گزارش مینی پروژه شماره ۱ درس مبانی سیستم های هوشمند

لینک گیت هاب پروژه

مقدمه

پاسخ به سوالات

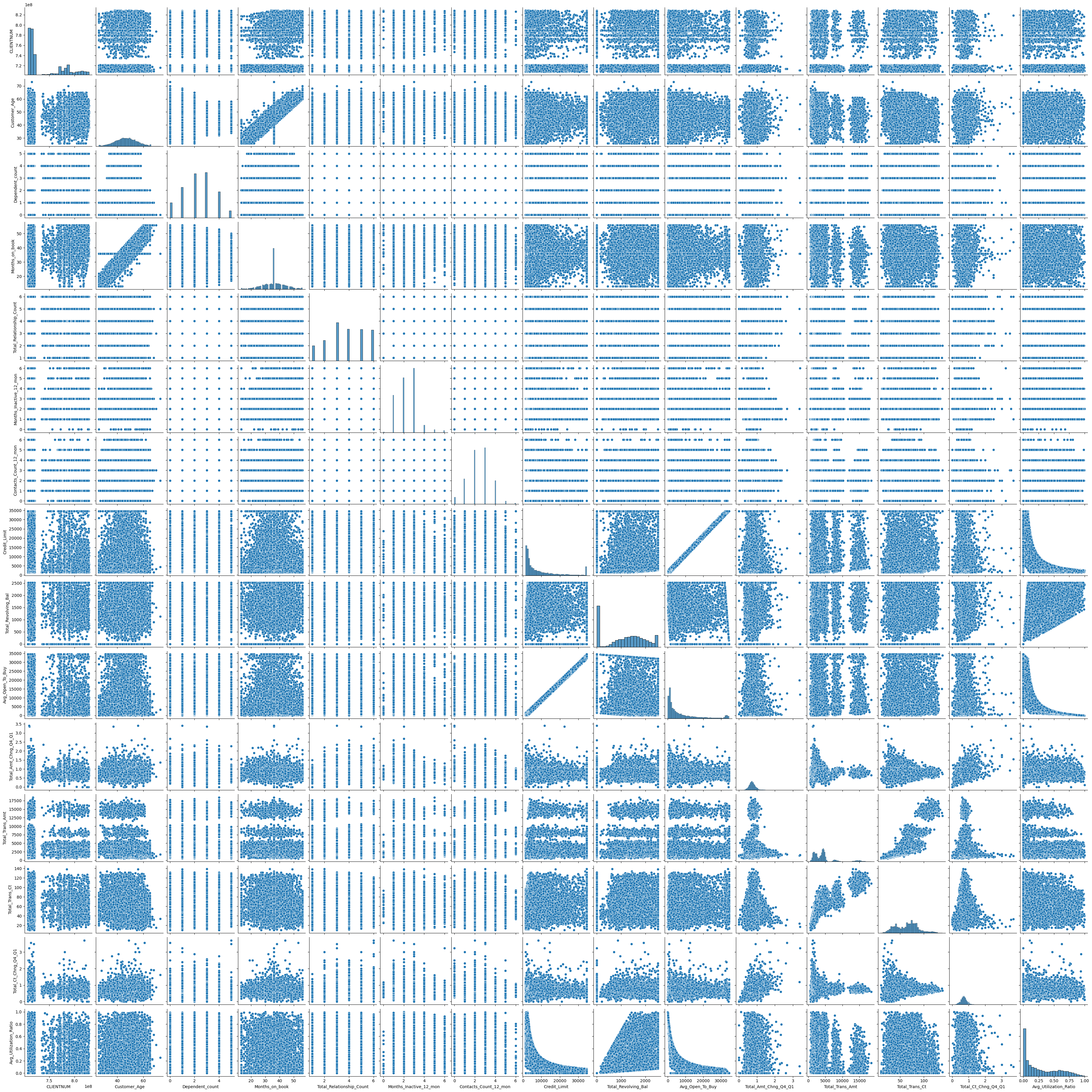
پرسش ۱

۱-۱

این مربوط به دیتای مشتریان یک بانک می باشد که از یک وبسایت با آدرس https://leaps.analyttica.com/home است. این سایت چگونگی حل مشکلات کسب و کار را توضیح می‌دهد. اکنون، این مجموعه داده شامل ۱۰۱۲۷مشتری است که شامل ۱۸ ویژگی از جمله سن، حقوق، وضعیت تأهل، سقف کارت اعتباری، دسته‌بندی کارت اعتباری، و غیره است. (تعداد داده و نام تمام دسته ها در کد خروجی گرفته شده)

