶، ۷ و ۸ استفاده می‌شود. مشابه مدل «ناهم‌ترازی»، ابتدا داده‌ها با استفاده از روش `train_test_split` به دو بخش آموزشی و آزمایشی تقسیم می‌شوند. سپس، نرمال‌سازی داده‌ها با استفاده از `StandardScaler` انجام شده و مقیاس‌گذار (Scaler) به داده‌های آزمایشی اعمال می‌شود.

پس از این مراحل، مدل `LogisticRegression` بر روی مجموعه داده‌های آموزش دیده آموزش داده می‌شود. این مدل را `linear_model_Stage2_Underhang` نام‌گذاری می‌کنیم.

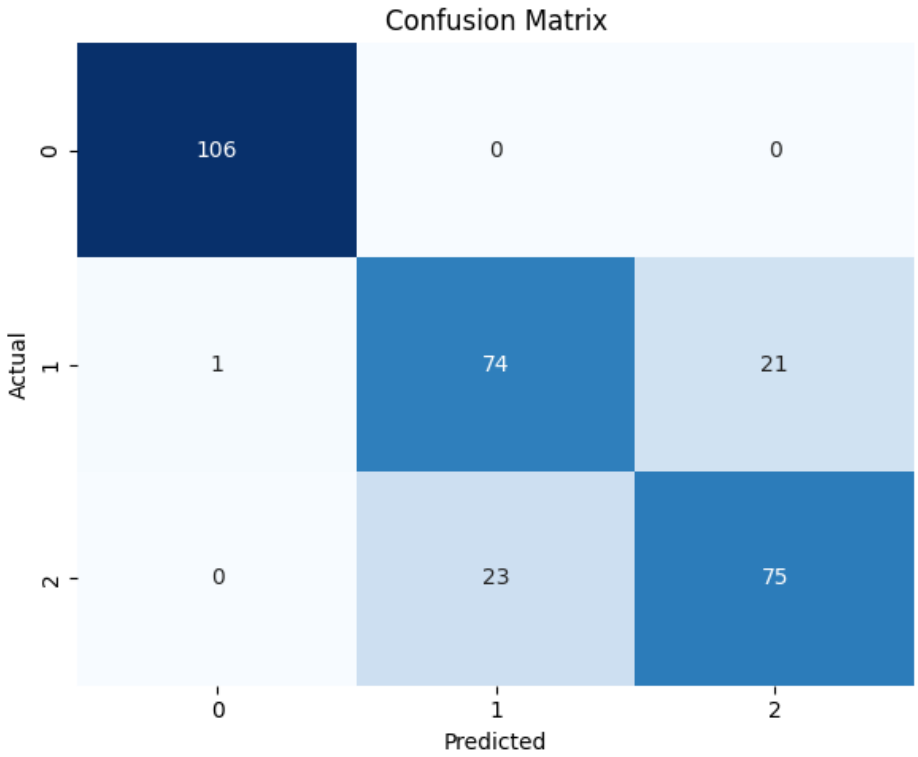
گزارش‌های دسته‌بندی و ماتریس اشتباهات این مدل در تصاویر زیر نشان داده شده است.

```
Accuracy: 0.85
Classification Report:
              precision    recall  f1-score   support

           3       0.99      1.00      1.00       106
           4       0.76      0.77      0.77        96
           5       0.78      0.77      0.77        98

    accuracy                           0.85       300
   macro avg       0.84      0.85      0.85       300
weighted avg       0.85      0.85      0.85       300
```

شکل ۳۶: REport classification Overhang



شکل ۳۷: Matrix confusion Overhang

۳.۵.۲ طبقه‌بندی سلسله‌مراتبی نهایی

پس از آموزش مدل‌هایی که در بخش‌های قبلی شرح داده شدند، یک ساختار طبقه‌بندی سلسله‌مراتبی طراحی شده است که بر پایه پیش‌بینی مدل مرحله اول عمل می‌کند.

