گزارش تمرین پنجم

یزدان زندیه وکیلی

610397193

**پکیج های استفاده شده**

قسمت ابتدایی کد شامل اضافه کردن پکیج های مورد نیاز است که علاوه بر پپکیج های روزمره کار با داده (Numpy, Pandas, Matplotlib, Seaborn) برای قسمت های پیش پردازش داده و مدل هایی نظیر Logistic Regression از کتاب خانه Scikit-Learn استفاده شده است که همانطور که مشاهده می کنید جدا شده اند تا درست به نظر برسند.

**بارگزاری دیتا**

همانطور که در ابتدا مشاهده می فرمایید، داده ها با استفاده از Pandas بارگزاری شده اند و در ادامه مراحل پیش پردازش داده به ترتیب بر روی آن ها اعمال شده است.

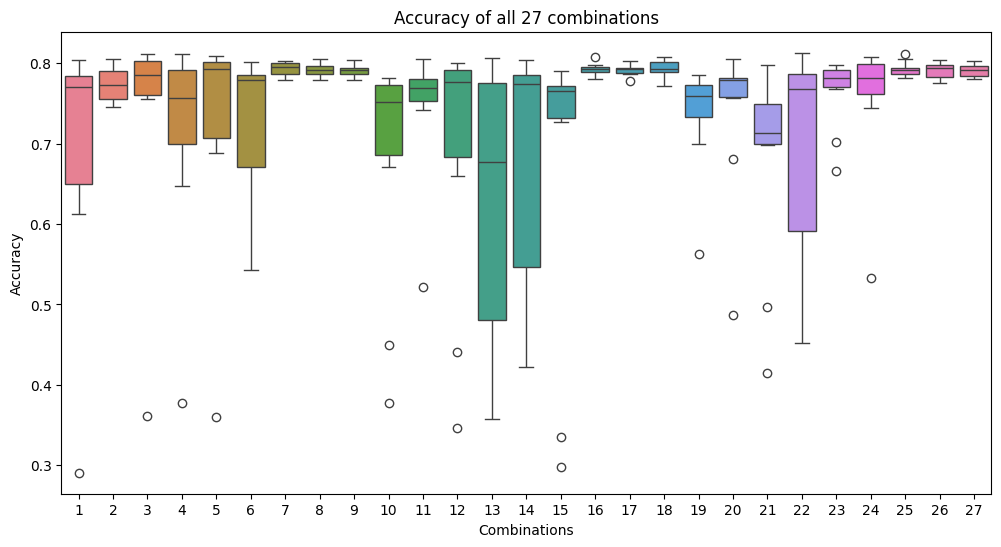
**سوالات 1 و 2**

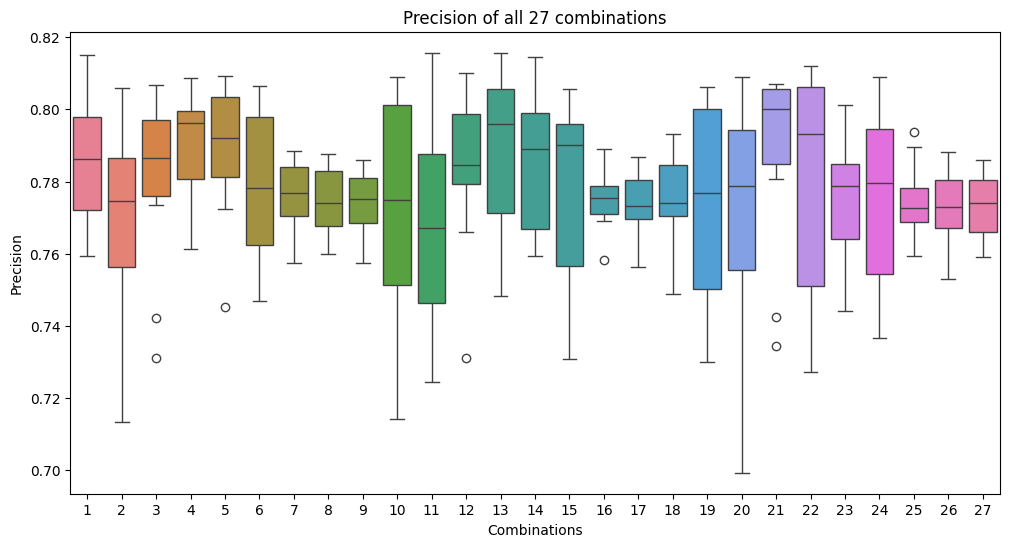
**پیش پردازش داده**

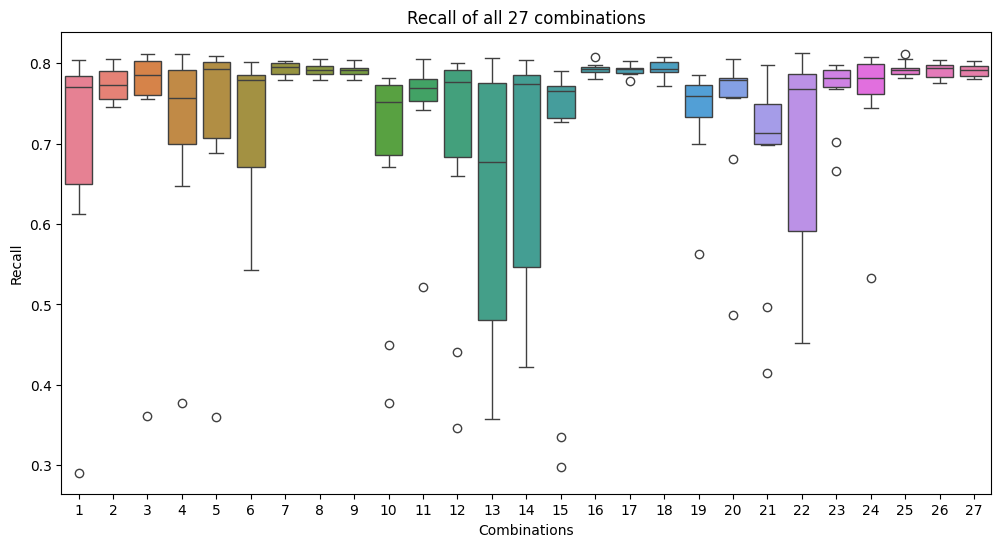
1. مقادیر گم شده : با استفاده از KNN مقادیری که علامت ؟ داشتند را پر کردم و مقادیرشان را بر حسب 3 تا از نزدیک ترین رکورد ها پیش بینی کردم.