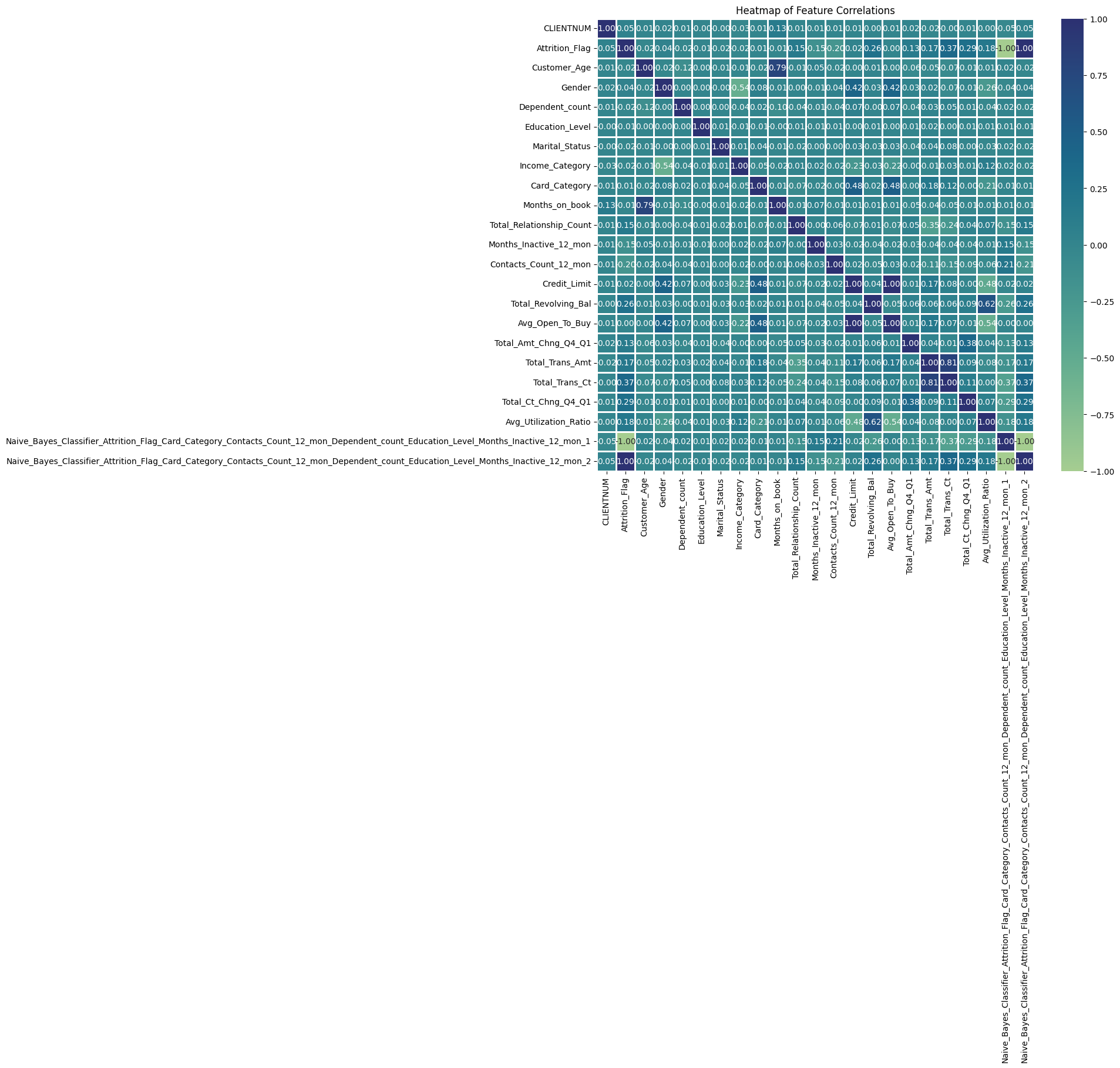
۱-۲

خروجی نمودار ها به شکل زیر است (برای نمایش دقیق تر به کد مراجعه شود):

