به نام خدا

****

مستندات پروژه اول

**مدل رگرسیون برای پیشبینی اعتبار مالی افراد**

مبانی یادگیری ماشین

رویا صالحی

4003623023

دکتر محمد کیانی

بهار 1403

**دیتاست:**

در ابتدا تمامی کتابخانه های مورد نیاز **install** و **import** شدند.

پس از خواندن دیتاست، برای مطلع شدن از وضعیت دیتا‌ها، از دستورات زیر استفاده شد:

df.info()

تعداد دیتای ویژگی‌هایی که مقادیر نال دارند، با دیتاتایپ آنان شناسایی شد.

df.describe()

اطلاعات کلی شامل میانگین، مقدار ماکسیمم، مینیمم و میزان پراکندگی داده ها و... شناسایی شد.

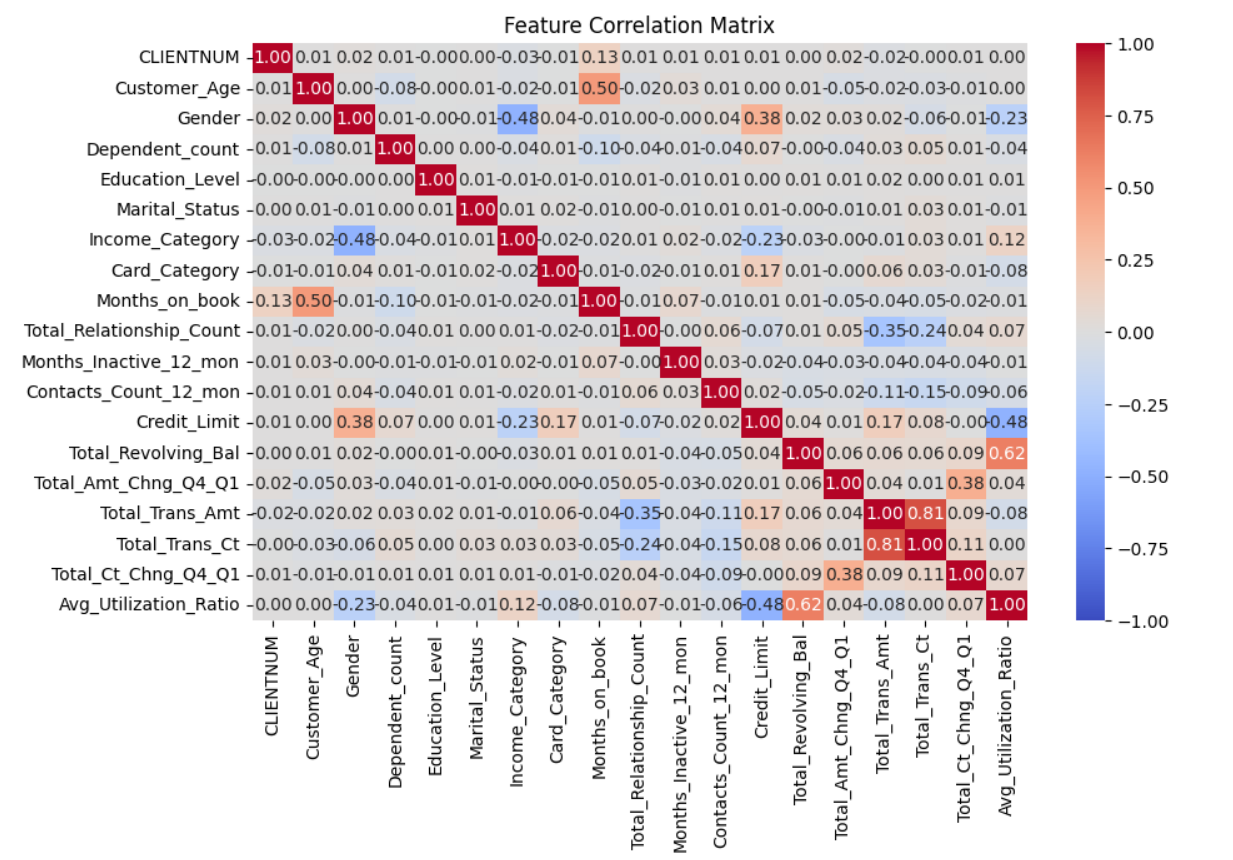
در بعضی اطلاعات داده شده به وضوح دیتا‌های پرت مشاهده می‌شود. برای مثال در ویژگی سن افراد به طور واضحی قابل مشاهده است که، در این دیتاست فردی با سن 352وجود دارد!

