محمد کاظم رجبی

پروژه سوم:دیتای این پروژه شامل 2 قسمت ترین و تست بوده که بر اساس transition id باهم جوین میزنیم و نتیجه به صورت زیر است

Train dataset has 590540 rows and 434 columns.

Test dataset has 506691 rows and 433 columns.

همان طور که دیدیم دارای 434 تا ستون است که نسبتا زیاد است برای پردازش روی داده ها میتوان از pca یاتبدیل نوع دیتا تایپ ها برای کم کردن حجم ان استفاده کرد .ما این جا از روش های group by برای کاهش تعداد ستون هااستفاده میکنیم که این مرحله از پیش پردازش روی داده ها مطابق مار انجام شده در پیوست میباشد.

کار های انجام شده روی دیتااز روی منبع زیر است

https://www.kaggle.com/artgor/eda-and-models

There are 414 columns in train dataset with missing values.

همان طور که میبینیم دیتا بسیار ناقص است و باید ستون ها ترکیب شوند

train['TransactionAmt\_to\_mean\_card1'] = train['TransactionAmt'] /

 train.groupby(['card1'])['TransactionAmt'].transform('mean')

مشابه بالا روی ستون های مختلف از group by استفاده کرده که در نوت بوک کامل انجام شده است حالا تعداد ستون های دارای اطلاعات خالی رو با isnull().sum() تحلیل میکنیم.

[col for col in train.columns if train[col].isnull().sum() / train.shape[0] > 0.9]

در واقع در ستون هایی که تعداد عناصر nan ان ها نسبت به تعداد کل سطر ها بیشتر از 0.9 باشد را انتخاب میکنیم.

در مرحله بعد ستون هایی که مقادیر بزرگ دارند نیز مشکل ساز هستند و ان ها را نیز کنار میگذاریم

برای این کار از value.count(dropna=false) استفاده میکنیم که ارایه ای به ترتیب بر حسب بیشترین داده ای که تکرار شده به ما میدهد در واقع اولین عنصر این ارایه بیشترین تکرار در ستون را دارد شامل nan هم هست .اگر مقدار این عنصر بیشتر از 0.9 باشد نیز ان ر کنار میگذاریم.

پس ستون هایی که دارای مقادیر خالی زیاد یا مقادیر بزرگ دارند همراه با ستون is froud کنار میگذاریم که در مجموع تعدادی از ستون شامل این 2 دسته شده و از دیتا ست حذف میشوند.

در مرحله بعد به بررسی ستون های کتگوریکال میپردازیم که شامل

id\_12...id 38

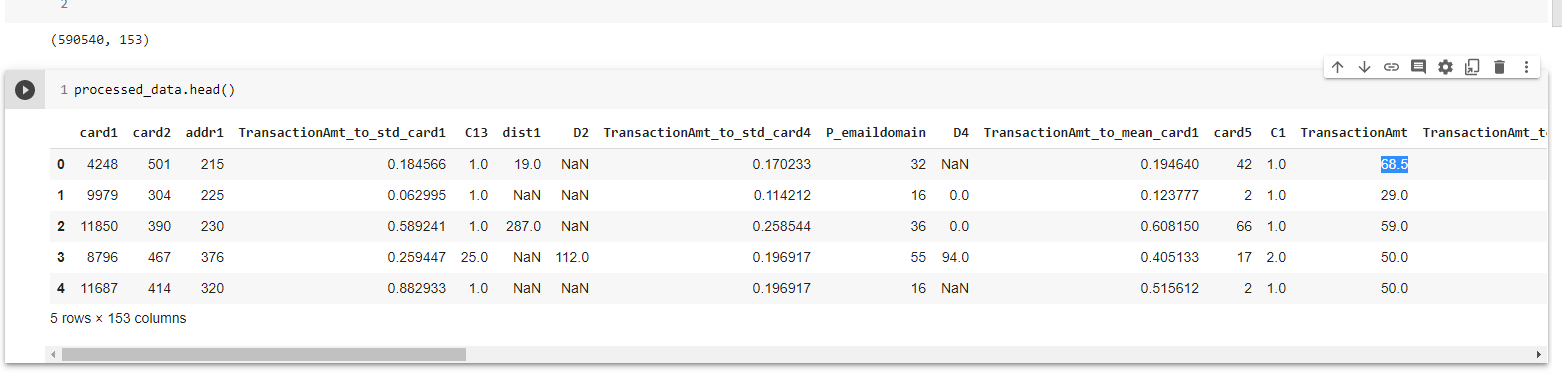
DeviceType, DeviceInfo, ProductCD, card4, card6, M4, P\_emaildomain,