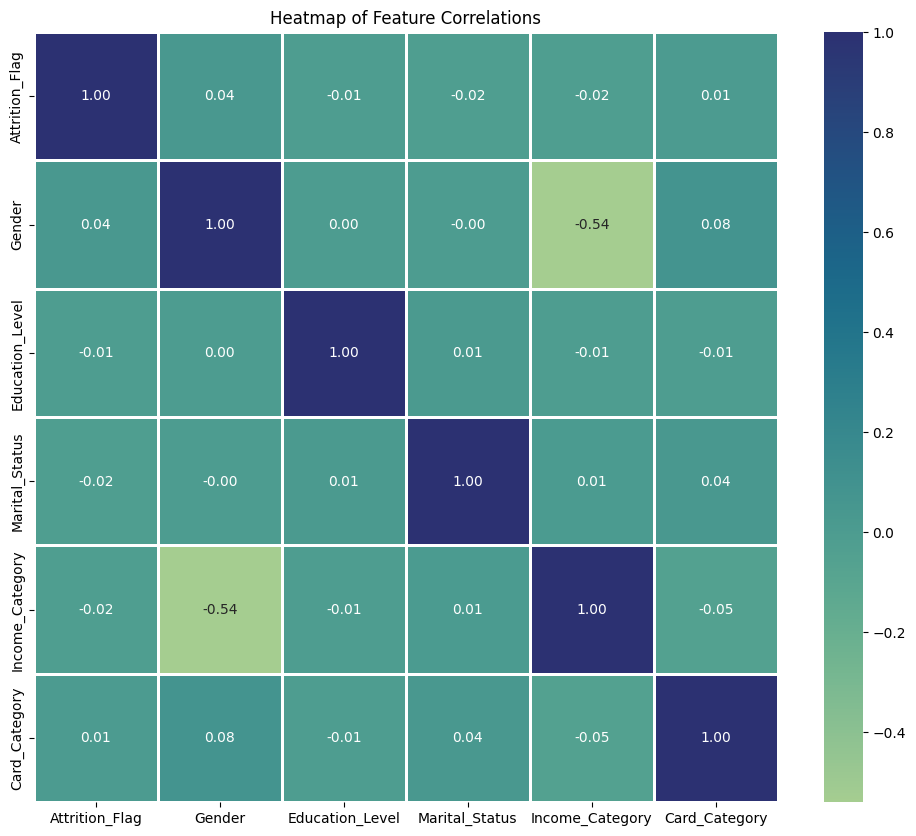


۱-۳

نمودار همبستگی میان داده ها به شکل زیر است که در آن عدد مثبت نشان دهنده ارتباط مستقیم و عدد منفی نشان دهنده رابطه عکس می باشد:



همچنین نمودار همبستگی داده های کیفی به شکل زیر است:

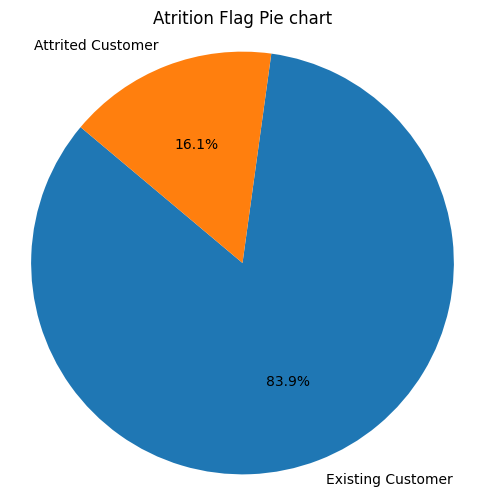


۱-۴

در این داده دیتای nan وجود ندارد ولی با این وجود کد حذف افرادی که دیتایشان ناقص است درج شده است

۱-۵

Attrition\_Flag دارای دو کلاس Attrited Customer و Existing Customer است که نمودار توزیع داده آن مطابق نودار زیر است:



همان طور که مشاهده می شود حدود۸۳ در صد داده ها مربوط به یک کلاس کلاس است. حال فرض کنیم یک مدل داریم که فقط کلاس اکثریت را به ما به عنوان خروجی بدهد. در این صورت دقت تئوری مدل ۸۳ درصد خواهد بود! در مدل خطی نیز نا میزانی چنین تاثیری میگذارد طوری که مدل در پیشبینی کلاس اکثریت بسیار قوی و در کلاس اقلیت ضعیف عمل خواهد کرد این امر با به دست آوردن Confusion Matrix می توان دید.

