مراحل انجام شده:

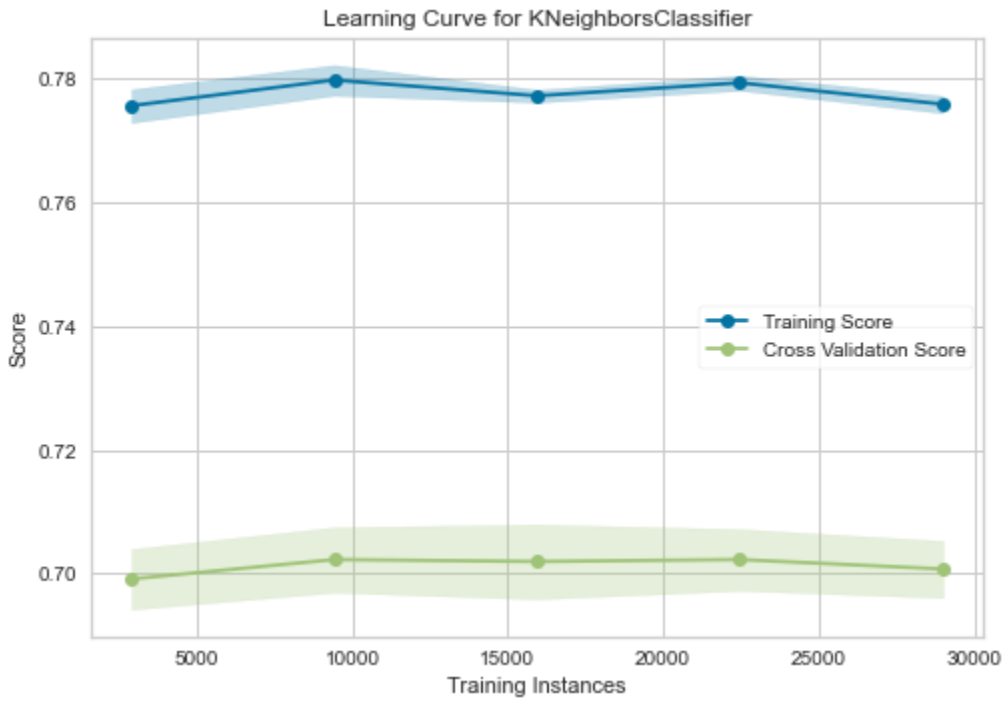
داده دارای ٤ ویژگی به نام های apps, games, queries و تاریخ تولد بود. در هیچ کدام از ویژگی های به جز birth_year داده NaN وجود نداشت.
ستون هدف هم gender به که آن را به صورت ۰ و ۱ تبدیل کردم.(۰ برای زن و ۱ برای مرد)
دردسر اصلی مدل داده، فیچر های دارای مقادیر لیستی بودند که در واقع همان سه ویژگی اصلی را تشکیل می دادند: بازی ها، برنامه ها و کوئری ها. برای حل این مشکل در ابتدا من از طریق تکه کد این ویژگی ها را به مجموعه ای از ویژگی های جدا تبدیل کردیم:

```
df = pd.concat([df.pop('queries').apply(pd.Series), df], axis=1)
```

بعد از این در یکسری از ستون ها های جدید تعداد زیادی NaN وجود داشت که با threshold هشتاد درصد برخی از این ستون ها را حذف کردم. در بقیه نیز NaN را با مقدار صفر جایگزین کردم.
همچنین سال تولد را با میانگین پر کردم.
این کار را برای دو ویژگی دیگر هم انجام دادم. سپس شروع به train کردن مدل های مختلف کردم. در زیر نتیجه را میتوان مشاهده کرد:

model	accuracy	precision	recall	F1	balanced_accuracy
Random Forest	0.74	0.74	1	0.85	0.5
Decision Tree	064	0.76	0.75	0.75	0.54
KNN	0.70	0.74	0.90	0.81	0.51

با توجه به اینکه عملا داده ها train نمی شدند، تصمیم به تغییر در داده های ورودی گرفتم. در زیر هم learning curve مربوط به مدل KNN را میبینیم که وضوح نشان دهنده train نشدن داده هاست:



در ایتدا ستون مربوط به تاریخ تولد را حذف کردم. اثری چندانی نداشت و در ادامه میزان threshold را هم تغییر دادم اما باز هم تغییری پیدا نکرد.