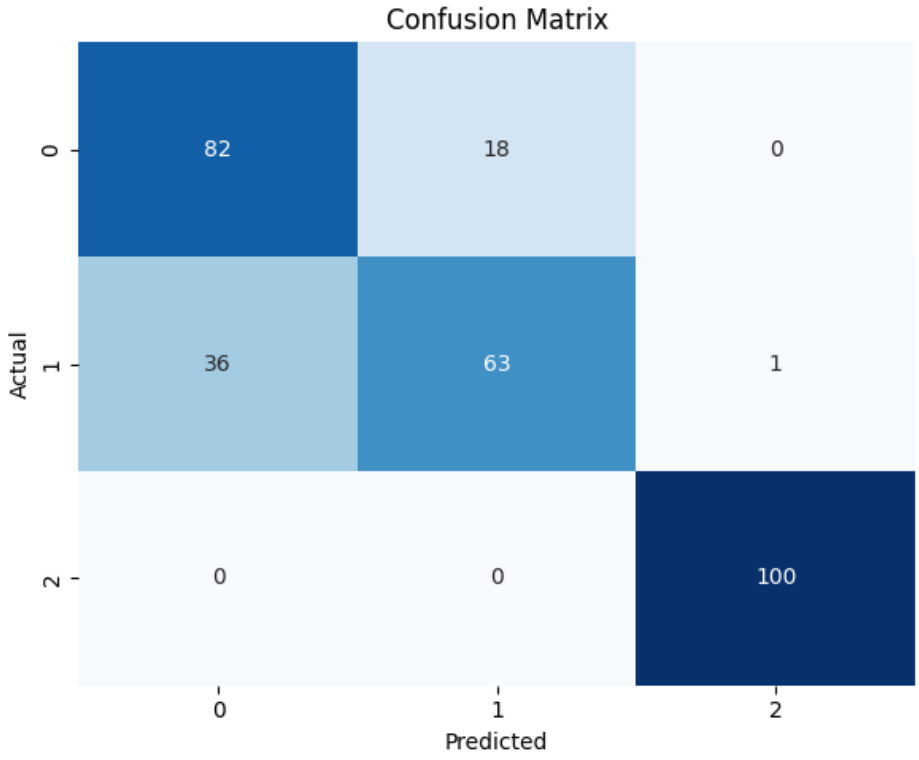
در این ساختار، ابتدا مدل مرحله اول سیگنال ورودی را تحلیل کرده و تشخیص می‌دهد که آیا سیگنال نرمال است یا دارای نوعی از خطاهای

```
Accuracy: 0.8166666666666667
Classification Report:
              precision    recall  f1-score   support

           6       0.69      0.82      0.75       100
           7       0.78      0.63      0.70       100
           8       0.99      1.00      1.00       100

    accuracy                           0.82       300
   macro avg       0.82      0.82      0.81       300
weighted avg       0.82      0.82      0.81       300
```