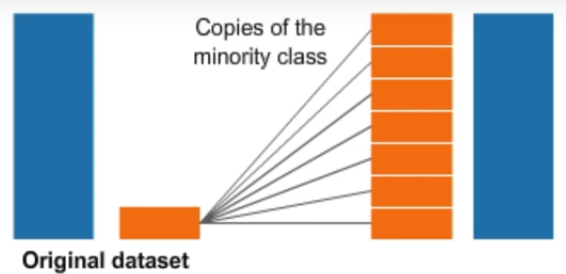
چندین راهکار وجود دارد که می‌توان برای مقابله با عدم تعادل داده‌ها استفاده کرد:

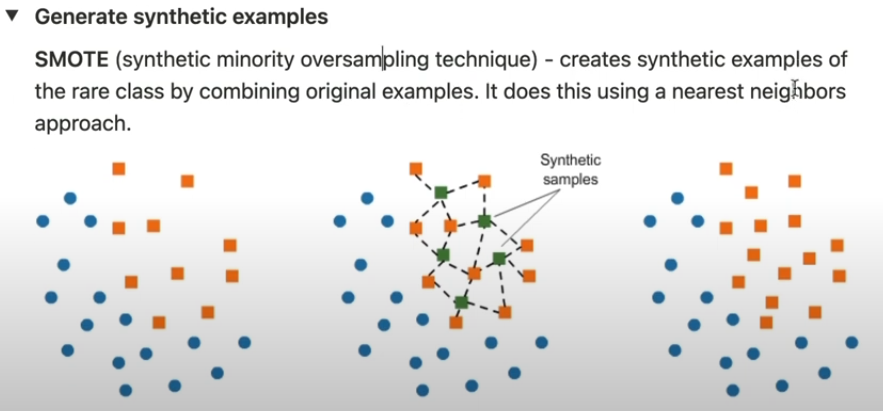
الف. روش‌های مبتنی بر داده‌ها:

افزایش نمونه‌های کلاس اقلیت (Oversampling):

تولید نمونه‌های جدید از کلاس اقلیت، معمولاً از طریق تکرار داده‌ها یا روش‌هایی مانند

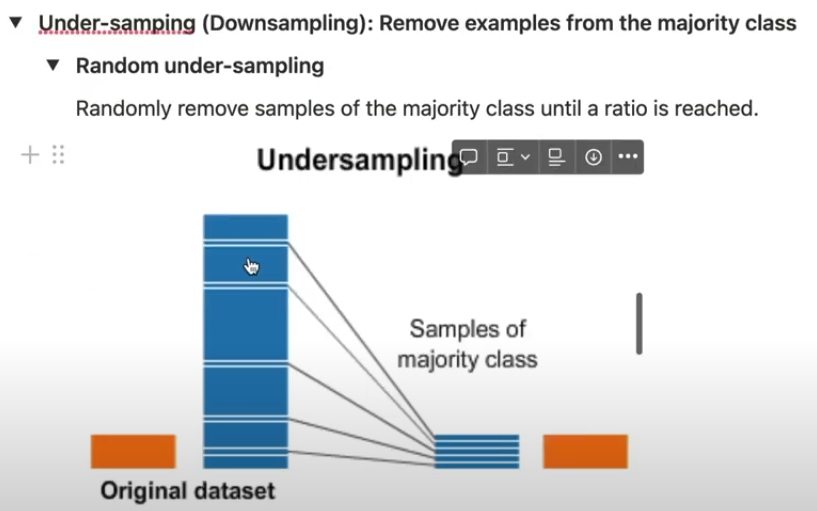
SMOTE (Synthetic Minority Oversampling Technique)





کاهش نمونه‌های کلاس غالب (Undersampling):

حذف بخشی از داده‌های کلاس غالب برای متعادل کردن تعداد نمونه‌ها، با این حال، ممکن است اطلاعات مفید را از دست بدهیم.



تولید داده‌های مصنوعی:

استفاده از الگوریتم‌هایی برای تولید داده‌های جدید با حفظ ویژگی‌های اصلی نمونه‌ها (مانند GANها)

ب. روش‌های مبتنی بر مدل:

استفاده از وزن‌دهی به کلاس‌ها:

تخصیص وزن بیشتر به نمونه‌های کلاس اقلیت در تابع هزینه مدل، تا خطاهای مربوط به این کلاس تأثیر بیشتری داشته باشد.

