

به نام خدا

(2)

PCA : principal component analysis یا PCA، یک روش کاهش ابعاد است که اغلب برای کاهش ابعاد مجموعه داده‌های بزرگ، با تبدیل مجموعه بزرگی از ویژگی‌ها به ویژگی‌های کوچک‌تر که همچنان حاوی بیشتر اطلاعات مجموعه بزرگ است، استفاده می‌شود. کاهش تعداد ویژگی‌های یک مجموعه داده به طور طبیعی به قیمت کاهش دقت تمام می‌شود، اما روش ما در کاهش ابعاد این است که کمی کاهش دقت را با سادگی داده عوض کنیم. زیرا مجموعه داده‌های کوچک‌تر تجزیه و تحلیل نقاط داده را برای الگوریتم‌های یادگیری ماشینی بدون ویژگی‌های اضافی برای پردازش آسان‌تر و سریع‌تر می‌کند. بنابراین، ایده PCA این است که تعداد ویژگی‌های یک مجموعه داده را کاهش دهید، در حالی که تا حد امکان اطلاعات را حفظ کنید.

آماده سازی داده ها :

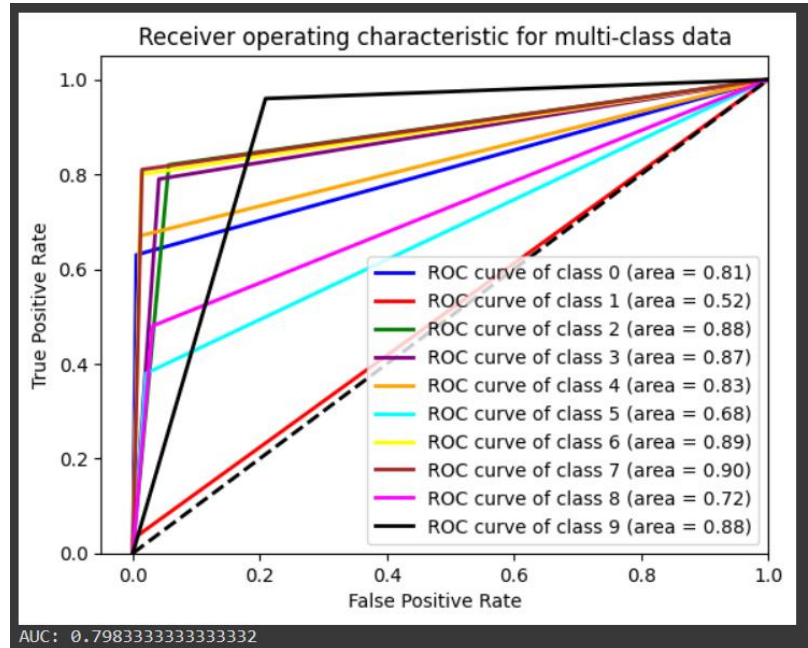
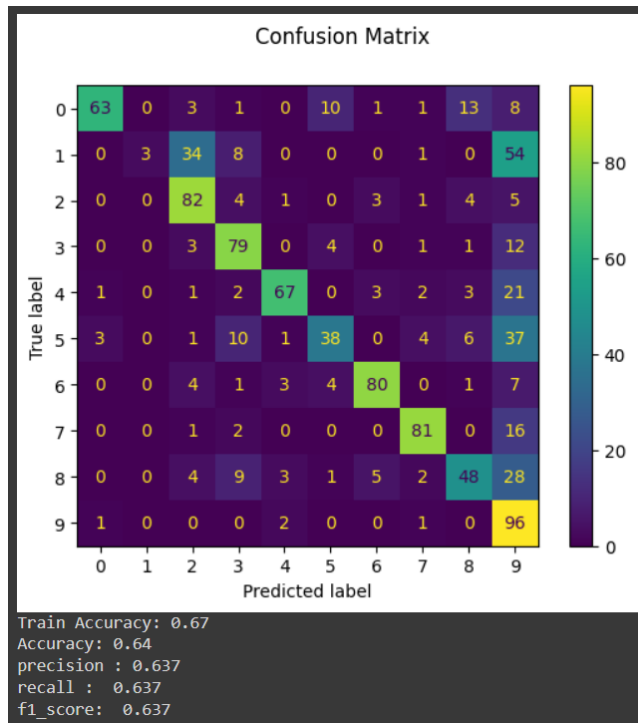
```
pca = PCA(n_components=30)
X_train_30 = pca.fit_transform(X_train)
X_test_30 = pca.transform(X_test)

pca = PCA(n_components=100)
X_train_100 = pca.fit_transform(X_train)
X_test_100 = pca.transform(X_test)
```

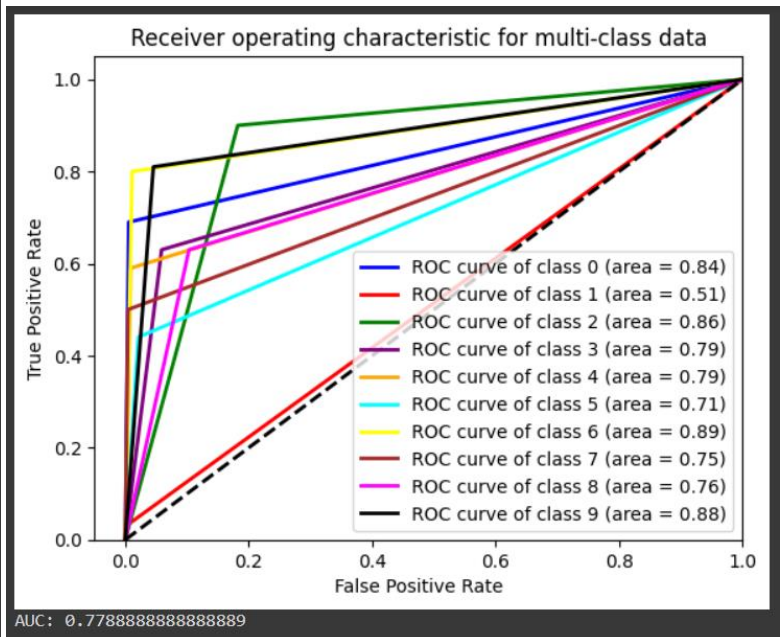
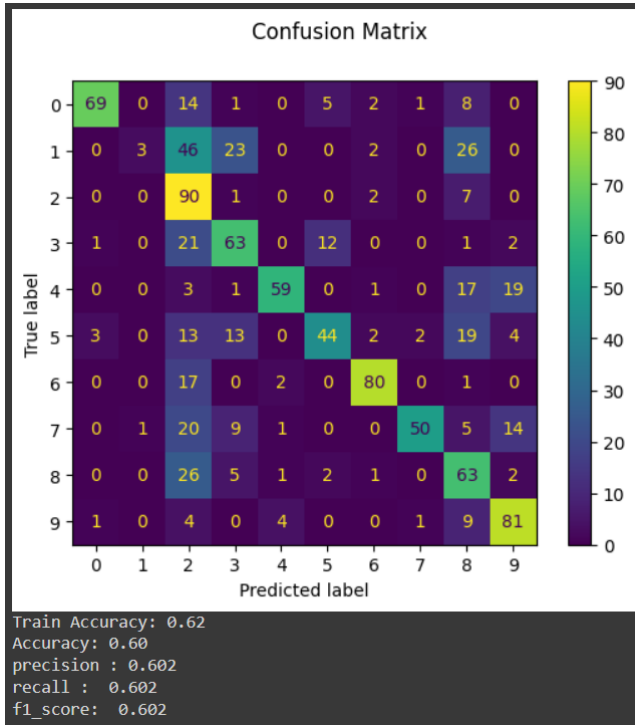
در اینجا n_components همان ابعاد نهایی مد نظر ماست.