در مرحله بعد دیتا‌های تکراری بر اساس **CLIENTNUM** سنجیده شده، و مقادیر تکراری حذف شدند.

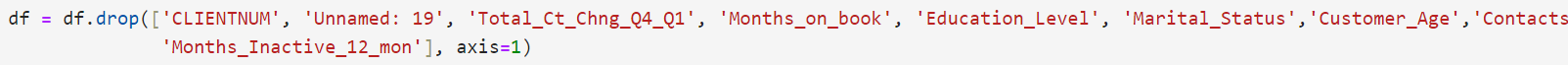
**انتخاب ویژگی:**

با استفاده از فانکشن dataset.corr()، ماتریس همبستگی بین تمامی جفت‌های ویژگی‌های موجود در دیتاست را محاسبه می‌شود. مقادیر همبستگی بین 1- و 1 قرار دارند، که 1- نشان‌دهنده همبستگی منفی کامل، 0 نشان‌دهنده عدم همبستگی، و 1 نشان‌دهنده همبستگی مثبت کامل است.

در واقع قدر مطلق مقادیر نشان دهده میزان همبستگی ویژگی‌ها هستند که با بیشتر شدن این مقدار، رنگ آن تیره تر نشان داده شده.



با تحلیل مقادیر به روش گفته شده به بی اهمیت بودن بعضی ویژگی‌ها پی می‌بریم. و سپس با حذف این ویژگی‌ها، به بهبود روند train کمک می‌کنیم.



**Z\_Score**

نمره z **(z-score)** یک مقدار استاندارد شده است که نشان می‌دهد یک داده چقدر از میانگین دور است، به واحد انحراف معیار. این مقدار به شما اجازه می‌دهد تا تشخیص دهید یک داده تا چه اندازه نسبت به میانگین داده‌های یک مجموعه داده‌ها پرت است. نمره z به صورت زیر محاسبه می‌شود:

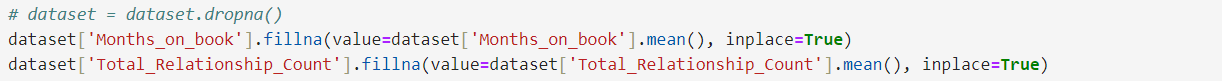
در مرحله بعد به منظور شناسایی و حذف داده‌های پرت (**outliers**) در یک دیتاست نمره z (**z-score**) محاسبه شده است. داده‌های پرت در اینجا به داده‌هایی اشاره دارد که مقدارشان بیش از 3 انحراف معیار از میانگین فاصله دارند.

در نهایت، دیتاست پس از حذف داده‌های پرت برای دو ویژگی **Total\_Trans\_Amt** و **Total\_Trans\_Ct** باقی می‌ماند. این فرآیند به بهبود کیفیت داده‌ها و کاهش تأثیرات منفی داده‌های پرت بر روی تحلیل‌ها و مدل‌های بعدی کمک می‌کند.

**مقدار‌دهی مقادیر مفقود:**

روش های مختلفی جهت انجام این مرحله صورت گرفت، مثل دراپ کردن، میانگین برای ویژگی های عددی و گرفتن مد برای ویژگی های دسته ای. سپس جهت کسب نتیجه بهتر از الگوریتم هایی برای پیش‌بینی این مقادیر استفاده شد.

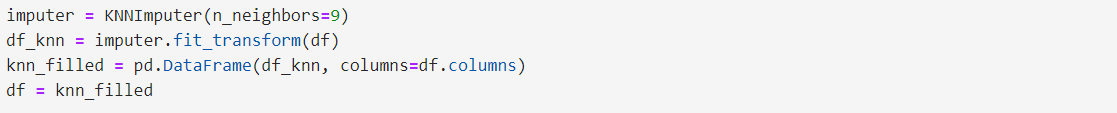
**Mean:**



**K-Means:**

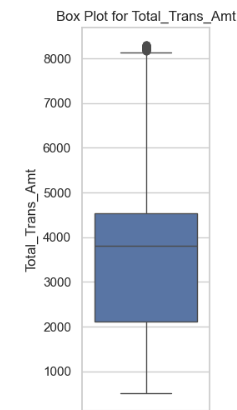
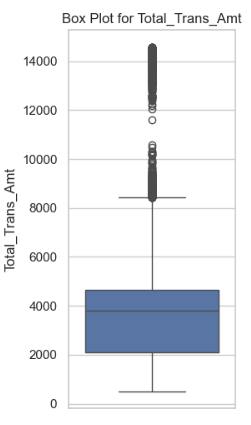


**KNN:**



اما در نهایت مشاهده شد که **KNN** بهترین نتیجه را خواهد داشت.