انتخاب مدل‌های مناسب:

ج. روش‌های ترکیبی

علاوه بر روش های فوق مدل‌هایی مانند درخت‌های تصمیم‌گیری (Decision Trees) یا الگوریتم‌هایی مانند جنگل تصادفی (Random Forest) می‌توانند تا حدی با عدم تعادل داده مقابله کنند.

۱-۶

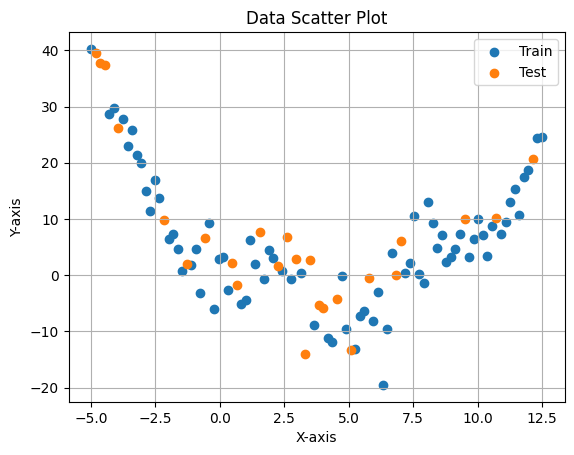
در مرحله اول که با دیتای نامتعادل آموزش صورت گرفته مدل به دقت صد در صد روی هر سه دسته داده رسیده یعنی نیازی به دستکاری داده های آموزش نمی باشد. ولی با این وجود می توانیم با استفده از دستور resample() دیتا را این روش همان SMOTE است که در آن به طور تصادفی از خود دیتا انتخاب می کند و از دیتای انتخاب شده برای افزایش تعداد داده استفاده می کند. دستور را تنها برای داده آموزش استفاده می کنی زیرا داده‌های آزمون باید توزیع طبیعی مجموعه داده اصلی را منعکس کنند تا ارزیابی عملکرد مدل واقعی باشد. اگر بعد از تقسیم داده، متعادل‌سازی انجام شود، داده‌های آزمون نیز تغییر خواهند کرد و نتایج ارزیابی مدل غیرواقعی خواهد شد.

از تحلیل ماتریس به هم ریختگی واضح است مدل در هر دو حالت داده ها را یه خوبی یاد گرفته و دقتش صد در صد شده. این به خاطر خوب بودن داده است که در ماتریس همبستگی نیز قابل مشاهده است.

پرسش ۲

۲-۱

با دستور اسپلیت این کار به راحتی قابل انجام است:



در اینجا دیتا به صورت بیست به هشتاد تقسیم شده.

۲-۲

Mean Square Error (MSE):

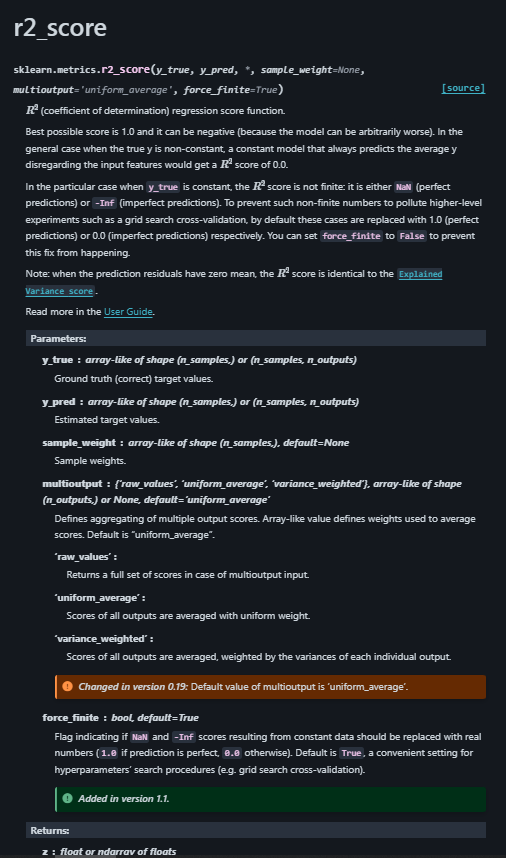
: Actual value.

: Predicted value.

