

1-1-1 پایش سیگنال قلب (ECG)

لیست تسک‌های تعریف و اجرا شده در این قسمت:

۱-۱-۱-۱ آشنایی با سیگنال قلب (ECG)

مطالعه درباره سیگنال Electrocardiogram و آشنایی با این سیگنال حیاتی، ویژگی‌های آن و شکل ظاهری و ریختی آن.

۱-۱-۱-۲ آشنایی نحوه ثبت کردن سیگنال قلب و فرمت‌های ذخیره‌سازی آن:

بررسی دستگاه‌ها و محل‌های قابل ثبت سیگنال به صورتی که کیفیت سیگنال بهتر باشد.
(برای مثال حین رانندگی از بین ساعت هوشمند و روی فرمان، دنده و روی کمربند ایمنی بهترین نوع آن روی کمربند است).

و بررسی فرمت‌های مختلف ذخیره سازی این سیگنال به نحوی که برای ما کاربرد داشته باشد.

۱-۱-۱-۳ کاربردهای سیگنال ECG:

چون سیگنال ECG یک سیگنال زیستی منحصربه‌فرد است پس شکل ظاهری آن برای هر فرد مانند اثر انگشت یکتا است پس می‌توان کاربردهای زیر را برای آن در نظر گرفت.
انتخاب کاربردهای "تشخیص هویت راننده"، "تشخیص خواب‌آلودگی راننده" و "نرمال بودن یا نبودن وضعیت قلب" با استفاده از ضربان قلب و پردازش سیگنال ECG

۴-۱-۱-۱ مطالعه مقالات و منابع مرتبط با پروژه و انتخاب منابع مفید و قابل بازتکرار به عنوان

نمونه اولیه پردازش سیگنال ECG

تعدادی از مقالات بررسی شده در این زمینه را در بخش مراجع آخر فصل می‌آوریم.

۵-۱-۱-۱ تست کردن یک نمونه کد همراه با مقاله با استفاده از دیتابیس (MIT_BIH)

با رویکرد بررسی مقاله

DEVELOPMENT OF MACHINE LEARNING MODELS TO DETECT ARRHYTHMIA
BASED ON ECG DATA – INTERPRETABILITY []

و بازتولید و بهبود نتایج بدست آمده توسط آن این بخش را آغاز می‌کنیم.

۱-۱-۱-۵-۱ دیتابیس MIT_BIH

دیتابیس MIT_BIH^۱ یا همان MIT_BIH Arrhythmia database یک دیتابیس از سیگنال‌های

ECG است که به صورت اپن سورس در اختیار افراد که نیاز به دیتای سیگنال ECG دارند قرار

دارد این دیتابیس شامل ۴۸ قطعه دیتا به طول زمانی نیم‌ساعت و از دو کانال که از ۴۷ مورد دیتاگیری

شده، دیتا با فرکانس سمپلینگ ۳۶۰ سمپل در هر ثانیه در هر کانال دیجیتالی شده است. و رزولوشن

۱۱ بیتی در رنج ولتاژی ۱۰ میلی‌ولت دارد. این دیتا توسط ۲ یا چند متخصص قلب لیبل‌گذاری شده

که این لیبل‌ها نیز در فایل‌های این دیتاست موجود است (اختلافی بین لیبل‌گذاری متخصصان وجود

ندارد).

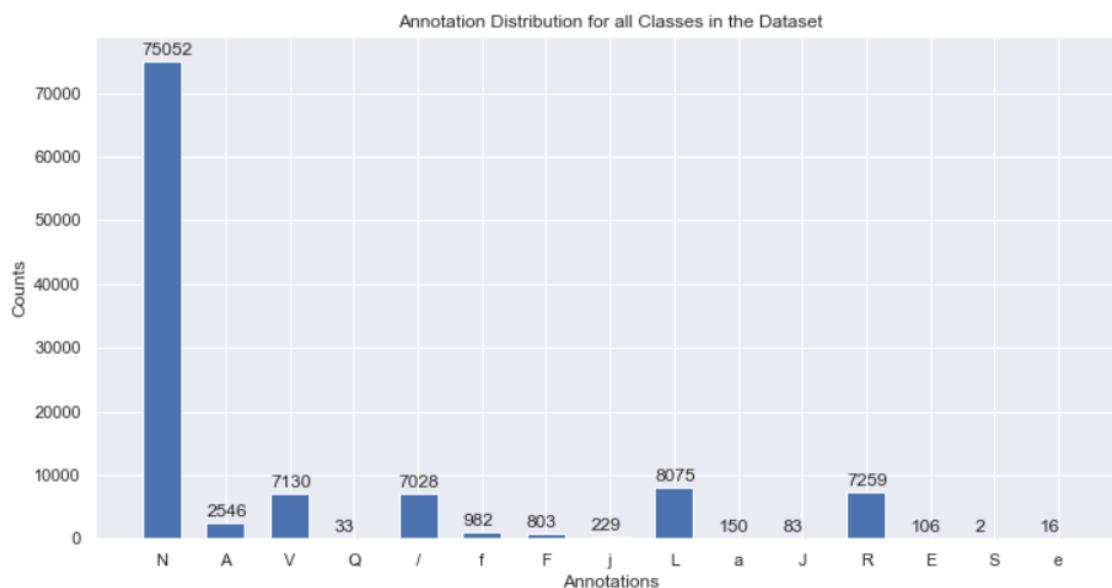
دیتا را از دیتاست MIT_BIH دانلود می کنیم سپس پردازش آن را شروع میکنیم دیتای ما دارای یادداشتهایی است که لیبل های سیگنال ما را تشکیل می دهند، با اسفاده از این یادداشتهای می توان لیبل های دیتا را به ۸ کلاس تقسیم کرد که در شکل زیر کلاس ها و توضیحات هر کدام را می بینیم.

