

به نام خدا

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر

پاسخ تمرین سری سوم یادگیری ماشین

استاد:

دکتر احسان ناظر فرد

دانشجو:

حلیمه رحیمی

شماره دانشجویی:

۹۹۱۳۱۰۴۳

زمستان ۱۳۹۹

۱- توضیح دهید که عمل smoothing در بیز ساده چیست و به چه منظور انجام می پذیرد؟

در برخی مواقع به ازای مقدار خاصی از یک ویژگی، هیچ داده ای متعلق به کلاس بخصوصی وجود ندارد. این به این معناست که احتمالاً داده کافی نداشته ایم. همچنین این مقدار صفر، باعث می شود تمام احتمالات ویژگی های دیگر نادیده گرفته شود و احتمال آن کلاس، صفر حاصل شود. بنابراین برای اینکه از این حالت جلوگیری شود، مطابق زیر مقداری را به فرمول محاسبه ی احتمال اضافه می کنیم. به عبارتی در اینجا نمونه (یا نمونه های) خیالی را به منظور از بین بردن احتمال صفر اضافه می کنیم. فرمول زیر از منبع معرفی شده در سوال دوم می باشد.

$$\hat{\theta}_{ijk} = \hat{P}(X_i = x_{ij} | Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\} + I}{\#D\{Y = y_k\} + IJ}$$

در اینجا I مقدار لاپلاس و J تعداد مقادیر مختلفی که ویژگی مورد نظر می تواند بگیرد می باشد.

در مورد این مسئله، در برخی منابع مقدار لاپلاس را فقط برای مقدار خاص یک ویژگی که موجب صفر شدن می شد، اضافه می کردند و در برخی منابع دیگر به همه اضافه می کردند. در حل تمرین ها، آن را به همه اعمال کرده ام.

۲- با استفاده از مراجع ۱ و ۲ و سایر مراجع، دسته بندهای بیز ساده و رگرسیون لاجیستیک را با هم مقایسه کنید.

در دسته بند بیز ساده از $P(Y)$ و $P(X|Y)$ برای رسیدن به $P(Y|X)$ استفاده می شود، درحالیکه در رگرسیون لاجیستیک به طور مستقیم به پیشبینی Y می رسم و یا به عبارتی به طور مستقیم از $P(Y|X)$ برای کلاس بندی استفاده می کنیم. بنابراین دست بند بیز Generative و رگرسیون لاجیستیک Discriminative می باشد.

با توجه به منابع معرفی شده، بیز ساده و رگرسیون لاجیستیک با داشتن تعداد داده های بسیار زیاد مانند یکدیگر عمل می کنند. در واقع رگرسیون لاجیستیک مشابه حالت خاصی از بیز ساده گاوسی است و در صورتی که این حالت خاص پیش نیامده باشد، این دو دسته بند مشابه عمل نمی کنند.

بیز ساده با سرعت بیشتری (با تعداد داده $\log n$) به تقریب مناسب دست پیدا می کند و رگرسیون لاجیستیک با سرعت پایین تر (با تعداد داده n). این مسئله باعث می شود به این نتیجه برسیم که در صورت داشتن تعداد داده های کمتر بهتر است از بیز ساده استفاده شود. همچنین از آن جهت که در بلند مدت (با تعداد داده های بیشتر) رگرسیون لاجیستیک نتیجه ای بهتر از بیز ساده می دهد، در صورت داشتن مجموعه داده ی بزرگتر، بهتر است از رگرسیون لاجیستیک استفاده شود.

طبق مسائل بالا، با اینکه به نظر می رسد رگرسیون لاجیستیک هم $P(X)$ ها را از هم مستقل در نظر بگیرد، در حقیقت اینطور نیست. بنابراین وقتی داده ای داریم که از قانون استقلال بیز ساده تبعیت نمی کند، رگرسیون لاجیستیک به راحتی اثر این داده را بر نتیجه ی خود اعمال می کند درحالیکه بیز ساده این طور نیست و استقلال را به طور پیش فرض ایجاد می کند.