شکل ۳۸: Report classification Underhang



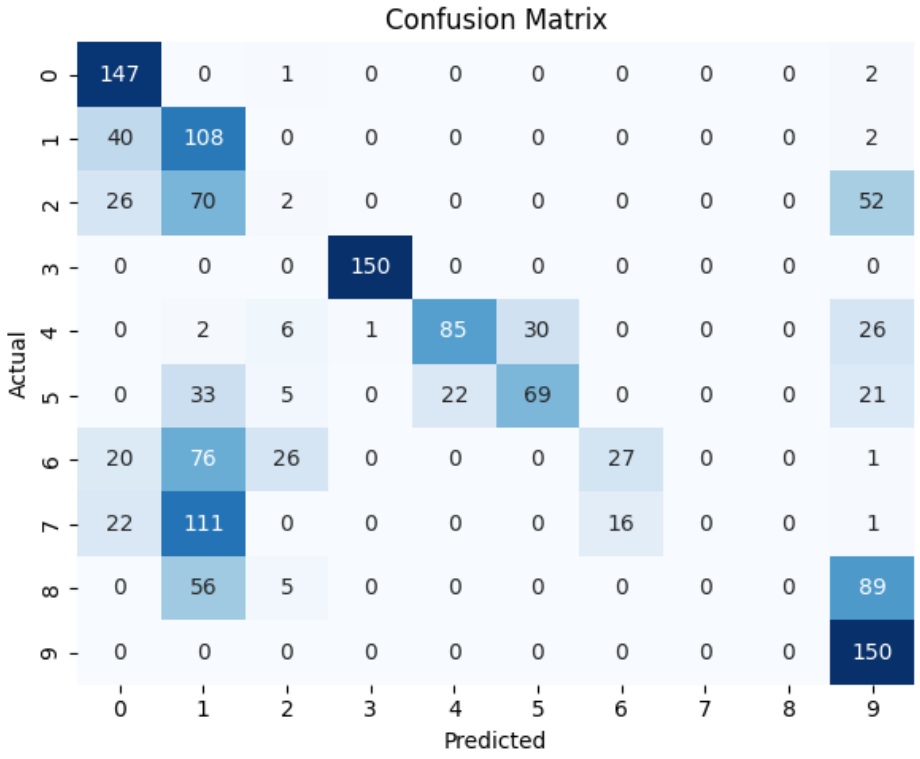
شکل ۳۹: Confusion Underhang

اصلی (مانند ناهم‌ترازی، کوتاهی محور، بلندی محور یا عدم تعادل) می‌باشد. در صورتی که سیگنال نرمال تشخیص داده شود، خروجی نهایی به عنوان وضعیت نرمال ثبت می‌گردد. اما در صورتی که نوع کلی خطا شناسایی شود، بسته به آن، یکی از مدل‌های آموزش‌دیده در مرحله دوم فراخوانی می‌شود تا نوع دقیق خطا مشخص گردد.

به طور خلاصه:

- اگر پیش‌بینی مرحله اول «نرمال» باشد، خروجی نهایی همان کلاس نرمال است.
- اگر پیش‌بینی «ناهم‌ترازی» باشد، نمونه به مدل مرحله دوم مربوطه داده می‌شود تا نوع دقیق ناهم‌ترازی مشخص شود.
- اگر پیش‌بینی «کوتاهی محور» باشد، نمونه به مدل مربوط به کوتاهی محور داده می‌شود.
- اگر پیش‌بینی «بلندی محور» باشد، مدل مرتبط با بلندی محور فراخوانی می‌شود.
- اگر پیش‌بینی «عدم تعادل» باشد، مستقیماً به عنوان آن کلاس در نظر گرفته می‌شود.

این منطق باعث می‌شود که تصمیم‌گیری به صورت سلسله‌مراتبی انجام شده و دقت تشخیص نوع خطا بهبود یابد.
در ادامه، نتایج به‌دست‌آمده از این طبقه‌بند سلسله‌مراتبی، شامل گزارش دسته‌بندی و ماتریس اشتباهات، ارائه می‌گردد.



شکل ۴۰: matrix confusion hierarchical

تحلیل نتایج و ماتریس اشتباهات

اگرچه هدف اصلی این پژوهش تمرکز بر طراحی و پیاده‌سازی ساختار سلسله‌مراتبی طبقه‌بندی است، اما لازم است مروری کلی بر عملکرد نهایی مدل‌ها و به‌ویژه تحلیل نتایج حاصل از ماتریس اشتباهات (Confusion Matrix) داشته باشیم.

بر اساس نتایج به‌دست‌آمده، می‌توان دریافت که مدل مرحله اول به‌طور قابل‌توجهی در تشخیص سیگنال‌های «نرمال» و همچنین سیگنال‌های مرتبط با «عدم تعادل» عملکرد دقیقی دارد؛ این موضوع در ماتریس اشتباهات به‌وضوح قابل مشاهده است.