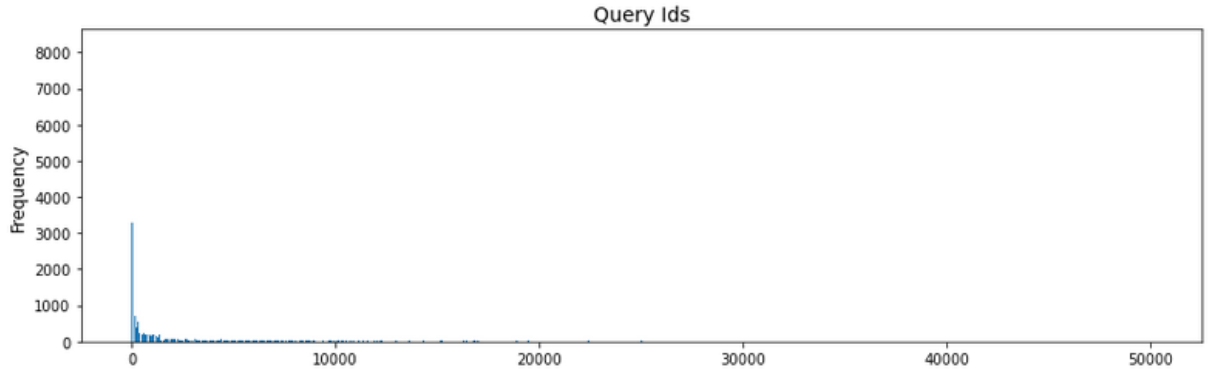
در ادامه کار ستون query را حذف کردم که نتیجه بدتر هم شد.

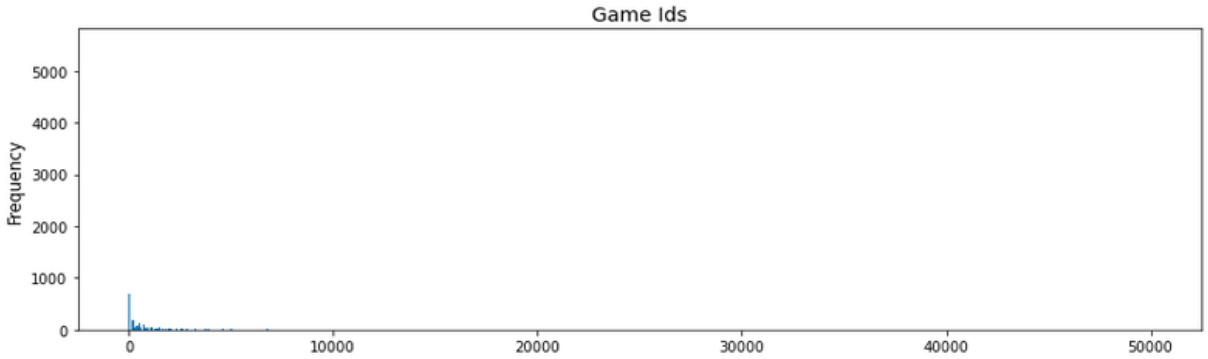
در ادامه با داده های قبلی اما بدون ویژگی تاریخ تولد کار را ادامه دادم. در اولین مرحله یک شبکه عصبی با استفاده از keras tuner به بهترین حالت برای داده train کردم اما باز هم به نتیجه خوبی نرسید. در این حالت دقت مدل چیزی حدود ٧٤ درصد بود. با این که سعی کردم درصد cross validation را جا به جا کنم این کار هم تاثیری نداشت.

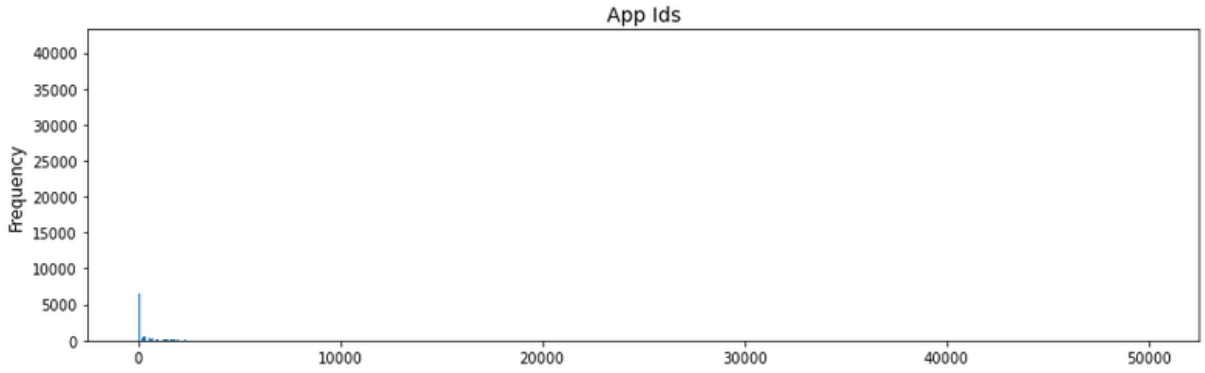
سپس با بررسی مدل روی داده تست متوجه شدم مدل دقیقا همه داده ها را مرد پیش بینی کرده است. البته این موضوع درباره Random Forest دقیق نبود و تفاوت هایی داشت در آن جا نسبت ٨٠٣٠/٢٥ را داشتیم. این در حالی بود که نسبت واقعی در داده تست برابر با ٥٩٤٧/٢٠٨١ بود. با توجه به اینکه مدل ها کاملا بر اساس hyperparameter ها tune شده بودند و از طرفی مدل های مختلفی بررسی شده بود پس به این نتیجه رسیدم که هنوز داده مشکل دارد.