2. ویژگی های اسمی : این ویژگی ها را که با احتساب برچسب ها 9 دسته می شدند با تکنیک Label Encoding به مقادیر عددی تبدیل نمودم.
3. نرمال سازی داده ها : با استفاده از نرمال سازی استاندارد و بردن داده ها به فضایی با میانگین صفر و انحراف معیار 1 ، آن ها را نرمال کردم. (البته که داده های عددی تنها نرمال شدند در ابتدا و داده های اسمی فقط یک بار انکود شدند و طبیعتا نرمال سازی ای هم روی آن ها اعمال نشد.)

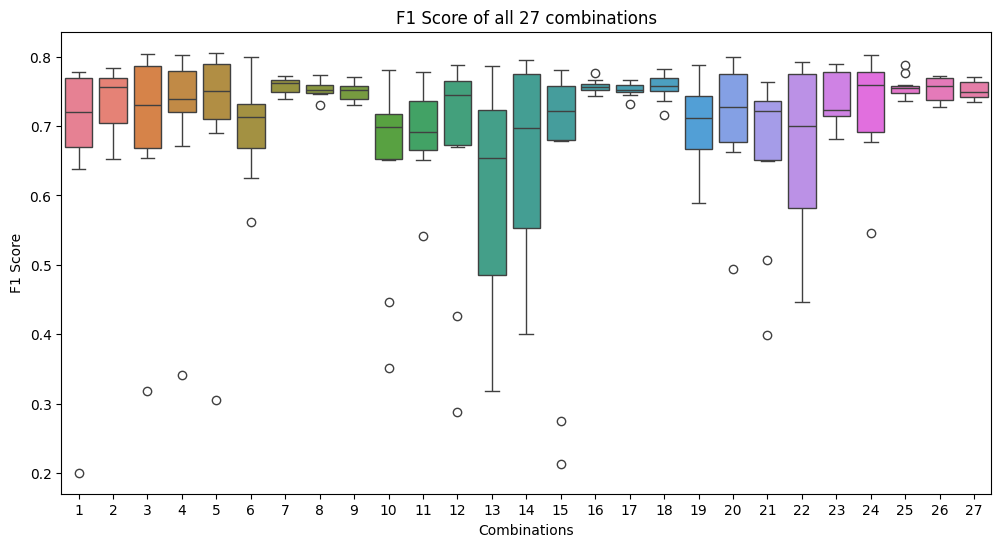
**آموزش Logistic Regressor**

در این قسمت با تعریف بازه تغیرات hyper parameters به 27 عدد مدل متفاوت رسیدیم که هر کدام 10 بار با استفاده از Repeated K Folds با 5 تکه کردن داده ها و دو بار تکرار آموزش داده شدند و در انتها باکس پلات ای برای هر کدام از 27 مدل بوججود آمده رسم نمودم و هر 4 معیار Accuracy, Precision, Recall, F1 Score را نمایش دادم.