R\_emaildomain, card1, card2, card3 card5, addr1, addr2, M1, M2, M3, M5, M6, M7, M8, M9,

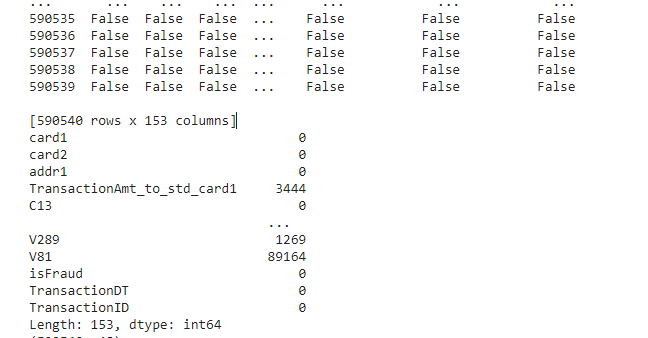
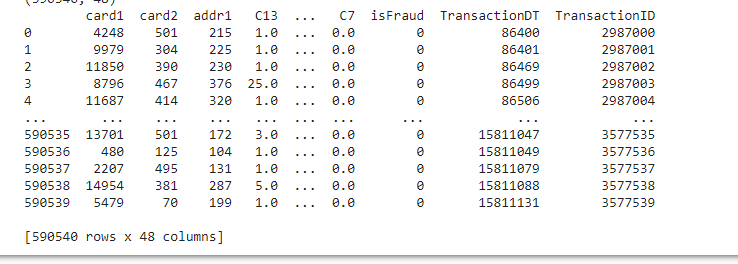
P\_emaildomain\_1, P\_emaildomain\_2, P\_emaildomain\_3, R\_emaildomain\_1, R\_emaildomain\_2, R\_emaildomain\_3

است و باید در واقع داده های کتگوریکال را به داده های لیبل یا one hot encoder تبدیل کنیم که در کد از lable استفاده شده برای این دسته از داده ها.

بعد از اعمال group by و کار های بالا دیتا ست نهایی دارای 158 ستون است

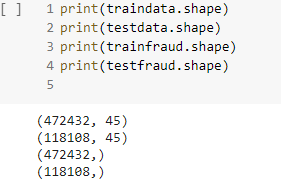
 در این مرحله ستون هایی که مقدار nan دارد رت بر حسب تعداد مشخص میکنیم همان طور که میبینیم ستون مثلا ستون trasactionAmt\_to\_std\_card1 دارای 3444 تا مقدار nan است

در این مرحله برای سادگی کار و کم شدن حجم دیتا این ستون ها را کنار میگذاریم و فقط ستون هایی رو در نظر میگیریم که isnull().sum()=0 باشد در واقع مثلا v289 وv81 ... کنار گذاشته میشوند

 پس از این کار 48 ستون باقی میماند و ان را سیو میکنیم در واقع با حذف ستون هایی ناقص حجم دیتا را کمتر کرده و مشکل حافظه ای که در ابتدا وجود داشت که نیاز بود با تبدیل نوع داده حجم کاهش یابد دیگر نیازی نیست .

بخش چردازش داده ها این جا تمتم شد و حالا به بخش fraud detection میپردازیم.

روی داد ها ستون های transition id وtransaction id را کنار میگذاریم به در اموزش شبکه مهم نیستند و صرفا ایدی هستند و مجموعه تست اموزشی و تست را از ان جدا میکنیم و از ستون isfraud به عنوان لیبل استفاده کرده که در مجموعه اموزش وجود دارد و ان را به مجموعه تست و اموزش به نسبت 80 به 20 تقسیم میکنیم.

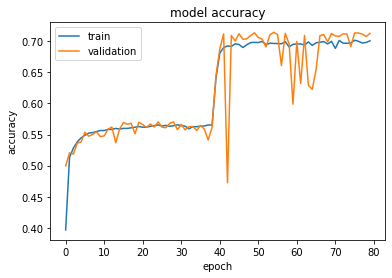
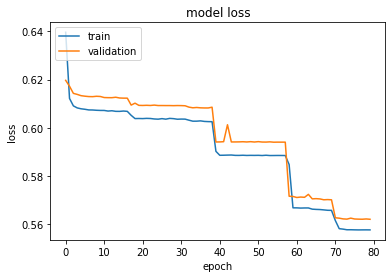


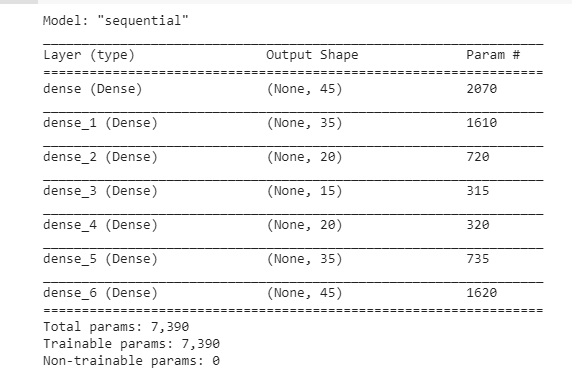
حالا باید داده ها را همسان سازی کنیم در واقع feature scaling انجام دهیم تا داده ها در یک رنج قرار گرفته و شبکه بهتر اموزش داده شود و با استفاده از standardscaler از پکیج sklearn.preprocessing این کار را انجام میدهیم و بعد از ان validation data را از دیتا ی اموزشی جدا کرده تا چک کنیم هنگام اموزش شبکه over fit رخ ندهد.

بعد از این کار ما از شبکه های اتواینکودر استفاده میکنیم با ساختار های مختلف :

در واقع لایه پنهان(latent space ) معمولا ابعاد کمتری دارد تا ویژگی ها ی خاص و مهم را بعد از یادگیری در خود نگه دارد میتواند فضای بین لایه ورودی تا لایه پنهان و بعد از ان تا لایه خروجی متقارن باشد یا لایه قبا از لایه پنهان تعداد بیشتری داشته باشد ی لایه از لایه پنهان تا خروجی تعداد لایه های بیشتری داشته باشد به همین منظور از 7 شبکه با ساختار های مختلف برای این کار استفاده میکنیم و با دیتا اموزش میدهیم و بهترین ان را برای fraud detection انتخاب میکنیم.

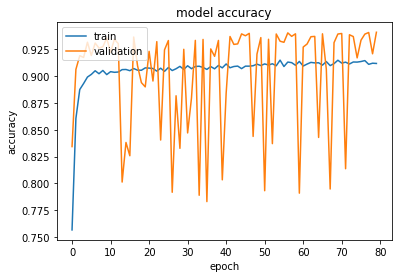
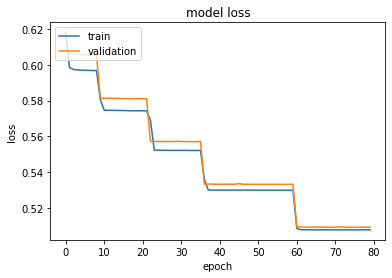
مدل اول :مدلی متقارن است با اندازه 15 نورون در لایه پنهان که دقت 69 درصد را بعد از 80 تا ایپاک به دست امده است.