Class	ID	Beat Description
N	1	Normal
L	2	Left Bundle Branch Block
R	3	Right Bundle Branch Block
V	4	Premature Ventricular Contraction
A	5	Atrial Premature
F	6	Fusion of Ventricular and Normal
f	7	Fusion of Paced and Normal
/	8	Paced

۱-۱-۵-۲:۱ لیبل ها و توضیحات هر کدام

کلاس N(Noraml) به وضعیت عادی و بقیه کلاس ها هر کدام به یک ناهنجاری در شکل سیگنال قلب اشاره دارند.

حال برای اطلاع از وضعیت از تعداد سیگنال های دارای ناهنجاری تعداد هر کدام از اعضای این کلاس ها را بدست آورده تا بهتر متوجه وضعیت دیتای خود شویم.



۲-۵-۱-۱-۱ تعداد اعضای هر کلاس

همانطور که مشاهده می‌کنید بیشتر دیتای ما در کلاس نرمال قرار می‌گیرد و تعداد کمتری از داده‌ها در کلاس‌های دیگر هستند که یک نوع عدم توازن بین داده‌ها است.

برای استفاده راحت‌تر از دیتا کلاس‌های خود را به جای حروف به اعداد تغییر می‌دهیم و مانند

عددگذاری در شکل ۲-۵-۱-۱-۱ اعداد ۱ تا ۸ را به کلاس‌ها نسبت می‌دهیم.

در این مرحله نیاز داریم تا دیتاست خود را برای آموزش شبکه و ساختن مدل برای یادگیری آماده

کنیم برای اینکار ۲ راه داریم راه اول این است که مجموعه دیتا را به صورت دیتای بزرگ ضربان

قلب و مستقل از بیماران در نظر بگیریم و از آن استفاده کنیم در این صورت احتمال شناسایی انواع

ناهنجاری در سیگنال ECG بالا می‌رود راه دیگر این است که بیماران را به صورت جداگانه به

صورت دیتا در نظر بگیریم در این صورت امکان دارد بعضی از ناهنجاری‌ها در روند آموزش شبکه

نباشند و در داده‌های تست قرار بگیرند که در نتیجه شبکه ما نمی‌تواند آن‌ها را به درستی تشخیص

بدهد. دقت کنید در هر دو مدل دیتاست ما ویژگی های مورد آموزش همان نقاط مختلف سیگنال

ECG در زمان هستند که به نسبت ۲۵/۷۵ داده های آموزشی و تست تقسیم می شوند.

برای حل مشکل عدم توازن بین کلاس ها یک راه حل داریم و آن هم دوباره نمونه برداری از دیتا

است به این صورت که روی کلاس هایی که تعداد اعضای آن زیاد است فرایند downsampling و