با این حال، در سایر دسته‌بندی‌ها دقت مدل با کاهش مواجه است. به‌طور خاص، مدل مرحله دوم مربوط به «ناهم‌ترازی» که در مرحله آموزش عملکرد خوبی از خود نشان داده بود، در فرآیند طبقه‌بندی سلسله‌مراتبی عملکرد قابل قبولی ندارد و به‌ویژه در تشخیص نوع دوم ناهم‌ترازی دچار خطا می‌شود.

در مورد مدل مربوط به «بلندی محور»، مشاهده می‌شود که این مدل به‌خوبی قادر به شناسایی نوع اول خطا (Ball Fault) است، اما در شناسایی سایر انواع دچار چندین مورد پیش‌بینی اشتباه است.

از طرف دیگر، مدل مربوط به «کوتاهی محور» عملکرد مناسبی ندارد. این ضعف به‌ویژه در تشخیص کلاس‌های ۸ و ۲ مشهود است که به اشتباه با کلاس ۹ (مربوط به عدم تعادل) اشتباه گرفته شده‌اند. این مسئله نشان‌دهنده‌ی هم‌پوشانی الگوهای ویژگی در برخی از انواع خطا و نیاز به بهبود بیشتر در طراحی مدل‌های مرحله دوم است.

در مجموع، می‌توان گفت که ساختار سلسله‌مراتبی پیشنهادی در شناسایی کلی نوع خطا عملکرد خوبی دارد، اما دقت در شناسایی نوع دقیق برخی از خطاها نیازمند بهبود و بازنگری است.

۶.۲ محصول

در این بخش، فرآیند پردازش داده‌ها و پیش‌بینی بر اساس مدل آموزش‌دیده‌شده به‌صورت گام‌به‌گام توضیح داده می‌شود. این فرآیند در قالب یک تابع کلی که تمامی مراحل را به‌طور خودکار انجام می‌دهد، به‌طور مختصر شرح داده خواهد شد. این تابع قادر است یک فایل CSV ورودی که شامل داده‌های نمونه است را پردازش کرده و برچسب پیش‌بینی شده برای هر نمونه را ارائه دهد.

گام اول: خواندن فایل CSV و پردازش اولیه داده‌ها در ابتدا، فایل CSV ورودی بارگذاری شده و داده‌های لازم استخراج می‌شوند. در این مرحله، ستون‌های غیرضروری مانند سیگنال تاکومتر و سیگنال میکروفون از داده‌ها حذف می‌گردند.

گام دوم: تقسیم‌بندی داده‌ها در این گام، داده‌ها با استفاده از همان کلاس‌هایی که قبلاً برای تقسیم‌بندی طراحی شده‌اند، بر اساس اندازه پنجره و اندازه گام مشابه به دسته‌های کوچک‌تر تقسیم می‌شوند.

گام سوم: استخراج ویژگی‌ها در این مرحله، ویژگی‌های مربوط به دامنه زمان و دامنه فرکانس از داده‌های تقسیم‌شده استخراج می‌شوند. این ویژگی‌ها با استفاده از کلاس‌هایی که پیش‌تر برای استخراج ویژگی‌ها طراحی شده‌اند، انجام می‌شود.

گام چهارم: انتخاب ویژگی‌ها در این گام، ویژگی‌های انتخاب‌شده با توجه به دانشی که از مدل LightGBM بدست آمده‌اند، فیلتر شده و فقط ستون‌های مهم از داده‌های استخراج شده انتخاب می‌شوند. این ویژگی‌های منتخب به‌طور دستی در کد ذخیره می‌شوند و فقط همین ویژگی‌ها از فریم داده‌ها استخراج و ذخیره می‌گردند.