در این مرحله به منظور شناسایی توزیع داده ها و مقادیر پرت، نمودارهای جعبه‌ای (**Box Plot**) برای تمامی ستون‌های موجود در دیتاست، رسم شد.



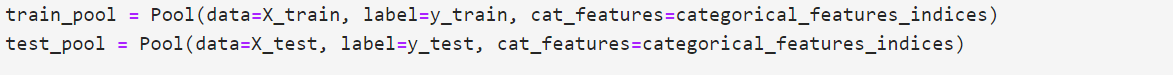
برای این منظور با استفاده از محدوده بین چارکی (**IQR**) استفاده شده است. به طور خاص، این کد از روش IQR برای تعیین مرزهای پایین و بالا استفاده می‌کند و سپس مقادیری را که خارج از این مرزها هستند حذف می‌کند.

چارک اول و سوم را محاسبه می‌کند.که این خطوط برای چارک اول (25 درصد پایین) و چارک سوم (75 درصد پایین) داده‌های ستون جاری را در نظر می‌گیرد. و سپس به محاسبه **IQR** محدوده بین چارکی پرداخته شده. سپس مرزهای پایین و بالا برای شناسایی مقادیر پرت محاسبه شد. هر مقداری که پایین‌تر از **lower\_bound** یا بالاتر از  **upper\_bound** باشد، به عنوان مقدار پرت در نظر گرفته می‌شود. پس اقدام به حذف آن می‌کنیم.

**مراحل آموزش مدل:**

در این مرحله به جداسازی ستون **credit limit** از ویژگی های دیگر، به عنوان **lable** می‌پردازیم. و تمامی ویژگی های باقی مانده را به عنوان فیچر در نظر می‌گیریم.

پس از آن با تقسیم داده ها به دو فاز **train** و **test** می‌پردازیم. این کار با **random\_state** مشخص، به منظور ثابت بودن تقسیم‌بندی جهت مقایسه مقادیر انجام شده. مقدار **test\_size**، 0.2 در نظر گرفته شده. یعنی 20 درصد داده ها به فاز **test** و سایرین به فاز **train** تخصیص داده شوند.



این کد مربوط به استفاده از کتابخانه **CatBoost** برای ایجاد مجموعه‌های داده‌های آموزشی و آزمایشی است.

**CatBoost**یکی از الگوریتم‌های یادگیری ماشین است که به طور خاص برای کار با داده‌های دسته‌ای (**categorical data**) بهینه شده است. این کتابخانه توسط **Yandex** توسعه داده شده و برای بسیاری از مسائل یادگیری نظارت‌شده (**supervised learning**) استفاده می‌شود.



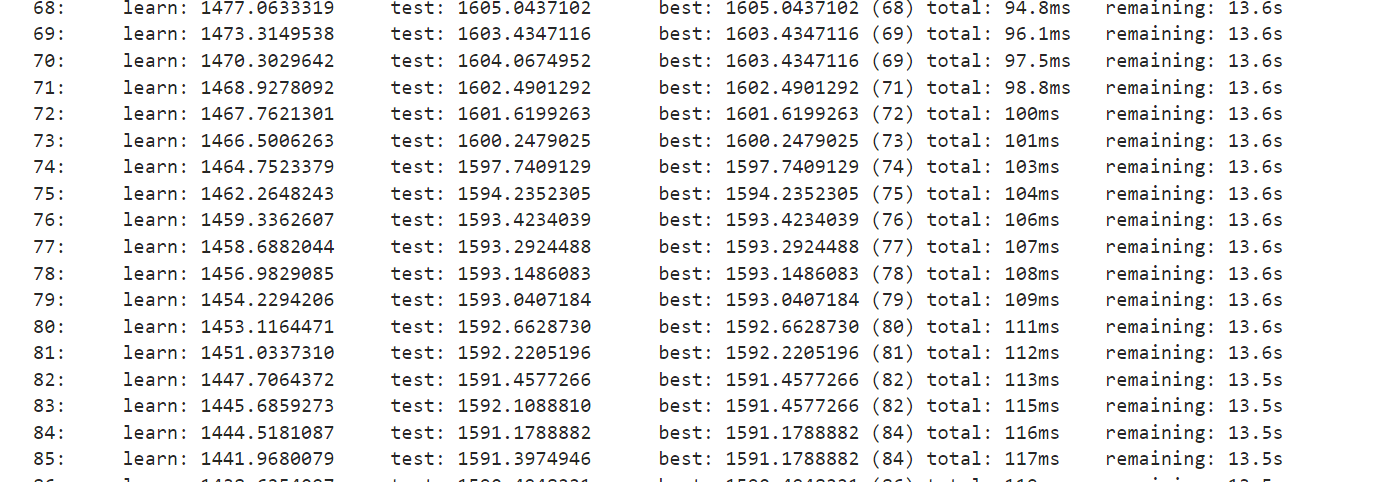
در اینجا مدل **catBoost** با پارامتر‌های مختلف ایجاد شده:

* تعداد تکرارها برای آموزش مدل در اینجا، **10000** تکرار انتخاب شده است.
* نرخ یادگیری برای آموزش مدل، تنظیم می‌کند که هر تکرار چقدر باید مدل را بهبود دهد.
* معیار ارزیابی مدل. در اینجا، معیار RMSE (**Root Mean Squared Error**) انتخاب شده است.
* به سطح گزارش‌دهی در طول آموزش، مقدار Verbos داده شده. بدین معناست که جزئیات آموزش به طور کامل نمایش داده شود.
* مقادیر **nun** با حداکثر مقدار ممکن جایگزین می‌شوند.
* مقداردهی بولین **use best model**، **True** در نظر گرفته شده به آن معناست که، مدل بهتری که در طول آموزش مشاهده شده را استفاده می‌کند. این پارامتر اطمینان می‌دهد که بهترین مدل مشاهده شده در طول آموزش برای پیش‌بینی‌ها استفاده شود.



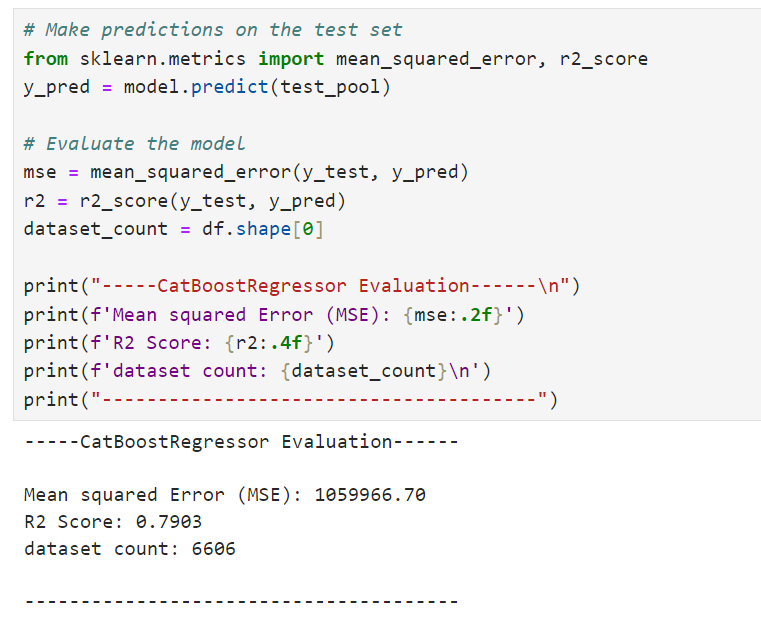
مقدار دهی **early\_stopping** بدان معناست که اگر معیار ارزیابی در طول مشخصی تکرار بهبود نیابد، آموزش متوقف می‌شود. که در اینجا 50 در نظر گرفته شده. این پارامتر کمک می‌کند تا مدل در صورتی که بهبود بیشتری در معیار ارزیابی مشاهده نشود، به زودی متوقف شود و از **overfitting** جلوگیری کند.