Hyper Parameter های بهترین مدل بدین گونه است:

1. نرخ منظم سازی کاهش وزن = 0
2. نرخ یادگیری ثابت =
3. روش نرخ یادگیری = ثابت
4. حداکثر تکرار = 500

**رسیدن به مدل بهتر**

با توجه به اینکه بقیه تنظیمات با Penalty = None غیر فعال شده بودند و بر روند آموزش تاثیری نگذاشتند، صد در صد می شد با وارد کار کردن آن ها به مدل یا مدل های بهتری هم رسید و به نظر من باید Hyper Parameter های بیشتری را وارد کار می کردیم و یا برای دقت بهتر مدل گزینه ها را بر روی همین 4 پارامتر بیشتر می کردیم. به طور مثال اگر نرخ یادگیری را به حالت پویا در می آوردیم به نظر من به احتمال خیلی زیاد به نتایج بهتری می رسیدیم و یا اگر Regularization ها را غیر فعال نمی کردیم.

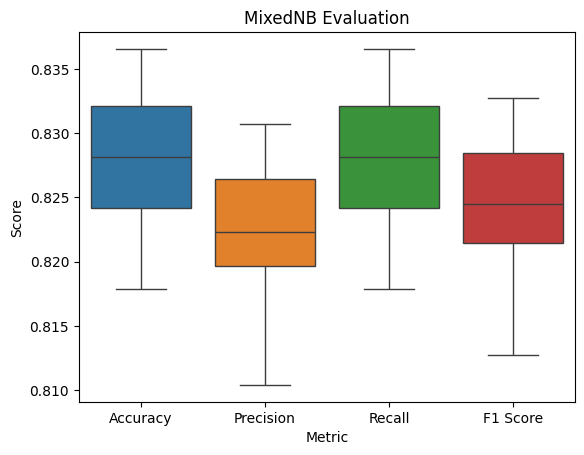
**ذخیر نتایج پیش بینی مدل**

در نهایت با نام خواسته شده در فولدر Dataset در کنار دیتاست اولیه، نتایج با نام خواسته شده prediction\_lr.csv ذخیره گردید.

**سوال 3**

**ساخت مدل جدید**

با پیاده سازی یک کلاس به نام NewBayesModel و استفاده از GaussianNB و CategoricalNB مدلی را پیاده سازی کردم که بر روی داده های عددی توزیع گوسی و بر روی داده های اسمی توزیع اسمی را اعمال می کرد. داده هایی که به مدل داده شده اند همان داده های سوال قبلی هستند و به همین دلیل نیازی به پیش پردازش دوباره نداشتیم. با استفاده از تکنیک Repeated K Folds و با 5 تکه کردن داده و دو بار تکرار در نهایت 10 مدل آموزش داده شده داشتم و باکس پلات مربوط به مدل جدید را در هر چهار معیار ارزیابی رسم نمودم که به شکل زیر است:



**مقایسه**

در مقایسه با مدل Logistic Regressor در سوال قبلی، به نتایج بهتری دست پیدا کردم و حدود 5 درصد پیشرفت در هر 4 معیار اندازه گیری شده داشتم که همانطور که در باکس پلات ها مشاهده می کنید به وضوح قابل مشاهده است.

**ذخیره سازی پیش بینی**

در نهایت همانطور که در توضیح پروژه خواسته شده بود، بهترین مدل را (که مدل در ایندکس 7 ام بود) گرفتم و برچسب داده های تست را پیش بینی کردم و در نهایت در فولدر Dataset با نام خواسته شده Prediction\_b.csv ذخیره کردم.

**سوال 4**

**ضعف ها و راه حل مربوطه**

1. **فرض مستقل بودن داده ها**: در مدل بیز فرض می کنیم که داده ها مستقل هستند که در واقع اینطور نیست و برای بهبود این قضیه با استفاده از تکنیک های feature engineering می توان داده ها را مستقل تر کرد.
2. **عملکرد ضعیف روی داده های نامربوط**: در واقع چون این مدل با همه ویژگی ها به یک شکل رفتار می کند، اگر داده های نامرتبط داشته باشیم عملکرد به شدت کاهش میابد و برای بهبود این قضیه باید از تکنیک های Feature Selection استفاده کرد.
3. احتمال صفر: اگر نمونه داده ای در داده های تست وجو داشته باشد که در داده های آموزشی موجود نباشد، این مدل احتمال آن را صفر فرض می کند در حالی که نیست و راه حل آن کاهش ابعاد یا استفاده از چند الگوریتم به طور همزمان است.
4. **عملکرد بد روی داده های با ابعاد بالا**: در این مواقع مدل به دلیل ضرب های زیاد عملکرد خوبی پیدا نمی کند که برای حل کردن آن نیاز به استفاده از تکنیک های کاهش ابعاد داریم.
5. **مواجه با داده های پیوسته**: این مدل همواره فرض می کند که داده ها از توزیع هایی مانند گوسی پیروی می کنند که ممکن است همیشه درست نباشد و برای مواجه با این مشکل می توان از Kernel Density Estimation ها استفاده کرد تا این تاثیرات کمتر شوند.
6. **عملکرد بد روی داده های کم**: اگر مقدار داده های ما کم باشد این مدل بر روی داده ها Bias می شود و نتایج خوبی نمی دهد و برای حل این مشکل نیاز به جمع آوری داده های بیشتر یا استفاده از تکنیک های Data Augmentation داریم.