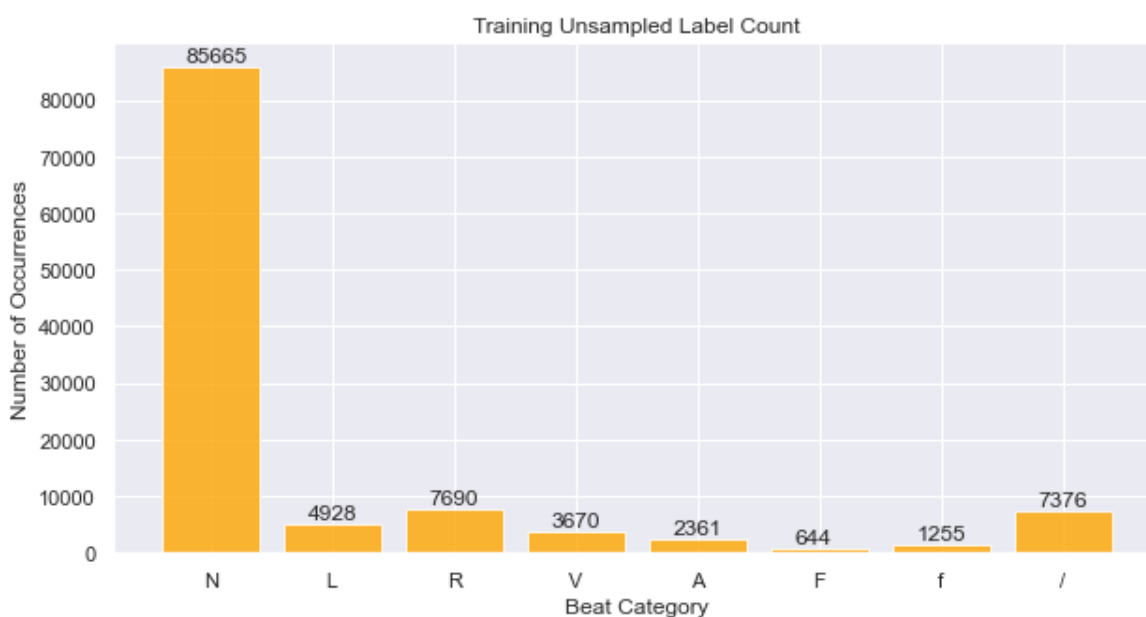
روی کلاس هایی که تعداد اعضای آن کم است فرایند upsampling را انجام دهیم.

تعدادی که برای مرجع تعداد داده ها در نظر میگیریم نیز تعداد میانگین اعضای کلاس های دارای

ناهنجاری هستند.

شکل های زیر تعداد اعضای هر کلاس را قبل و بعد از دوباره نمونه برداری در دیتاست نوع اول

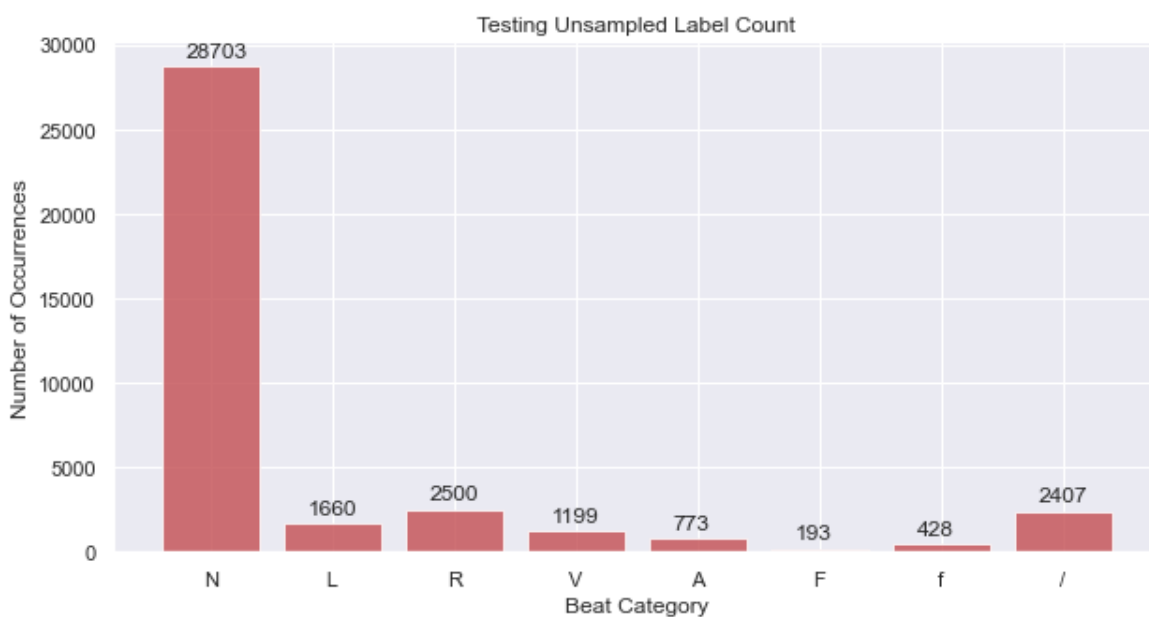
(دیتای مستقل از بیماران و مجموعه ضربان قلب) را می بینیم.