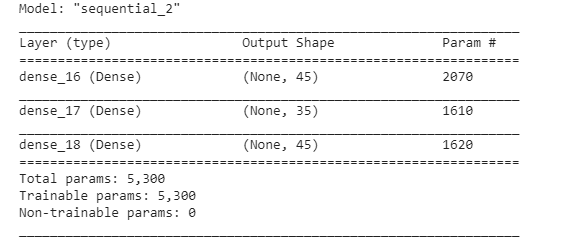


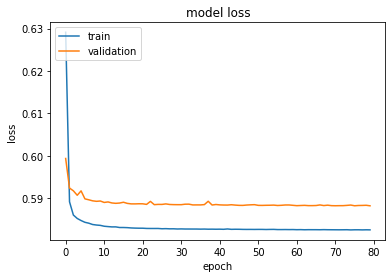


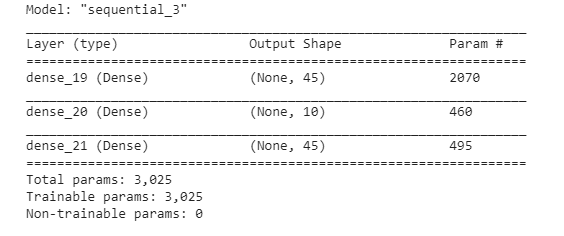
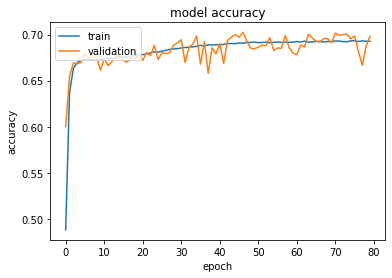
مدل دوم:یه اتو اینکودر متقارن است با این تفاوت نسبت به مدل اول که تعداد لایه های بیشتر دارد تا رسیدن به لایه پنهان وجود دارد و لایه پنهان فشرده تر در نظر گرفتیم با 5 نورون که بهتر شد وبه دقت 76 درصد رسید.

مدل سوم:مدل سوم ساده ترین حالت ممکن با 1 لایه پنهان با تعداد نورون 35 تا است و فقط 1 لایه وجود دارد بین ورودی و خروجی که همان لایه است که به دقت درصد 90 رسید.

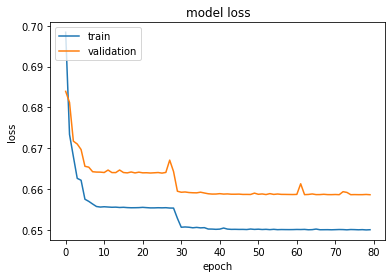


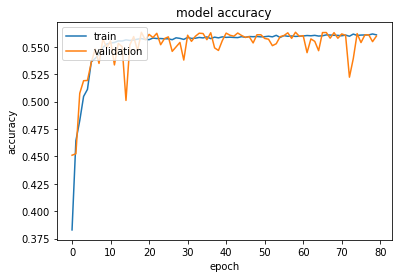


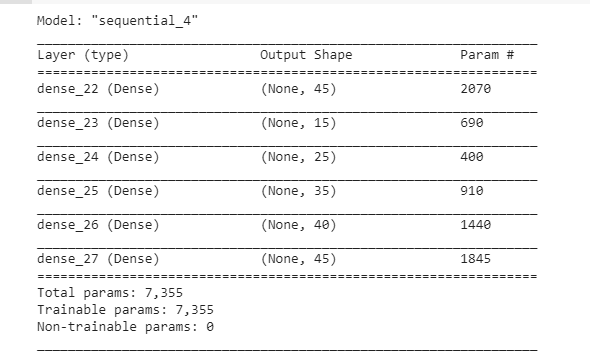
مدل چهارم:این مدل مانندمدل 3وم متقارن و فقط 1 لایه دارد و تفاوت ان در این است که تعداد نوروتن های ان در لایه پنهان 10 تاست و در واقع با کم کردن تعداد نورون ها میخواستیم ببینیم که فشرده سازی بیشتر بهتر است یا نه که به درصد دقت 69 رسیدیم.

