در این تمرین یک دیتاست در اختیار ما بود که شامل ویژگی‌های زیر بود:

سن: سن ذینفع اصلی.

جنسیت: جنسیت دارنده بیمه، مرد یا زن.

BMI: شاخص توده بدنی فرد، معیاری که وزن را با قد مرتبط می کند. محدوده ایده آل 18.5 تا 24.9 است.

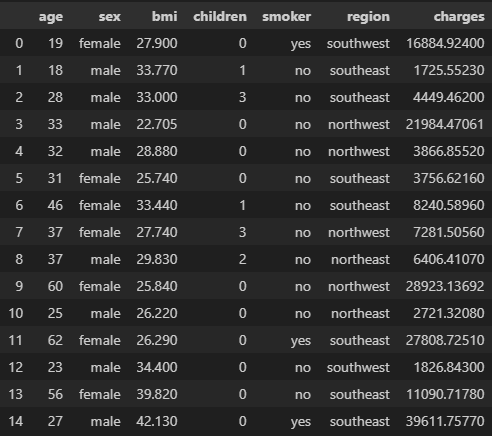
فرزندان: تعداد افراد تحت تکفل تحت پوشش بیمه نامه درمانی.

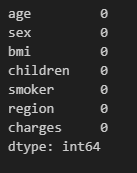
سیگاری: اینکه آیا فرد سیگاری است (بله/خیر).

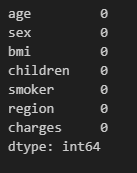
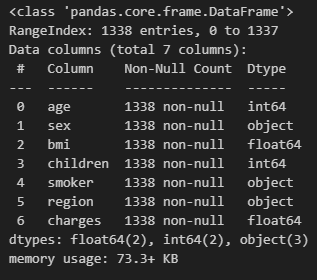
منطقه: منطقه جغرافیایی بیمه گذار در ایالات متحده (شمال شرقی، جنوب شرقی، جنوب غربی، شمال غربی).

هزینه ها: هزینه های پزشکی واقعی که توسط بیمه سلامت صورتحساب می شود. این متغیر هدفی است که ما قصد داریم آن را پیش بینی کنیم.

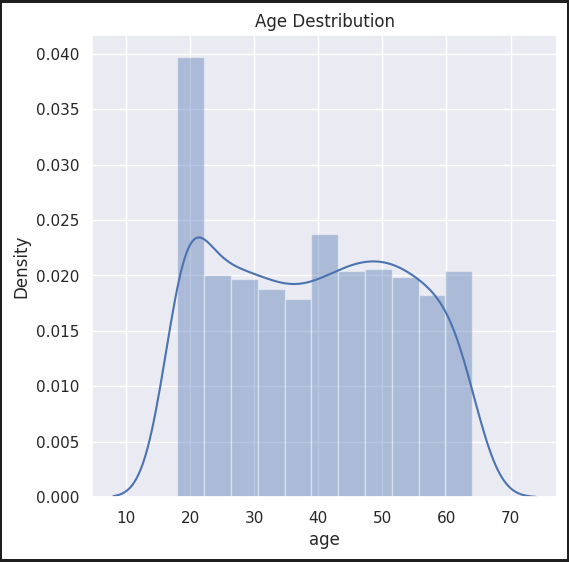
همانطور که از ویژگی‌ها مشخص است، هدف ما این بود که هزینه‌ها را با استفاده از ویژگی‌های دیگر پیش‌بینی کنیم.