۳- برای هر کدام از داده های زیر که دارای دو ویژگی با مقدار حقیقی X_1 و X_2 هستند یک دسته بند بیز ساده گاوسی را آموزش داده ایم. با ذکر دلیل تعیین کنید که برچسب داده تست (که با علامت سوال مشخص شده است) چه خواهد بود؟

با توجه به تصویر می توان گفت به طور تقریبی میانگین متغیر دوم هر دو کلاس برابر با مقدار متغیر دوم داده ی تست می باشد، بنابراین در هر جایی از محاسبه مرز تصمیم با مشاهده ی تفاضل این اعداد با یکدیگر، عبارت صفر می شود. می توان گفت برای تصمیم گیری می توان تنها به میانگین متغیر اول و ماتریس کوواریانس کلاس ها توجه کرد.

در صورتی که ماتریس کواریانس دو کلاس مشابه می بود، این دو کلاس با یک خط بین دو مجموعه داده A و B از یکدیگر جدا می شدند اما با توجه به تعداد داده معدودی که مشاهده می کنیم، این ماتریس در دو کلاس مشابه یکدیگر نیست. از تصویر بر می آید دو متغیر از یکدیگر مستقل اند، واریانس متغیر اول کلاس A بسیار کوچک و نزدیک به صفر می باشد و همچنین احتمال پیشین کلاس B از A بیشتر است.

طبق آنچه گفته شد، بنظر می رسد داده تست جزو کلاس B باشد، از آن جهت که مقدار $P(Z=X_1)$ برای این دو کلاس تقریباً نزدیک به هم می باشد ولی با بالاتر بودن احتمال پیشین B، این کلاس را پاسخ سوال در نظر می گیریم.

باید بگوییم نتیجه یک hyperbola خواهد بود که کلاس B را از A جدا کند و داده ی تست در بخش B قرار گیرد.

۴- احتمال $P(B|D=T)$ را در شبکه بیزین زیر حساب کنید.

$$P(B = T|D = T) = \frac{P(B = T, D = T)}{P(D = T)}$$

$$P(B = F|D = T) = \frac{P(B = F, D = T)}{P(D = T)} = 1 - P(B = T|D = T)$$

$$\begin{aligned} P(D = T) &= P(A)P(B|A)[P(C|A)P(D|B, C) + P(\neg C|A)P(D|B, \neg C)] \\ &\quad + P(\neg A)P(B|\neg A)[P(C|\neg A)P(D|B, C) + P(\neg C|\neg A)P(D|B, \neg C)] \\ &\quad + P(A)P(\neg B|A)[P(C|A)P(D|\neg B, C) + P(\neg C|A)P(D|\neg B, \neg C)] \\ &\quad + P(\neg A)P(\neg B|\neg A)[P(C|\neg A)P(D|\neg B, C) + P(\neg C|\neg A)P(D|\neg B, \neg C)] \end{aligned}$$

$$\begin{aligned} P(D = T) &= 0.2 \times 0.37[0.3 \times 0.5 + 0.7 \times 0.15] + 0.8 \times 0.21[0.25 \times 0.5 + 0.75 \times 0.15] + 0.2 \\ &\quad \times 0.63[0.3 \times 0.67 + 0.7 \times 0.95] + 0.8 \times 0.79[0.25 \times 0.67 + 0.75 \times 0.95] \\ &= 0.724046 \end{aligned}$$

$$\begin{aligned} P(B = T, D = T) &= P(A)P(B|A)[P(C|A)P(D|B, C) + P(\neg C|A)P(D|B, \neg C)] \\ &\quad + P(\neg A)P(B|\neg A)[P(C|\neg A)P(D|B, C) + P(\neg C|\neg A)P(D|B, \neg C)] \end{aligned}$$

$$\begin{aligned} P(B = T, D = T) &= 0.2 \times 0.37[0.3 \times 0.5 + 0.7 \times 0.15] + 0.8 \times 0.21[0.25 \times 0.5 + 0.75 \times 0.15] \\ &= 0.05877 \end{aligned}$$

$$P(B = T|D = T) = \frac{0.05877}{0.724046} = 0.08169$$

$$P(B = F|D = T) = 1 - 0.08169 = 0.91831$$

۵- نحوه انتخاب نقطه cut-off در یک مدل رگرسیون لاجیستیک را شرح دهید.

از آن جهت که با تغییر نقطه cut-off مقدار TPR و FPR تغییر می کند، می توان برای انتخاب این نقطه از محور ROC استفاده کرد و متناسب با مسئله به تصمیم گیری پرداخت. در حالت کلی ما در پی نقطه ای هستیم که TPR بالا و FPR پایین داشته باشد. در صورتی که هزینه FP بیشتر از FN باشد، باید دید در چه نقطه ای با FPR پایین، به مقدار دلخواهی از TPR می رسیم و بالعکس در صورت بالاتر بودن هزینه FN از FP، باید دید در چه نقطه ای با TP بالا، به مقدار مناسبی از FPR می رسیم.

در صورتی که مجموعه داده هایمان imbalance باشد، بهتر است از نمودار Precision-Recall استفاده کنیم.

۶- نسبت بخت چیست؟ شرح دهید و نحوه ی استفاده آن را در رگرسیون لاجیستیک بیان کنید.

نسبت بخت، نسبت یک اتفاق به اتفاق دیگر را نشان می دهد. Odds در اینجا به این معنی است که چقدر محتمل است پاسخ کلاس مورد نظر باشد تا آنکه نباشد؛ به عبارتی نسبت پیروزی به شکست خواهد بود. Odds Ratio نسبت این مقدار برای یک اتفاق را با اتفاق دیگر می سنجد.

آنچه در اینجا خواهیم داشت میزان تاثیرگذاری هر ویژگی بر انتخاب یک کلاس است. نمی توان این مسئله را با استفاده از احتمال تنها با یک عدد نمایش داد در حالیکه Odds Ratio با یک عدد این تاثیر را به ما نمایش می دهد. در صورتی که این مقدار برای یک کلاس بیشتر از دیگری باشد، می توان گفت احتمال اینکه داده به آن کلاس تعلق داشته باشد بیشتر است.

در رگرسیون لاجیستیک به عمل لگاریتم گرفتن از Odds Ratio، Logit گفته می شود. یکی از دلایل این عمل این است که نتیجه بین منفی بی نهایت تا مثبت بی نهایت خواهد بود و بنابراین مدل سازی بر اساس آن آسان تر از مدل سازی براساس مقدار احتمال که بین صفر و یک است می باشد. دلیل دیگر آن است که تفسیر آن آسان تر از راه های نگاشت دیگر می باشد.

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k.$$

سپس به تخمین β ها پرداخته می شود. در نهایت احتمال مثبت بودن کلاس، برابر با فرمول زیر خواهد بود.

$$p = \frac{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)}.$$

۷- داده های آموزشی زیر که مربوط به افراد مختلفی است را در اختیار داریم. ستون Buy مشخص می کند که آیا فرد مورد نظر یک جنس مشخص (مثلا کامپیوتر) را خریداری می کند یا خیر. با استفاده از دسته بند بیز ساده مشخص کنید که آیا افراد با مشخصات زیر، جنس مورد نظر را خریداری می کنند یا خیر؟

$X_1 = (\text{age} = \text{youth}, \text{income} = \text{high}, \text{student} = \text{yes}, \text{credit} = \text{fair})$

$X_2 = (\text{age} = \text{senior}, \text{income} = \text{low}, \text{student} = \text{no}, \text{credit} = \text{excellent})$

$X_3 = (\text{age} = \text{middle-aged}, \text{income} = \text{medium}, \text{student} = \text{no}, \text{credit} = \text{fair})$

$$P(Y = +) = \frac{9}{14}, \quad P(Y = -) = \frac{5}{14}$$

		Buy	
		+	-
Age	Youth	2/9 → 3/12	3/5 → 4/8
	Middle-aged	4/9 → 5/12	0/5 → 1/8
	Senior	3/9 → 4/12	2/5 → 3/8
Income	Low	3/9 → 4/12	1/5 → 2/8
	Medium	4/9 → 5/12	2/5 → 3/8
	High	2/9 → 3/12	2/5 → 3/8
Student	Yes	6/9 → 7/11	1/5 → 2/7
	No	3/9 → 4/11	4/5 → 5/7
Credit	Fair	6/9 → 7/11	2/5 → 3/7
	Excellent	3/9 → 4/11	3/5 → 4/7

$$P(Y = +|X_1) = P(\text{youth}|+)P(\text{high}|+)P(\text{yes}|+)P(\text{fair}|+)P(+)$$

$$= \frac{3}{12} \times \frac{3}{12} \times \frac{7}{11} \times \frac{7}{11} \times \frac{9}{14} = 0.0163$$

$$P(Y = -|X_1) = P(\text{youth}|-)P(\text{high}|-)P(\text{yes}|-)P(\text{fair}|-)P(-) = \frac{4}{8} \times \frac{3}{8} \times \frac{2}{7} \times \frac{3}{7} \times \frac{5}{14}$$

$$= 0.0057$$

$$P(Y = +|X_1) > P(Y = -|X_1)$$

$$P(Y = +|X_2) = P(\text{senior}|+)P(\text{low}|+)P(\text{no}|+)P(\text{excellent}|+)P(+)$$

$$= \frac{4}{12} \times \frac{4}{12} \times \frac{4}{11} \times \frac{4}{11} \times \frac{9}{14} = 0.0082$$

$$P(Y = -|X_2) = P(\text{senior}|-)P(\text{low}|-)P(\text{no}|-)P(\text{excellent}|-)P(-) = \frac{3}{8} \times \frac{2}{8} \times \frac{5}{7} \times \frac{4}{7} \times \frac{5}{14}$$

$$= 0.0137$$

$$P(Y = +|X_2) < P(Y = -|X_2)$$

$$P(Y = +|X_3) = P(\text{middle}|+)P(\text{medium}|+)P(\text{no}|+)P(\text{fair}|+)P(+)$$

$$= \frac{5}{12} \times \frac{5}{12} \times \frac{4}{11} \times \frac{7}{11} \times \frac{9}{14} = 0.0258$$

$$P(Y = -|X_3) = P(middle|-)P(medium| -)P(no| -)P(fair|-)P(-) = \frac{1}{8} \times \frac{3}{8} \times \frac{5}{7} \times \frac{3}{7} \times \frac{5}{14}$$

$$= 0.0051$$

$$P(Y = +|X_3) > P(Y = -|X_3)$$

سوالات پیاده سازی

۱- مجموعه داده ی Car Evaluation را دانلود کنید.

این مجموعه داده را در برنامه ی خود بارگزاری کرده و به دو قسمت آموزش و آزمون تقسیم کنید.

مجموعه داده شامل ستون های ['buying', 'maint', 'doors', 'persons', 'lug_boot', 'safety', 'class'] می باشد که به عنوان header در دیتافریم اضافه کردم.

پس از شافل کردن، ۷۰٪ داده ها را به عنوان داده های آموزش و ۳۰٪ باقی را به عنوان داده های آزمون قرار دادم.

تابعی بنویسید که دسته بند بیز ساده را با دریافت داده ها و پارامتر smoothing اجرا کند.

در بخش Calculating Probabilities با استفاده از تابع Probs که تعریف کردم، احتمالات شرطی و پیشین لازم را به دست آوردم. در بخش Prediction, Computing Confusion Matrix and Rates و ROC به ترتیب توابع لازم برای پیشبینی کلاس داده ی تست و احتمال تعلق به کلاس ها، جدول درهم ریختگی و نتایج خواسته شده در بخش های الف و ب سوال و نمودار ROC را تعریف کرده ام.

