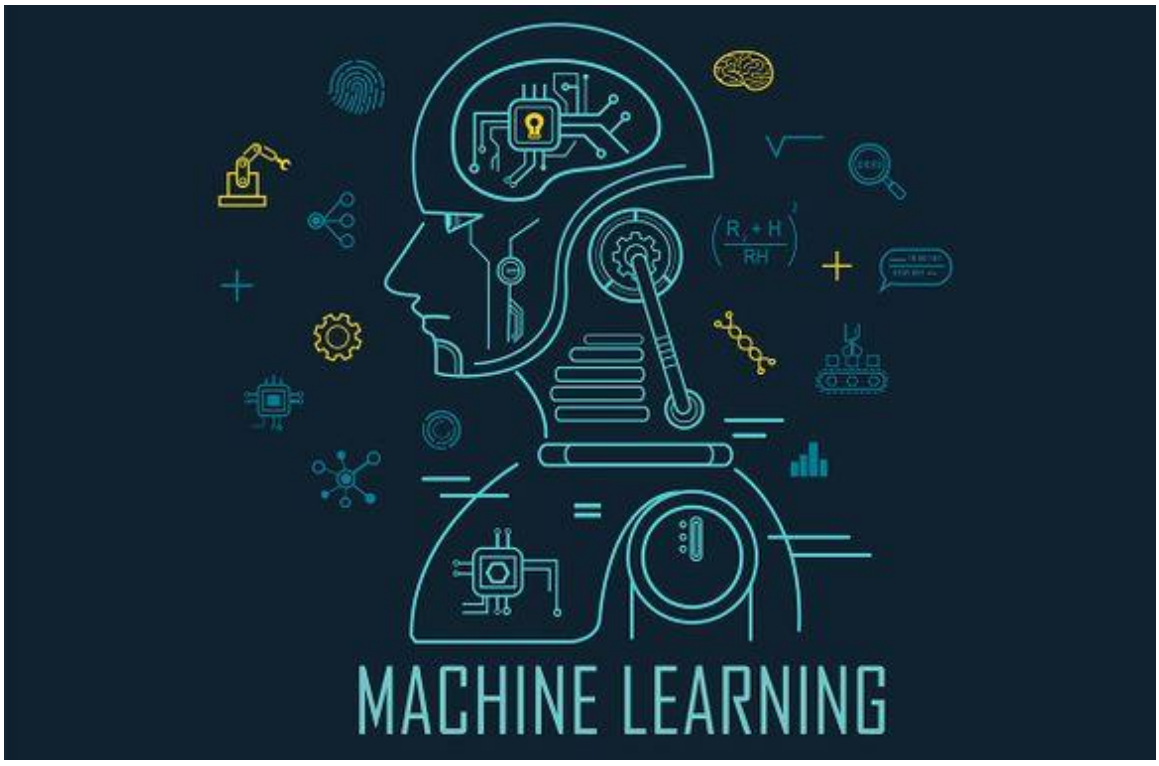


پروژه نهایی درس یادگیری ماشین

ترم پاییز 00-01

استاد عباس حسینی

ارشیا سلطانی {98109623}

پانیذ حلواچی {98109729}

فهرست مطالب

1- تحلیل اکتشافی داده

ابتدا به دو ستون SalesAmountInEuro و time_delay_for_conversion دقت کنید. این دو ستون در صورتی که کلیک منجر به خرید نشده باشد مقدار منفی یک دارند و از همین‌رو همبستگی بسیار زیادی با Sale دارند و رابطه‌ی علت و معلولی با ستون Sale دارند و بهتر است حدف شوند.

در گام بعدی به ستون product_price توجه کنید. همانطور که در کد آمده است به ازای تمامی سطرها که مقدار Sale برابر صفر است مقدار product_price نیز برابر صفر است و از همین رو این ستون نیز رابطه علت و معلولی با ستون Sale دارد و بهتر است حذف شود.

ابتدا با دستور info() اطلاعاتی درباره‌ی دیتاست داده شده به دست می آوریم که در شکل 1.1 آمده است.

```
0   Sale                       80000 non-null  int64
1   SalesAmountInEuro          10929 non-null  float64
2   time_delay_for_conversion  10886 non-null  float64
3   click_timestamp            80000 non-null  object
4   nb_clicks_1week            43188 non-null  float64
5   product_price              80000 non-null  float64
6   product_age_group          19455 non-null  object
7   device_type                79968 non-null  object
8   audience_id                22633 non-null  object
9   product_gender             19464 non-null  object
10  product_brand              27414 non-null  object
11  product_category(1)        43628 non-null  object
12  product_category(2)        43618 non-null  object
13  product_category(3)        38177 non-null  object
14  product_category(4)        23103 non-null  object
15  product_category(5)        6370 non-null   object
16  product_category(6)        947 non-null    object
17  product_category(7)        0 non-null      float64
18  product_country            60981 non-null  object
19  product_id                 61010 non-null  object
...
20  product_title              43413 non-null  object
21  partner_id                 80000 non-null  object
22  user_id                    80000 non-null  object
```

شکل 1.1: اطلاعات مربوط به دیتاست مسئله

همانطور که مشاهده میکنید تعداد مقادیر غیر null ستون product category(7) برابر صفر است و از همین‌رو این ستون به ما آگاهی‌ای نمیدهد و باید حذف شود.

از طرفی اگر دقت کنید ستون های product category (5) و product category (6) تعداد مقادیر غیر null شان نسبت به کل تعداد داده‌ها بسیار کم است و از همین رو کاندید حذف شدن هستند.

برای بررسی بیشتر، نسبت تعداد مقادیر یکتا به تعداد داده‌های موجود (داده‌های گم نشده) در هر ستون را به دست آوردیم که در شکل 1.2 آورده شده است.

```
Sale 0.0% #unique:2   unmissing:100.0
SalesAmountInEuro 67.99% #unique:7431   unmissing:13.66
time_delay_for_conversion 70.24% #unique:7647   unmissing:13.61
click_timestamp 72.06% #unique:57646   unmissing:100.0
nb_clicks_1week 2.62% #unique:1133   unmissing:53.99
product_price 4.87% #unique:3893   unmissing:100.0
product_age_group 0.05% #unique:9   unmissing:24.32
device_type 0.01% #unique:4   unmissing:99.96
audience_id 12.87% #unique:2914   unmissing:28.29
product_gender 0.06% #unique:11   unmissing:24.33
product_brand 15.73% #unique:4312   unmissing:34.27
product_category(1) 0.05% #unique:22   unmissing:54.54
product_category(2) 0.33% #unique:145   unmissing:54.52
product_category(3) 1.78% #unique:679   unmissing:47.72
product_category(4) 3.78% #unique:874   unmissing:28.88
product_category(5) 6.61% #unique:421   unmissing:7.96
product_category(6) 8.86% #unique:84   unmissing:1.18
product_category(7) 100.0% #unique:1   unmissing:0.0
product_country 0.03% #unique:17   unmissing:76.23
product_id 62.63% #unique:38212   unmissing:76.26
product_title 54.22% #unique:23537   unmissing:54.27
partner_id 0.23% #unique:183   unmissing:100.0
user_id 97.3% #unique:77838   unmissing:100.0
```

شکل 1.2: نسبت تعداد مقادیر یکتا به تعداد کل مقادیر در هر ستون (تعداد داده‌های گم نشده)

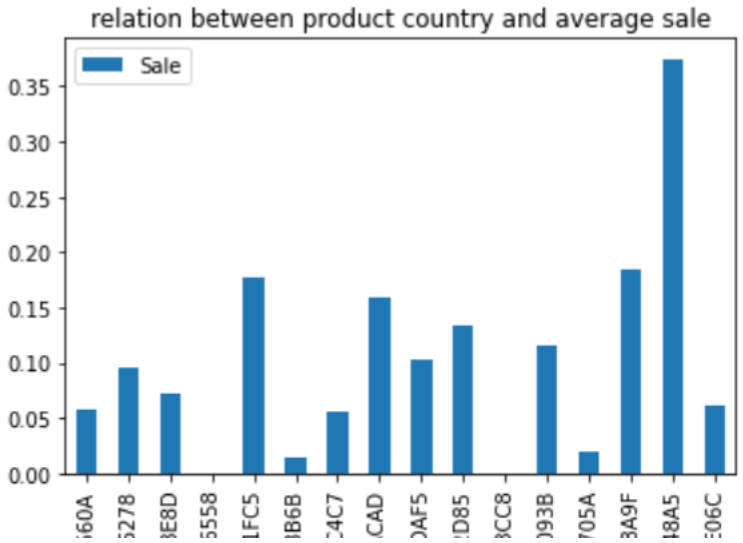
از شکل 1.2 مشاهده میشود که نسبت محاسبه شده برای ستون‌های user-id، product-id و product_title بیش از 50% است و این نشاندهنده‌ی آن است که این سه سطر نیز دانش زیادی درباره‌ی لیبلمان (Sale) به مدل نمی‌افزایند و بهتر است حذف شوند.

اکنون به صورت جداگانه ستون‌های باقی‌مانده را بررسی میکنیم.