گام پنجم: نرمال‌سازی داده‌ها داده‌های ویژگی‌ها نرمال‌سازی می‌شوند. در این مرحله، از اسکیلری که در طی فرایند آموزش مدل ذخیره شده است، برای نرمال‌سازی داده‌های ورودی استفاده می‌شود. این کار به این منظور است که هیچ‌گونه اطلاعات جدیدی از داده‌ها به مدل منتقل نشود. در اینجا از بسته `joblib` برای ذخیره و بارگذاری اسکیلر استفاده می‌شود.

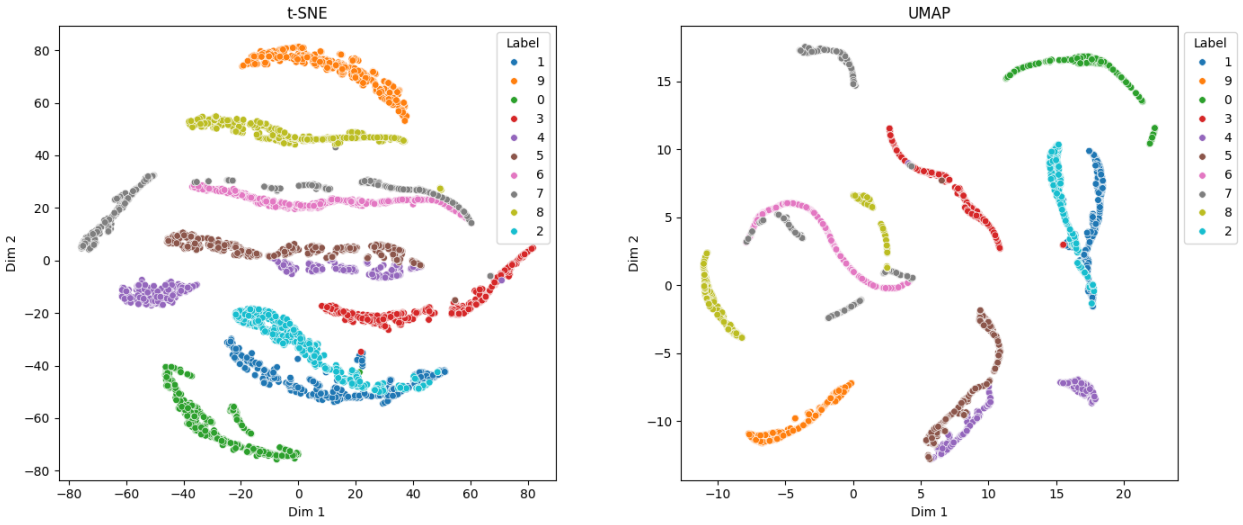
گام ششم: پیش‌بینی برچسب‌ها پس از انجام تمام مراحل بالا، داده‌ها به‌طور کامل پردازش شده و ویژگی‌های نرمال‌شده آماده پیش‌بینی هستند. در این مرحله، مدل ذخیره‌شده که پیش‌تر با استفاده از بسته `joblib` ذخیره شده است، بارگذاری شده و پیش‌بینی برچسب‌ها انجام می‌شود.

تابع Pipeline Full تمامی این مراحل در یک تابع کلی به نام `Pipeline Full` جمع‌آوری شده‌اند که به‌طور خودکار تمامی مراحل پردازش را انجام می‌دهد. این تابع ابتدا داده‌های خام را بارگذاری کرده، سپس آن‌ها را تقسیم‌بندی کرده، ویژگی‌ها را استخراج و فیلتر می‌کند، سپس داده‌ها را نرمال می‌سازد و در نهایت مدل ذخیره‌شده را برای پیش‌بینی بارگذاری می‌کند.

در نهایت، با استفاده از این روش، یک فریم داده به نام `features normalized` تولید می‌شود که شامل ویژگی‌های نرمال‌شده است و می‌تواند برای پیش‌بینی استفاده شود.

۷.۲ نمایش t-SNE و UMap

در این بخش، با استفاده از دیتافریم فیچر هایی که در بخش قبل در فرایند استخراج ویژگی به دست آمده اند، نمودار t-sne و Umap آن را برای بررسی شباهت و تفکیک پذیری کلاس های مورد استفاده در این دیتاست، رسم می کنیم.



شکل ۴۱: Umap and TSNE

۱.۷.۲ تحلیل تصویری با استفاده از t-SNE و UMAP

برای ارزیابی کیفیت ویژگی‌های استخراج‌شده و بررسی قابلیت جداسازی کلاس‌ها، از دو تکنیک تصویری رایج، یعنی t-SNE و UMAP استفاده شده است. این روش‌ها با نگاشت داده‌های چندبعدی به فضای دوبعدی، امکان درک بهتر ساختار داده و رفتار مدل‌ها را فراهم می‌کنند.

تحلیل t-SNE

نتایج حاصل از تحلیل t-SNE نشان‌دهنده شکل‌گیری خوشه‌های فشرده و خمیده برای اکثر کلاس‌هاست که نشان‌دهنده حفظ ساختار محلی داده‌ها توسط این روش است. برخی از مشاهدات کلیدی عبارتند از:

- کلاس ۹ (نارنجی) در بالای تصویر کاملاً مجزا قرار دارد و به خوبی تفکیک شده است.
- کلاس ۰ (سبز) در پایین-چپ تصویر به وضوح قابل شناسایی است.
- کلاس‌های ۱ (آبی روشن) و ۲ (خاکستری) در مجاورت یکدیگر قرار دارند اما هم‌پوشانی زیادی ندارند.
- کلاس‌های ۶، ۷ و ۲ دارای هم‌پوشانی نسبی هستند که می‌تواند به دلیل ویژگی‌های مشابه یا محدودیت روش t-SNE در نمایش ساختار جهانی داده‌ها باشد.

همچنین، t-SNE به دلیل فشرده‌سازی ساختار جهانی، نمی‌تواند فاصله‌ی بین خوشه‌ها را به‌درستی تفسیر کند. به‌عنوان مثال، نزدیکی کلاس‌های ۳ و ۴ در تصویر لزوماً به معنای شباهت آن‌ها در فضای ویژگی اصلی نیست.

تحلیل UMAP

در مقابل، نتایج حاصل از UMAP نشان می‌دهد که این روش علاوه بر حفظ ساختار محلی، ساختار جهانی را نیز بهتر حفظ می‌کند. در این نگاشت، خوشه‌ها شکلی کشیده‌تر دارند که نشان‌دهنده‌ی انتقال تدریجی نمونه‌ها در فضای اصلی است. برخی نکات مهم عبارتند از:

- کلاس‌های ۱، ۳ و ۹ به‌صورت خوشه‌های متمایز و واضح قابل مشاهده هستند.
- کلاس‌های ۲، ۶ و ۷ اگرچه پراکندگی دارند، اما هنوز قابل تفکیک هستند.
- بین کلاس‌های ۶ و ۷ تماس‌هایی وجود دارد که ممکن است در طبقه‌بندی نیز منجر به اشتباه شود.
- کلاس ۲ دارای رفتار چندحالته (Multi-modal) است و در قالب چند ناحیه جداگانه ظاهر می‌شود.

برداشت نهایی و کاربردها

این تحلیل‌ها نشان می‌دهند که ویژگی‌های مورد استفاده در مدل‌ها از قدرت تفکیک خوبی برای بسیاری از کلاس‌ها برخوردارند. با این حال، برخی کلاس‌ها به‌خصوص در t-SNE دارای هم‌پوشانی هستند که می‌تواند توضیح‌دهنده‌ی چالش‌های مدل در طبقه‌بندی صحیح آن‌ها باشد. برای مثال، هم‌پوشانی بین کلاس‌های ۶ و ۷ یا شباهت کلاس‌های ۲ و سایر کلاس‌ها، می‌تواند دلیلی بر عملکرد ضعیف مدل در طبقه‌بندی دقیق برخی موارد باشد.