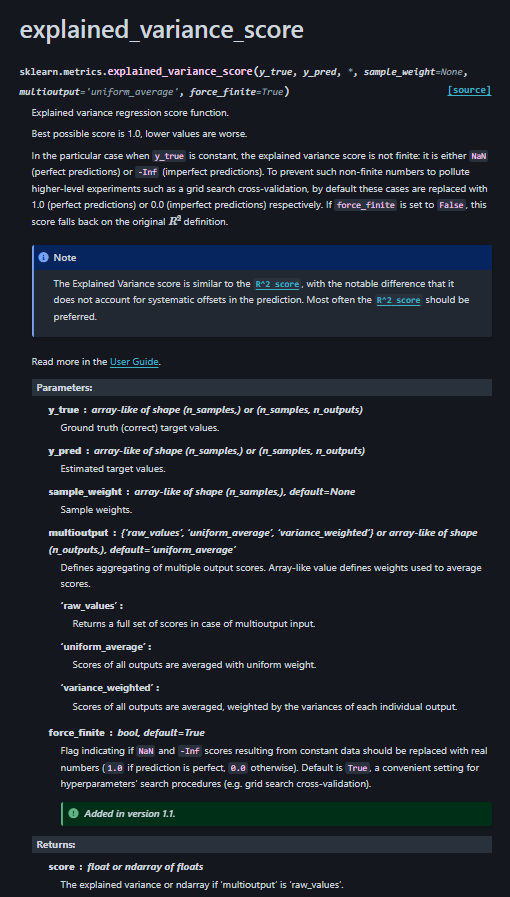
: Number of samples.



Coefficient of Determination(R^2):



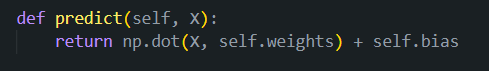
Explained Variance Score:

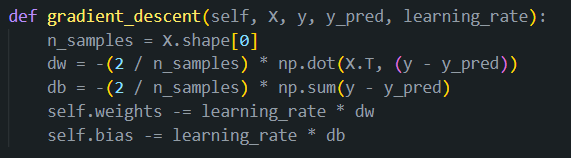


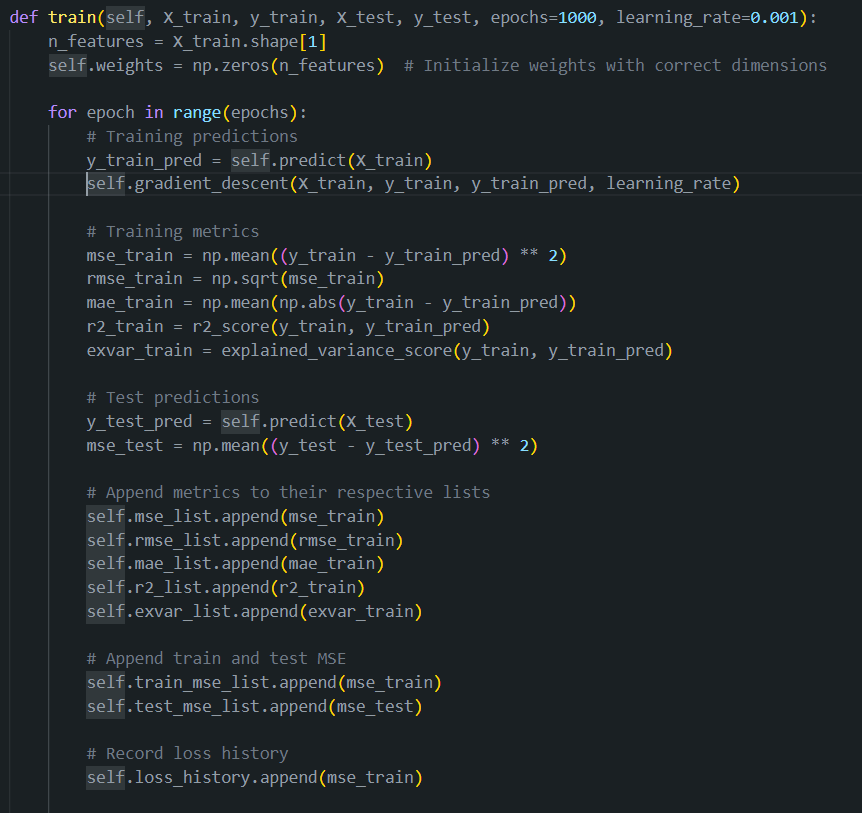
۲-۳

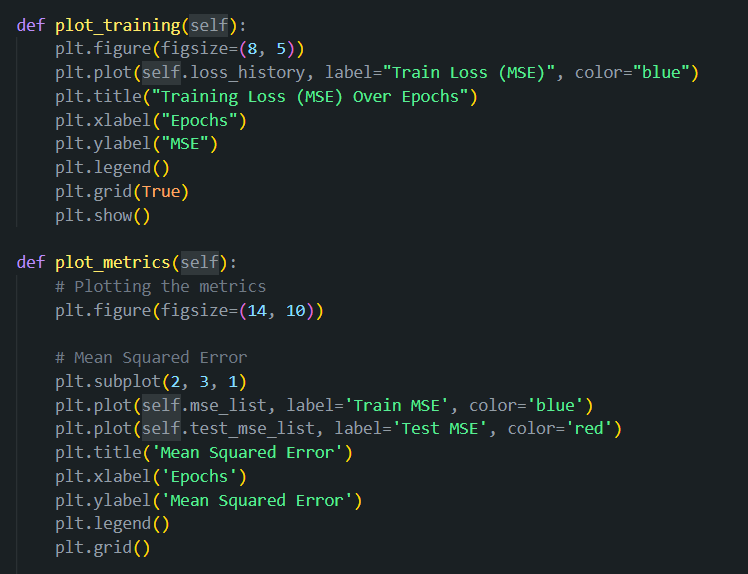
Initialization:











با آموزش داده ها به این مدل مشاهده می شود عملکرد مدل بسیار دقت پایینی دارد

۲-۴

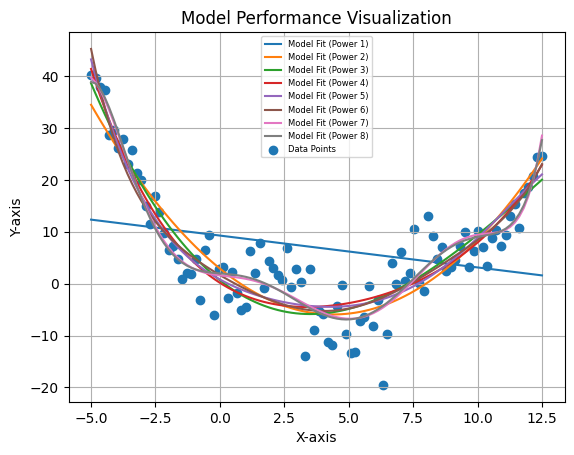
با توجه به نمودار های زیر

۲-۵

اگر ملاک و دیتای ترین ما نمونه فعالیت انسان باشد خطای دیتاست تابع خطای انسان است اگر مدل ترین شده دقت پایین تری دارد میتوان با افزایش تعداد داده دقت را بالا برد ولی چون دیتای ترین به طور ذاتی خدای انسانی دارد دقت مدل با افزایش تعداد داده به خطای انسان نزدیک می سود ولی هیچ وقت از آن بالاتر نمی رود.

۲-۶

خطا کاهش پیدا می کند البته به معنای این نیست که مدل بهتر شده ممکن است اورفیت شده باشد من با حلقه آموزش با کتابخانه scikit-learn توان بالای پنج را هم امتحان کردم و خطای مدل بعد از توان هفتم ثابت می شود. البته واضح است که مدل اورفیت شده.



۲-۷

رگراسیون درخت تصمیمگیری:

الگوریتم درخت تصمیم‌گیری یک روش یادگیری ماشین نظارت‌شده است که برای وظایف دسته‌بندی و رگرسیون استفاده می‌شود. این الگوریتم با تقسیم بازگشتی مجموعه داده به زیربخش‌هایی بر اساس ویژگی‌ای که به بهترین شکل داده‌ها را جدا می‌کند، کار می‌کند. در هر گره از درخت، یک تصمیم بر اساس مقدار یک ویژگی گرفته می‌شود و مجموعه داده به دو یا چند شاخه تقسیم می‌شود، که یک ساختار شبیه به درخت ایجاد می‌کند. معیار تقسیم‌بندی معمولاً بر اساس اندازه‌ای مانند ناخالصی جینی (برای دسته‌بندی)، کسب اطلاعات (IG)، یا میانگین خطای مربعی (MSE) (برای رگرسیون) انتخاب می‌شود.  
درخت حاصل به صورت یک نمودار جریان قابل تصور است، جایی که هر گره داخلی یک تصمیم بر اساس ویژگی را نشان می‌دهد و هر گره برگ نتیجه‌ای را (یک برچسب دسته‌بندی برای دسته‌بندی یا یک مقدار پیوسته برای رگرسیون) نشان می‌دهد.