{توجه: در ادامه در نمودارهای آورده شده، مقادیر میانگین تعداد کالای فروخته شده برتعداد کل در محور y محاسبه شده است.}

1.1 Product country:

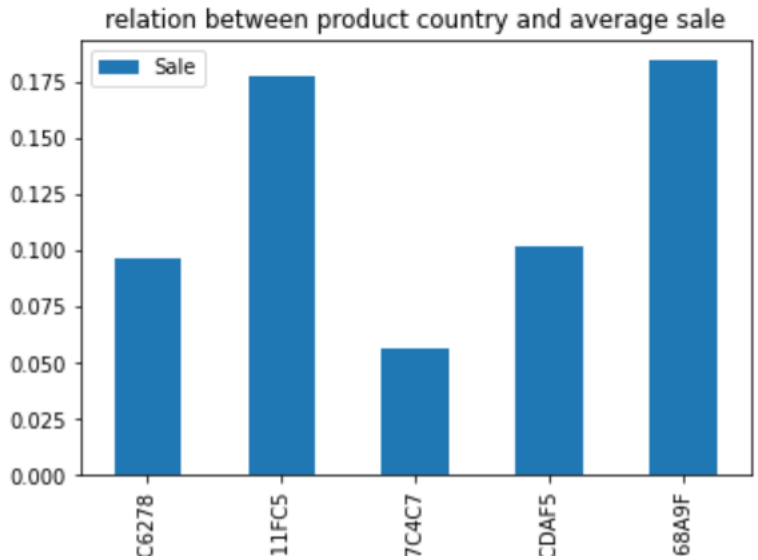
ابتدا نمودار میله‌ای (bar plot) ستون product_country را برحسب میانگین sale (میانگین تعداد کالای فروخته شده) رسم میکنیم. نمودار حاصل در شکل 1.3 آمده است.



شکل 1.3: نمودار میله‌ای product_country برحسب میانگین تعداد کالای فروخته شده

اگر دقت کنید 5 دسته از بقیه دسته ها تعداد کالای فروخته شده‌شان بیشتر است. اکنون برای تعداد کالاهای موجود در هر دسته، حد پایینی قرار میدهیم (ما 1000 درنظر گرفتیم) و مجددا نمودار بالا را رسم میکنیم که در شکل 1.4 آمده است.

اگر به شکل 1.4 نگاه کنید نسبت به قبل مقادیر بیشتر بهم نزدیکند و مقادیر در بازه‌ی کوچکتری هستند. از همین رو بهتر است country محصول‌ها را 6 دسته درنظر بگیریم: 5 دسته با بیشترین میانگین Sale و یک دسته با عنوان other که شامل دیگر product_country ها است.



شکل 1.4: نمودار میله‌ای 5 country اصلی برحسب میانگین تعداد کالای فروخته شده

1.2 Product brand:

نمودار میله‌ای ستون product_brand را برحسب میانگین sale رسم میکنیم. نمودار حاصل در شکل 1.5 آمده است .

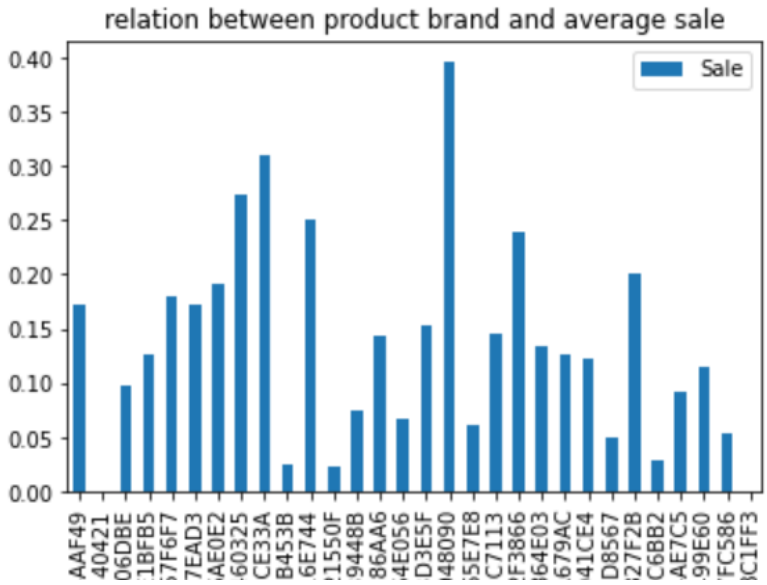
برای آنکه بازه‌ی اطمینان میانگین ستون‌ها کوچک باشد، برندهایی که تعداد کالای موجودشان کم است را درنظر نمی‌گیریم و حد پایین تعداد کالاهای موجود در هر دسته را 100 درنظر میگیریم.

مقادیر میانگین تعداد کالای فروخته‌شده اکثر brand ها در بازه‌ی یکسانی هستند و از همین رو تمام برندهای موجود را در نظر میگیریم.

1.3 Product age group:

نمودار میله‌ای ستون product_age_group را برحسب میانگین تعداد کالای فروخته شده رسم میکنیم. نمودار حاصل در شکل 1.6 آمده است.

مقادیر میانگین تعداد کالاهای فروخته شده در بازه‌های سنی مختلف بهم نزدیکند از همین رو تمام بازه‌های سنی را در نظر میگیریم.



شکل 1.5: نمودار میله‌ای product_brand برحسب میانگین تعداد کالای فروخته شده

1.4 Product category:

باتوجه به اینکه هنگامی که دو مقدار در ستون product_category(i) با هم برابر باشند مشاهده میکنیم که این دو سطر در تمام product_category های قبلی نیز برابرند، متوجه میشویم که product_category ساختار درختی دارد.

برای مثال product_category(1) نشاندهنده‌ی نوع کالا است (مانند کالای دیجیتال، خودرو و ...) .

و در دسته‌ی دوم، دسته‌بندی های ریزتر قرار گرفته است.

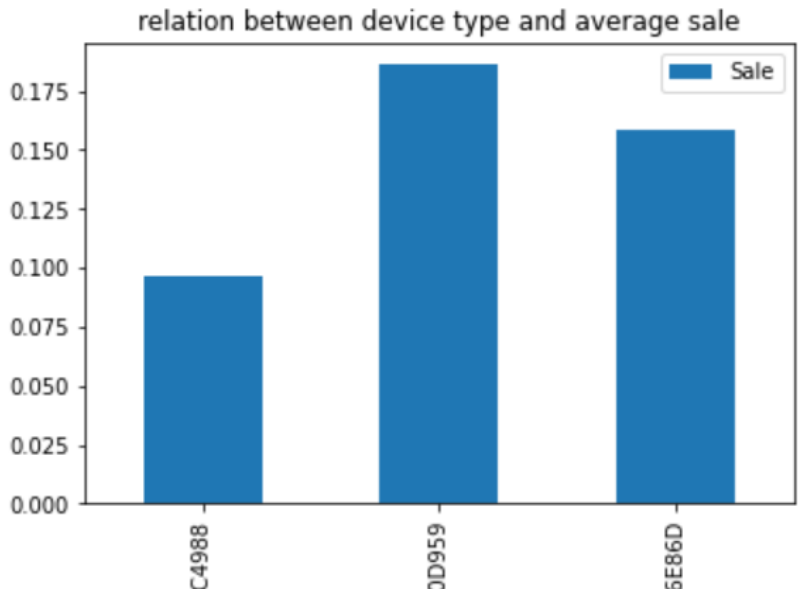
از همین رو، درنظرگرفتن product_category(1) به تنهایی کافیست و میتوان از دسته‌بندی های فرعی صرف‌نظر کرد.

در صفحه‌ی بعد نمونه‌ای از این نوع دسته‌بندی محصول که مربوط به دیجی‌کالا است، در شکل 1.6 آمده است.

شکل 1.6: عکس ها به ترتیب نشاندهنده‌ی دسته بندی اصلی، دسته بندی فرعی اول و دسته‌بندی فرعی دوم هستند.

1.5 Device type:

نمودار میله‌ای ستون device_type را برحسب میانگین تعداد کالای فروخته شده رسم میکنیم. نمودار حاصل در شکل 1.7 آمده است. این ستون میتواند اطلاعات خوبی به ما دهد پس آن را حذف نکرده و تمام دسته‌هایش را در نظر میگیریم.



شکل 1.7: نمودار میله‌ای device type برحسب میانگین تعداد کالای فروخته شده

1.6 Product gender:

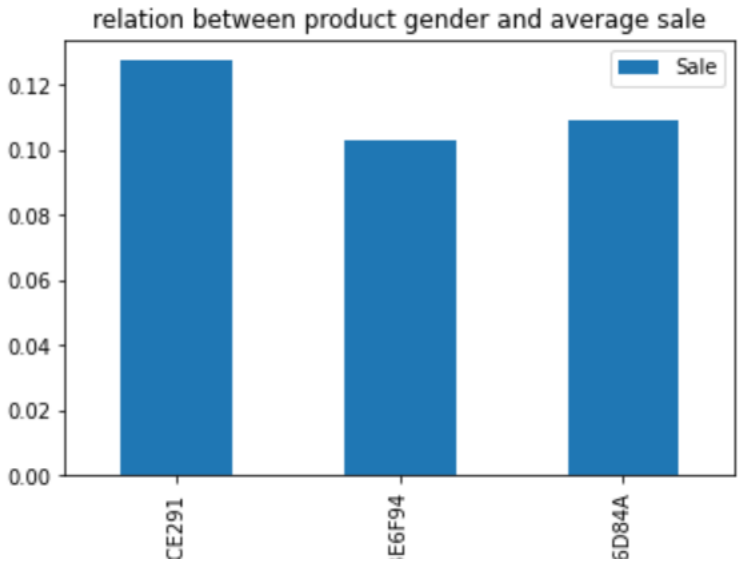
ابتدا میانگین تعداد کلاهای فروخته شده در هر دسته را محاسبه میکنیم. حاصل در جدول 1.1 آورده شده است.

همانطور که مشاهده میشود چند دسته مقادیرشان با گپ مشهودی از مقادیر دیگر دسته‌ها بیشتر است. از همین رو خوب است تنها این سه دسته (به ترتیب دسته‌های دوم، هفتم و هشتم جدول) را به همراه یک دسته با نام other شامل دسته‌های دیگر، در نظر بگیریم.

حال نموار میله‌ای product_gender را برحسب میانگین تعداد کالاهای فروخته شده برای این سه دسته اصلی رسم میکنیم. نمودار حاصل در شکل 1.8 آمده است.

جدول 1.1: جدول میانگین تعداد کالای فروخته شده برای هر دسته جنسیت

product_gender	Sale
0FB06F1EAC1E00A436B336C5DF3C14AF	4
1B491180398E2F0390E6A588B3BCE291	6769
26FE89E9DD2E6FC18AA5BE1F6D5A6870	19
28F311FA00BD3B4D076659D87EE3AE8D	142
6EFABCEDA36A931DBF760F88970BAF0E	110
86E2AFBF909EC95B069893FF0BBC5B26	46
A5D15FC386510762EC0DDFF54ABE6F94	9305
C45A9AC6D102ACAEEDF0D6F78636D84A	2587
D7B42B0C4D807EF1C13F79948743E9DA	457
D894202FD3ABAE0B55F9FAE133412DE5	25

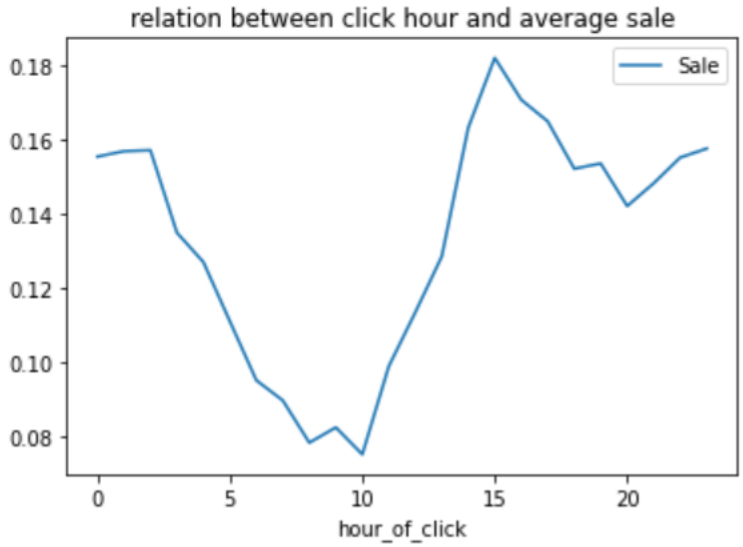


شکل 1.8: نمودار میله‌ای product_gender را برحسب میانگین تعداد کالاهای فروخته شده

1.7 Click timestamp:

این ستون تمام مقادیرش مشخص است و none ندارید. این ستون شامل روز و ساعت است که زمان انجام شدن کلیک را نشان میدهد ولی باتوجه به اینکه داده‌های داده شده تنها دو روز را دربرمیگیردند دانستن روز نمیتواند اطلاعات مفیدی به مدل دهد از همین رو ما تنها ساعت را درنظر گرفتیم.

نمودار میانگین تعدادکالاهای فروخته شده برحسب ساعت انجام شدن کلیک در شکل 1.9 آمده است.

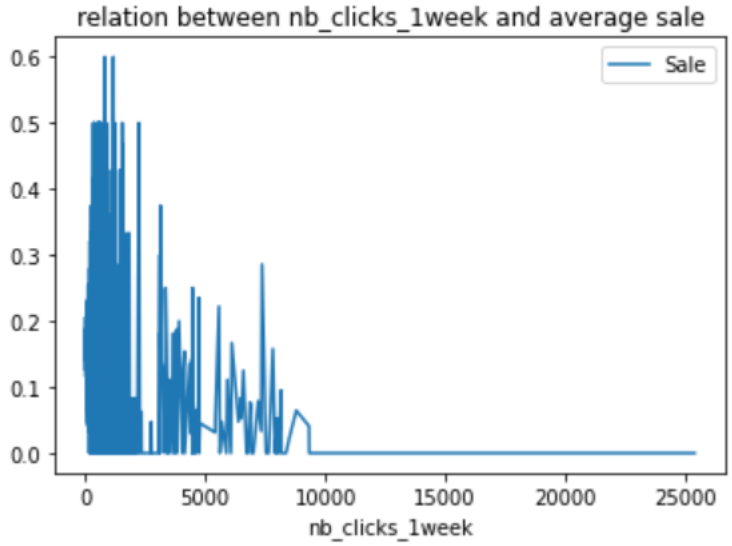


شکل 1.9: نمودار میانگین تعداد کالاهای فروخته‌شده برحسب ساعت انجام شدن کلیک

برای سهولت، 8 بازه برای زمان درنظر میگیریم که هر بازه شامل سه ساعت است و هر ساعت را به بازه‌ی مربوطش مپ میکنیم.

1.8 Nb clicks 1week:

این ستون نشان‌دهنده‌ی تعداد کلیک‌هایی است که یک تبلیغ در هفته‌ی آخر بازه دو 90 روزه دریافت کرده است. نمودار میانگین تعداد کالاهای فروخته شده برحسب nb_clicks_1week در شکل 1.10 آمده است.

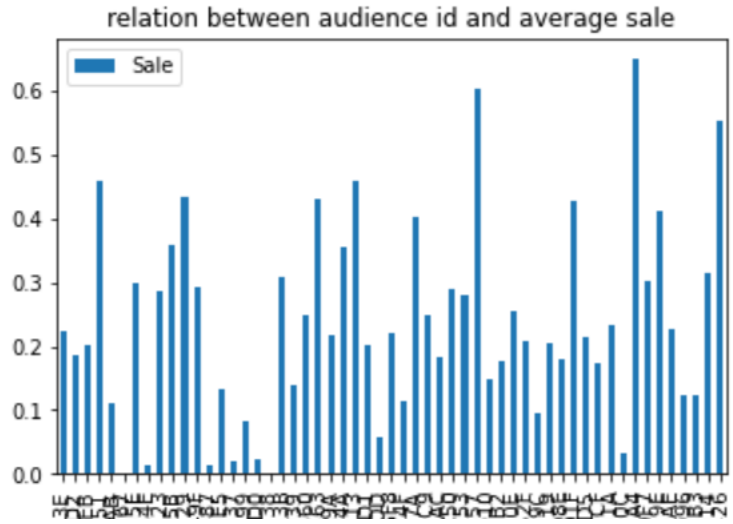


شکل 1.10: نمودار میانگین تعداد کالاهای فروخته شده برحسب nb_clicks_1week

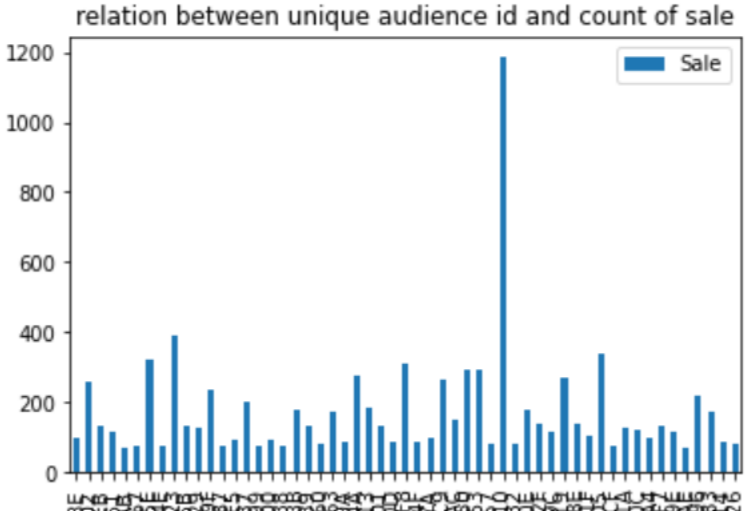
مورد جالبی که در شکل 1.10 مشاهده میشود این است که با افزایش nb_clicks_1week درصد کالاهای خریداری شده کاهش میابد و حتی به صفر میرسد. دلیل این اتفاق به خاظر آن است که این تبلیغ ها clickbait بوده‌اند. به این منظور که در تبلیغ موردی بیان میشود که برای اکثر مخاطبان جذاب است و افراد بیشتری روی تبلیغ کلیک میکنند ولی چون آنچه در تبلیغ آورده شده است، حقیقی نیست افراد آن را خریداری نمیکنند.

1.9 Audience id:

این ستون ویژگی‌های کاربر را نشان میدهد که به خاطر حفظ حریم خصوصی کاربران کدگذاری شده است. برای آنکه شهود بیشتری از این ویژگی بیابیم، نمودار میله‌ای میانگین تعداد کالاهای فروخته شده برحسب audience id های یکتا در شکل 1.11 و نمودار میله‌ای تعداد کالاها برحسب audience id های یکتا در شکل 1.12 آمده است.



شکل 1.11: نمودار میله‌ای میانگین تعداد کالاهای فروخته شده برحسب audience id های یکتا

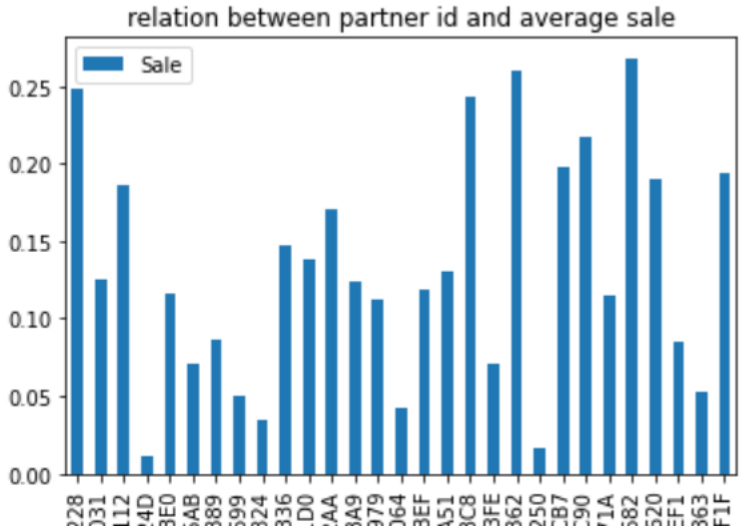


شکل 1.12: نمودار میله‌ای تعداد کالاها برحسب audience id های یکتا

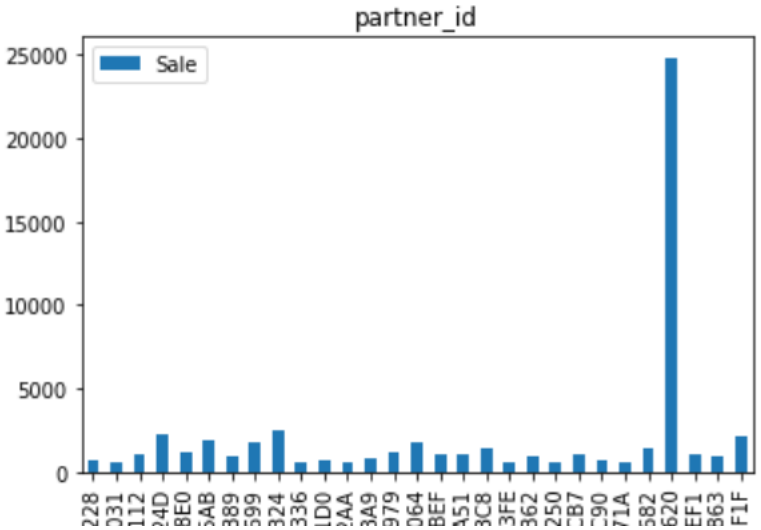
تعداد مقادیر یکتا برای audience id زیاد است و تمامی آنها اطلاعات مفیدی به مدل نخواهند داد از همین رو بهتر است شرطی را برای audience id هایی که به عنوان دسته‌های اصلی درنظر میگیریم، قرار دهیم. ما برای تعداد کالاهای مربوط به هر audience id حدپایینی میگذاریم و کتگوری‌هایی از audience id را که تعداد کالاهایشان کمتر از این حد پایین باشند را یک دسته درنظر میگیریم. ما حد پایین را در اینجا 70 درنظر گرفتیم.

1.10 Partner id:

این ویژگی، شناسه فروشنده محصول را نشان میدهد. نمودار میله‌ای میانگین تعداد کالاهای فروخته شده برحسب partner id های یکتا در شکل 1.13 و نمودار میله‌ای تعداد کالاها برحسب partner id های یکتا در شکل 1.14 آمده است.



شکل 1.13: نمودار میله‌ای میانگین تعداد کالاهای فروخته شده برحسب partner id های یکتا



شکل 1.14: نمودار میله‌ای تعداد کالاها برحسب partner id های یکتا

برای آنکه بازه‌ی اطمینان میانگین ستون‌ها کوچک باشد، partner idهایی که تعداد کالای موجودشان کم است را درنظر نمی‌گیریم و حد پایین تعداد کالاهای موجود در هر دسته را 500 درنظر میگیریم.

2- مهندسی ویژگی‌ها

پس از آنکه در مرحله‌ی قبل داده ها را بررسی کردیم حال در این بخش، تغییراتی که برای ماشین‌فهم تر کردن و اصلاح داده‌ها باید انجام دهیم را شرح میدهیم.

2.1 حذف ستون‌های زائد:

باتوجه به بررسی های بخش قبل، ستون های زیر را حذف کردیم:

$$Removed\ Columns = \{SalesAmountInEuro,\ time\_delay\_for\_conversion,\ product\_price,\ product\_id,\ product\_title,\ user\_id,\ product\_category(2),\ product\_category(3),\ product\_category(4),\ product\_category(5),\ product\_category(6),\ product\_category(7)\}$$

2.2 رمزگذاری(encoding) داده‌های categorical:

دو شیوه برای رمزگذاری داده‌های categorical در نظر گرفتیم:

استفاده از onehot_encoder:

برای ویژگی‌های زیر از one hot encoder استفاده کردیم به اینصورت که برای هر دسته‌ی یکتا یک ستون درنظر گرفتیم که برای هر داده ستون متناظر با دسته‌ی ویژگی‌اش برابر یک میشود و ستون‌های دیگر که مربوط به دسته‌های دیگر ویژگی categorical هستند برابر صفر میشوند.

$$one\ hot\ encoded\ columns = \{product\_category(1),\ device\_type,\ product\_country,\ product\_gender, click\_timestamp\}$$

1- استفاده از میانگین درصد تعداد کالاهای فروخته‌شده محاسبه شده برای داده‌های آموزشی:

در برخی از ویژگی‌ها میانگین درصد کالاهای فروخته شده برای هر دسته اطلاعات بیشتری نسبت به اینکه داده در کدام دسته است به ما میدهد. از همین رو برای برخی از ویژگی ها (که در زیر آمده است) به جای استفاده از one hot encod از این روش زیر برای رمزگذاری استفاده کردیم:

در این روش ابتدا روی هر کدام از دسته‌های مختلف ویژگی، groupby میکنیم. سپس مقدار mean کالاهای فروخته شده را محاسبه میکنیم و برای هر دسته مقدار mean متناطرش را قرار میدهیم. اگر ویژگی موردبررسی در میان ویژگی‌هایی که فیلتر کردیم بود در اینصورت مقدارش را برابر میانگین sale قرار میدهیم.

$$sale\ percentage\ encoded\ columns = \{product\_brand,\ \ audience\_id,\ partner\_id,\ \ product\_age\_group\}$$

2.3 مقداردهی به مقادیر نامشخص و Noneها:

داده‌های عددی :

تنها ستون عددی که در مرحله اول حذف نشده است Nb_clicks_1week است که مقادیر درایه هایی از این ستون که None هستند را برابر میانگین مقادیر موجود در این ستون میگذاریم.

داده‌های رمزنگاری شده با one hot encoder:

برای این ستون‌ها، خانه‌هایی که مقدار گمشده یا None دارند را در دسته‌ای جدا میگیریم که دسته‌های فیلترشده (دسته‌های فرعی) را نیز در بردارد.

داده‌های رمزنگاری شده با sale percentage:

برای این ستون‌ها مقدار میانگین sale را برای خانه‌هایی که مقادیرشان گمشده هستند، در نظر میگیریم.

2.4 نرمالایز کردن ستون‌ها:

برای نرمالایز کردن ستون ها ابتدا ستون ها به دو دسته‌ی ستون هایی با مقادیر پیوسته و ستون های categorical تقسیم میشوند .

ستون های catergorical چون به صورت one-hot درآمده اند نیاز به نرمالایز کردن ندارند با این حال ستون های پیوسته به دلیل نوعشان نیاز به normilize کردن دارند که با صفر کردن میانگین و یک کردن std انجام شده است. Mean و std هر ستون بر اساس دیتای train محاسبه شده است و از این دید مشکلی مجود ندارد.

از همین رو ، میانگین را از همه‌ی داده‌های عددی و رمزنگاری شده با Sale percentage را کاسته و سپس بر انحراف معیار تقسیم کردیم.

$$z = \frac{x - \mu}{\sigma}$$

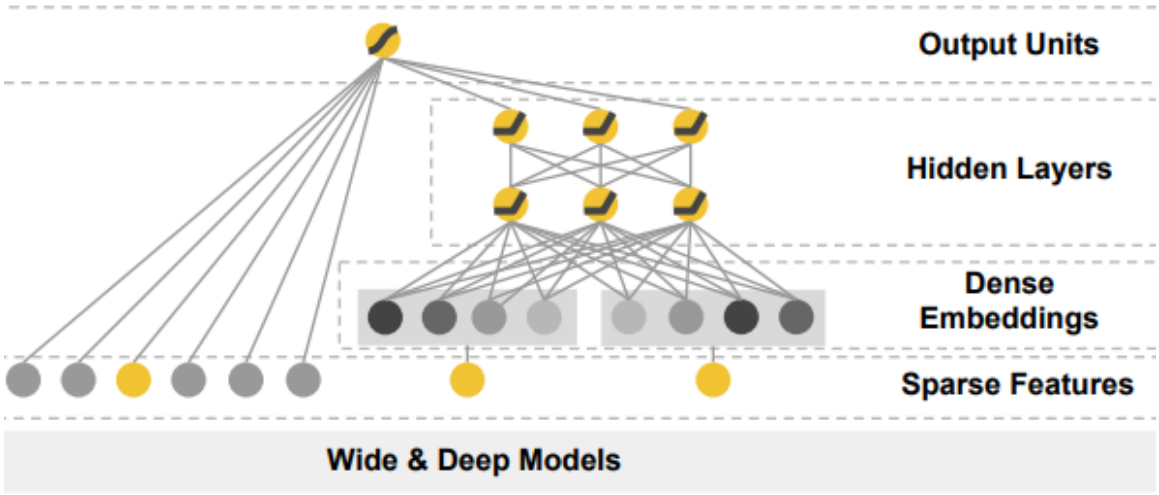
2- مدل‌های پیاده‌سازی شده:

در زیربخش‌های زیر تفسیر مدل طراحی شده در پروژه آمده است.

3.1 مدل wide:

این مدل بر اساس مقاله‌ی Wide & Deep Learning for Recommender Systems, Heng-Tze Cheng, نوشته شده است. در این مدل ابتدا داده‌ی categorical به embedding ها تبدیل میشود و سپس با دیتای پیوسته ترکیب میشوند.

پس از این ترکیب شدن از یک شبکه‌ی عادی عبور میکنند. شمای کلی در تصویر 3.1 وجود دارد.

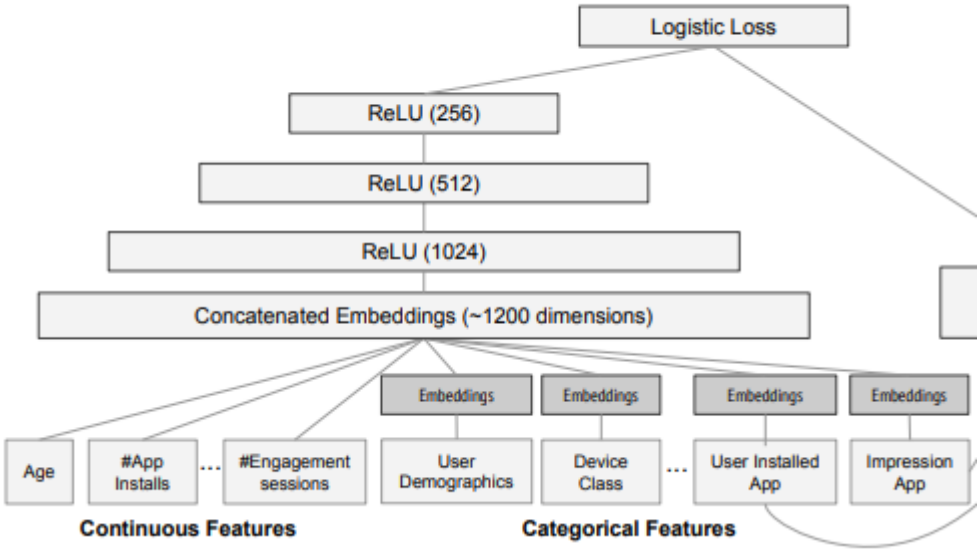


شکل 3.1: نمای کلی مدل wide

همان طور که مشخص است در سمت راست پایین ورودی های one-hot را میبینیم و در سمت چپ پایین ورودی های پیوسته را، که ورودی های one-hot به embeddingها تبدیل میشوند و از یک سری hidden layer عبور میکنند.

البته این تنها شمای کلی است و در تصویر دقیق تر (شکل 3.2) میبینیم که همه‌ی ورودی ها من جمله ورودی های پیوسته به hidden layers ها فرستاده میشوند.

در پیاده سازی، ما برای افزایش سرعت یادگیری بعد از هر ReLU یک Batch norm هم اضافه شده است تا با پخش کردن دیتا دور صفر مشکل کند بودن یادگیری را برطرف کند.



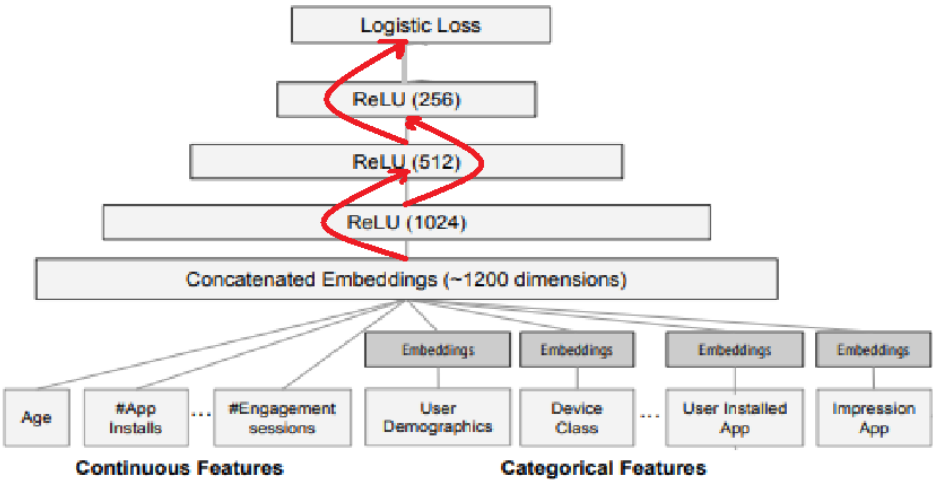
شکل 3.2: تصویر دقیق‌تر از شمای کلی مدل به همراه ورودی‌ها

3.2 مدل Residual wide deep:

با توجه به اهمیت ارتباطات residual که در مقاله ی Deep Residual Learning for Image Recognitio, Kaiming He مورد بررسی قرار گرفته است، تصمیم گرفتیم که نوعی ارتباط مشابه با این ارتباط را به مدل اضافه کنیم، انگیزه ی اصلی این کار این بود که به نظر می‌آمد مدل به دیتای لایه های اول نیاز دارد و این دیتا از اهمیت بالایی برخوردار باشد. این تحلیل با توجه به تحلیل های صورت گرفته در بالا و مثلا اهمیت بالای ساعت کلیلک کردن روی تبلیغ است. برای حل این مشکل این ارتباطات به مدل اضافه شد.

شمای کلی این مدل در شکل 3.3 آورده شده است.

اارتباطات قرمز نشان‌داده شده در شکل 3.3 به مدل کمک میکنند که مدل اطلاعات مهم که در لایه های اول موجود است بیشتر ببیند و در نتیجه بتواند بیشتر از آن ها استفاده کند.

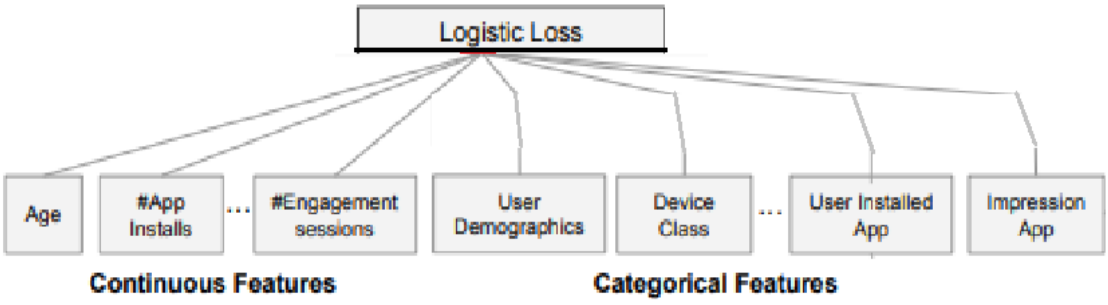


شکل 3.3: شمای کلی مدل Residual wide deep

3.1 مدل Linear:

برای تست مدل های غیر عمیق از مدل خطی استفاده شده است این مدل به شدت ساده است و دیگر نه embedding دارد و نه چیز دیگری، در واقع ساختار آن مشابه این شکل است.

نمای کلی مدل خطی در شکل 3.4 آمده است.



3- TUNE کردن پارامترها :

برای Tune کردن مدل ها دوبه دو روش grid search و random فکر کردیم که با توجه به کم بودن تعداد پارامتر هایی که نیاز به tune شدن داشتند (learning rate , batch size) .تصمیم گرفتیم که از روش grid search استفاده کنیم.

در این بخش با توجه به این که مدل تنها دو عدد دارد، یعنی learning rate و batch size از روش grid search استفاده میکنیم.

مقادیری که برای پارامتر learning rate و batch size برای grid search در نظر گرفته شده است برابرند با:

$$learning\ rate \in \{0.000001, 0.0001, 0.001, 0.01, 0.1\}$$

$$batch\ size \in \{100, 500, 4000\}$$

در ادامه نتایج به تفکیک مدل آمده است.