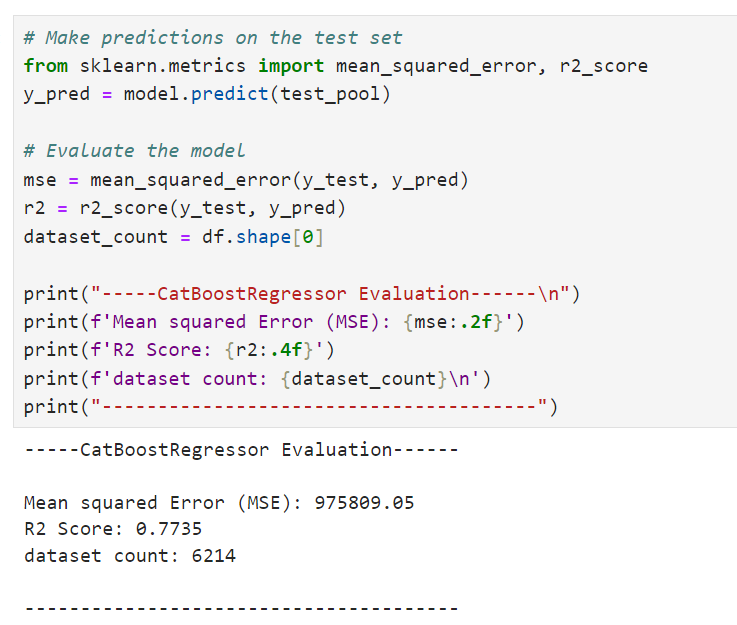
**گزارشات حین آموزش:**

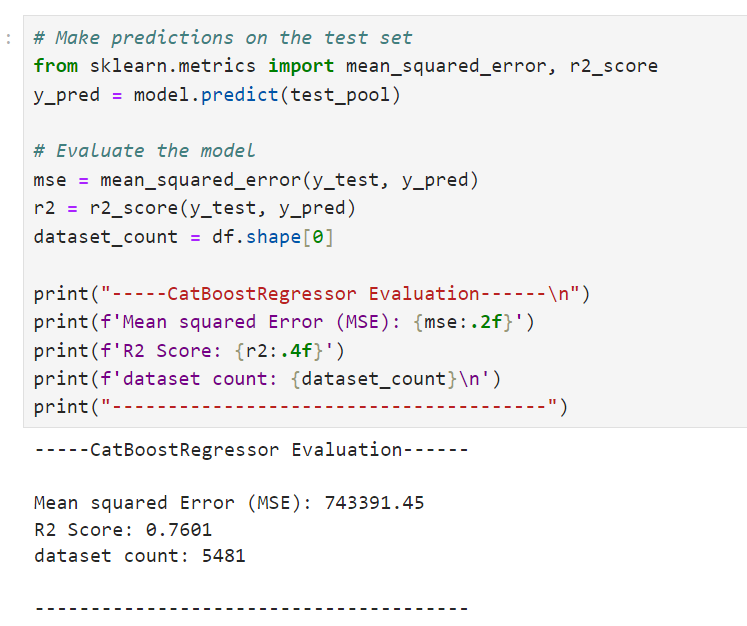


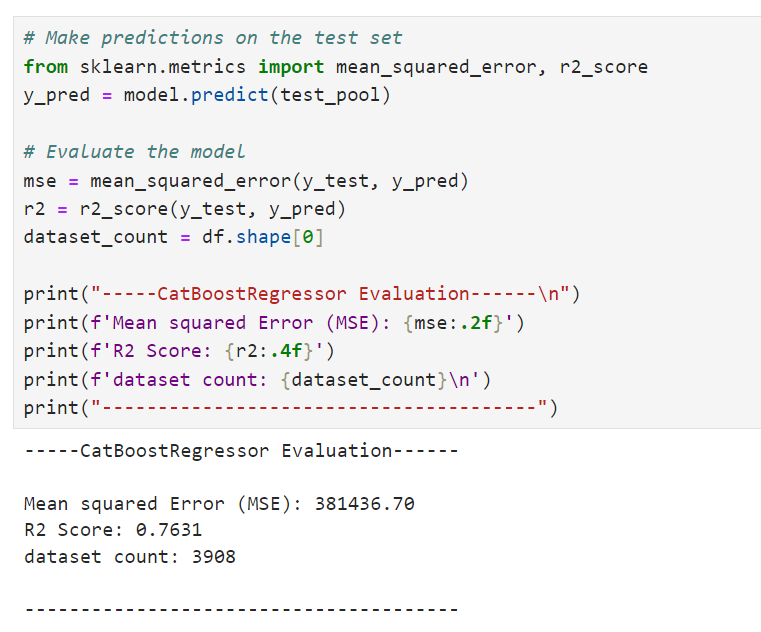
**نتایج نهایی یه دست آمده از الگوریتم CatBoostRegressor:**

مقادیر متقاوتی برای تعداد داده های متفاوتی به دست می‌آید.

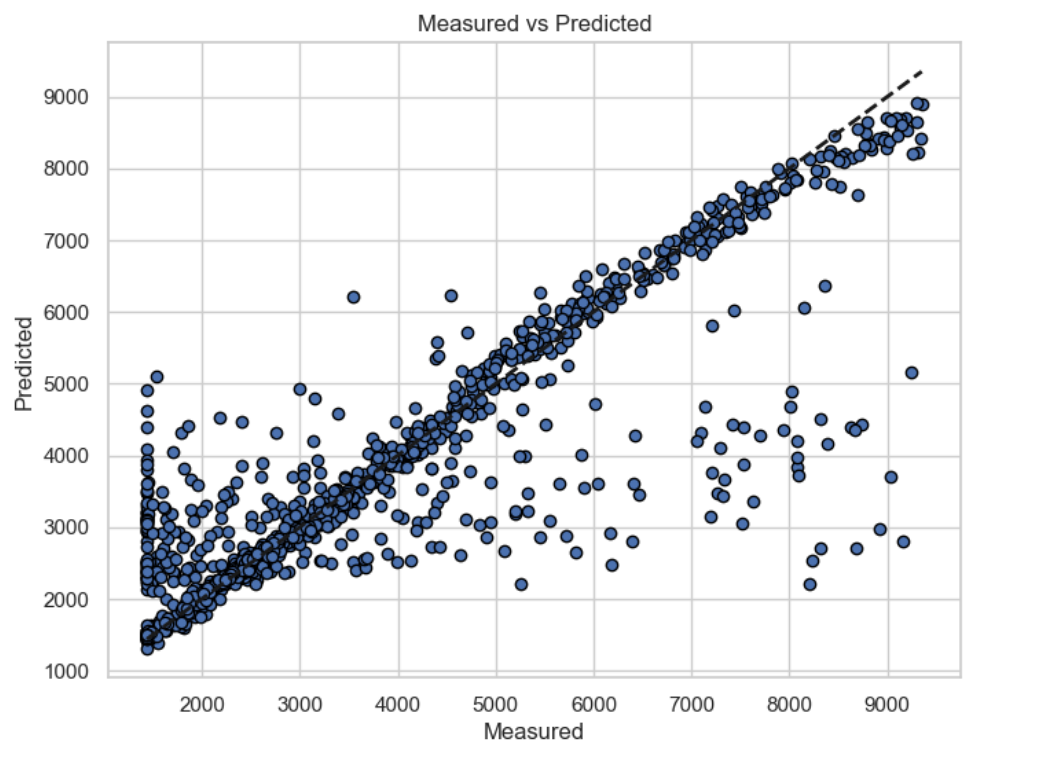








**خط فیت شده با استفاده از مدل train شده:**



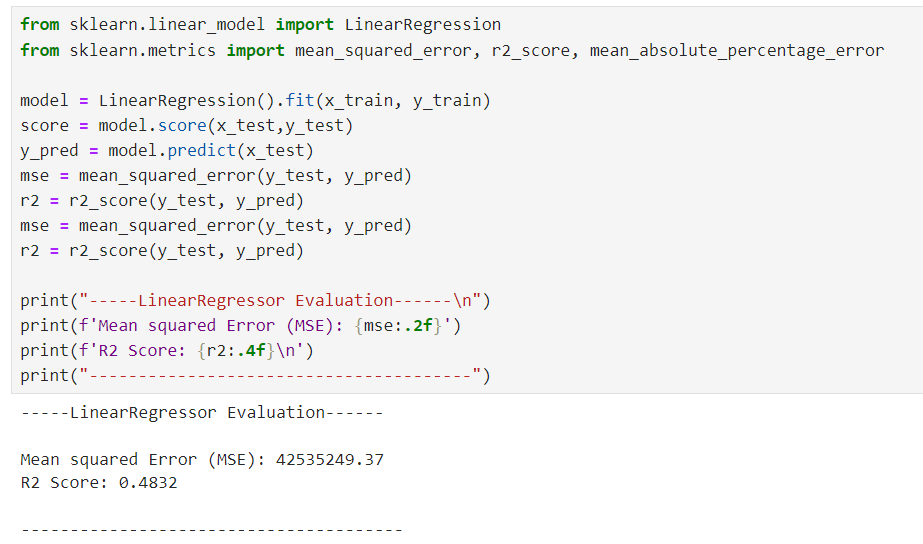
قبل از انتخاب الگوریتم **catBoost** به عنوان بهترین نتیجه، انواع الگوریتم‌های **Regression** برای این دیتاست بررسی شد.

در فایل های دیگر از انواع این الگوریتم‌ها استفاده شد.

**نتایج یک مدل شبکه عصبی با استفاده از tensorflow:**



**نتایج یک مدل Linear Regression:**

****

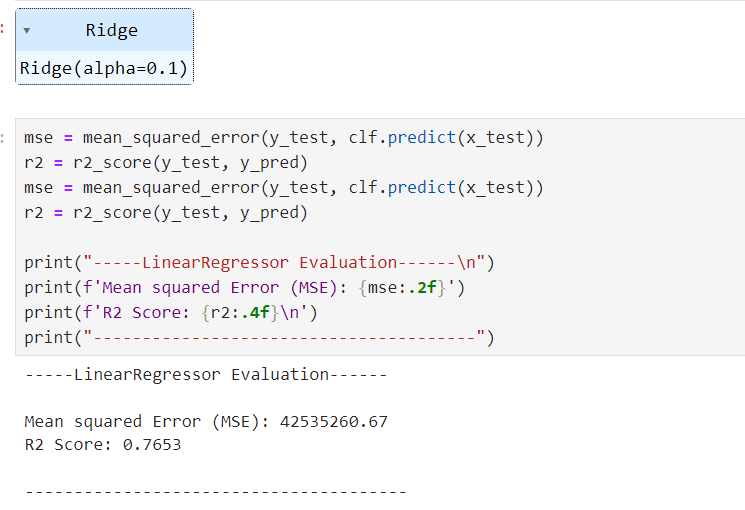
**نتایج یک مدل Gradient Boosting Regression:**

****

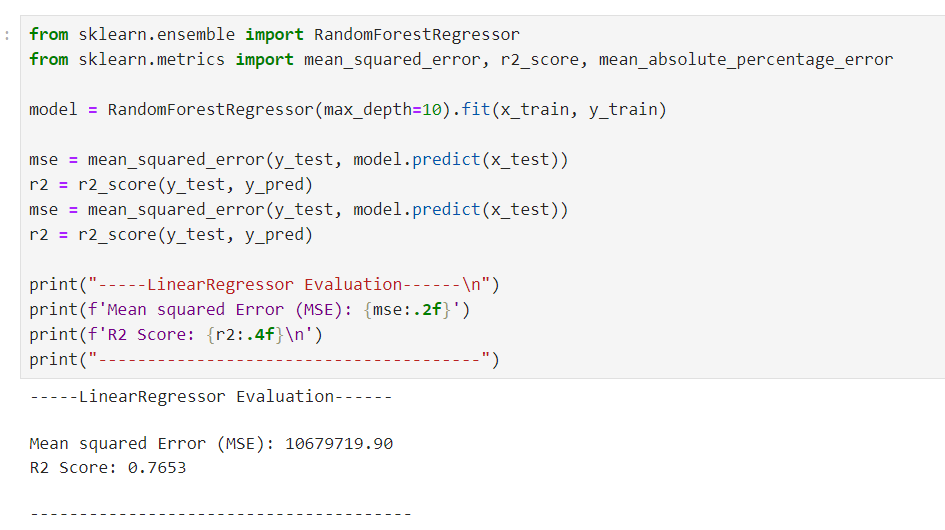
**نتایج یک مدل Polynomial Regression:**

****

**نتایج یک مدل Ridge Regression:**

****

**نتایج یک مدل RandomForest Regression:**

****