رگراسیون جنگل تصادفی:

جنگل تصادفی (Random Forest) یک الگوریتم یادگیری ماشین تجمعی برای وظایف طبقه‌بندی و رگرسیون است که از ترکیب چندین درخت تصمیم‌گیری به عنوان "یادگیرنده‌های ضعیف" استفاده می‌کند تا یک مدل قوی و دقیق ایجاد کند.

ایده اصلی در جنگل تصادفی این است که هر درخت به صورت مستقل بر روی یک زیرمجموعه تصادفی از داده‌ها و ویژگی‌ها آموزش می‌بیند. این فرآیند تصادفی‌سازی کمک می‌کند که مدل نهایی از قدرت تعمیم‌پذیری بالایی برخوردار باشد و به مشکلاتی مانند بیش‌برازش (Overfitting) که در درخت‌های تکی رایج است، دچار نشود.

مراحل کار جنگل تصادفی:

1. نمونه‌گیری تصادفی از داده‌ها: الگوریتم از روش Bootstrap استفاده می‌کند که در آن زیرمجموعه‌ای از داده‌ها با جایگزینی انتخاب می‌شود. این زیرمجموعه‌ها به عنوان داده‌های آموزشی برای هر درخت استفاده می‌شوند.
2. انتخاب تصادفی ویژگی‌ها: در هر گره از درخت، الگوریتم به جای استفاده از تمام ویژگی‌ها، به صورت تصادفی تعداد محدودی از ویژگی‌ها را بررسی می‌کند. این کار از غالب شدن ویژگی‌های خاص جلوگیری می‌کند.
3. ساخت درخت‌های تصمیم‌گیری: هر درخت به صورت مستقل از دیگر درخت‌ها آموزش داده می‌شود و تا عمق کامل رشد می‌کند (بدون هرس).
4. ادغام پیش‌بینی‌ها

Gradient Boosting Regressor:

رگرسیون گرادیان بوستینگ (GBR) یک الگوریتم یادگیری ماشین تجمعی برای وظایف رگرسیون است که یک مدل پیش‌بینی قوی را با ترکیب چندین یادگیرنده ضعیف (معمولاً درخت‌های تصمیم‌گیری) می‌سازد. ایده اصلی در گرادیان بوستینگ این است که مدل‌های جدید به صورت تکراری آموزش داده می‌شوند تا خطاهای مدل‌های قبلی در مجموعه را اصلاح کنند.  
در هر مرحله، مدل بر روی خطاهای باقی‌مانده (اختلاف بین مقادیر پیش‌بینی‌شده و واقعی) مدل قبلی تمرکز می‌کند و یک درخت جدید برای پیش‌بینی این خطاها آموزش داده می‌شود. این فرآیند تا تعداد معین از تکرارها یا رسیدن به شرایط توقف ادامه پیدا می‌کند. پیش‌بینی‌های تمامی درخت‌ها با استفاده از میانگین وزنی برای ایجاد پیش‌بینی نهایی ترکیب می‌شوند.  
ویژگی متمایز گرادیان بوستینگ از دیگر روش‌های تجمعی، مانند جنگل‌های تصادفی، این است که این یک الگوریتم ترتیبی است. مدل به گونه‌ای ساخته می‌شود که هر درخت اشتباهات درخت قبلی را اصلاح کند و از تکنیک بهینه‌سازی نزول گرادیان استفاده می‌کند. این به این معناست که الگوریتم مدل جدید را با گرادیان منفی تابع زیان متناسب می‌کند، به طور مؤثر خطا را در هر مرحله به حداقل می‌رساند.  
پارامتر نرخ یادگیری کنترل می‌کند که هر مدل چقدر به پیش‌بینی نهایی کمک می‌کند.

امتیازی

به دلیل ماهیت مسئله هر چقدر توان را بالا می بریم شاهد یک ارور بودم بعد از تحقیق متوجه شدم توان های بالا باعث بهم خوردن الگوریتم می شدند و ارور مربوط به صرب اعدا بیش از حد برگ بود که با کاهش گام و افزایش تعداد ایپاک حل می شد ولی آموزش بعد از درجه سوم بسیار طولانی بود بنابر این به صورت پیش فرض از اسکیل استفاده کردم.

چند مثال از اسکیل های موجود در کتابخانه scikit-learn: