مقدمه:
محاسبه ی عمر مفید قطعات (که منتج به اگاهی از زمان تعمیر و تعویض انها می شود) موضوع مورد بحث در این پروژه می باشد.
هدف از انجام این پروژه محاسبه ی عمر مفید موتور ها بر اساس ویژگی های متعدد انهاست.

مرحله ی اول:
فراخوانی کتابخانه های مورد نیاز از جمله tqdm، plt، seabor، pandas، numpy،و ...

مرحله ی دوم (خوانش فایل ها):
فانکشن load_data:
این فانکشن ادرس فایل هایی که شامل input data ما هستند را به عنوان ورودی می گیرد. سپس خوانش فایل های متعدد train و test و labels که به فرمت txt هستند را انجام می دهد. داده ها همگی در دیتافریم های جداگانه برای train و test و label ها در یک ارایه ی numpy قرار داده شده و اسم ستون های دیتا فریم نیز به ان ها داده می شود. همچنین datatype دو ستون unit_number و time به

int64 تبدیل شده است.
در اخر دیتاست train و test و ارایه ای از label ها به عنوان خروجی ارائه می شود.

مرحله ی سوم (فانکشن run_to_failure_aux):
هدف این فانکشن تغییر دیتاست اولیه و تبدیل ان به دیتاستی جدید و قابل استفاده است.
این فانکشن در ابتدا دیتاست های train و test را به عنوان ورودی دریافت کرده و ستونی به نام broken را به انها می افزاید .اگر مقادیر ستون life time کمتر از حدی تعیین شده باشند ، به مقدار broken انها صفر تعلق می گیرد و بالعکس.
سپس ستون های sensor استخراج می شوند و محاسبه ی mean، std، max، min را برای انها انجام می دهد .
برای ساختار دیتا فریم اصلاحاتی انجام می دهد مانند تبدیل دیتافریم از حالت long به wide و ساختن اسم ستون ها مانند sensor1_max.
در نهایت دیتا فریم جدیدی از تمامی مواردی را که محاسبه کرده و تغییر داده بود به عنوان خروجی ارائه می دهد.

مرحله ی چهارم (فانکشن

censoring_augmentation):
این فانکشن داده ها را به صورت رندوم به دسته هایی تقسیم می کند و به صورت دسته دسته به فانکشن run_to_failure_aux می دهد در نهایت دیتا فریم هایی که از خروجی این فانکشن به دست می اید به هم متصل می کند و به صورت یک دیتا فریم منسجم ارائه می کند.

مرحله ی پنجم (generate_run_to_faliure):
این فانکشن run_to_failure sample تولید می کند و در صورت نیاز برای تولید sample های اضافه از فانکشن censoring_augmentation استفاده می کند.

مرحله ی ششم (فانکشن leave_one_out):
در این فانکشن، داده های train و test به این صورت تعیین می شوند که برای هر ۴ سمپل generate_run_to_failure اجرا می شود و به عنوان یک subset ارائه می شود ،به هر subset یک مقدار به نام fold تعلق می گیرد .
در اخر تاپل هایی ایجاد می شود که داده های train و test در انها دسته بندی می شوند . subset ای از train که fold مشابه subset ای از test دارد در تاپل

قرار نمی گیرد.

مرحله ی هفتم (فانکشن generate_validation_set):
در این فانکشن به کمک فانکشن leave one out یک validation_set ایجاد می شود . مقادیر train و test درون ان در دو فایل csv جداگانه ذخیره شده و validation_set به عنوان خروجی ارائه می شود.

مرحله ی هشتم(تابع main):
در این تابع ابتدا اسامی ستون ها و بخشی از دیتافریم train و test را چاپ می کند و سپس مقادیر nan دیتا فریم train با مقدار 0 پر می شود.
سپس مقادیر x و y برای train و test به صورت جداگانه ایجاد می شود .
برای محاسبه ی طول عمر قطعات که مقداری پیوسته است یک مدل رگرسیونی مورد نیاز است لذا مدل LinearRegression تعریف شده و پس از ان بر روی داده های train ما ، fit می شود. پس از ان مقادیر prediction از x_test گرفته می شود و برای بررسی عملکرد مدل که شامل مقایسه ی پیش بینی های مدل و جواب های حقیقی MSE ان محاسبه می شود.

در نهایت مدل به دو صورت استفاده از کتابخانه ی mlflow و ذخیره در یک دایرکتوری مشخص ، ذخیره می شود.

*توجه: بدلیل عدم دسترسی microsoft azure به کشور ایران برای اپلود مدل در workspace ، این بخش کامنت شده و مدل در یک فایل در نهایت ذخیره می شود.

*توجه :داده های ورودی در فایل application_2Fzip قرار دارند . همچنین لینک دانلود داده های ورودی در جوپیتر نوت بوک قرار داده شده است.
کد ها در فایل جوپیتر نوت بوک ، script و train.yaml قرار دارند.