برای حل مشکل خودم به صورت دستی داده های به صورت لیست را به ویژگی های جدید تبدیل کردم. در این جا با توجه به اینکه برخی از برنامه ها یا بازی ها به تعداد کمی در داده ها وجود داشتند، ستون هایی که در کمتر از ٥ درصد تعداد داده ورودی بودند را حذف کردم.

می توانید توزیع داده ها را از طریق بخش زیر مشاهده کنید:







در این حالت به ۲۱۹ ویژگی رسیدم. مدل را بر این اساس train کردم اما در اینجا نتایج بسیار بهتر شد. نتایج را در زیر میتوانید مشاهده کنید:

model	accuracy	precision	recall	F1	balanced_accuracy

Random Forest	0.81	0.81	0.97	0.88	0.66
Decision Tree	0.79	0.81	0.92	0.86	0.66
KNN	0.73	0.78	0.88	0.83	0.58
ANN	0.81	0.86	0.88	0.87	0.74
Logistic Regression	0.82	0.84	0.94	0.88	0.72
Gaussian NB	0.68	0.88	0.66	0.75	0.71

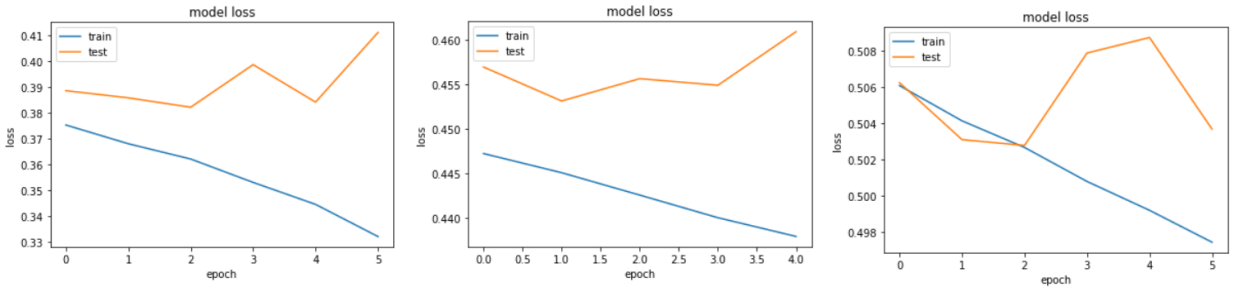
نسبت تعداد زن به مرد هم در ANN به نسبت ٦٠٨٦/١٩٦٩ رسید که خیلی نزدیک به نسبت واقعی بود.

مدل های تماما tune شده اند و از early stopping هم استفاده شده است. برای بررسی دلیل دقت نه چندان بالا تغییراتی در threshold انتخاب ویژگی ها کردم. به جای ٥ درصد از موارد ٣ درصد، ١٠ درصد و ٢٠ درصد هم استفاده کردم.

نتایج هر حالت را در زیر میبینید:

data	accuracy	precision	recall	F1	balanced_accuracy
3 percent threshold	0.81	0.78	0.40	0.53	0.68
10 percent threshold	0.78	0.73	0.26	0.39	0.61
20 percent threshold	0.74	0.57	0.53	0.20	0.29

بر این اساس به نظر بهترین حالت همان ٥ درصد بوده است. همچین در learning curve برای مدل های زیر را به ترتیب از چب به راست برای ٣ درصد، ١٠ درصد و ٢٠ درصد میبینیم.(از early stopping استفاده شده است)



البته برای ۲۰ با توجه به اینکه loss برای test در حال کاهش بود early stopping را برای بررسی بیشتر حذف کردم اما داده به طور کلی به روند سینوسی خود ادامه داد.

با توجه به توضیحات داده شده به نظر برای حل مشکل نیاز داده و همچنین ویژگی های بیشتر برای رسیدن به دقت بیشتر نیاز خواهد بود.