(4) داده های اصلی

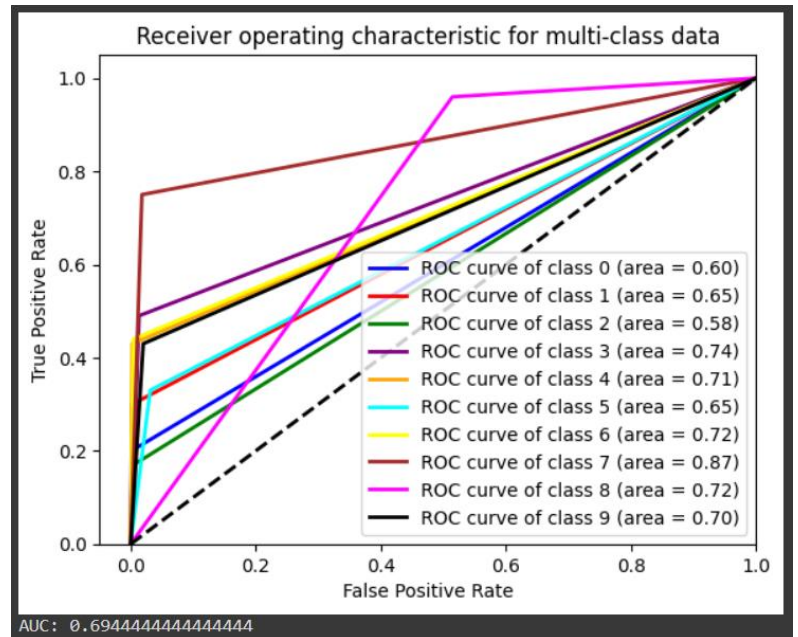
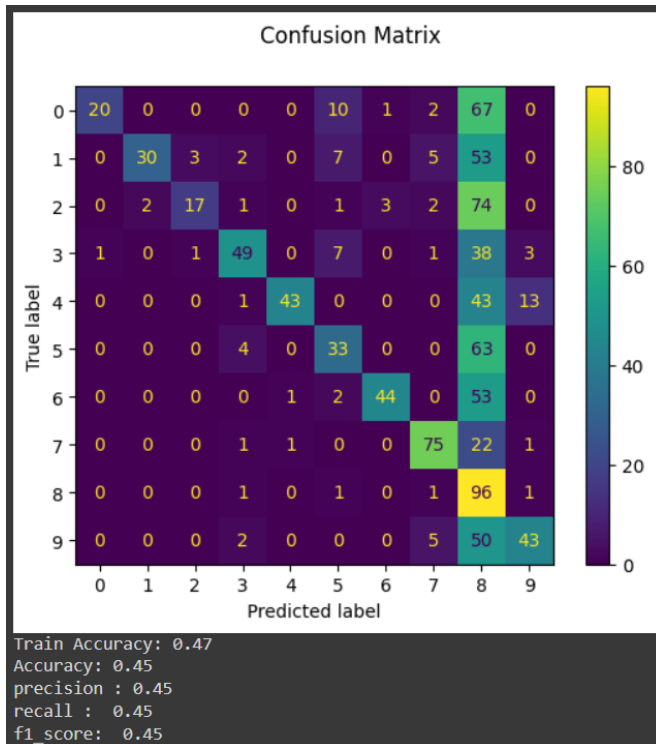
: linear , beta = 0.01



: linear , beta = 1

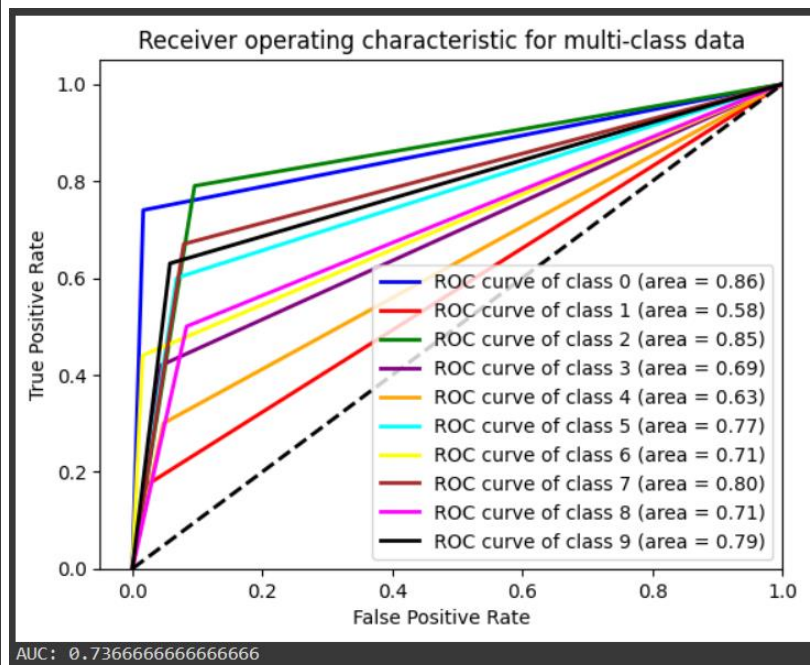
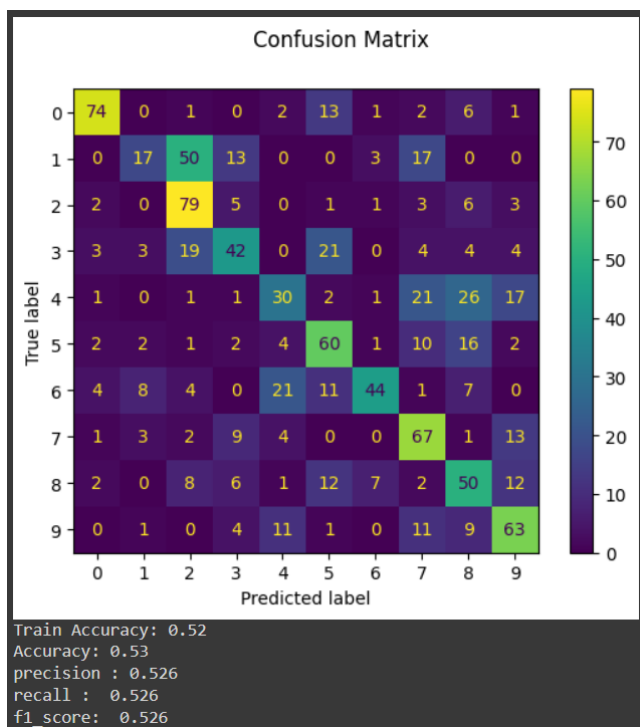


: linear , beta = 100

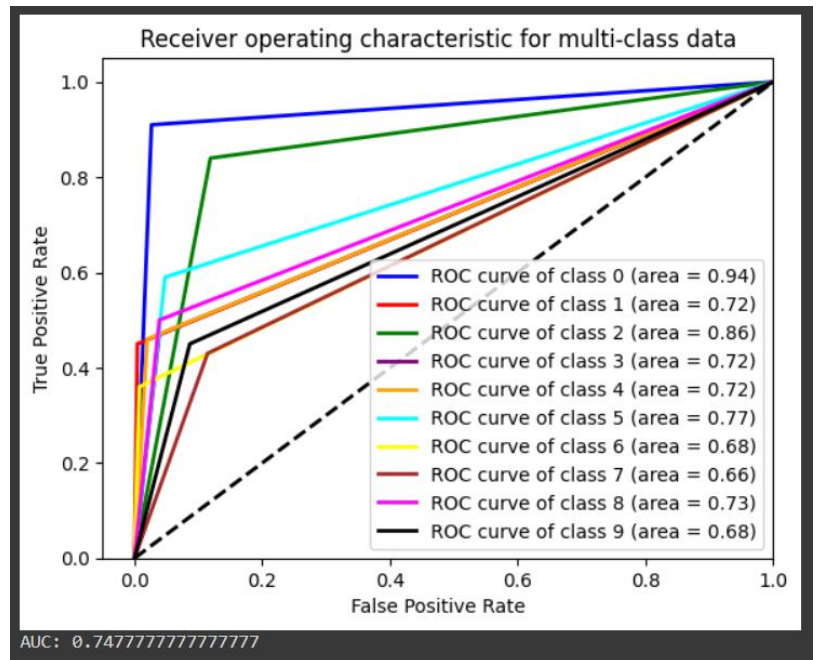
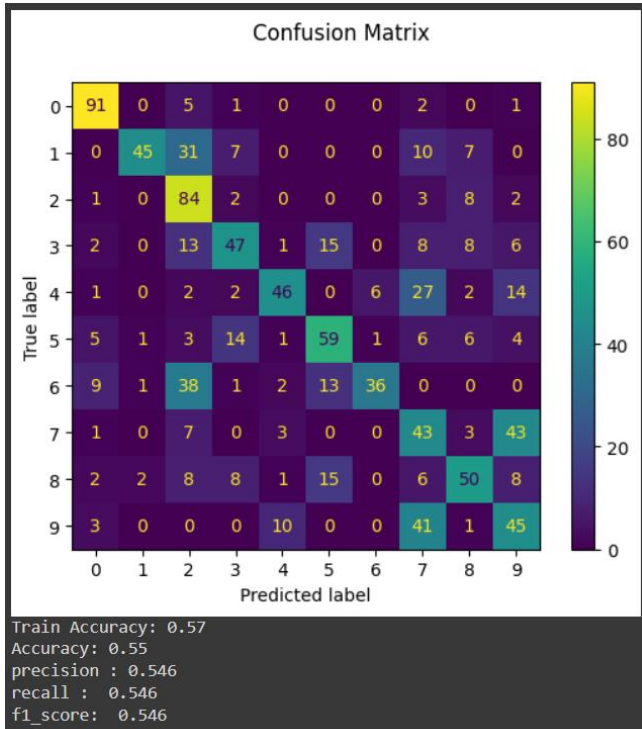


(5) داده های 100 بعدی

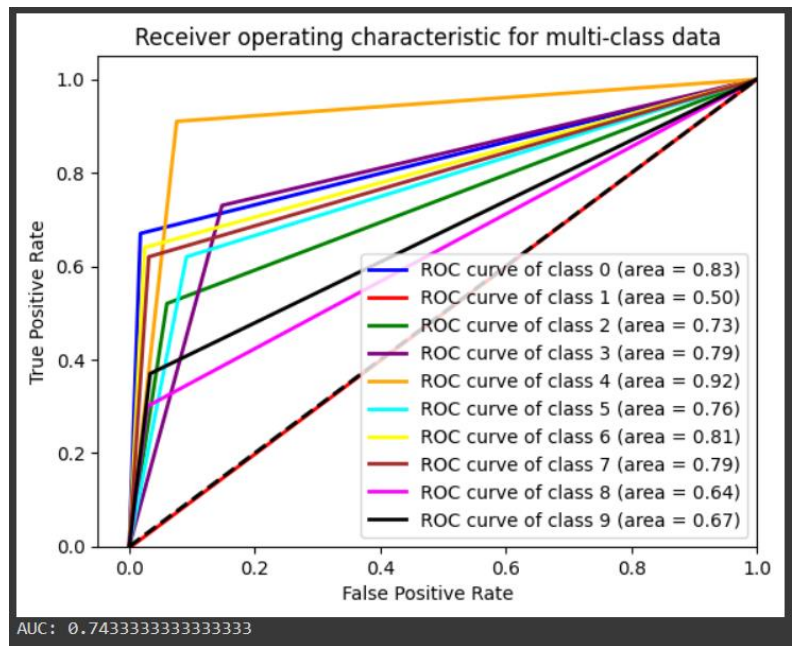
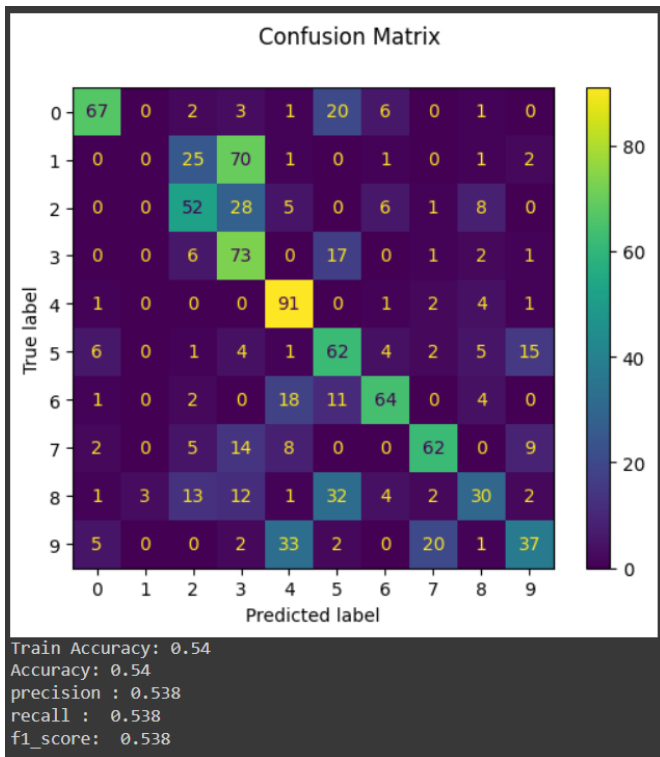
: linear , beta = 0.01



: linear , beta = 1

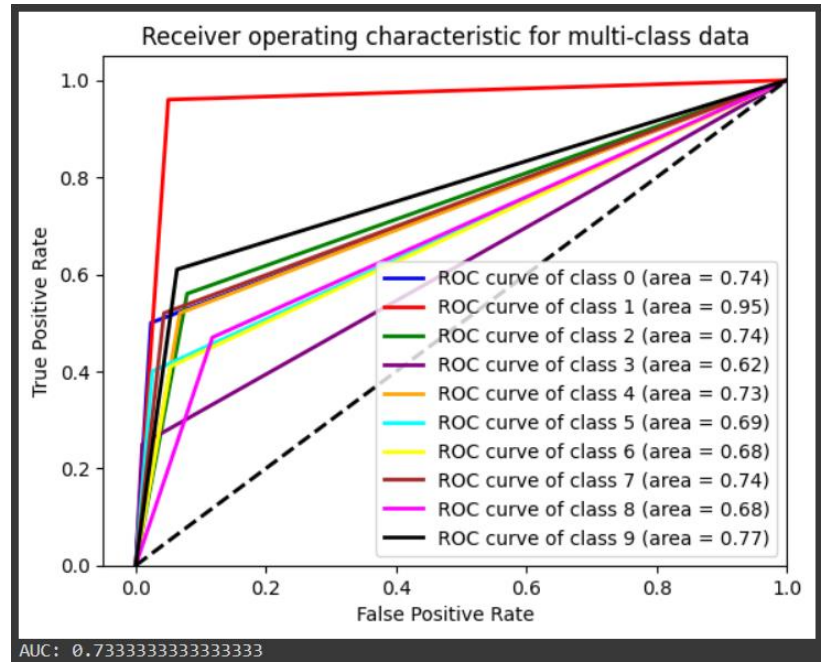
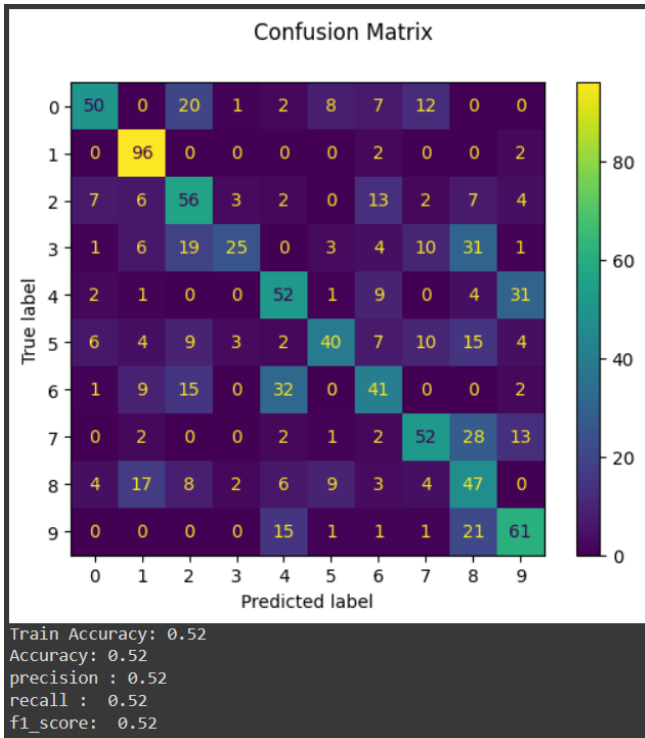


: linear , beta = 100

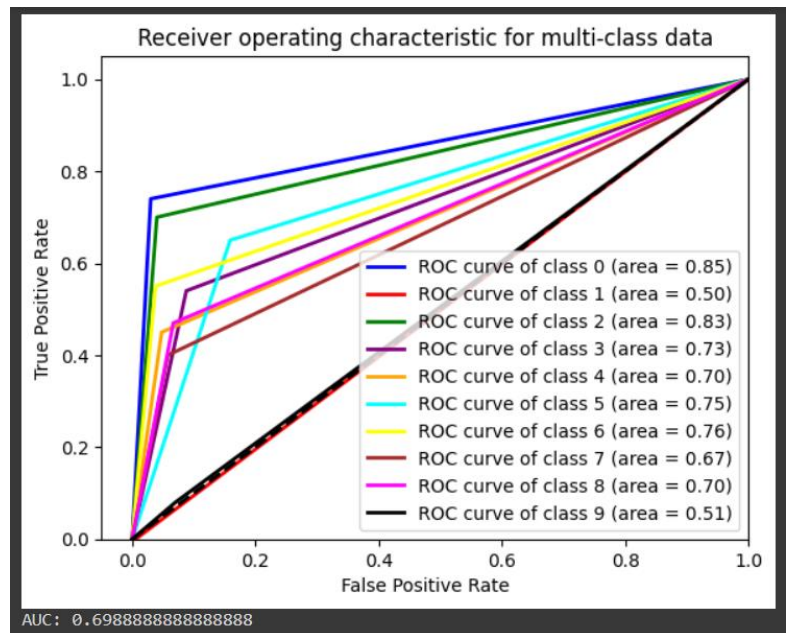
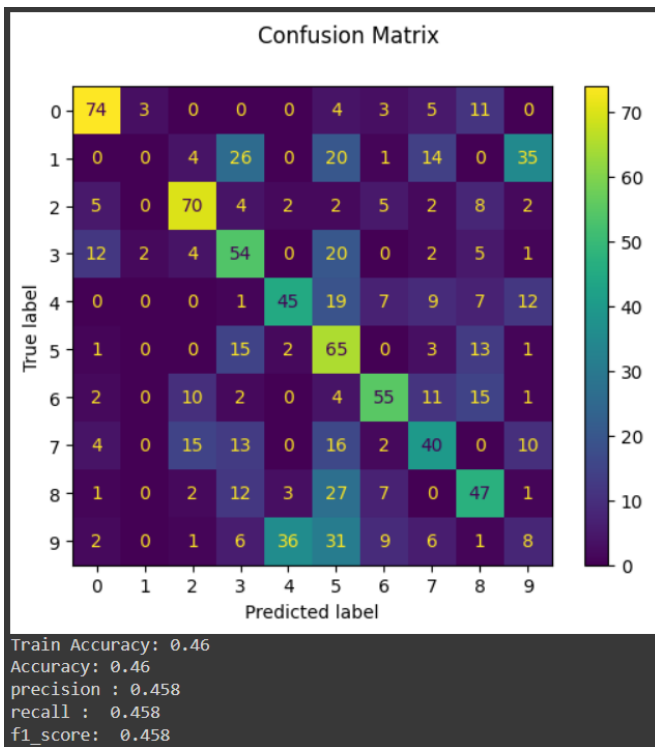


داده های ۴۰ بعدی

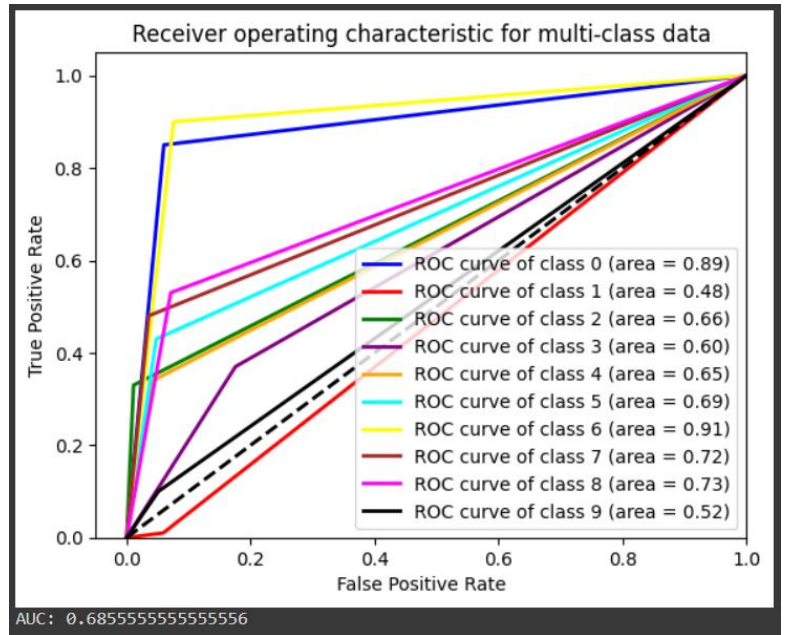
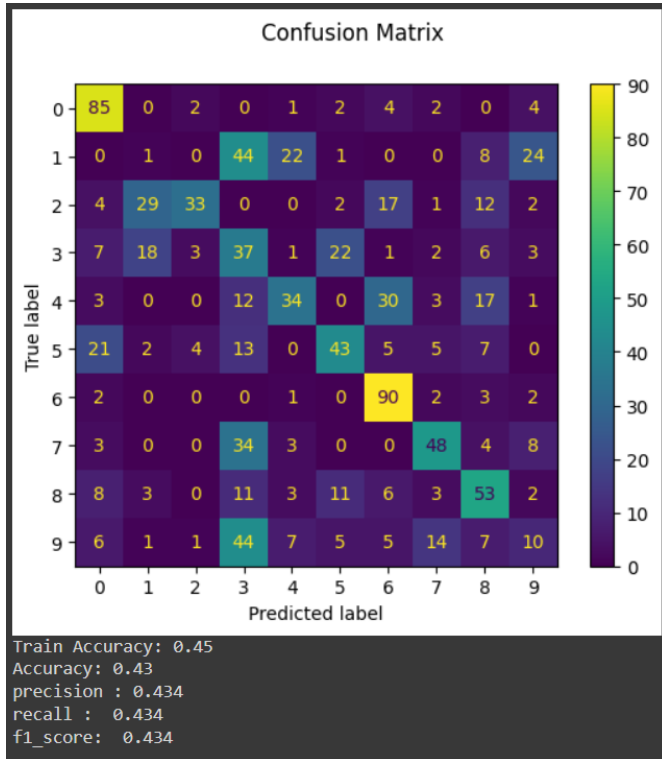
: linear , beta = 0.01



: linear , beta = 1

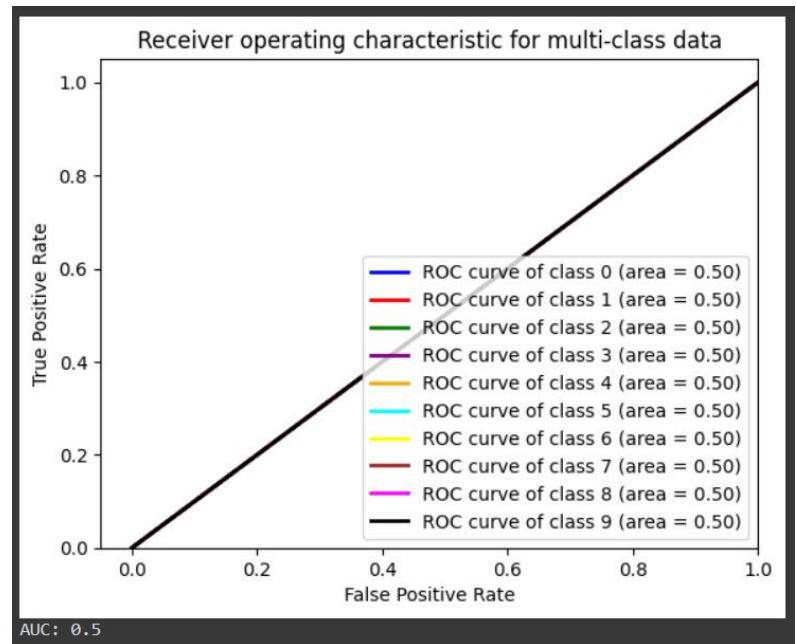
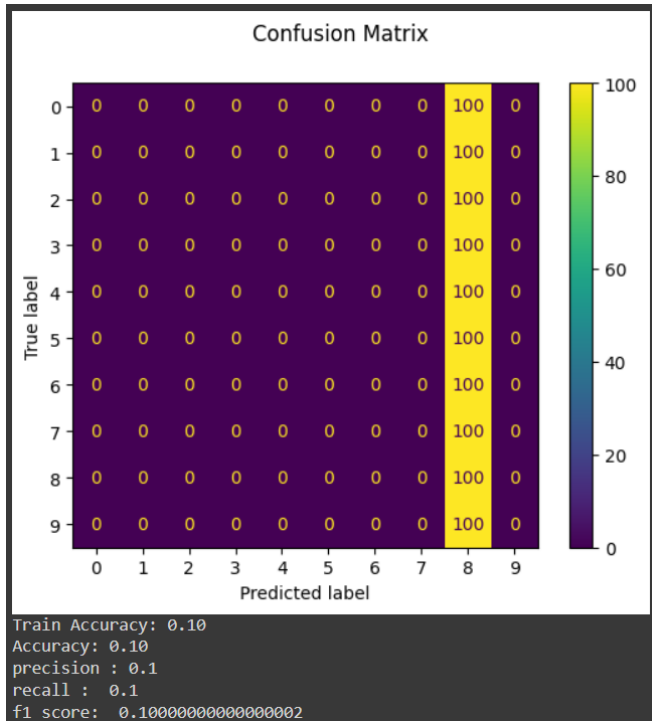


: linear , beta = 100

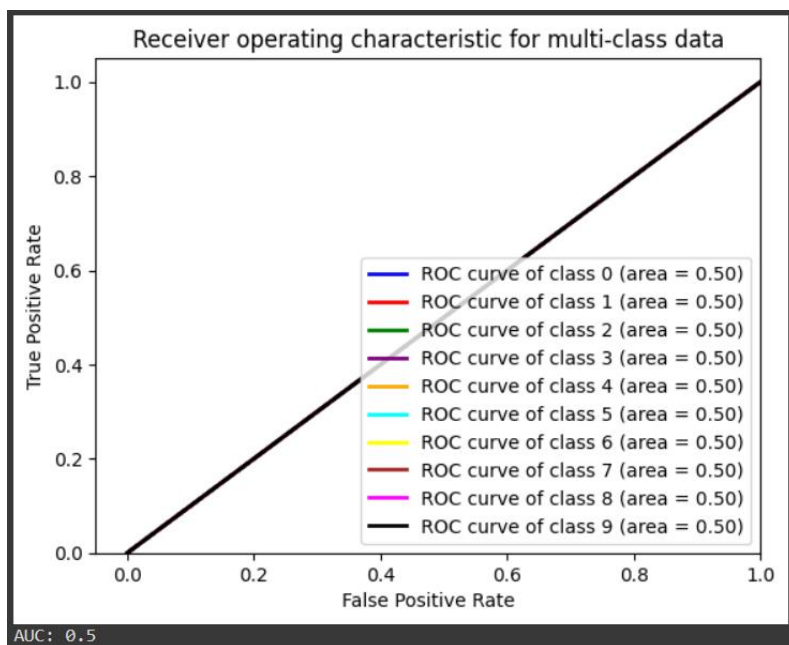
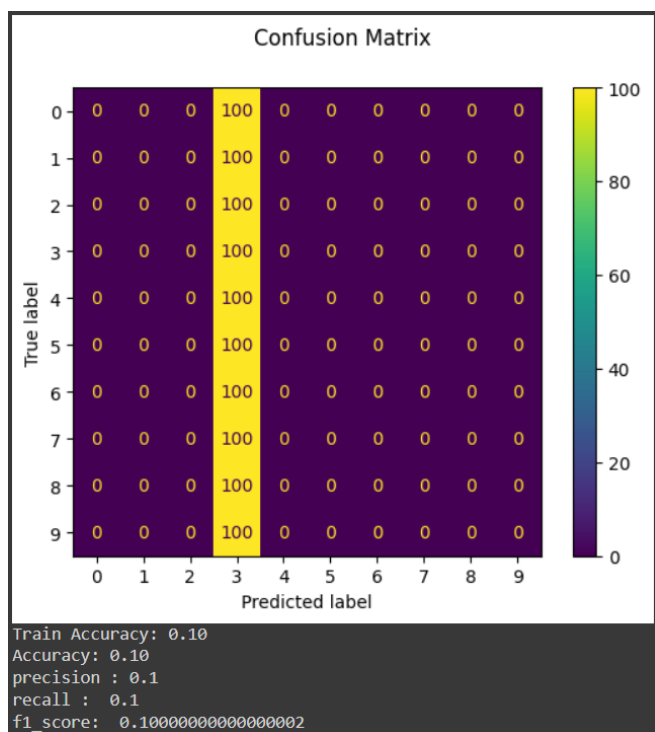


داده های اصلی :

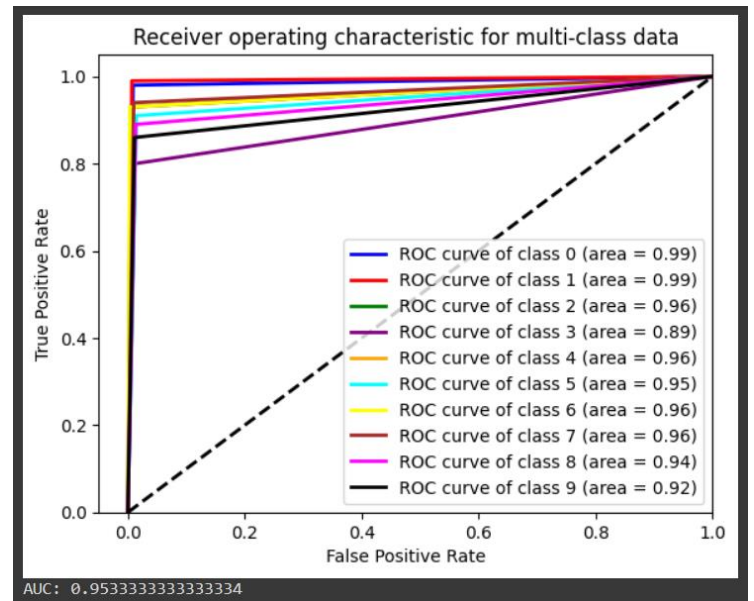
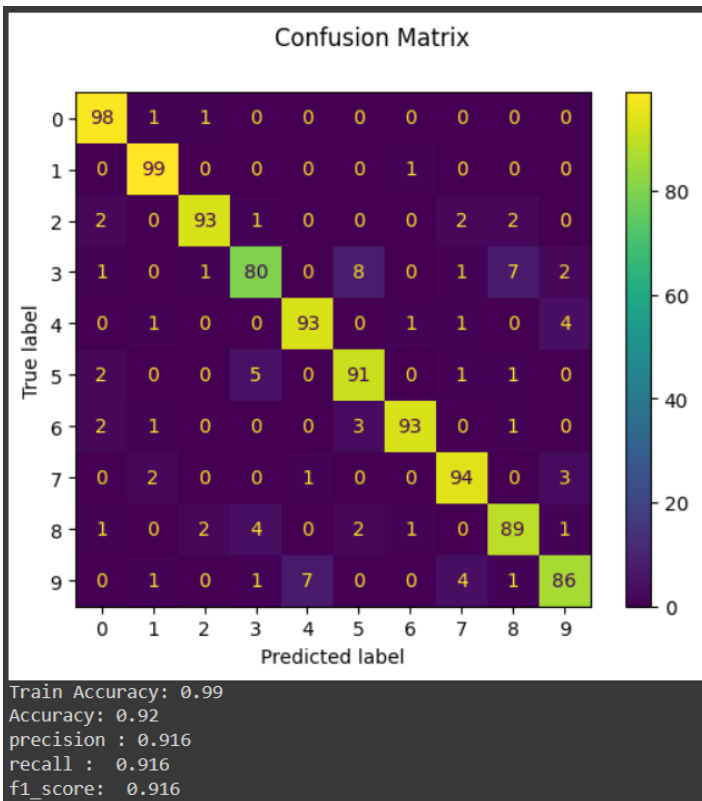
: RBF , $\beta = 0.1$, $\gamma = 1e-9$



: RBF , beta = 0.1 , gamma = 1e-6

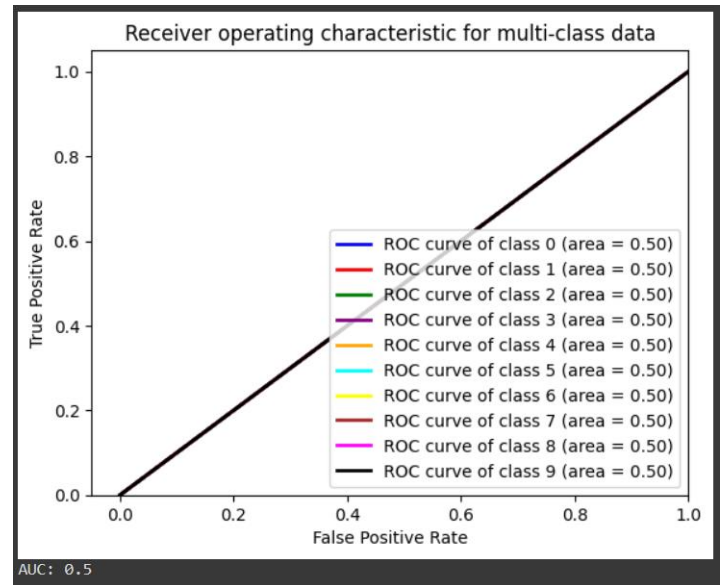
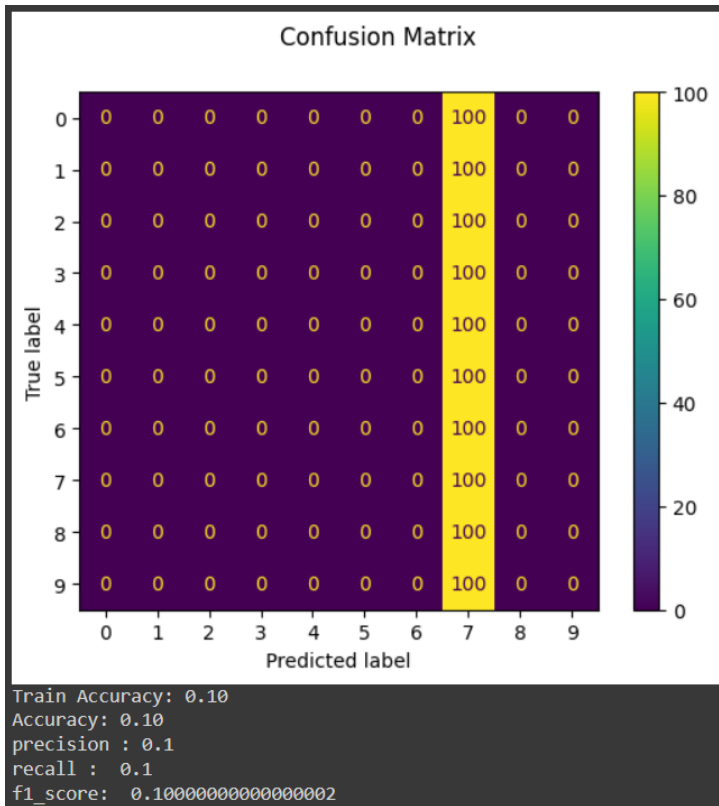


: RBF , beta = 0.1 , gamma = 1

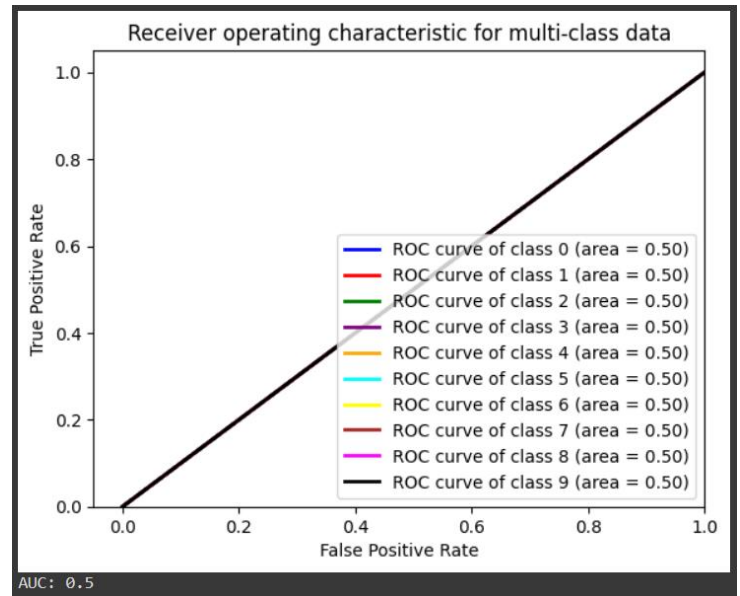
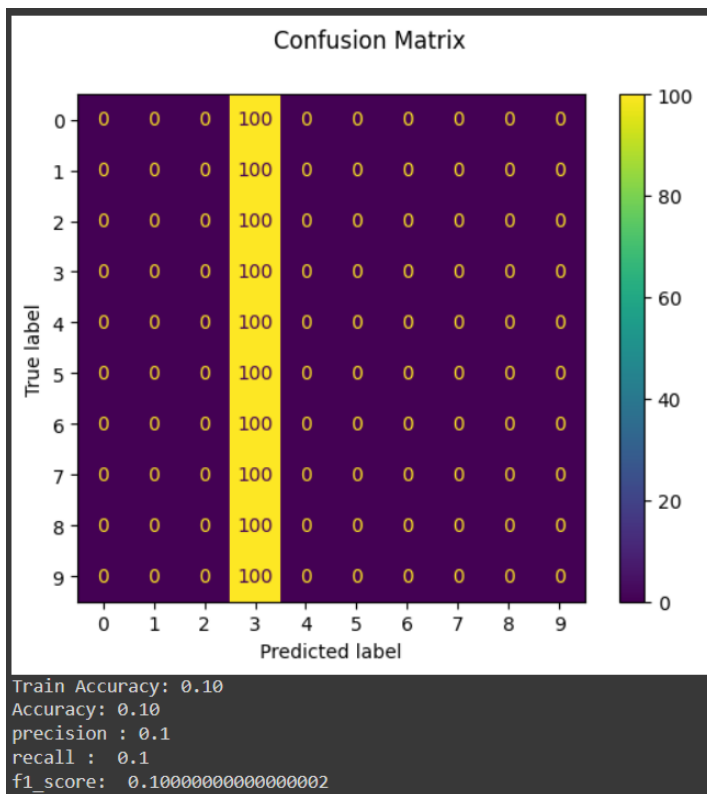


داده های 100 بعدی :

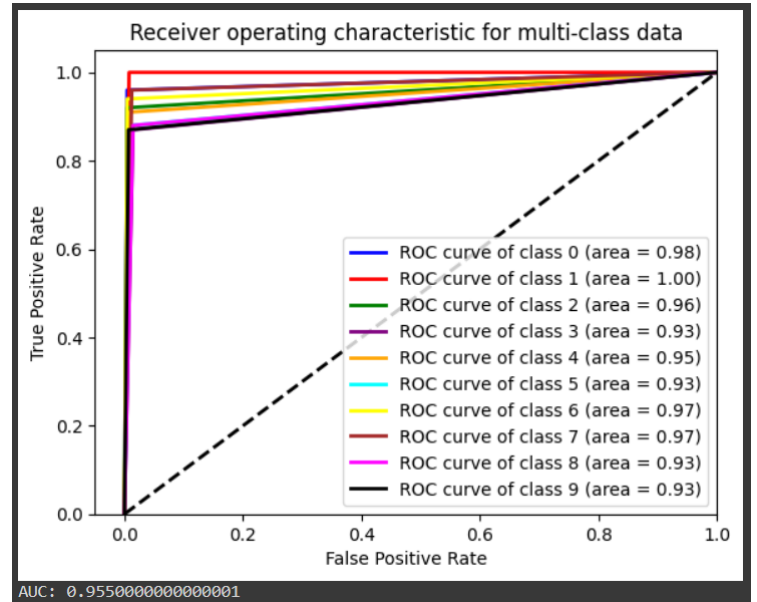
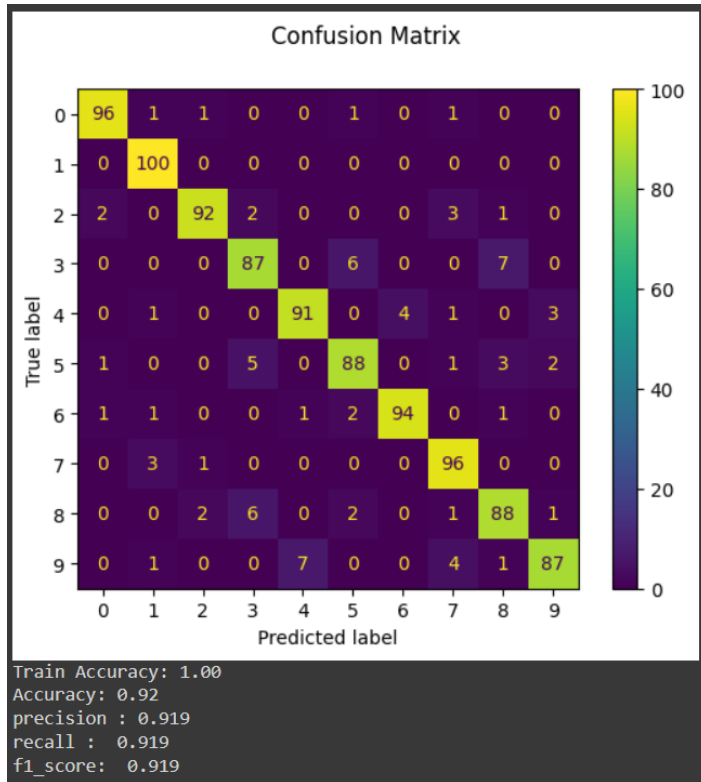
: RBF , $\beta = 0.1$, $\gamma = 1e-9$



: RBF , beta = 0.1 , gamma = 1e-6

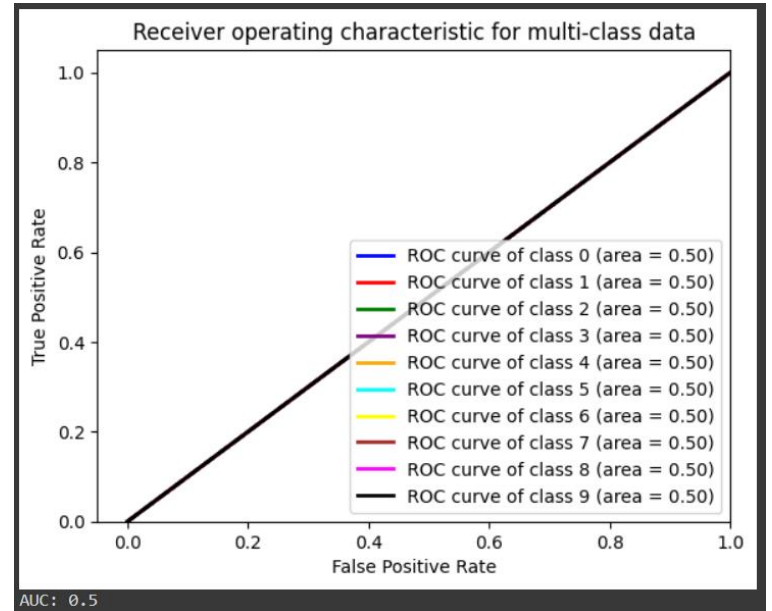
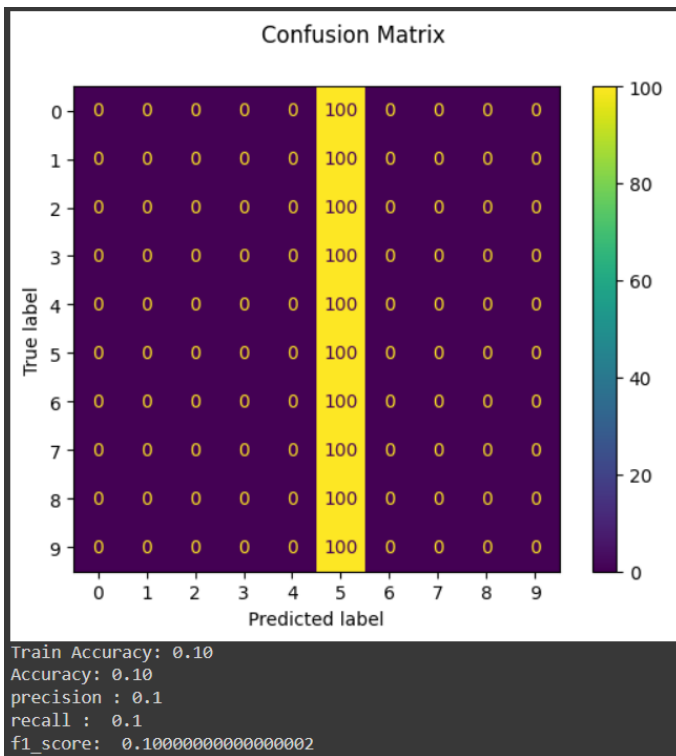


: RBF , beta = 0.1 , gamma = 1

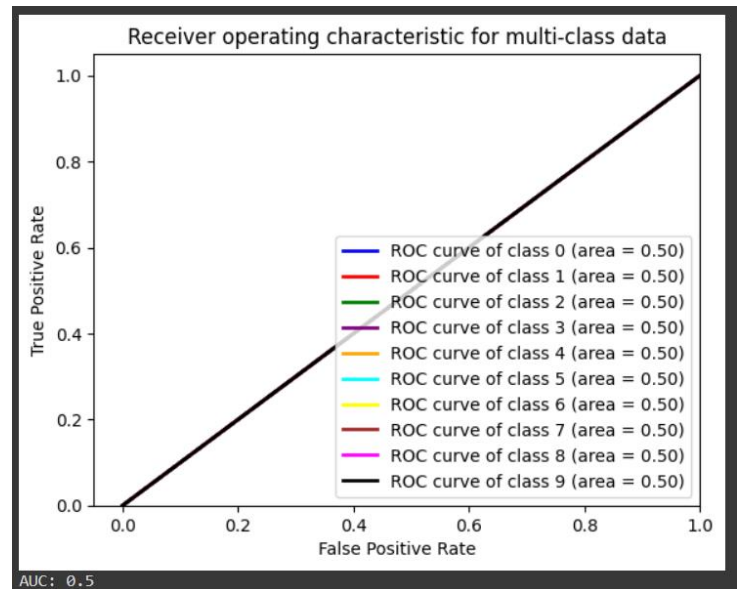
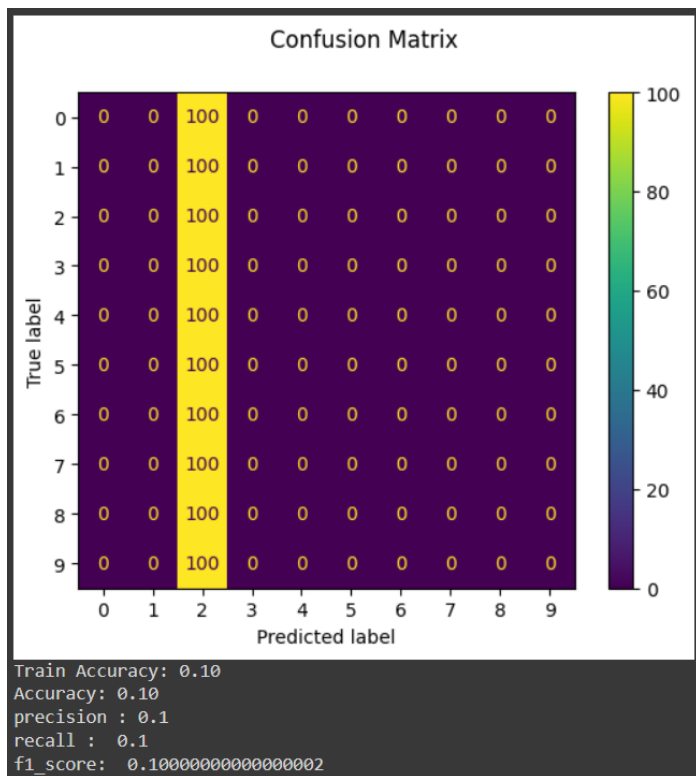


داده های ۴۰ بعدی :

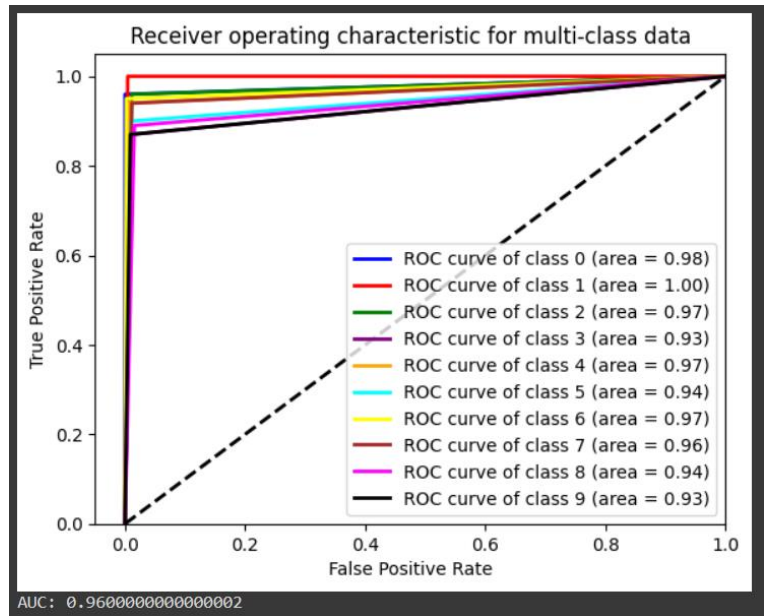
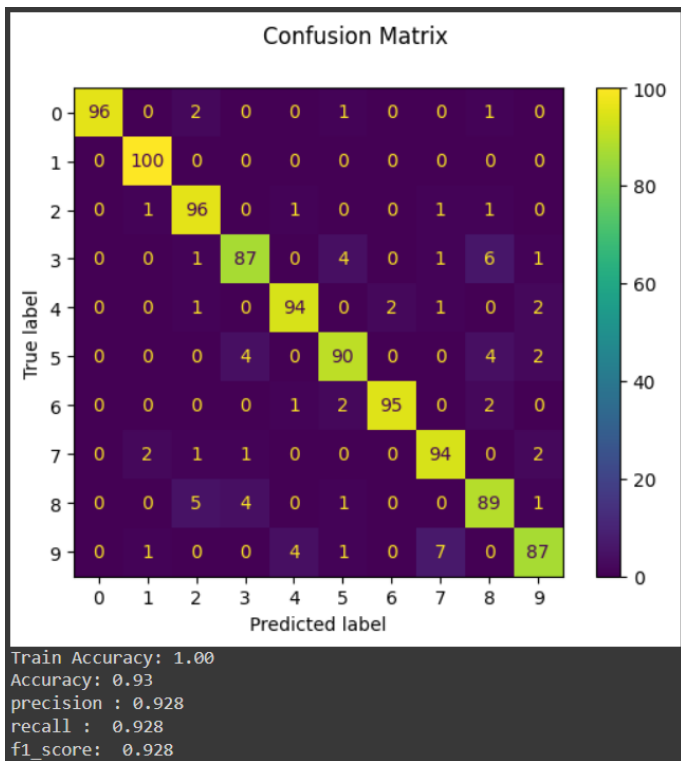
: RBF , beta = 0.1 , gamma = 1e-9



: RBF , beta = 0.1 , gamma = 1e-6



: RBF , beta = 0.1 , gamma = 1



داده های اصلی :

linear			
beta	1e-2	1	100
Train acc	0.67	0.62	0.47
Test acc	0.64	0.60	0.45

RBF , beta = 0.1			
gamma	1e-2	0.1	1
Train acc	1	1	0.99
Test acc	0.94	0.95	0.92

دقت تست و ترین به هم نزدیک هستند پس اورفیت رخ نداده است.
همانطور که میبینیم کرنل RBF دقت بیشتری از کرنل خطی دارد. در کرنل های خطی مقدار کمتر بتا نیز موثرتر عمل می کند.

داده های 100 بعدی :

linear			
beta	1e-5	1e-3	1
Train acc	0.70	0.56	0.57
Test acc	0.70	0.56	0.55

RBF , beta = 0.1			
gamma	1e-9	1e-6	1
Train acc	0.1	0.1	1
Test acc	0.1	0.1	0.92

با کاهش ابعاد طبیعتاً بخشی از ویژگی های داده ها را از دست می‌دهیم. همانطور که انتظار می‌رفت در کرنل خطی با مقدار بتا ثابت با کاهش ابعاد بخشی از دقت را از دست داده ایم. کاهش گاما نیز تاثیر منفی در دقت مدل دارد.

داده های 40 بعدی :

linear			
beta	1e-5	1e-3	1
Train acc	0.57	0.54	0.46
Test acc	0.59	0.55	0.46

RBF , beta = 0.1			
gamma	1e-9	1e-6	1
Train acc	0.1	0.1	1
Test acc	0.1	0.1	0.93

با کاهش ابعاد به 40 دقت بسیار افت می کند و ویژگی های بسیاری را از دست می دهیم.

(7)

استفاده از PCA 100 بهترین انتخاب است زیرا در این حالت داده ها ویژگی های مفیدی وجود دارد و مدل می تواند با این ویژگی ها بسیار خوب آموزش ببیند و بسیار سریعتر از استفاده از ویژگی های اصلی برای آموزش مدل SVM است، اما PCA 40 نامناسب است. زیرا با کاهش ابعاد داده ها به 40 ویژگی های مفید را از دست می دهد و در نتیجه دقت مدل بسیار کاهش می یابد.

همچنین، استفاده از SVM با کرنل RBF بسیار بهتر از Linear است و می تواند دقت تست را افزایش دهد. SVM با کرنل RBF می تواند به خوبی با مجموعه داده PCA 40 کار کند و به نظر من بهترین انتخاب این سوال استفاده از PCA 40 است زیرا عملکرد مدل آنقدر کاهش پیدا نکرده و همچنین سرعت آموزش بسیار بالاتر از استفاده از PCA 100 یا داده های اصلی است.

با افزایش پارامتر گاما قدرت مدل افزایش می یابد تا بتواند داده های آموزشی را به خاطر بسپارد و سپس اورفیت اتفاق می افتد، بنابراین دقت تست بسیار کاهش می یابد، بنابراین بهترین مقدار برای گاما برای این مثال 0.1 است.

(8) یافتن عدد 2:

$Recall = TP / (TP + FN)$ بهترین معیار است چرا که به اعدادی که اشتباه 2 شناسایی شده اند کاری ندارد و اعداد 2 را که درست پیش بینی شده اند را نسبت به کل 2 ها می سنجد.

(9) داده های 100 بعدی

$Beta : 1e-5 \leq 419$: مقدار بسیار کوچک بتا به SVM اجازه می دهد تا با طبقه بندی های نادرست ملایم تر باشد. در نتیجه، بردارهای پشتیبانی بیشتری برای گرفتن پیچیدگی های داده های آموزشی استفاده می شود.

$Beta : 1e-3 \leq 215$: در این حالت "بتا" کمی بزرگتر است و طبقه بندی های اشتباه را بیشتر جریمه می کند، که باعث کاهش تعداد بردارهای پشتیبانی می شود. مدل ممکن است سعی کند بین به حداقل رساندن خطای طبقه بندی و حداکثر کردن مارجین تعادل برقرار کند.

$Beta : 1 \leq 468$: در اینجا با تنظیم «بتا» روی 1، SVM ممکن است بیشتر بر روی حداکثر کردن مارجین تمرکز کند و در عین حال دسته بندی های اشتباه را جریمه کند. تعداد بردارهای پشتیبانی در مقایسه با $Beta = 1e-3$ بیشتر است، که نشان دهنده رویکرد متعادل تر است.

$Beta : 0.1 \text{ and } \gamma : 1e-6 \leq 992$: در RBF SVM، انتخاب «گاما» تأثیر نقاط آموزش را به صورت جداگانه تعیین می کند. مقدار «گاما» کوچک تر باعث می شود هسته RBF گسترده تر شود و تعداد بیشتری از بردارهای پشتیبانی برای بررسی پیچیدگی داده ها مورد نیاز است.

بین اندازه مارجین و تعداد بردارهای پشتیبان یک تریدآف وجود دارد. تنظیم ابرپارامترها امکان کنترل این تریدآف را بر اساس اولویت‌ها به ما می‌دهد.

(10) خیر با کاهش ابعاد این حساسیت نیز کاهش می‌یابد. در داده‌های اصلی، تعداد ویژگی‌ها زیاد است و مدل‌ها ابهامات بیشتری دارد. این ابهامات ممکن است باعث شود مدل‌ها حساس‌تر به تغییرات پارامترها باشند. در داده‌های با کاهش بعد، تعداد ویژگی‌ها کاهش یافته و مدل‌ها ساده‌تر می‌شوند. این سادگی ممکن است باعث شود که حساسیت کمتری به تغییرات داشته باشند.

مدل‌های با تعمیم بالا معمولاً حساسیت کمتری به تغییرات در پارامترها دارند. چون این مدل‌ها تلاش می‌کنند نمونه‌های عمومی را به خوبی توصیف کنند، تغییرات کوچک در پارامترها ممکن است تأثیر کمتری داشته باشد. دیگر روش‌ها: رگولاریزاسیون و افزایش داده (Data Augmentation)

(11)

ایده اصلی در SVM، ایجاد یک حد مرز به نحوی است که بیشتر نقاط داده از یک طرف این حد مرز یا خط قرار بگیرند و تا حد امکان فاصله (مارجین) بین این حد مرز و نقاط مختلف بیشینه شود. مارجین در SVM به فاصله نزدیک‌ترین نقاط با حد مرز یا هایپرپلین اشاره دارد. هدف اصلی در SVM این است که مارجین را بیشینه کند. SVM از توابع هسته برای بردن داده‌ها به فضاهای با ابعاد بالاتر استفاده می‌کند. این تبدیلات به SVM این امکان را می‌دهند که با استفاده از حد مرزها یا هایپرپلین‌هایی پیچیده‌تر، داده‌های غیرخطی را هم بتواند تشخیص دهد. هدف یافتن وزن‌ها و ضرایب تابع هایپرپلین به نحوی است که مارجین بیشینه شود و در عین حال خطا (نقاطی که در داخل مارجین یا اشتباه دسته‌بندی شده‌اند) کمینه گردد.

(12)

روش "One-vs-Rest" یا "One-vs-All" یکی از روش‌های متداول در رده‌بندی چند کلاسه با استفاده از مدل‌های دودویی (binary classifiers) است. در این روش، برای هر کلاس یک مدل دودویی آموزش داده می‌شود تا بین اعضای کلاس مورد نظر و بقیه کلاس‌ها تمایز ایجاد کند. سپس در مرحله تست، کلاسی که امتیاز بیشتری را کسب کند، به عنوان کلاس نهایی برای نمونه در نظر گرفته می‌شود. در این روش، اگر یک مدل دودویی برای هر کلاس استفاده شود، تعداد کل مدل‌ها برابر با تعداد کلاس‌ها است.

مثال:

اگر ۳ کلاس داشته باشیم، سه مدل دودویی به صورت زیر آموزش داده می‌شوند:

1. مدل برای تشخیص کلاس 1 در مقابل کلاس‌های 2 و 3 (کلاس 1 vs. کلاس 2&3)
2. مدل برای تشخیص کلاس 2 در مقابل کلاس‌های 1 و 3 (کلاس 2 vs. کلاس 1&3)
3. مدل برای تشخیص کلاس 3 در مقابل کلاس‌های 1 و 2 (کلاس 3 vs. کلاس 1&2)

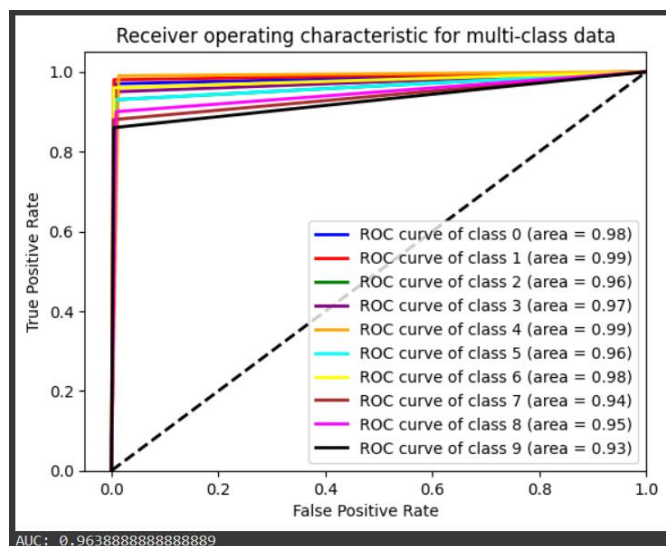
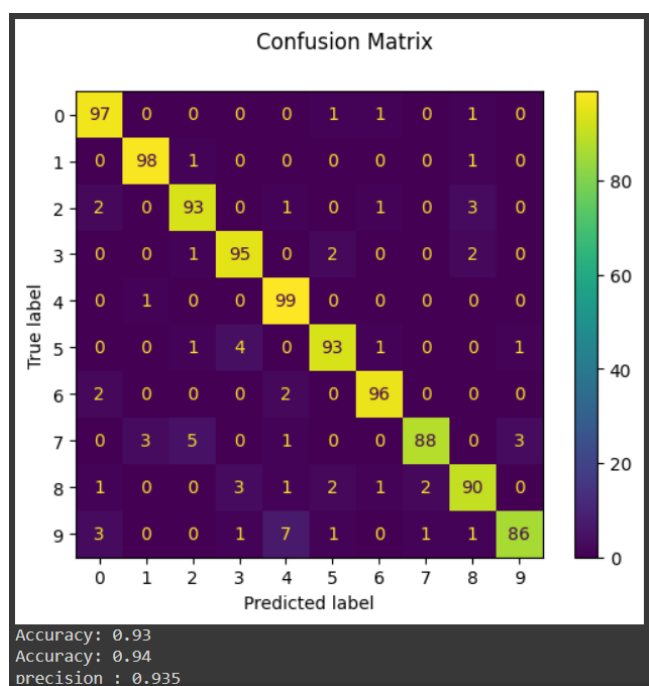
در مرحله تست، نمونه به تمام مدل‌ها وارد شده و امتیازهای هر مدل بررسی می‌شوند. کلاسی که مدل مربوط به آن امتیاز بیشتری داشته باشد، به عنوان کلاس نهایی انتخاب می‌شود.

C در SVM نشان‌دهنده تعیین میزان تسلیم به خطا است. هرچه مقدار C بزرگتر باشد، مدل با داده‌های آموزشی با دقت بیشتری مطابقت دارد ولی ممکن است باعث Overfitting شود.

در حالت Crammer Singer، بتا یک پارامتر اضافی است که در تابع هدف (Objective Function) مربوط به مسئله بهینه‌سازی آموزش ماشین بردار پشتیبان Crammer Singer اضافه می‌شود. این پارامتر به بهبود عملکرد در شرایطی کمک می‌کند که نمونه‌ها به‌طور همزمان به چند کلاس تعلق داشته باشند. در این موارد، Crammer Singer به صورت یک مسئله بهینه‌سازی چندکلاسه با محدودیت‌های مشخص می‌شود.

در کل، C در SVM و بتا در Crammer Singer هر دو مربوط به میزان تحمل خطا و Overfitting هستند، اما تأثیر مستقیم بر یکدیگر ندارند.

(13)



همانطور که می‌بینیم دقت داده‌های آموزشی کاهش یافته اما تمامی معیارها و دقت تست در این روش افزایش داشته است.

(14)

یک ماشین svm یک کلاسه را با داده‌های تمیز آموزش می‌دهیم. از این ماشین برای پیش‌بینی اینکه نقاط جدید پرت هستند یا نه استفاده می‌کنیم. معیار مقایسه نیز فاصله نقاط با بردارهای پشتیبانی آموخته شده می‌باشد. نقاطی با فاصله زیاد به احتمال زیاد پرت هستند.