4.1 مدل Wide and deep model:

```
model_classes=WideDeepModel lr=1e-05 batch_size=100 f1=0.323 recall=0.757 precision=0.205
model_classes=WideDeepModel lr=1e-05 batch_size=500 f1=0.322 recall=0.741 precision=0.206
model_classes=WideDeepModel lr=1e-05 batch_size=4000 f1=0.214 recall=0.189 precision=0.248
model_classes=WideDeepModel lr=0.0001 batch_size=100 f1=0.318 recall=0.78 precision=0.199
model_classes=WideDeepModel lr=0.0001 batch_size=500 f1=0.32 recall=0.749 precision=0.204
model_classes=WideDeepModel lr=0.0001 batch_size=4000 f1=0.283 recall=0.292 precision=0.274
model_classes=WideDeepModel lr=0.001 batch_size=100 f1=0.32 recall=0.637 precision=0.213
model_classes=WideDeepModel lr=0.001 batch_size=500 f1=0.321 recall=0.649 precision=0.213
model_classes=WideDeepModel lr=0.001 batch_size=4000 f1=0.261 recall=0.221 precision=0.318
model_classes=WideDeepModel lr=0.01 batch_size=100 f1=0.308 recall=0.659 precision=0.201
model_classes=WideDeepModel lr=0.01 batch_size=500 f1=0.313 recall=0.797 precision=0.194
model_classes=WideDeepModel lr=0.01 batch_size=4000 f1=0.31 recall=0.581 precision=0.212
model_classes=WideDeepModel lr=0.1 batch_size=100 f1=nan recall=0.0 precision=nan
model_classes=WideDeepModel lr=0.1 batch_size=500 f1=0.267 recall=0.901 precision=0.157
model_classes=WideDeepModel lr=0.1 batch_size=4000 f1=0.242 recall=1.0 precision=0.137
```

همان طور که میبینیم بهترین نتیجه را با

$$learning\ rate = 0.00001 \text{ AND batch size} = 100$$

میگیریم.

4.2 مدل Res Wide and deep model:

```
model_classes=ResWideDeepModel lr=1e-05 batch_size=100 f1=0.313 recall=0.763 precision=0.197
model_classes=ResWideDeepModel lr=1e-05 batch_size=500 f1=0.313 recall=0.741 precision=0.198
model_classes=ResWideDeepModel lr=1e-05 batch_size=4000 f1=0.302 recall=0.642 precision=0.197
model_classes=ResWideDeepModel lr=0.0001 batch_size=100 f1=0.313 recall=0.735 precision=0.199
model_classes=ResWideDeepModel lr=0.0001 batch_size=500 f1=0.311 recall=0.779 precision=0.194
model_classes=ResWideDeepModel lr=0.0001 batch_size=4000 f1=0.325 recall=0.686 precision=0.213
model_classes=ResWideDeepModel lr=0.001 batch_size=100 f1=0.319 recall=0.454 precision=0.246
model_classes=ResWideDeepModel lr=0.001 batch_size=500 f1=0.318 recall=0.761 precision=0.201
model_classes=ResWideDeepModel lr=0.001 batch_size=4000 f1=0.327 recall=0.659 precision=0.217
model_classes=ResWideDeepModel lr=0.01 batch_size=100 f1=0.32 recall=0.562 precision=0.224
model_classes=ResWideDeepModel lr=0.01 batch_size=500 f1=0.321 recall=0.696 precision=0.209
model_classes=ResWideDeepModel lr=0.01 batch_size=4000 f1=0.323 recall=0.495 precision=0.24
model_classes=ResWideDeepModel lr=0.1 batch_size=100 f1=0.285 recall=0.832 precision=0.172
model_classes=ResWideDeepModel lr=0.1 batch_size=500 f1=0.308 recall=0.608 precision=0.207
model_classes=ResWideDeepModel lr=0.1 batch_size=4000 f1=0.242 recall=0.999 precision=0.138
```

در اینجا مدل به طور کلی بهتر عمل میکند و بهترین parameter ها برای آن

$$learning\ rate\ =\ 0.001\ \text{AND}\ \text{batch size}\ =\ 4000$$

است.

4.3 مدل Linear:

```
model_classes=LinearModel lr=1e-05 batch_size=100 f1=0.216 recall=0.213 precision=0.218
model_classes=LinearModel lr=1e-05 batch_size=500 f1=0.204 recall=0.193 precision=0.216
model_classes=LinearModel lr=1e-05 batch_size=4000 f1=0.196 recall=0.182 precision=0.211
model_classes=LinearModel lr=0.0001 batch_size=100 f1=0.294 recall=0.477 precision=0.213
model_classes=LinearModel lr=0.0001 batch_size=500 f1=0.245 recall=0.268 precision=0.225
model_classes=LinearModel lr=0.0001 batch_size=4000 f1=0.205 recall=0.195 precision=0.216
model_classes=LinearModel lr=0.001 batch_size=100 f1=0.321 recall=0.763 precision=0.203
model_classes=LinearModel lr=0.001 batch_size=500 f1=0.312 recall=0.714 precision=0.2
model_classes=LinearModel lr=0.001 batch_size=4000 f1=0.265 recall=0.308 precision=0.233
model_classes=LinearModel lr=0.01 batch_size=100 f1=0.321 recall=0.696 precision=0.208
model_classes=LinearModel lr=0.01 batch_size=500 f1=0.322 recall=0.721 precision=0.207
model_classes=LinearModel lr=0.01 batch_size=4000 f1=0.317 recall=0.752 precision=0.201
model_classes=LinearModel lr=0.1 batch_size=100 f1=0.308 recall=0.441 precision=0.236
model_classes=LinearModel lr=0.1 batch_size=500 f1=0.319 recall=0.682 precision=0.208
model_classes=LinearModel lr=0.1 batch_size=4000 f1=0.315 recall=0.771 precision=0.198
```

مدل linear هم بهترین پارامترهایش

$$learning\ rate\ =\ 0.01\ \text{AND batch size}\ =\ 500$$

میگیرد.

نتیجه‌گیری:

با تجمیع تمام مدل ها و hyper tuning روی تمام آن ها بهترین مدل، مدل ResWideDeep است.

با توجه به بالاتر بودن F1 score این مدل تصمیم گرفتیم که از این مدل به عنوان مدل نهایی استفاده کنیم.

این مدل با توجه به زمان کوتاه train از دید زمانی مدل سریعی است و f1 score بالایی هم کسب میکند.

4- نتایج مدل نهایی:

5.1 قدرت و ضعف های مدل:

ضعف‌ها:

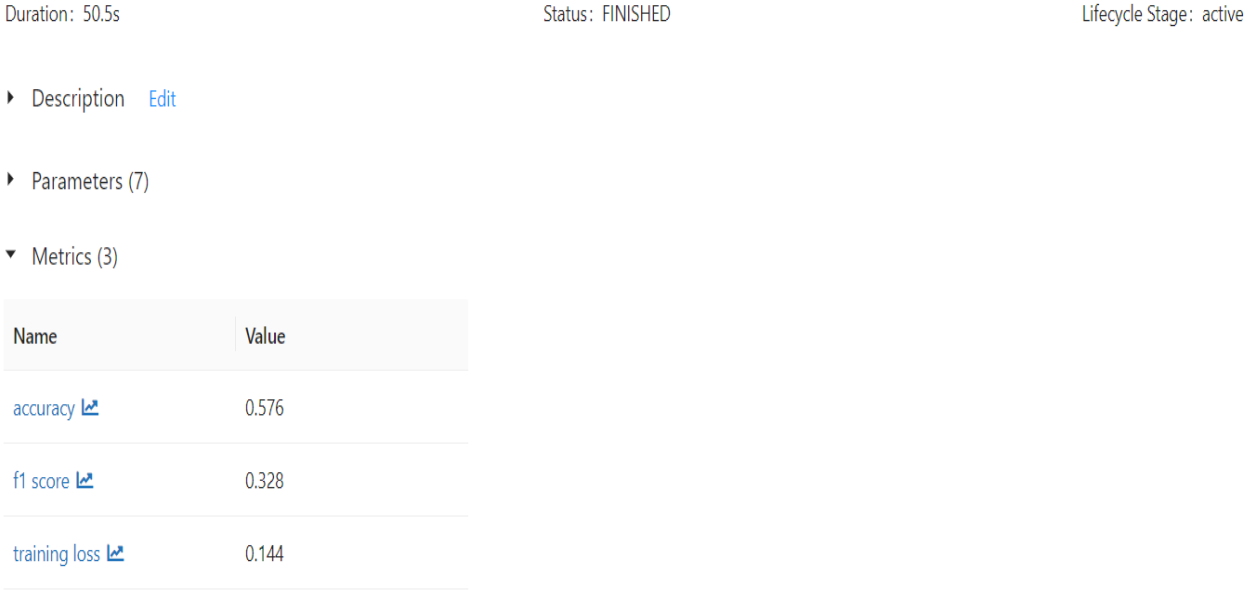
یکی از ضعف های مدل این است که نسبت به null در دیتاست مقاوم نیست و f1 score آن با زیاد شدن تعداد nun ها به شدت کاهش پیدا میکند.

شماره 5.1: توزیع f1 score نسبت به تعداد null های دیتا

همان طور که در عکس 5.1 مشخص است اگر تعداد none ها از 4 بیشتر شود دقت مدل به سرعت شروع به کاهش میکند.

قدرت مدل:

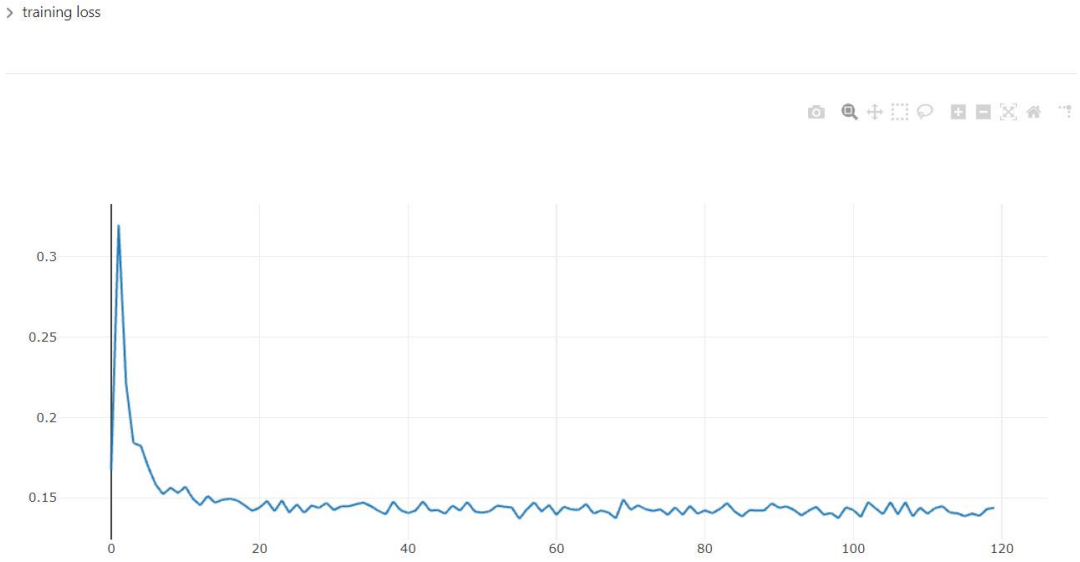
سرعت مدل بسیار بالا است سرعت مدل را میتوان در mlflow مشاهده کرد به طوری که کل فرایند training در 50.5 ثانیه به پایان میرسد.



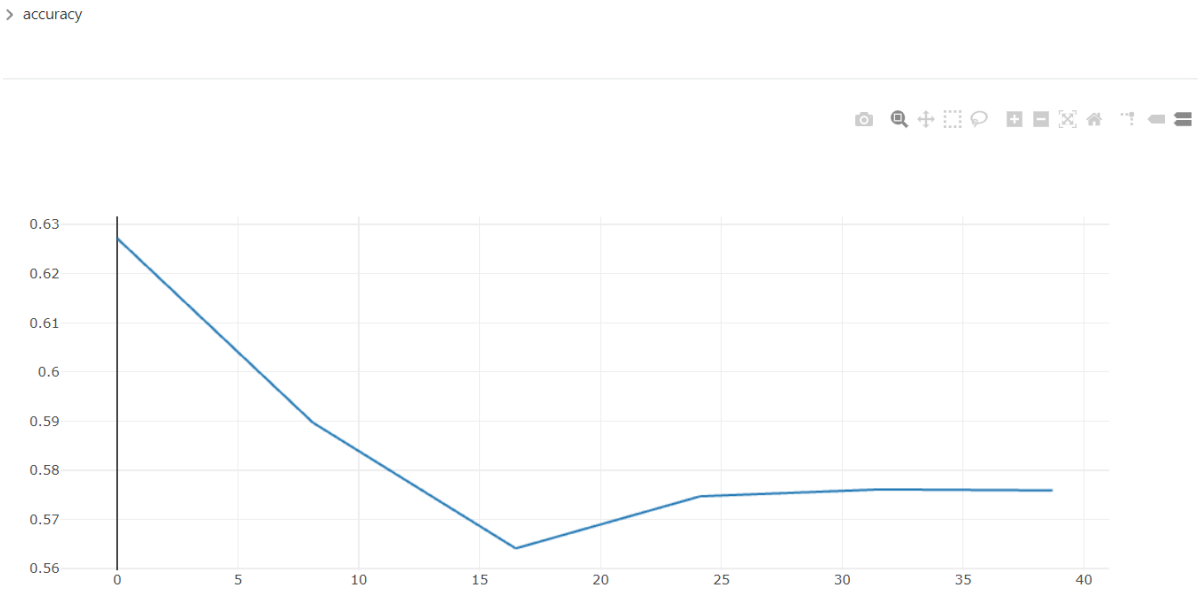
شکل 5.2: خروجی mlflow برای یک ران و زمان اجرای کوتاهش

5.2 بررسی پیشرفت مدل در هنگام training:

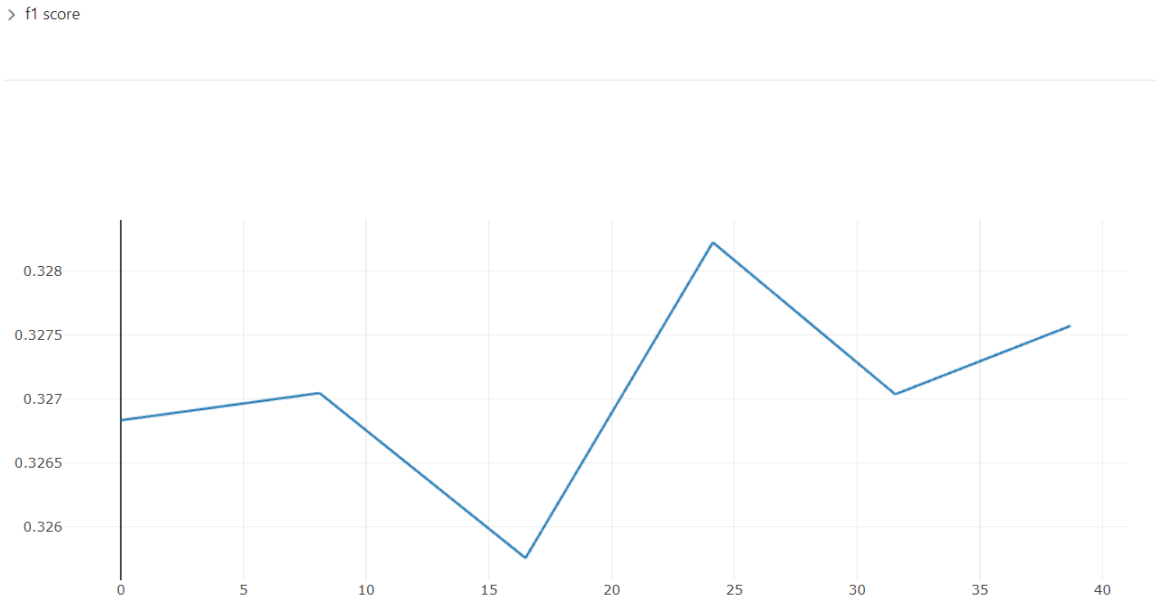
برای این بررسی از mlflow استفاده شده است و به وِیژه از آن استفاده شده است تا 3 پارامتر را لاگ کنیم accuracy , fıscore , بعد از هر ایپاک و loss در طول train این مقادیر برای مدل انتخاب شده به شرح زیر است. (در شکل‌های 5.5 و 5.4 و 5.3)



شکل 5.3: training loss



شکل 5.4: accuracy



شکل 5.5: f1 score

نتایج مدل روی test set:

مدل ResWideDeep را انتخاب میکنیم و با توجه به این که مدل انتخاب شده است، حال دقت آن را روی دیتای test (که تا به حال هیچ تاثیری در هیچ جایی نداشته اند) میسنجیم.
مدل روی این دیتا به نتایج نشان داده شده در شکل 5.6 میرسد.

manual f1 score= 0.3151012361049652

	precision	recall	f1-score	support
False	0.93	0.55	0.69	8641
True	0.20	0.73	0.32	1359
accuracy			0.57	10000
macro avg	0.56	0.64	0.50	10000
weighted avg	0.83	0.57	0.64	10000

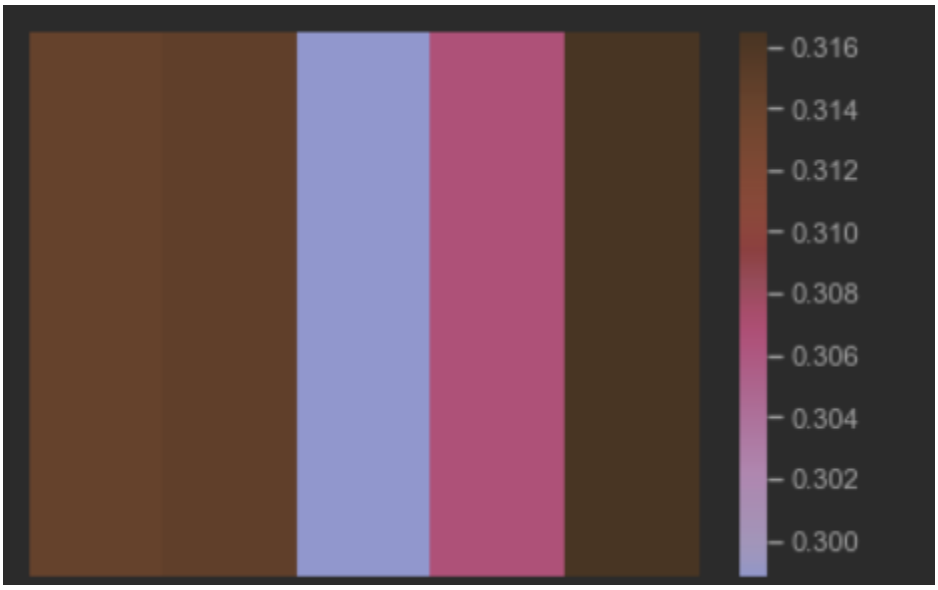
شکل5.6: نتایج مدل روی داده‌های test

5- DATA STORY TELLING:

روشی که مورد استفاده قرار گرفته است این است که یک ستون را مقادیرش را حالت پیشفرضش میگذاریم و بعد میبینیم f1 score چه مقدار کاهش پیدا میکند، این کار ابتدا برای ستون هایی با مقادیر عددی صورت میگیرد.