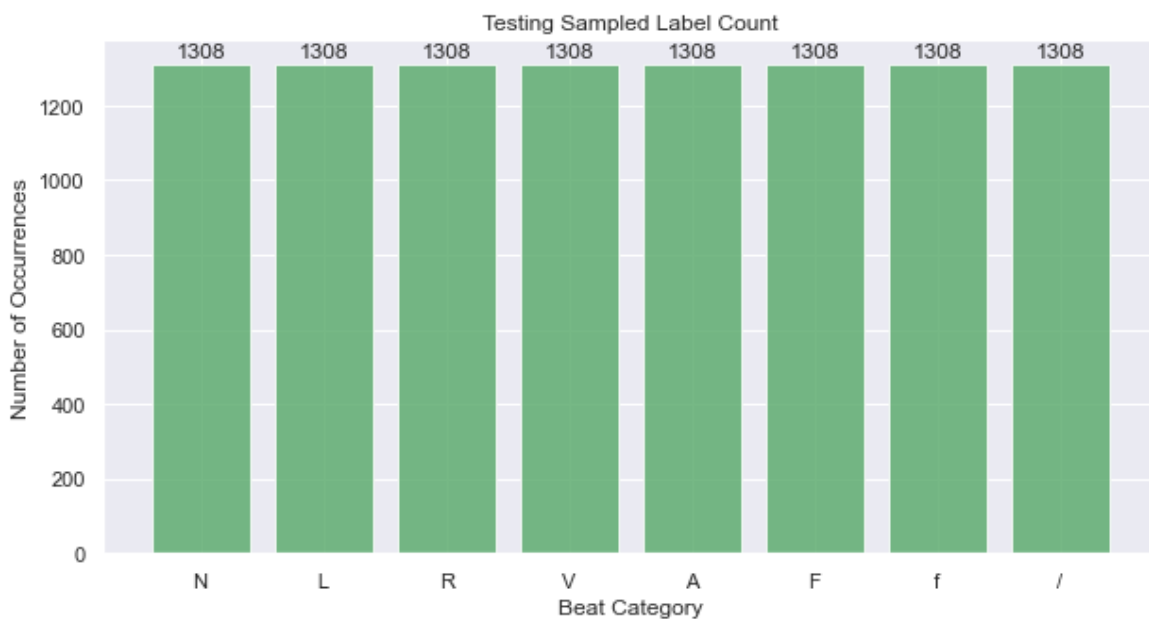
۴:۲-۵-۱-۱-۱ داده های آموزشی قبل از دوباره نمونه برداری



