به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

تمرین دوم درس یادگیری ماشین

گزارش سوال‌های پیاده سازی

استاد درس: دکتر احسان ناظرفرد

دانشجو: فاطمه غلام‌زاده

۹۹۱۳۱۰۰۳

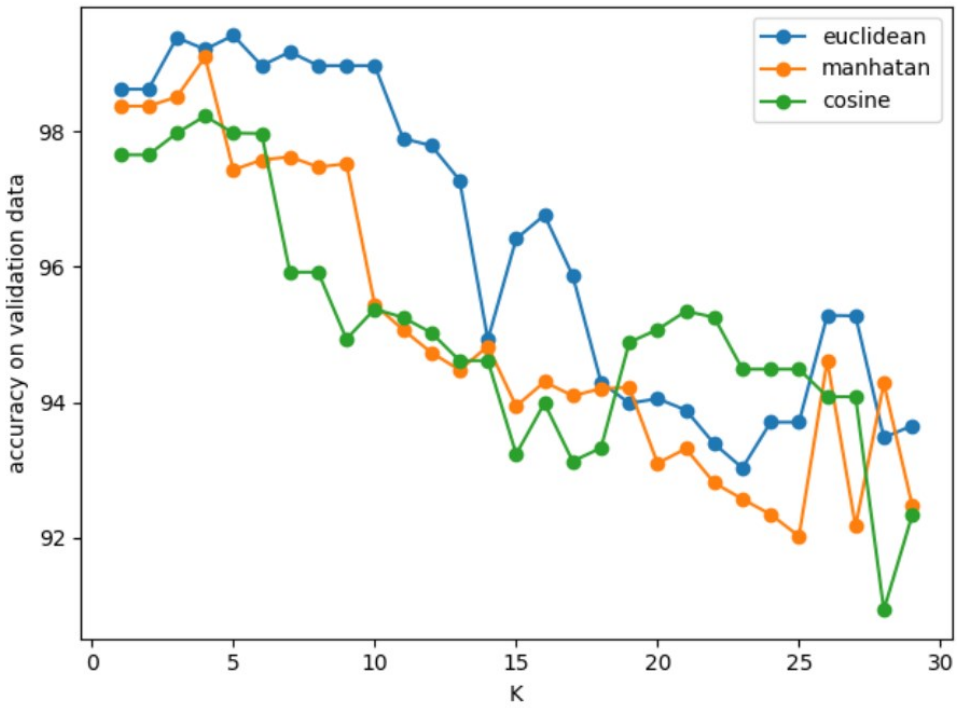
نیم سال دوم ۱۳۹۹-۱۴۰۰

سوال‌های پیاده‌سازی

سوال اول:

الف) دیتا به سه دسته آموزش، ارزیابی و تست که هر کدام به ترتیب سهم ۶۰ و ۲۰ و ۲۰ درصدی دارند تقسیم شده است.

برای یافتن بهترین مدل، آموزش و اندازه گیری دقت مدل بر روی داده های ارزیابی انجام گرفت بدین صورت که برای مقادیر k از ۱ تا ۳۰ و برای سه تابع فاصله اقلیدسی، منهتن و کسینوسی، دقت اندازه گیری شد. نمودار دقت برای kها و توابع فاصله مختلف رسم شده است :



بهترین مدل، مدلی است که از فاصله اقلیدسی و با k=5 استفاده می‌کند که دقت آن برابر ۹۹,۴۱٪ شد. بنابراین آموزش مدل را با این پارامترها انجام داده و دقت را برای داده‌های تست اندازه گیری می‌کنیم. میزان دقت بر روی داده‌های آموزش، ارزیابی و تست در جدول زیر آورده شده است:

Model parameters	Train accuracy	Validation accuracy	Test accuracy
K=4 & Euclidean	۹۸,۴۲	۹۹,۴۱	۹۸,۳۳

میزان خطای آموزش، ارزیابی و تست:

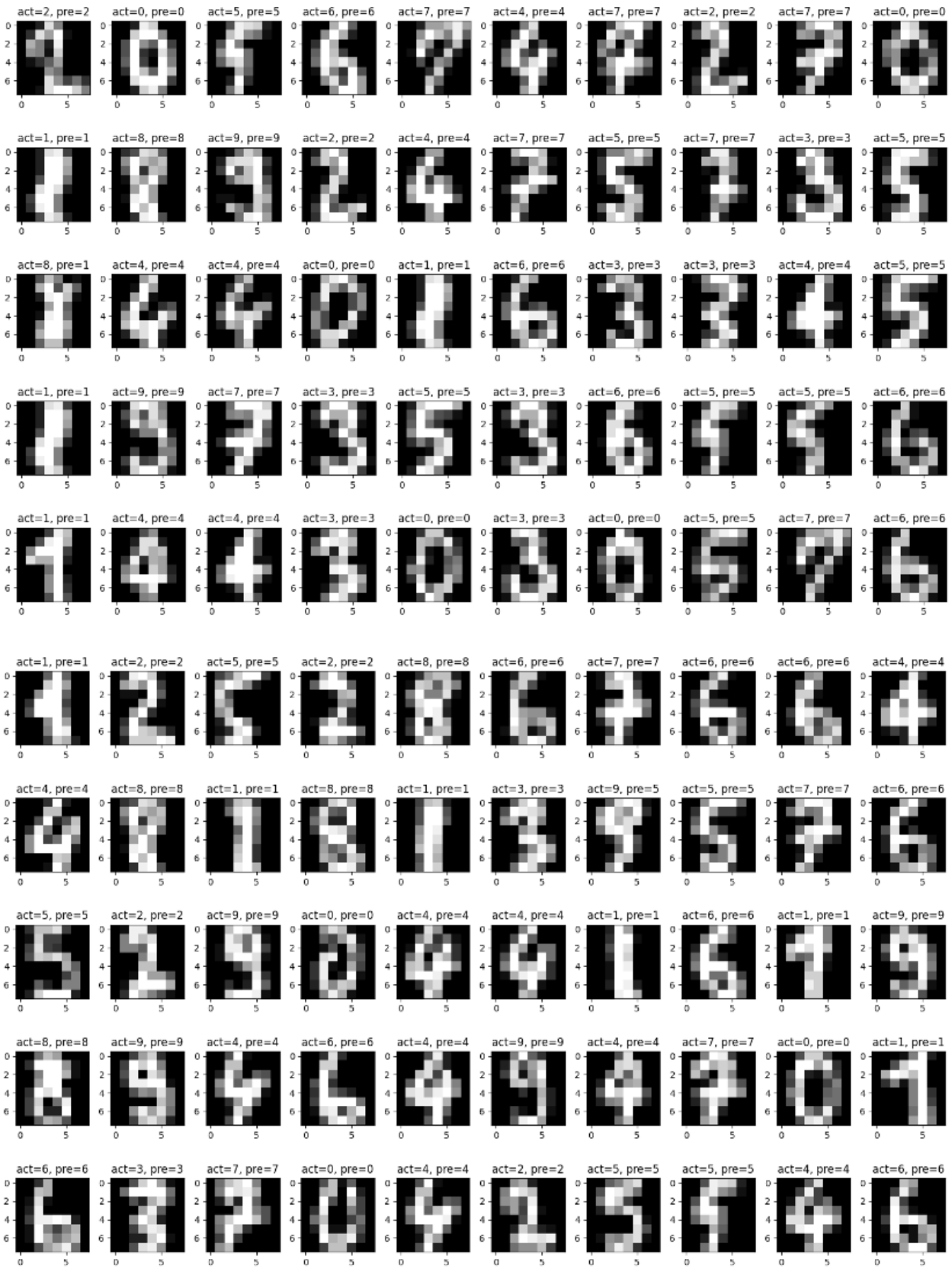
Model parameters	Train Error	Validation Error	Test Error
K=4 & Euclidean	1.58%	0.59%	1.67%

ماتریس در هم ریختگی:

Predicted Class

True Class	0	1	2	3	4	5	6	7	8	9
0	33	0	0	0	0	0	0	0	0	0
1	0	28	0	0	0	0	0	0	0	0
2	0	0	33	0	0	0	0	0	0	0
3	0	0	0	34	0	0	0	0	0	0
4	0	0	0	0	46	0	0	0	0	0
5	0	0	0	0	0	45	1	0	0	1
6	0	0	0	0	0	0	35	0	0	0
7	0	0	0	0	0	0	0	33	0	1
8	0	1	0	0	0	0	0	0	29	0
9	0	0	0	0	1	1	0	0	0	38

مقدار واقعی و مقدار پیش بینی شده برای ۱۰۰ رقم که به طور تصادفی از داده های تست انتخاب شده اند رسم شده است:


act=2, pre=2
act=0, pre=0
act=5, pre=5
act=6, pre=6
act=7, pre=7
act=4, pre=4
act=7, pre=7
act=2, pre=2
act=7, pre=7
act=0, pre=0
act=1, pre=1
act=8, pre=8
act=9, pre=9
act=2, pre=2
act=4, pre=4
act=7, pre=7
act=5, pre=5
act=7, pre=7
act=3, pre=3
act=5, pre=5
act=8, pre=1
act=4, pre=4
act=4, pre=4
act=0, pre=0
act=1, pre=1
act=6, pre=6
act=3, pre=3
act=3, pre=3
act=4, pre=4
act=5, pre=5
act=1, pre=1
act=9, pre=9
act=7, pre=7
act=3, pre=3
act=5, pre=5
act=3, pre=3
act=6, pre=6
act=5, pre=5
act=5, pre=5
act=6, pre=6
act=1, pre=1
act=4, pre=4
act=4, pre=4
act=3, pre=3
act=0, pre=0
act=3, pre=3
act=0, pre=0
act=5, pre=5
act=7, pre=7
act=6, pre=6
act=1, pre=1
act=2, pre=2
act=5, pre=5
act=2, pre=2
act=8, pre=8
act=6, pre=6
act=7, pre=7
act=6, pre=6
act=6, pre=6
act=4, pre=4
act=4, pre=4
act=8, pre=8
act=1, pre=1
act=8, pre=8
act=1, pre=1
act=3, pre=3
act=9, pre=5
act=5, pre=5
act=7, pre=7
act=6, pre=6
act=5, pre=5
act=2, pre=2
act=9, pre=9
act=0, pre=0
act=4, pre=4
act=4, pre=4
act=1, pre=1
act=6, pre=6
act=1, pre=1
act=9, pre=9
act=8, pre=8
act=9, pre=9
act=4, pre=4
act=6, pre=6
act=4, pre=4
act=9, pre=9
act=4, pre=4
act=7, pre=7
act=0, pre=0
act=1, pre=1
act=6, pre=6
act=3, pre=3
act=7, pre=7
act=0, pre=0
act=4, pre=4
act=2, pre=2
act=5, pre=5
act=5, pre=5
act=4, pre=4
act=6, pre=6

ب)

در این قسمت از کتابخانه‌ی sklearn و KNeighborsClassifier استفاده شده است.

بهترین مدل برای k=3 به دست آمد که میزان دقت آن روی داده های ارزیابی ٪۹۹,۱۷ شد.

میزان دقت :

Model parameters	Train accuracy	Validation accuracy	Test accuracy
K=3 & Euclidean	۹۸,۹۷	۹۹,۱۷	۹۸,۶۱

میزان خطای آموزش، ارزیابی و تست:

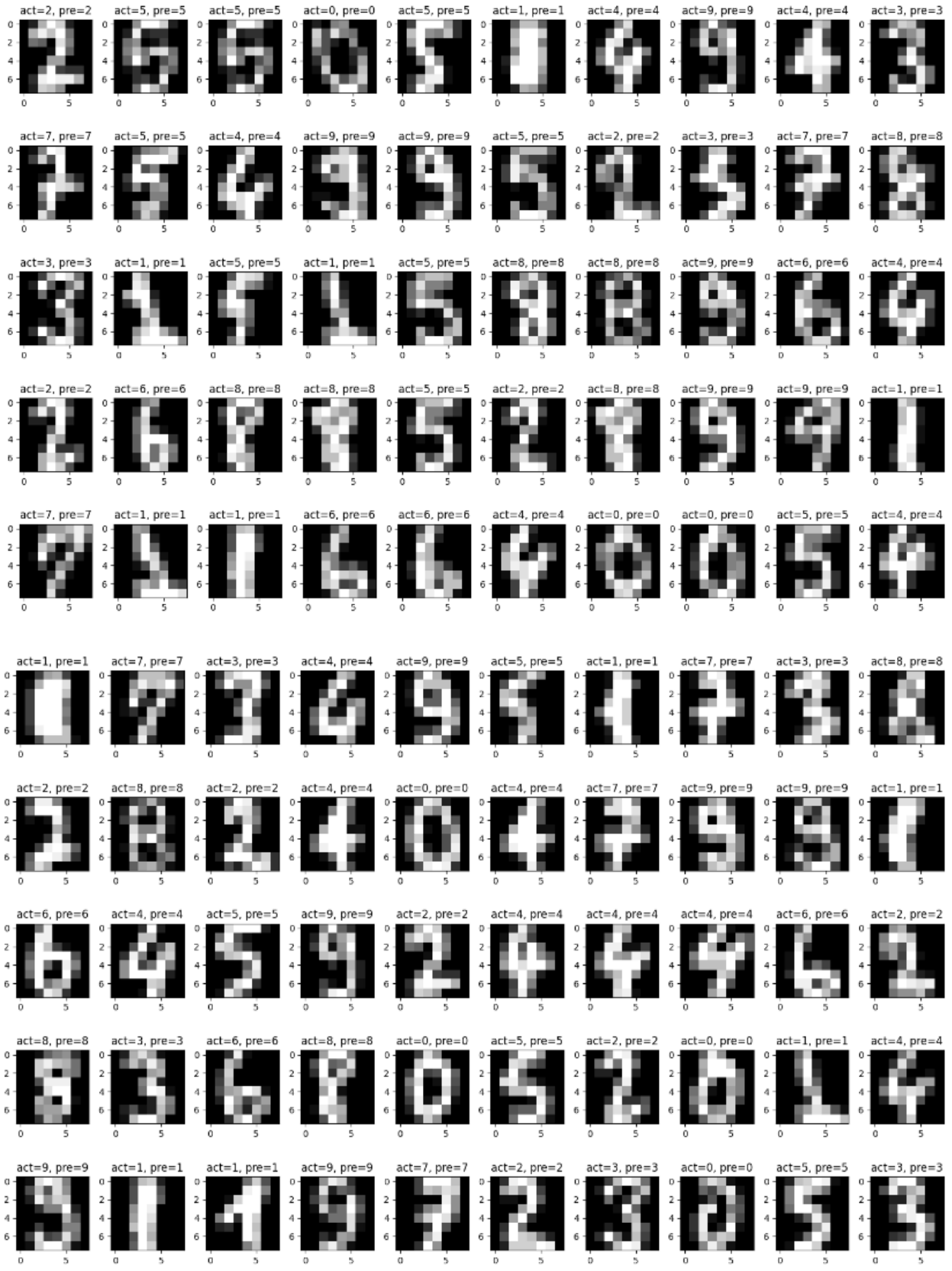
Model parameters	Train Error	Validation Error	Test Error
K=3 & Euclidean	1.03%	0.87%	1.39%

ماتریس در هم ریختگی:

Predicted Class

True Class	0	1	2	3	4	5	6	7	8	9
0	33	0	0	0	0	0	0	0	0	0
1	0	28	0	0	0	0	0	0	0	0
2	0	0	33	0	0	0	0	0	0	0
3	0	0	0	34	0	0	0	0	0	0
4	0	0	0	0	46	0	0	0	0	0
5	0	0	0	0	0	46	0	0	0	1
6	0	0	0	0	0	0	35	0	0	0
7	0	0	0	0	0	0	0	33	0	1
8	0	1	0	0	0	0	0	0	29	0
9	0	0	0	0	1	1	0	0	0	38

مقدار واقعی و مقدار پیش بینی شده برای ۱۰۰ رقم که به طور تصادفی از داده های تست انتخاب شده اند رسم شده است:


act=2, pre=2
act=5, pre=5
act=5, pre=5
act=0, pre=0
act=5, pre=5
act=1, pre=1
act=4, pre=4
act=9, pre=9
act=4, pre=4
act=3, pre=3
act=7, pre=7
act=5, pre=5
act=4, pre=4
act=9, pre=9
act=9, pre=9
act=5, pre=5
act=2, pre=2
act=3, pre=3
act=7, pre=7
act=8, pre=8
act=3, pre=3
act=1, pre=1
act=5, pre=5
act=1, pre=1
act=5, pre=5
act=8, pre=8
act=8, pre=8
act=9, pre=9
act=6, pre=6
act=4, pre=4
act=2, pre=2
act=6, pre=6
act=8, pre=8
act=8, pre=8
act=5, pre=5
act=2, pre=2
act=8, pre=8
act=9, pre=9
act=9, pre=9
act=1, pre=1
act=7, pre=7
act=1, pre=1
act=1, pre=1
act=6, pre=6
act=6, pre=6
act=4, pre=4
act=0, pre=0
act=0, pre=0
act=5, pre=5
act=4, pre=4
act=1, pre=1
act=7, pre=7
act=3, pre=3
act=4, pre=4
act=9, pre=9
act=5, pre=5
act=1, pre=1
act=7, pre=7
act=3, pre=3
act=8, pre=8
act=2, pre=2
act=8, pre=8
act=2, pre=2
act=4, pre=4
act=0, pre=0
act=4, pre=4
act=7, pre=7
act=9, pre=9
act=9, pre=9
act=1, pre=1
act=6, pre=6
act=4, pre=4
act=5, pre=5
act=9, pre=9
act=2, pre=2
act=4, pre=4
act=4, pre=4
act=4, pre=4
act=6, pre=6
act=2, pre=2
act=8, pre=8
act=3, pre=3
act=6, pre=6
act=8, pre=8
act=0, pre=0
act=5, pre=5
act=2, pre=2
act=0, pre=0
act=1, pre=1
act=4, pre=4
act=9, pre=9
act=1, pre=1
act=1, pre=1
act=9, pre=9
act=7, pre=7
act=2, pre=2
act=3, pre=3
act=0, pre=0
act=5, pre=5
act=3, pre=3

مقایسه نتایج قسمت الف و ب:

همانطور که مشاهده می‌شود با استفاده از کتابخانه آماده خطای آموزش و تست مقدار کمی کاهش پیدا کرده است. خطای آموزش از 1.58% به 1.03% و خطای تست از 1.67% به 1.39% کاهش یافته است.

با مقایسه ماتریس‌های درهم‌ریختگی هم می توان مشاهده کرد که با استفاده از کتابخانه آماده، تعداد داده های درست دسته بندی شده فقط ۱ مورد افزایش یافته است.

سوال دوم:

الف)

پیش پردازش: ابتدا داده ها را شافل می کنیم چون کلاس ها به ترتیب در داده ها ظاهر شده اند و این موجب کاهش دقت دسته بند می شود. هم چنین برخی از ستون های داده ها را استاندارد می کنیم.

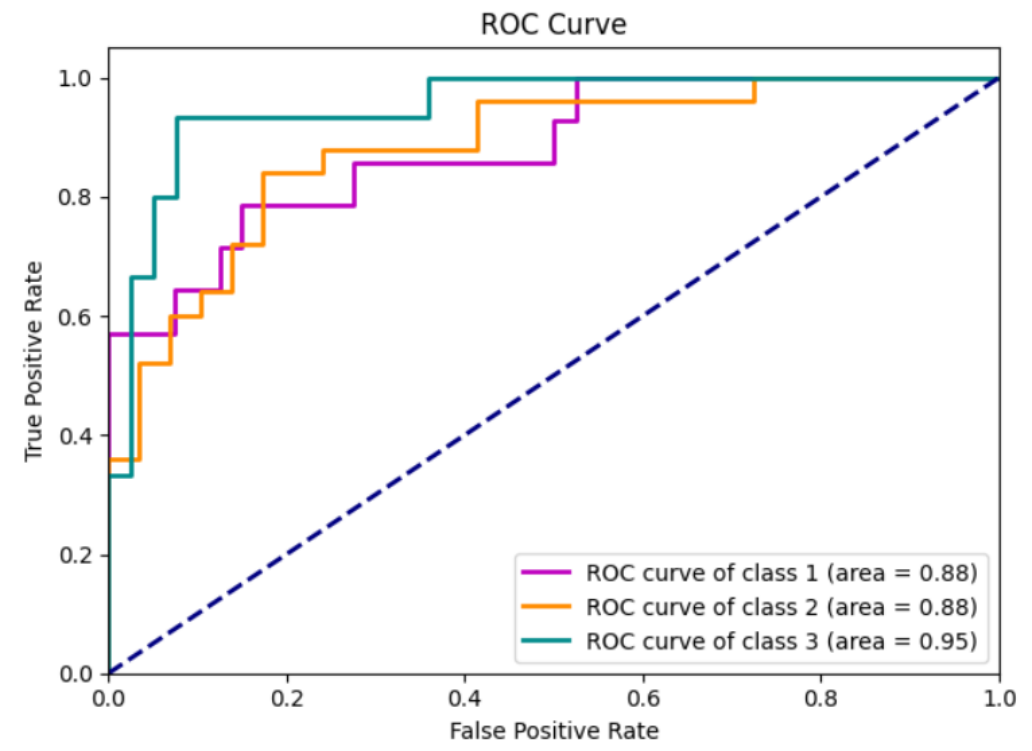
دقت به دست آمده با روش 6-fold cross validation : ٪۹۳,۴۵

ب)

در این قسمت ۷۰ درصد از داده ها را برای آموزش و ۳۰٪ را برای تست در نظر گرفتیم.

دقت به دست آمده : ۹۴,۳۳

نمودار ROC :



تحلیل نمودار ROC:

هر چقدر نمودار ROC از خط y=x فاصله بیشتری داشته باشد بهتر است، یعنی هر چقدر مساحت زیر منحنی ROC بیشتر باشد دقت بالاتر است، با توجه به نمودار بالا می‌توان مشاهده کرد که دسته بندی برای کلاس ۳ با دقت بیشتری انجام شده چون مساحت زیر منحنی ROC برای آن بزرگتر است.

سوال سوم:

الف)

در این قسمت ۷۰٪ از داده‌ها برای آموزش و ۳۰٪ برای تست در نظر گرفته شده است. آموزش به این صورت است که ۱۰ تا دسته بند (به ازای هر رقم یک دسته بند) آموزش می‌دهیم. برای آموزش هر دسته بند برچسب داده‌ی ترین مخصوص خود آن دسته بند تولید می کنیم بدین صورت که اگر برچسب داده ترین برابر آن رقم بود آن را ۱ می کنیم و اگر غیر از آن رقم بود آن را ۰ می کنیم. بدین صورت برای هر رقم یک دسته بند باینری آموزش می دهیم. وقتی یک داده تست می آید هر ۱۰ تا دسته بند را بر روی آن اعمال می کنیم و احتمال تعلق آن برای هر دسته که بیشتر بود نتیجه می گیریم رقم مربوط به آن دسته است.

دقت مدل بر روی داده‌های تست: ۹۵,۵۵

خطای آموزش و تست:

Model	Train Error	Test Error
Logistic Regression	0.4 %	4.44%

ماتریس در هم ریختگی:

Predicted Class

True Class	0	1	2	3	4	5	6	7	8	9
0	53	0	0	0	0	0	0	0	0	0
1	0	42	1	0	0	0	0	0	7	0
2	0	0	47	0	0	0	0	0	0	0
3	0	0	0	52	0	1	0	0	1	0
4	0	1	0	0	59	0	0	0	0	0
5	0	1	1	0	0	62	0	1	1	0
6	0	0	0	0	0	1	52	0	0	0
7	0	0	0	0	0	0	0	54	0	1
8	0	1	0	0	0	1	0	0	41	0
9	0	0	0	0	0	0	0	0	5	54

ب)

مقدار واقعی و مقدار پیش بینی شده برای ۲۵ رقم که به طور تصادفی از داده های تست انتخاب شده اند رسم شده است:

ج)

مقایسه عملکرد KNN با Logistic Regression :

مرحله training در KNN بسیار سریع‌تر از logistic Regression است اما مرحله test در KNN بسیار کندتر است زیرا KNN یک lazy learner است و بیشتر محاسبات مربوط به دسته‌بندی را در زمانی که داده تست به آن داده می‌شود انجام می‌گیرد.

دقت به دست آمده در روش KNN از روش Logistic Regression بیشتر است.

د)

توضیح یادگیری نامتوازن:

به این معناست که فرکانس کلاس‌های هدف در داده‌ها به طرز چشمگیری نامتوازن است. به عنوان مثال وقوع یکی از کلاس‌ها در داده‌ها در مقایسه با سایر کلاس‌ها بسیار بیشتر است و الگوریتم دسته‌بندی تمایل پیدا می‌کند که داده‌های تست را در کلاسی که تکرار بالایی دارد جای دهد. فرض کنید یک مسئله دسته بندی دو کلاسه با کلاس های A وB داریم. فرض کنید داده‌های آموزشی به گونه ای است که ۹۵٪ آن متعلق به کلاس A وفقط 5% آن متعلق به کلاس B است. وقتی مدل را روی این داده‌های آموزشی، آموزش بدهیم اگر حتی همه داده‌ها را در کلاس A دسته بندی کند دقت مدل 95% است و دقت قابل قبولی است اما داده‌های کلاس B نادیده گرفته شده اند! به بیان دیگر تعداد سمپل‌های کلاس B به اندازه ای کافی نیستند که مدل بتواند بفهمد چه زمانی یک داده را در کلاس B جای دهد. حالا فرض کنید در یک مسئله دسته‌بندی ۵ کلاس دارید که در متوازن‌ترین حالت ممکن هر کلاس دارای ۲۰٪ از داده ها است و اگر از روش one-vs-all استفاده کنیم در هر بار استفاده از مدل، یک توزیع ۲۰-۸۰ درصدی از داده‌ها داریم و یک کلاس بندی نامتوازن است. (البته به بدی مثال اول نیست اما باز هم نامتوازن است) اگر تعداد کلاس‌ها بیشتر باشد این مشکل نامتوازن بودن هم افزایش می‌یابد. به این مسئله، یادگیری متوازن گفته می‌شود که یکی از مشکلات روش one-vs-all است.

راه حل:

۱- یک راه برای حل این نسبت دادن وزن‌هایی به کلاس اقلیت و اکثریت است. بدین صورت که اهمیت کلاس اقلیت را زیادتر و اهمیت کلاس اکثریت را کمتر جلوه دهد و فرآیند آموزش مدل با توجه به این وزن‌دهی صورت گیرد.

۲- Random Under-Sampling: به طور رندوم بعضی از سمپل‌های کلاسی که اکثریت دارد را حذف کنیم تا جایی که توازن میان کلاس‌ها برقرار شود.

۳- Random Over-Sampling: افزایش تعداد داده‌های کلاس اقلیت با جایگزین کردن و تکرار آن‌ها به صورت رندوم.

۴- Cluster-Based Over Sampling: الگوریتم خوشه‌بندی k-means به طور مجزا بر روی داده‌های کلاس اقلیت و اکثریت اجرا می‌شود تا خوشه‌های موجود در دیتاست مشخص شوند. پس از آن هر خوشه به گونه‌ای oversample می‌شود که تمام خوشه‌هایی که کلاس یکسان دارند دارای تعداد یکسانی داده باشند و سایز همه کلاس‌ها هم باهم برابر باشد.