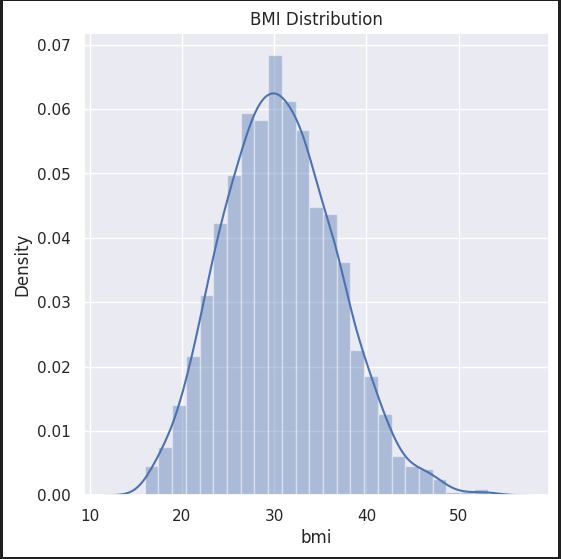




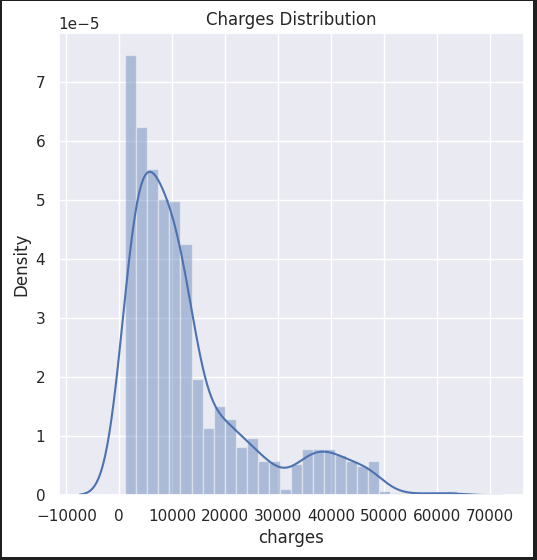
همانطو که از تصاویر بالا مشخص است، دیتاست ما فاقد داده‌ی گمشده است. ویژگی جنسیت ما فقط شامل دو مقدار مرد یا زن است. ویژگی سیگاری بودن یا نبودن هم فقط شامل دو مقدار بله یا خیر است. ویژگی منطقه هم فقط شامل چهار مقدار است که پیشتر ذکر شد. این ویژگی ها طبیعتا نیاز به encoding دارند. بقیه‌ی مقادیر هم عددی هستند که با توجه به هویتشان در فرم درستی هستند و با توجه به تصویر اول، می‌توان گفت که فاقد ناهنجاری نیز هستند.



این تصویر توزیع سنین مختلف را در داده‌های ما نشان می‌دهد. همانطور که مشخص است، بیشترین نمونه‌ها افراد با سنین نزدیک به ۲۰ سال هستند. بقیه‌ی سنین تقریبا تعداد برابری دارند.



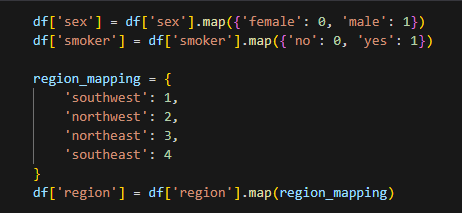
این نمودار هم توزیع شاخص BMI را نشان می‌دهد که از توزیع نرمال پیروی می‌کند و اکثر نمونه‌های ما دارای BMI نزدیک به ۳۰ هستند.



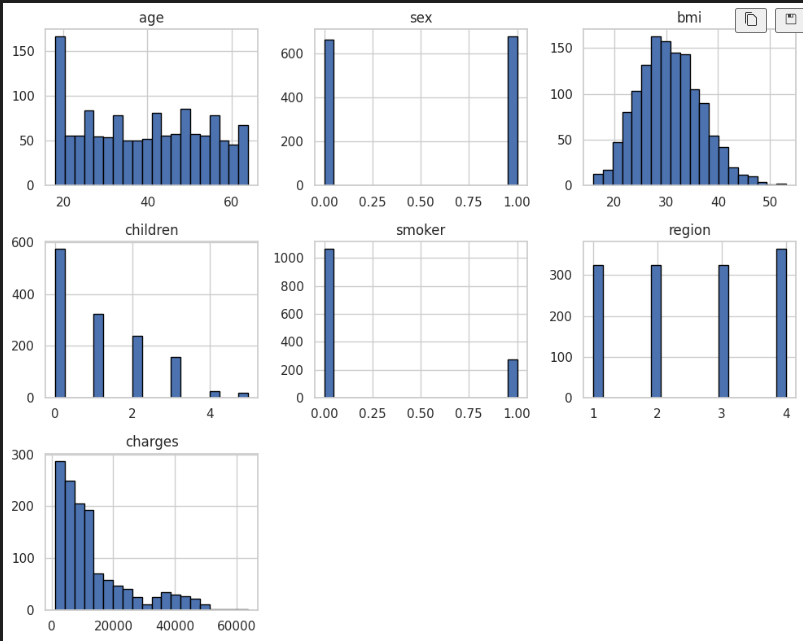
این نمودار نیز توزیع توزیع مقادیر هدف ما را نشان می‌دهد که اکثر نمونه‌های ما حول ۱۰۰۰۰ است.