۱-۱-۵-۲:۵ داده های آموزشی بعد از دوباره نمونه برداری



۱-۱-۵-۲:۶ داده های تست قبل از دوباره نمونه برداری

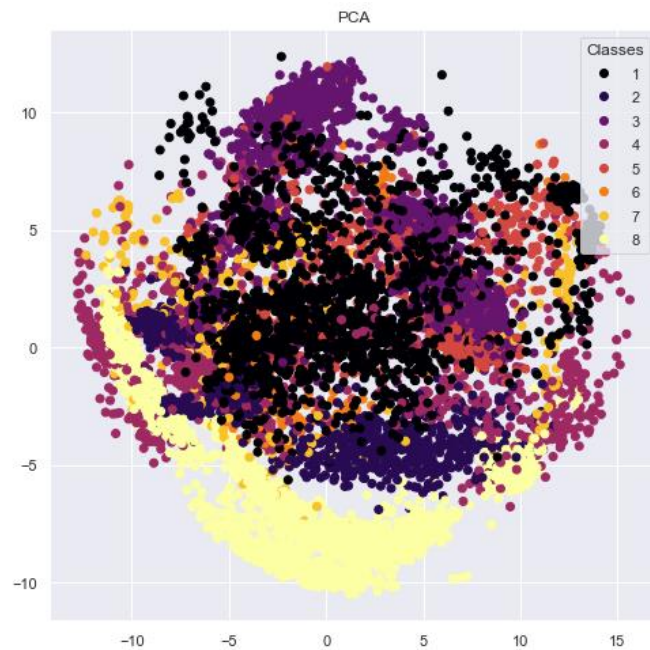


۱-۱-۵-۲:۷ داده‌های تست قبل از دوباره نمونه‌برداری

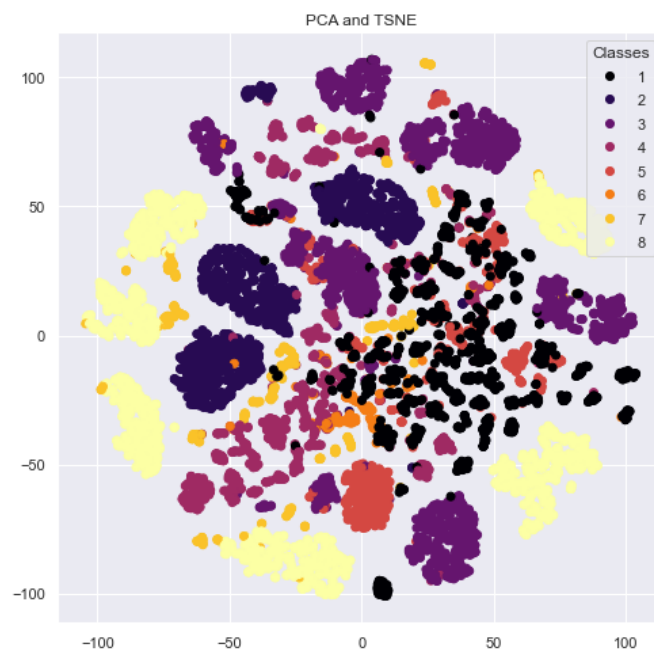
۱-۱-۵-۳ کاهش بعد دیتا

دیتای ورودی ما حجم زیادی دارد و ویژگی‌های انتخابی ما نیز زیاد هستند به همین خاطر برای دیدن دیتای خود و کلاس‌ها به صورت تصویری و درک بهتر آن چند الگوریتم کاهش بعد را روی آن تست می‌کنیم تا وضعیت دیتا را قبل و بعد دوباره نمونه‌برداری مشاهده کنیم.

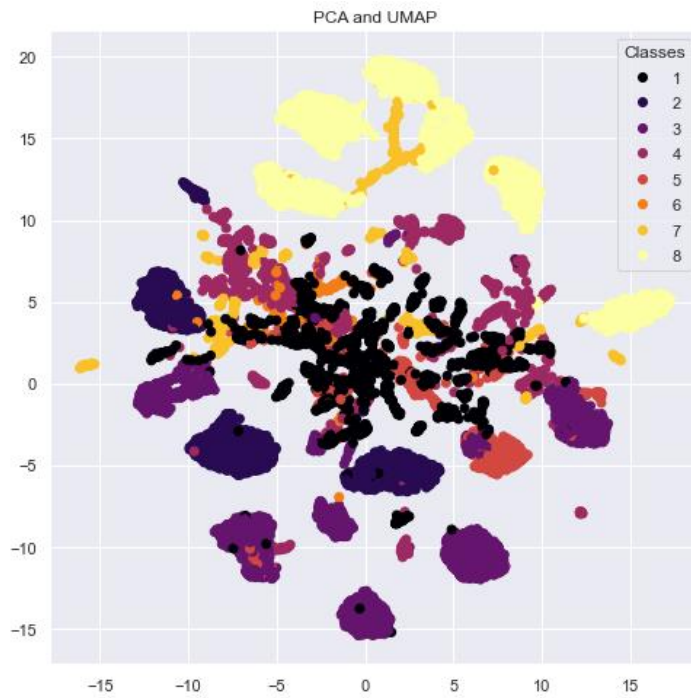
برای دیتا قبل نمونه برداری داریم:



۱-۱-۱-۵-۳:۱ خوشه‌بندی دیتای تست سمپل نشده PCA

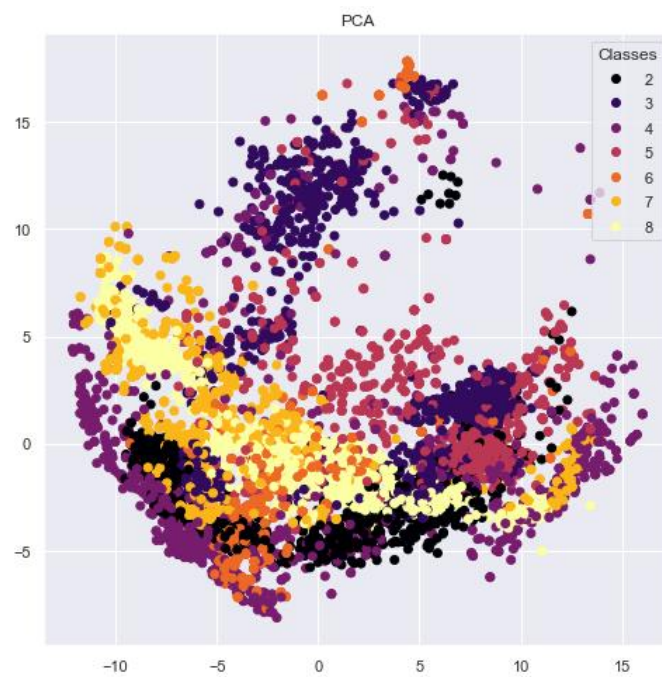


۱-۱-۱-۵-۳:۲ خوشه‌بندی دیتای تست سمپل نشده PCA-TSNE

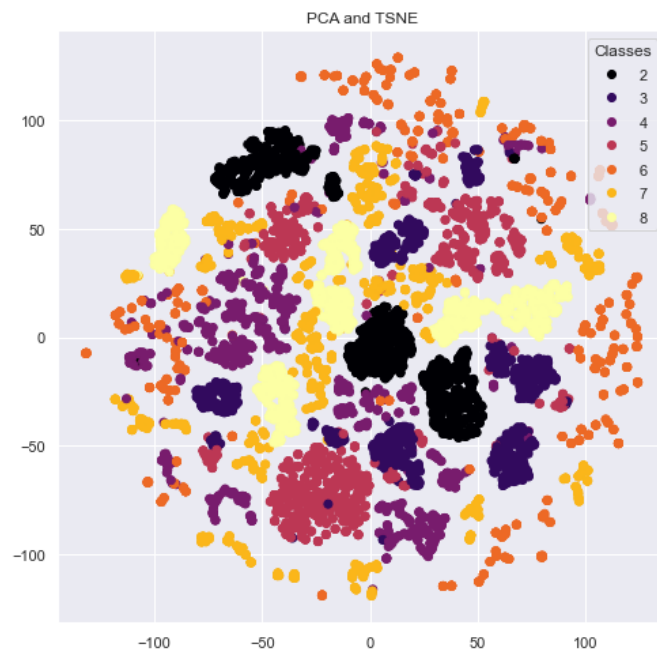


۳:۱-۱-۵-۱ خوشه‌بندی دیتای تست سمپل نشده PCA-UMAP

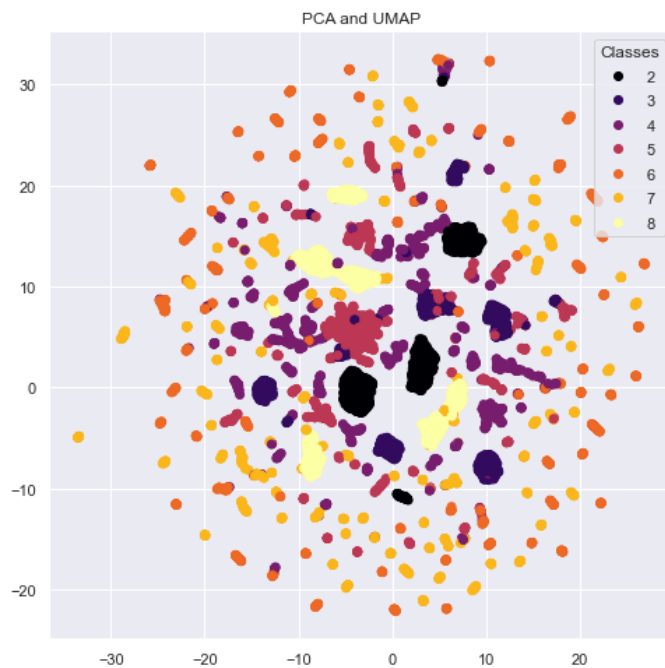
برای دیتا بعد نمونه برداری داریم:



۴:۱-۱-۵-۱ خوشه‌بندی دیتای تست سمپل شده PCA



۱-۱-۵-۳:۵ خوشه‌بندی دیتای تست سمپل شده PCA-TSNE



۱-۱-۵-۳:۶ خوشه‌بندی دیتای تست سمپل شده PCA-UMAP

همانطور که می‌بینید بعد از نمونه‌برداری دیتا بیشتر قابل جداسازی است.

برای رسیدن به بهترین طبقه بندی روی داده‌هایمان از چندین شبکه استفاده می‌کنیم و نتایج آن‌ها را

با یکدیگر مقایسه می‌کنیم، شبکه‌های انتخاب به شرح زیر هستند:

Gradient boost classifier(GBC)

Adaboost classifier(ABC)

Random forest classifier(RFC)

Naive bayes(NB)

NN Multi Layer Perceptron classifier(MLP)

Support vector classifier(SVC)

Convolutional neural network(CNN)

Long short-term memory(LSTM)

نتایج استفاده از هریک از مدل‌های بالا به همراه ماتریس درهم‌ریختگی (confusion matrix) آن‌ها