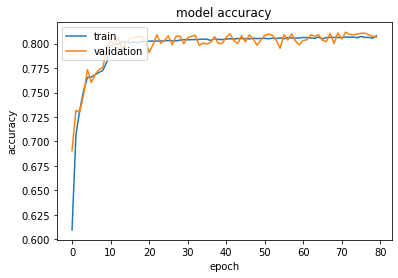


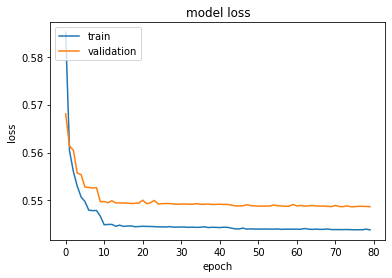
مدل پنجم:این مدل نامتقارن است بعد از ورودی لایه latent space اینکودر قرار دارد بعد ان 3 لایه قرار دادیم تا به خروجی برسیم در این مدل به دقت 65 درصد رسیدیم.

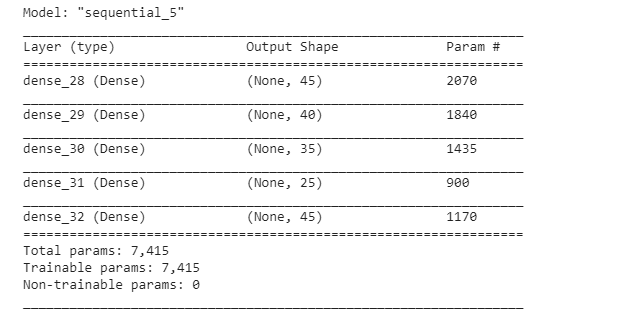




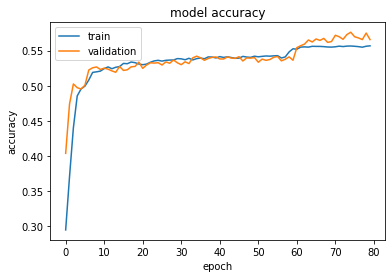
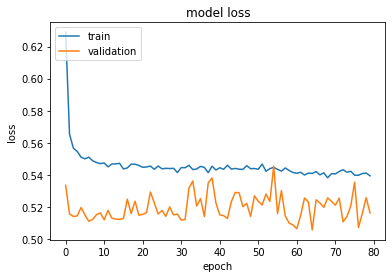


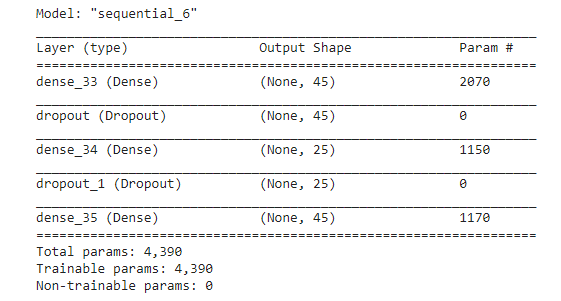
مدل ششم:این مدل نیز نامتقارن است به طوری که تعداد لایه های قبل از latent space ما بیشتر است و در واقع 3 لایه قرار دادیم تا به دقت 54 درصد رسیدیم





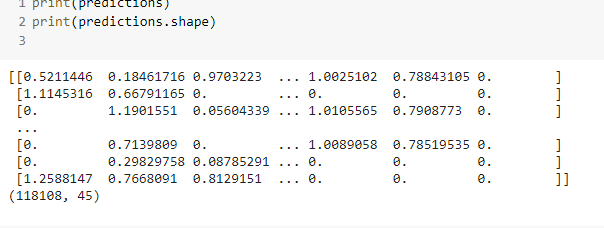
مدل هفتم:این مدل مدلی متقارن که latent space ان یا لایه کد ان 25 نورون داشته و از دراپ اوت در ان استفاده شده که به دقت 53 درصد رسیدیم





بهترین مدل مدل 3 است و کار را روی ان ادامه میدهیم. در همه ی مدل ها تست شده تعداد ایپاک ها برابر 80 و bath-size=256 استفاده شده و همه تابع عای فعالیت relu هستند و از loss=mean\_squar\_error استقاده شده است.