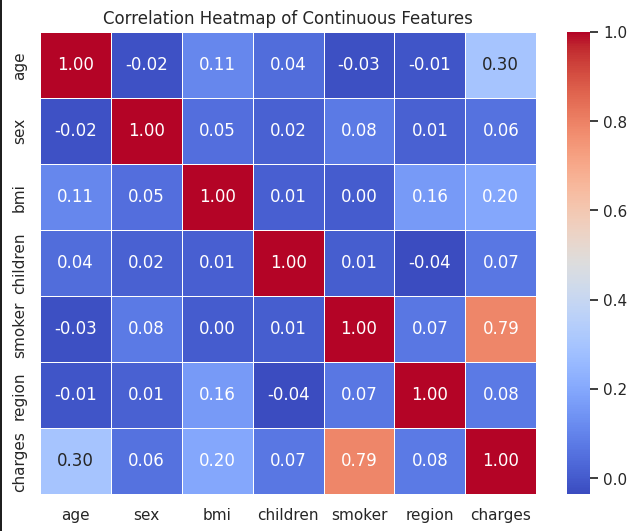
در این تصویر می‌توان یک شما از پارامتر‌های آماری دیتاست را مشاهده کرد.



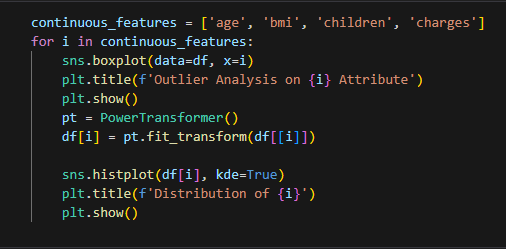
در این قسمت ما داده‌های categorical مان را به داده‌های عددی تبدیل کردیم. می‌توانستیم از one-hot encoding استفاده کنیم ولی در آن صورت ابعاد دیتاست ما بسیار بزرگ می‌شد. بنابراین از lable encoding استفاده کردیم.



بعد از تبدیل داده‌های طبقه‌بندی شده‌مان به داده‌های عددی بار دیگر نمودارهای هیستوگرام را برای داده‌هایمان رسم می‌کنیم. نمودار مربوط به تعداد بجه‌ها قبلا بررسی نشد ولی همانطور که مشخص است اکثر داده‌های ما مربوط به کسانی است که بچه‌ای ندارند. همچنین داده‌های مربوط به زنان و مردان و مناطق مختلف تقریبا برابر اند. همچنین اکثر داده‌های مربوط به کسانی است که سیگاری نیستند.