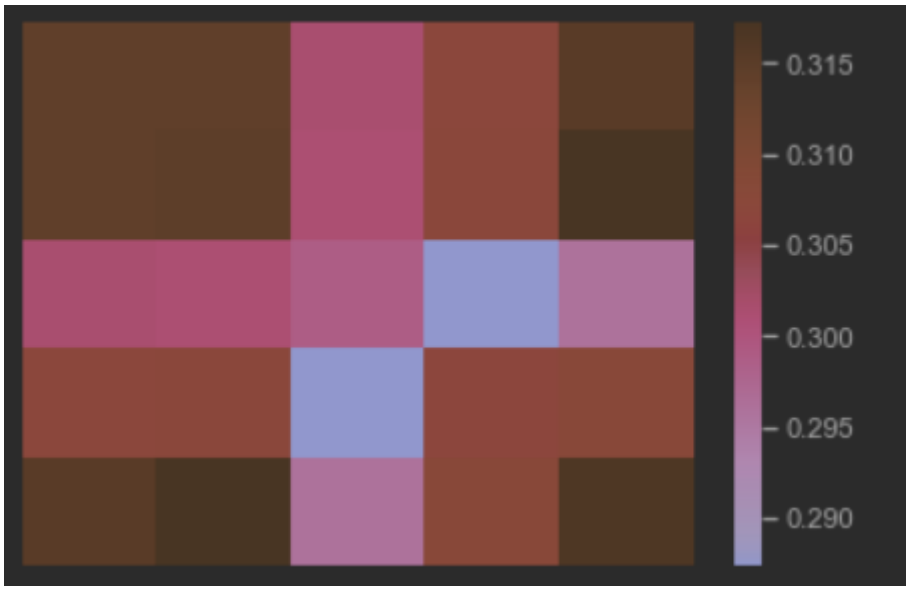
```
normal f1=0.315
remove nb_clicks_1week f1=0.314
remove product_age_group_percentage f1=0.315
remove partner_id_percentage f1=0.299
remove audience_id_percentage f1=0.307
remove brand_percentage f1=0.316
```

شکل 6.1: مقادیر f1 score متناظر با حذف هر یک از ویژگی ها



شکل 6.2: رنگبندی مقادیر نشان داده شده در شکل 6.1 (برای ایجاد شهود بیشتر)

همان طور که از تصویر 6.2 میبینیم ستون های audience_id , parent_id برای مدل بسیار مهم هستند، همچنین حدس زده میشود که مقادیر ستون ها به صورت دوتایی تاثیر گذار باشند.



شکل 6.3: مقادیر f1 score های حاصل با حذف audience_id , parent_id

همان طور که مشخص است همچنان parent_id , audience_id خیلی مهم هستند و وقتی هر دوی آن ها را حذف کنیم F1 به 0.29 میرسد.

6.1 تاثیر فیچرهای categorical:

در این بخش به بررسی تاثیر ستون های کتگوریکال میپردازیم، با توجه به این که این ستون ها یک مقدار none دارند (در واقع یکی از ستون هایشان نشان دهنده مقادیر null در دیتاست است) با قرار دادن مقادیر این وان هات به آن مقدار بررسی میکنیم که اگر مدل این فیچر را نداند f1 score چه مقدار افت میکند.

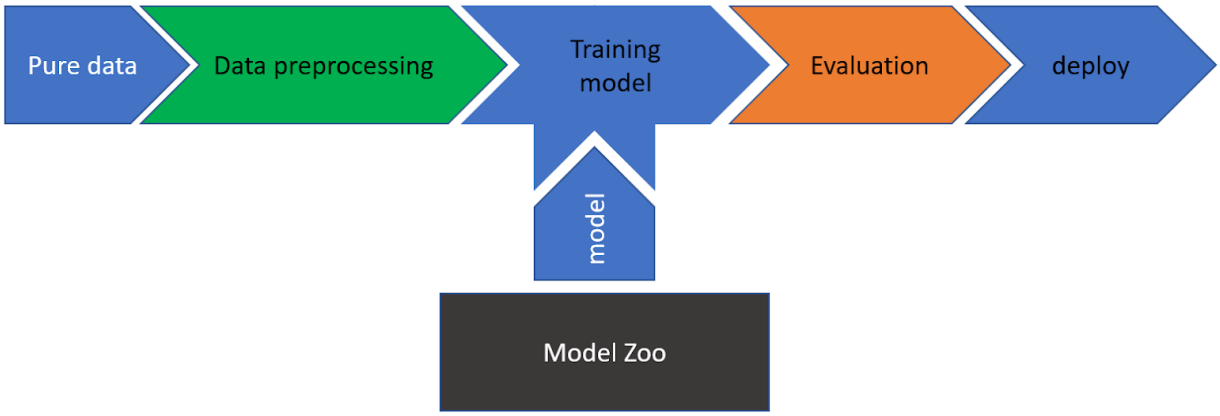
```
normal f1=0.315
remove device_type f1=0.289
remove gender f1=0.316
remove country f1=0.315
remove sub_category f1=0.318
remove period f1=0.312
```

شکل 6.4: مقادیر f1 score متناظر با حذف ویژگی‌های categorical

شکل 6.5: رنگبندی مقادیر نشان داده شده در شکل 6.4

همان طور که میبینیم ستون device_type از اهمیت بالایی برخوردار است و مدل بدون این ستون به مشکل میخورد.

6- DEPLOYMENT:



این pipeline برای این بخش در نظر گرفته شده است، یعنی در ابتدا pure data وارد میشود و سپس در بخش data preproccessing تمیز میشود و سپس با استفاده از یکی از مدل های model zoo عملیات training روی آن انجام میپذیرد. پس از اتمام training عملیات evaluation انجام میگیرد و سپس مدل deploy میشود.

چیزی که ارتباط میان این بخش های مختلف را برقرار میکند یک فایل config است، یعنی در بخش اول آدرش دیتا از فایل کانفیگ خوانده میشود و دیتای اصلی پس از شکستن به train, test, dev در فایل هایی با آدرس هایی که در فایل config وجود دارد ذخیره میشوند، در بخش data preprocessing هم فایل های train, test, dev با توجه به فایل config خوانده میشوندو در دیتای clean در فایل هایی که در config مشخص شده است ذخیره میشود. در بخش training model هم دیتای تمیز باز با توجه به فایل config خوانده میشود و اصلاعات مدل هم از همان فایل میاید، یعنی اسم مدل، epochs , batch size , learning rate و همهی این ها از همین فایل میاید، سپس مدل ذخیره میشود (این ذخیره سازی با استفاده از mlflow انجام میگیرید.)

سپس در بخش evaluation با توجه به اطلاعات mlflow.run که در بخش training بود، مدل را لود کرده و آن را evaluate میکند، سپس در بخش deploy هم همین اتفاق می افتد یعنی مدل دوباره لود میشود و آماده ی سرویس دهی میشود.

```
with mlflow.start_run() as run:
    config = {
        "epochs": 1,
        "lr": 0.001,
        "batch_size": 4000,
        "device": "cpu",
        "model_name": "res wide deep",
        "run info": run.info,

        'data address': "train_dataset.csv",
        'train pure address': "data/train_pure_dataset.pickle",
        'dev pure address': "data/dev_pure_dataset.pickle",
        'test pure address': "data/test_pure_dataset.pickle",
        'train clean address': "data/train_clean_dataset.pickle",
        'dev clean address': "data/dev_clean_dataset.pickle",
        'test clean address': "data/test_clean_dataset.pickle",
    }
```

شکل 7.1: نمونه ای از فایل config

7.1 نحوه‌ی سرو کردن مدل

برای این کار از یک mlflow.pyfunc..PythonModel استفاده شده است. در بخش load_contex که نیاز است اطلاعات اولیه لود شود ابتدا مدل ترین شده با توجه به شماره run_id لود میشود و سپس preproccessing یونیت لود میشود.

در بخش predict هم ابتدا دیتا تمیز میشود و سپس با استفاده از مدل prediction ها حساب میشود.

این کار برای این صورت گرفته است که mlflow توانایی serve کردن این گونه مدل ها را دارد.

این کد در pipeline_example.json وجود دارد.

البته در این پروژه با توجه به ارور هایی که گرفتیم روشی ساده تر برای serve کردن مدل انتخاب شده است، در این روش که با اجرای کد main.py میتوان مشاهده کرد، کد ادرش فایل pickle تست ست را میخواهد و سپس هم پیشبینی ها را خروجی میدهد و هم آن ها را در prediction.csv ذخیره میکند، فرمت فایل test باید مانند ترینینگ باشد با این تفاوت که ستون Sale میتواند هر مقداری داشته باشد و مدل به آن توجهی نمیکند.