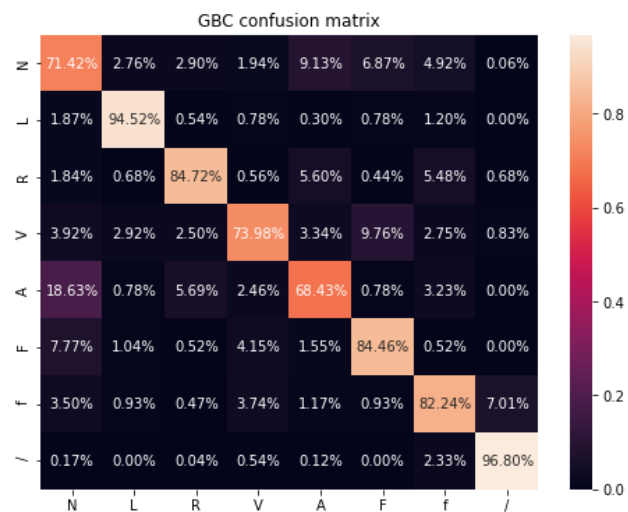
را در ادامه آورده و عملکرد شبکه‌ها را بررسی می‌کنیم.

```
Gradient Boosting Results
0.7513667696695983
Ada Boosting Results
0.33655547632253124
Random Forest Results
0.8667300530861263
Naive Bayes Results
0.35612603333069226
```

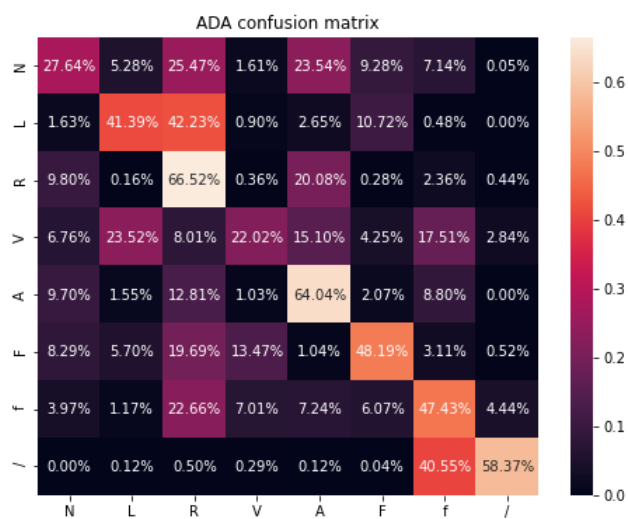
```
C:\Users\Asus\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:582: ConvergenceWarning:
Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't converged yet.
  warnings.warn(
```

```
NNMLP Results
0.8925071969997095
Support Vector Results
0.8899189182051079
```

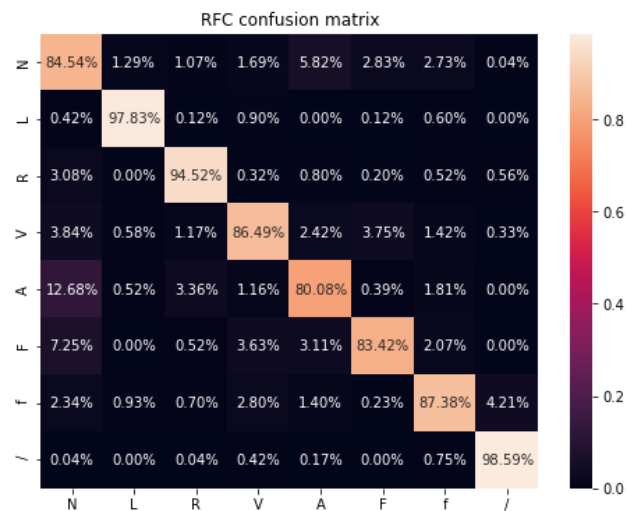
۱-۱-۱-۵-۴: دقت مدل‌های GBC, ABC, RFC, NB, NNMLP, SVC