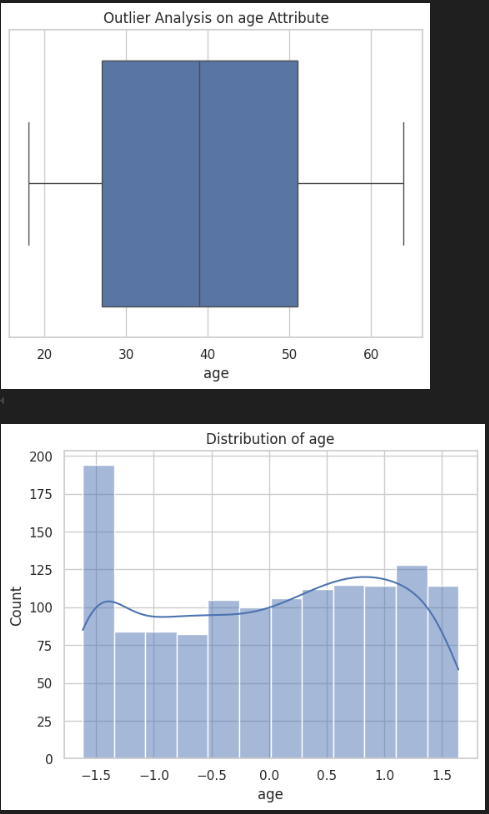


همانطور که در نمودار حرارتی بالا مشخص است، متغیر‌های غیر هدف ما ارتباط قوی‌ای با هم ندارند، بنابراین نیازی نیست که ویژگی‌هایی که ارتباط قوی‌ای با هم دارند را حذف کنیم. ولی سیگاری بودن یا نبودن ارتباط قوی‌ای با ویژگی هدف ما دارد. احتمال زیاد این موضوع به علت اختلاف بسیار زیاد تعداد افراد سیگاری و غیرسیگاری در دیتاست ما است.

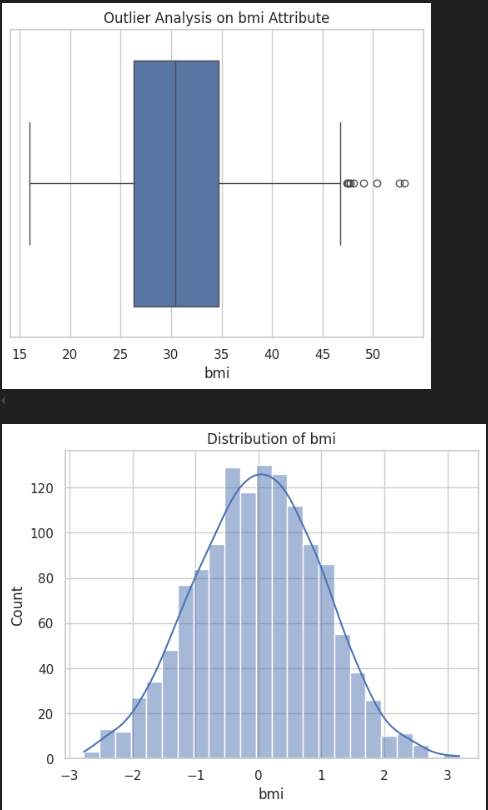


در این قسمت از کد ما با استفاده از نمودار جعبه‌ای outlier ها را تشخیص داده و سپس با استفاده از power transformer داده‌های outlier را هندل کرده و سپس نمودار هیستوگرام را برای بررسی دوباره‌ی داده‌ها رسم می‌کنیم. همچنین استفاده از روش power transform می‌توانیم داده‌هایمان را Normalize کنیم، یعنی همزمان هم داده‌های outlier را هندل کنیم و هم داده‌هایمان را Normalize کنیم. بسیاری از روش‌های آماری برای داده‌های نرمال شده به بهترین شیوه‌ی ممکن عمل می‌کنند و نرمال کردن داده‌های‌ ما باعث می‌شود که کارایی مدل ما بالا برود. همچنین استفاده از روش power transform باعث می‌شود که چولگی و واریانس از بین برود همین موضوع می‌تواند داده‌های outlier ما را هندل کند.