با مدل سوم روی داده های تست پردیکت انجام میدهیم

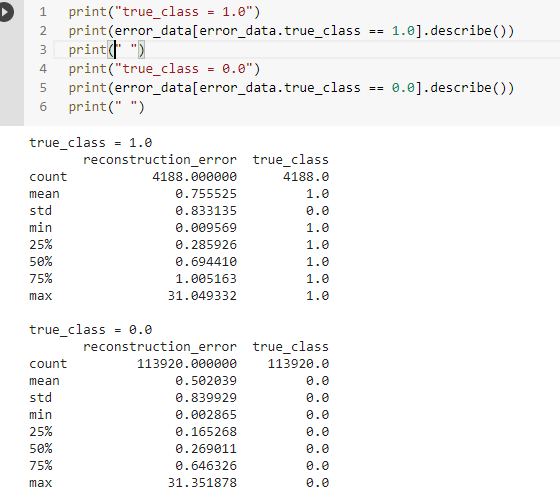
****

تستب برای ادامه کار خطایی را تعریف میکنیم بین مقدار واقعی و مقدار پردیکت

Mean-error=np.power(testdata-predictrion,2)

یک دیتا فریم از خطای بالا و مقدار is fraud که از دیتا اولیه جدا شده بود ایجاد میکنیم و نام ستون ها را reconstruction-error و true-class قرار مدهیم

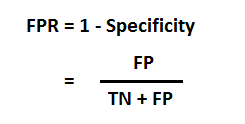
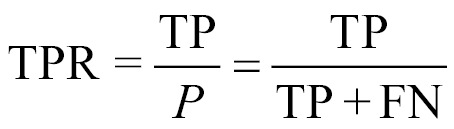
بر ساس frad به 2 قسمت دیتا فریم را تقسیم میکنیم



برای مثایل کلسیفیکیشن و قدرت ان ها در تشخیص که این جا 0و1 به معنا isfraud در نظر میگیریم از نمودار های roc و aoc استفاده میکنیم

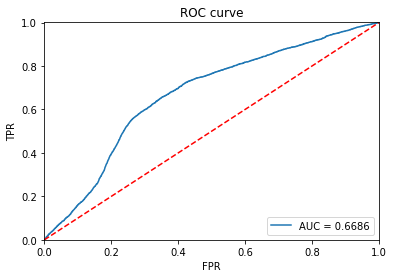


به طوری که

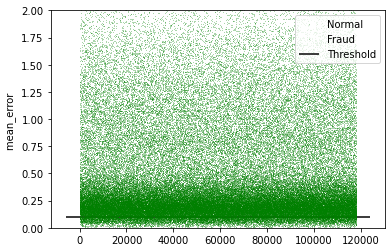


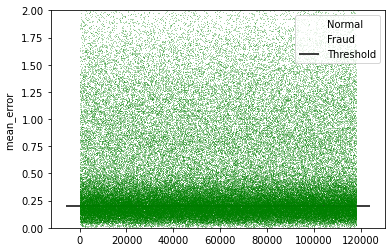
به این معنا که هرچه aoc عدد بیشتری باشد احتمال کلسیفای کردن بیشتر است و اگر 50 % باشد عملا قدرتی ندارن و مانند پرتاپ سکه عمل میکند.

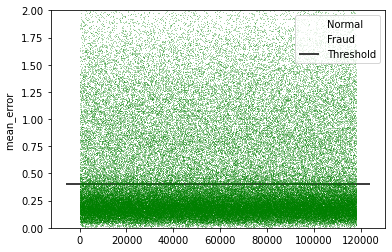
به صورت دیفالت treshhod که توضیع احتمال کلسیفیکیشن است در ابتدا و به صورت پیشفرض 0.5 است و در نودار زمانی بهترین نتیجه را میگیریم که به نقطه (0و1) نزدسک تر باشیم یا به عبارتی ماکسیمم باشدو در نقطه ای از treshhold که نمودار ماکسیمم است بهترین قابلیت کلسیفیکیشن را داریم

با توجه به داده ها و فریمی که ساختیم و 2 قسمت شد   


همان طور که میبینیم auc برابر 66 درصد است و بهترین threshold در نزدیکی 0.4 اتفاق میافتد که با تست کردن مقادیر مختلف به این نتیجه رسیده و در نمودار هم مشخص است که نزدیک به 0.4 ماکسیمم است







و برای threshold=0.4 نتیجه به دست امد