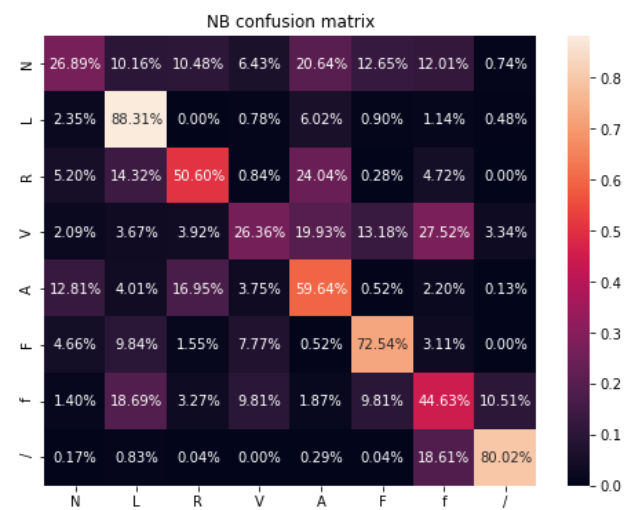
۲:۴-۵-۱-۱-۱ ماتریس درهم‌ریختگی مدل GBC



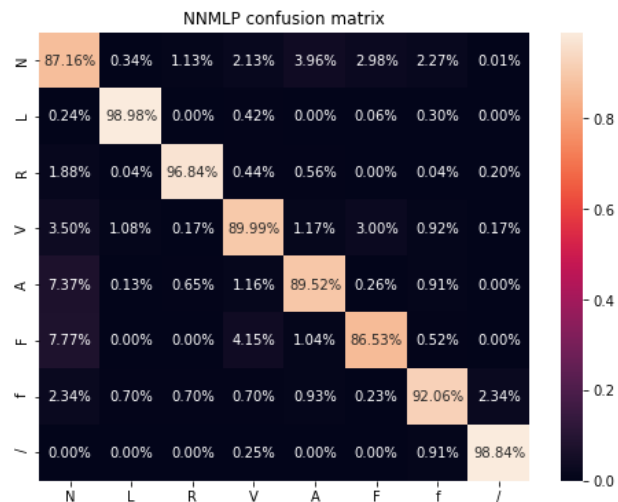
۳:۴-۵-۱-۱-۱ ماتریس درهم‌ریختگی مدل ABC



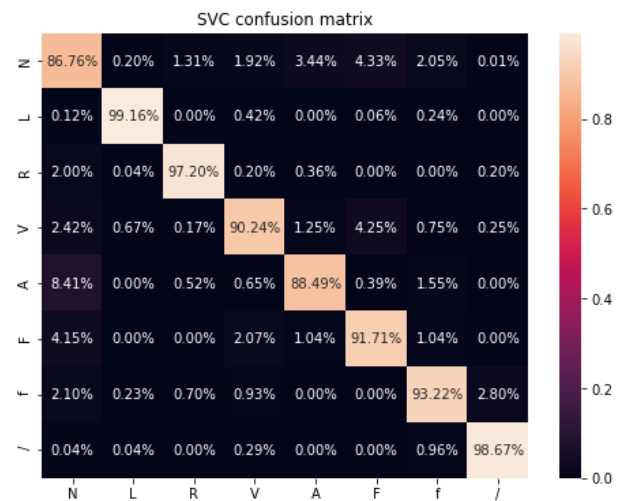
۴:۵-۱-۱-۱ ماتریس درهم‌ریختگی مدل RFC



۵:۵-۱-۱-۱ ماتریس درهم‌ریختگی مدل NB



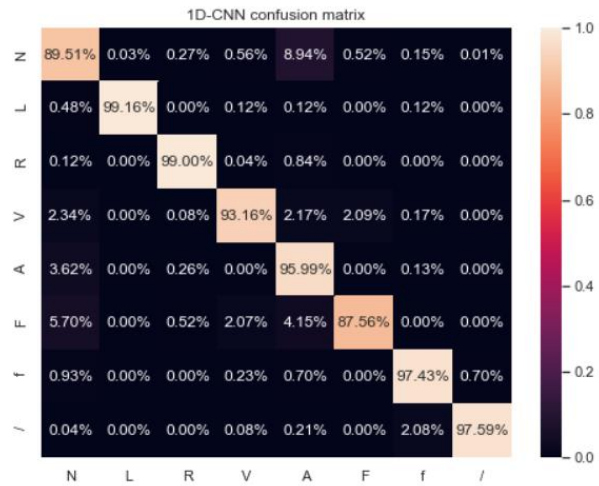
۱-۱-۱-۵-۴:۶ ماتریس درهم‌ریختگی مدل NNMLP



۱-۱-۱-۵-۴:۷ ماتریس درهم‌ریختگی مدل SVC

Accuracy : 0.9140321686078757
Precision : 0.7933107810336006
Recall : 0.9492569635140528
f1score : 0.8341760780385417

۱-۱-۱-۵-۴:۸ دقت مدل CNN



۹:۴-۱-۱-۱-۱ ماتریس درهم‌ریختگی مدل CNN

Accuracy : 0.9461009174311926
Precision : 0.9474958560203636
Recall : 0.9461009174311926
f1score : 0.9462364148590681

۱۰:۴-۱-۱-۱-۱ دقت مدل LSTM



۱۱:۴-۱-۱-۱-۱ ماتریس درهم‌ریختگی مدل LSTM

نکته قابل توجه در بررسی دقت مدل‌های بالا این است که ما در در داده تست دارای عدم توازن

هستیم و تعداد داده های نرمال از بقیه داده ها بیشتر است پس شبکه‌ای بهتر است که در همه موارد

ماتریس درهم‌ریختگی عملکرد خوبی داشته باشد بر این اساس بدترین مدل‌های ما NB , ABC

هستند و بقیه مدل ها عملکرد به نسبت قابل قبولی دارند و مدل‌های

SVC, CNN, LSTM, NNMLP بهترین عملکردها را دارند.