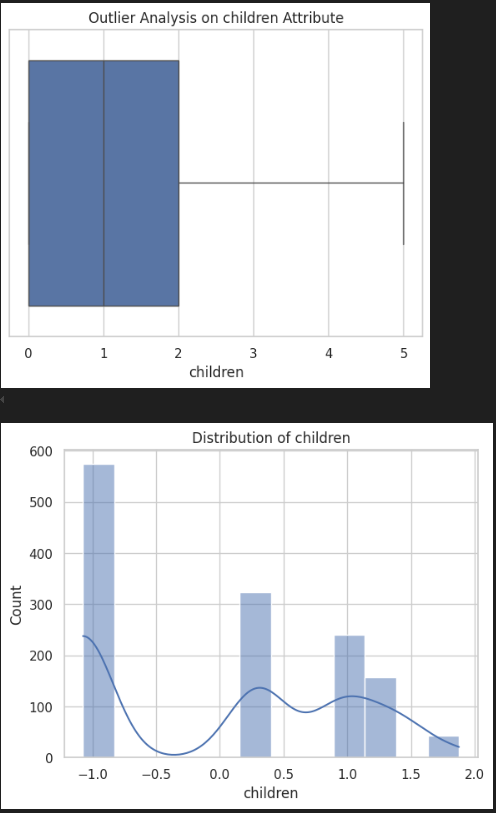
در نمودار‌های زیر می‌توانید نمودار جعبه‌ای قبل از عملیات power transformation و نمودار‌های هیستوگرام بعد از عملیات را مشاهده نمایید.



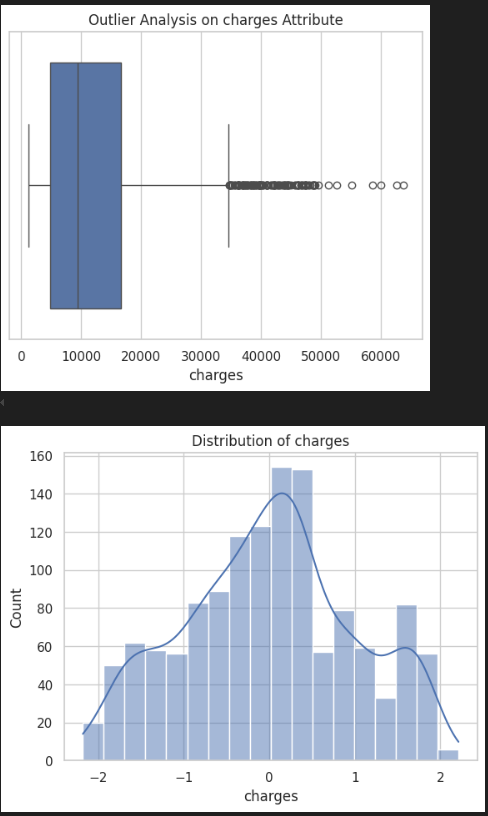
—----------------------------------------------------------------------------



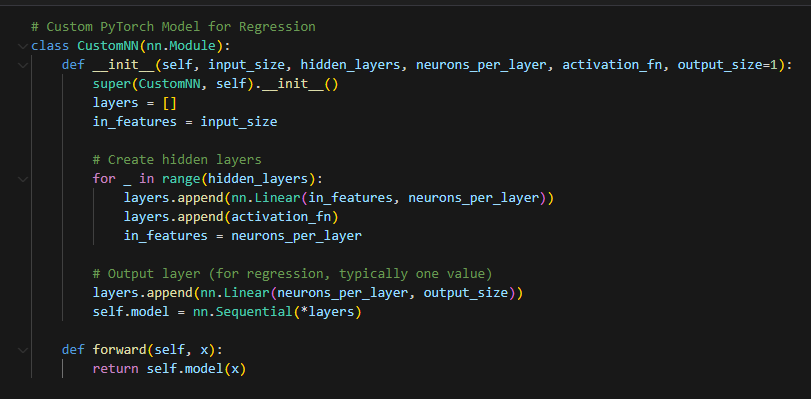
—----------------------------------------------------------------------------

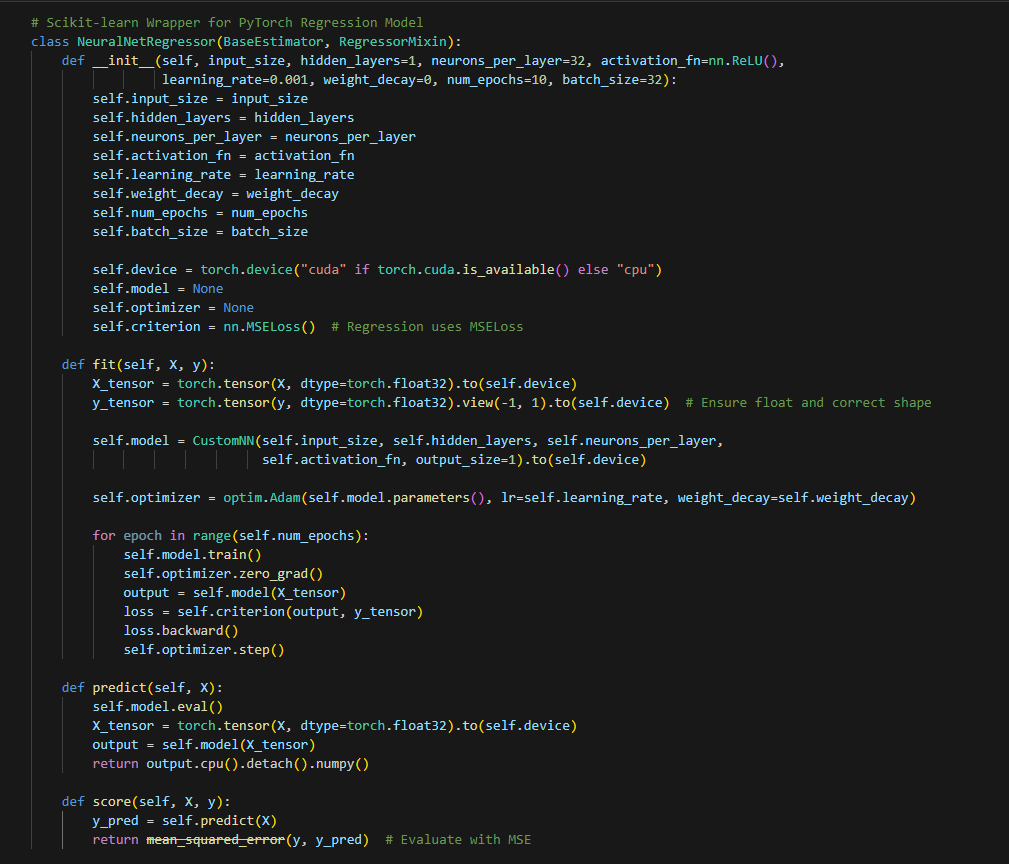


—----------------------------------------------------------------------------

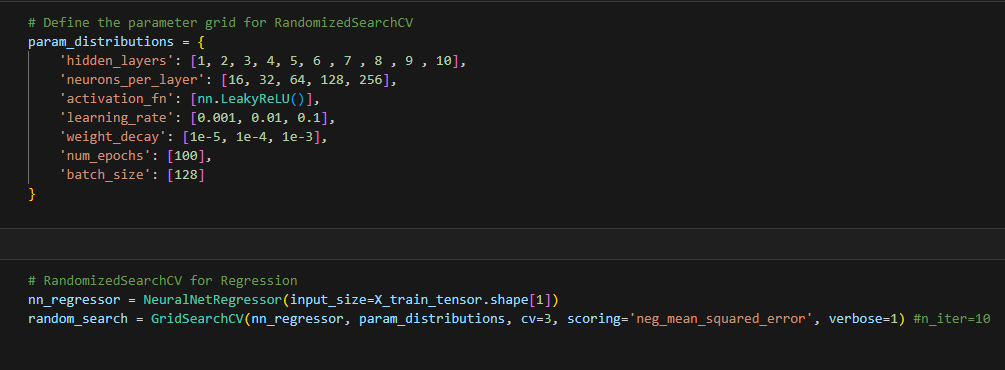


همانطور که می‌توانید مشاهده نمایید، چولگی داده‌‌های ما بعد از انجام عملیات تا حد خوبی از بین رفته است.

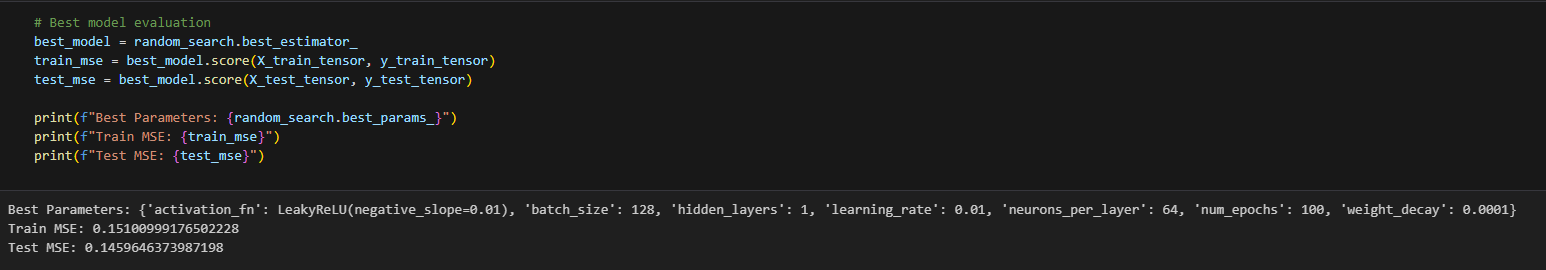


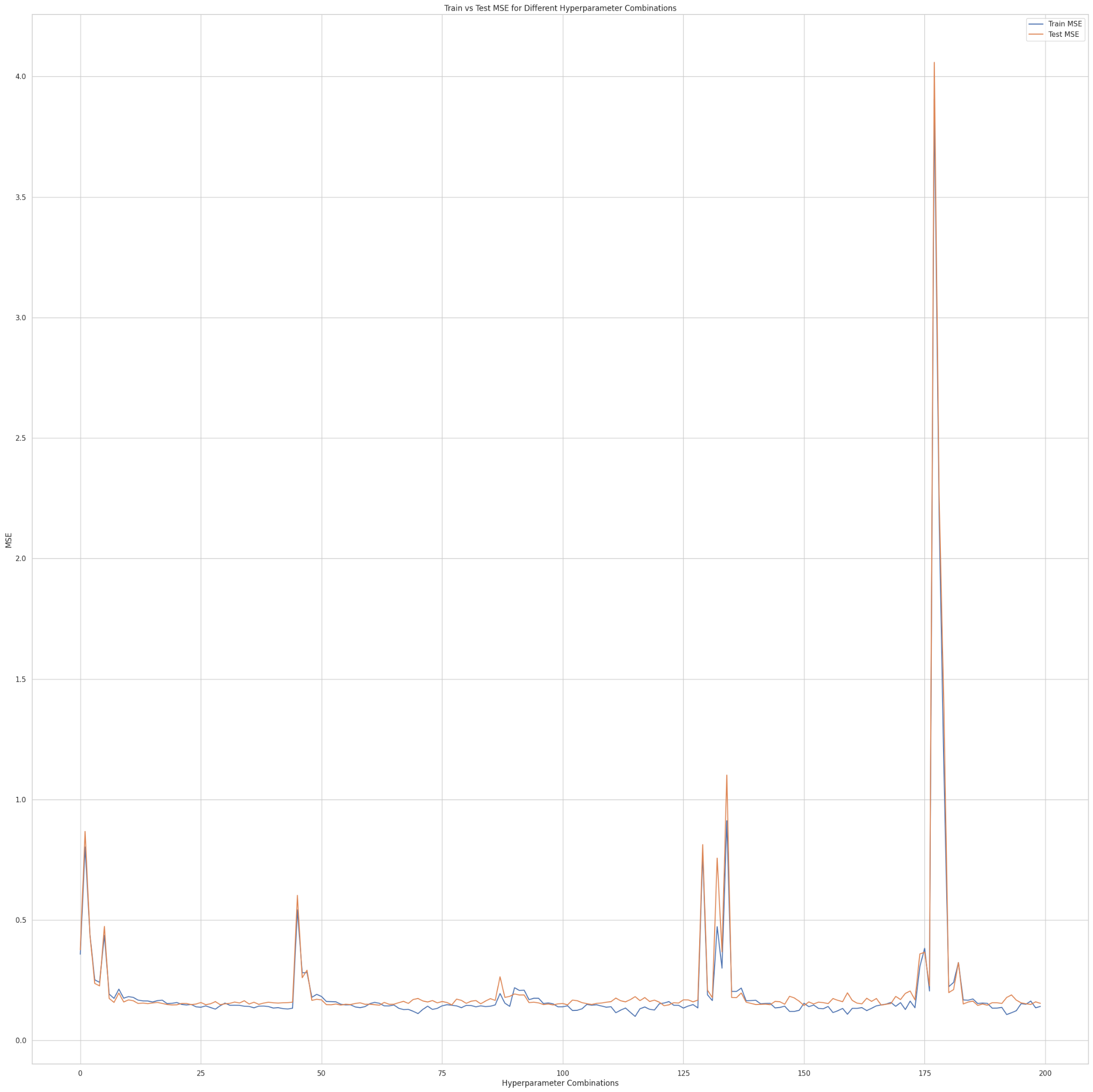


در دو تصویر بالا ما اقدام ساخت شبکه‌ی عصبی خود کرده‌ایم. نکته‌ای که قابل توجه است این است که ما تعداد لایه‌های پنهان را به صورت یک عدد ثابت قرار نداده‌ایم، بلکه آن را به صورت متغیر قرار داده‌ایم تا در صورت نیاز آن را تغییر دهیم. همچنین تعداد نورون‌ها در هر لایه نیز یک عدد ثابت نیست بلکه به صورت یک متغیر برای هر لایه تغییر می‌کند تا بهترین تعداد نورون در هر لایه را پیدا کنیم. همچنین تابع فعالیت ما نیز متغیر است و می‌توانیم از هر تابع فعالیت دلخواه استفاده کنیم(برای لایه‌ی پنهان) ولی بهتر است که از تابع leaky relu استفاده کنیم چون تابع relu از دیگر توابع برای تابع فعالیت لایه‌ی پنهان بهتر است و همچنین تابع leaky relu مشکلات تابع relu که در واقع dying relu است را ندارد. همچنین learning rate و weight decay و number of epochs و batch size هم متغیر است. همچنین ما از adam optimizer استفاده کرده‌ایم تا از dynamic learning rate بهره ببریم. ما به عنوان تابع هزینه از تابع MSE استفاده کرده‌ایم که برای مسائل رگرسیون مناسب است.



در این قسمت ما یک pipeline ایجاد کرده‌ایم و انواع مقادیر را برای hyperparameter هایمان انتخاب کرده‌ایم تا با استفاده از Grid Search CV بهترین مقدار برای hyperparameter هایمان را انتخاب کنیم(البته برای batch size و number of epochs تنها یک مقدار انتخاب کردیم، چون تست کردن انواع مختلف برای این hyperparameter ها زمان بسیار زیادی را نیاز داشت و همچنین به اندازه‌ی دیگر hyperparameter ها در مدل ما تاثیر گذار نبودند. همچینی ما از 3fold-cross validation استفاده کردیم تا خطر بیش‌برازش را کاهش دهیم.

 در این تصویر می‌توانید نتایج را مشاهده نماید. به علت فاصله‌ی بسیار کم بین مقدار MSE در داده‌های train و test می‌توان از این قضیه مطمئن بود که دچار بیش‌برازش نشده‌ایم.



در این تصویر می‌توان خطای داده‌های train و test را مشاهده نمود. در برخی نمونه ها با وجود این که خطای داده‌های آموزشی پایین است ولی خطای داده‌های تست بالا است که نشان دهنده‌ی بیش برازش است. برخی نمونه ها هم دارای خطای بسیار بالایی هستند. به همین دلیل pipeline ما آن‌ها را رد کرده است. ولی در best estimator ما بهترین نمونه را با خطای پایین چه در داده‌های آموزشی و چه داده‌های تست داریم که میزان اختلاف‌ آن‌ها پایین بوده و می‌توان مطمئن بود که دچار بیش‌برازش نشده‌ایم.