در بخش Answers تابع Classify با دریافت smoothing و داده ها، دسته بند را آموزش داده و نتایج آزمون را برمی گرداند.

الف) با استفاده از تابع بالا یک دسته بند بدون استفاده از smoothing ایجاد کنید.

با توجه به اینکه استفاده نکردن از smoothing مانند قراردادن مقدار صفر برای عدد smooth در فرمول است، در اینجا چنین کاری را انجام داده ام.

نتایج را در زیر می توانید ببینید.

مسلم است از آن جهت که تعداد داده های کلاس unacc بیشتر بود، مقدار احتمال پیشین برای این کلاس بیشتر از بقیه می شد و بنابراین در بسیاری از مواقع نتیجه را تحت تاثیر قرار داده و یادگیرنده کلاس را به غلط unacc تشخیص می داد. به دلیل مشابه، تشخیص صحیح این کلاس بسیار بهتر از کلاس های دیگر بود. می توانید این نکات را در جدول در هم ریختگی مشاهده کنید.

برای به دست آوردن TPR کل، آن را برای هر یک از کلاس ها به طور جداگانه به دست آورده و سپس میانگین گرفتیم. همین عمل را برای به دست آوردن سایر نتایج به کار بردم.

TPR و FPR تحت تاثیر آنچه بالاتر بیان شد قرار گرفته اند و به همین دلیل نتایج به این شکل درآمده اند. در صورت استفاده از میانگین وزنی، مسلماً نتایج بسیار بهتر می شد.

نتایج برای مجموعه آموزش:

		Predicted Class			
		ACC	GOOD	UNACC	VGOOD
Actual Class	ACC	55	3	223	0
	GOOD	12	8	28	1
	UNACC	40	9	780	4
	VGOOD	21	6	15	4

TPR	0.34558147876201384
FPR	0.20148176728317296
TNR	0.798518232716827
FNR	0.6544185212379862

نتایج برای مجموعه آزمون:

		Predicted Class			
		ACC	GOOD	UNACC	VGOOD
Actual Class	ACC	13	0	90	0
	GOOD	3	4	9	4
	UNACC	18	3	355	1
	VGOOD	9	3	6	1

TPR	0.33012243337864894
FPR	0.20839401310747172
TNR	0.7916059868925283
FNR	0.6698775666213511

ب) با استفاده از تابع بالا یک دسته بند با استفاده از **smoothing** ایجاد کنید.

به علت احتمال پیشین بسیار بالای یکی از کلاس ها، smoothing با مقدار ۱ اثر چندانی نگذاشته است. با تغییر این مقدار نتایج تغییر بیشتری دارند اما این تغییرات باعث بدتر شدن TPR برای کلاس Unacc می شود که با توجه به تعداد بالای داده های آن، اثر نامناسبی بر نتایج می گذارد.

نتایج برای مجموعه آموزش:

		Predicted Class			
		ACC	GOOD	UNACC	VGOOD
Actual Class	ACC	53	2	226	0
	GOOD	12	6	30	1
	UNACC	39	6	784	4
	VGOOD	21	3	19	3

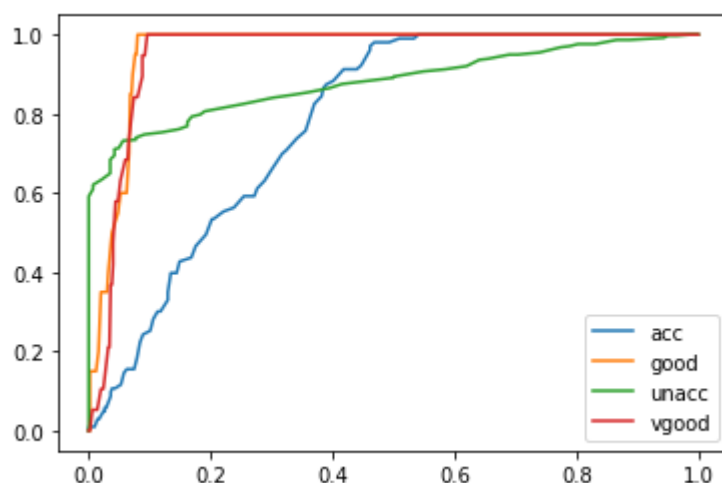
TPR	0.32936373528213697
FPR	0.20568779259498515
TNR	0.7943122074050148
FNR	0.670636264717863

نتایج برای مجموعه آزمون:

		Predicted Class			
		ACC	GOOD	UNACC	VGOOD
Actual Class	ACC	11	0	92	0
	GOOD	3	0	14	3
	UNACC	17	0	360	0
	VGOOD	9	1	8	1

TPR	0.27858371431398404
FPR	0.2201331119715053
TNR	0.7798668880284947
FNR	0.721416285686016

ج) برای مجموعه ی آزمون نمودار ROC را برای مدل آموزش داده شده ی قسمت الف رسم کنید و نتیجه را تحلیل کنید.



با توجه به آنکه کلاس **acc** و **unacc** شامل تعداد داده ی بیشتری می شوند، می توان درک کرد که چرا ROC این دو کلاس دچار تغییرات آهسته تری هستند. **good** و **vgood** شامل تعداد کمتری از داده هستند و مقدار احتمال پیشینشان از دو کلاس دیگر کوچکتر است و بنابراین مقدار احتمال برای انتخاب آن کلاس ها به عنوان کلاس پیشبینی شده، کم است و به سرعت به نقطه ای می رسند که **threshold** مرتبط با آن نقطه تمام داده های آن کلاس ها را درست تشخیص می دهد و بنابراین **TPR** برای این کلاس ها سریعتر به یک می رسد. حال آنکه دو کلاس با تعداد داده ی بیشتر سرعت کمتری در رسیدن به **TPR** برابر یک دارند.

Unacc به علت احتمال پیشین بالاتر از بقیه با سرعت بیشتری به **TPR** یکسان (حدود ۰/۶) می رسد و پس از آنکه دو کلاس با کمترین تعداد داده و احتمال پیشین کم به یک رسیدند، سرعت رشد آن کاهش می یابد، از آن جهت که تعداد داده های آن بسیار بیشتر از بقیه است. این وضعیت با توجه به آنچه در قسمت الف در مورد کلاس ها گفتیم قابل توجیه و منطقی است.

کلاس **acc** نیز به علت آنکه در شرایطی میانه ی این کلاس ها قرار دارد، سریعتر از **Unacc** و دیرتر از دو کلاس دیگر به **TPR** برابر یک می رسد.

۲- مجموعه داده ی **MNIST** را دانلود کنید.

برنامه ای بنویسید که با استفاده از روش **One-vs-All** داده ها را دسته بندی کند.

برای خواندن داده ها از کتابخانه ی `mlxtend` استفاده کردم.^۱ برای پیاده سازی `One-vs-All` نیز هر بار به ازای هر یک از کلاس ها، برچسب باقی کلاس ها را ۱- در نظر گرفته و با استفاده از `LogisticRegression` خطی کتابخانه `SKLearn` به آموزش و سپس دریافت مقدار احتمال پیشبینی هر کلاس پرداختم.

مقدار احتمال به دست آمده برای هر یک از کلاس ها را در ماتریسی از مقادیر قرار دادم و برای تعیین پیشبینی نهایی، کلاس مرتبط با بیشترین مقدار را به عنوان برچسب نهایی در نظر گرفتم.

خطا را برای هر کلاس محاسبه کرده و سپس میانگین را به دست آوردم. همانطور که قابل مشاهده است، خطا برای مجموعه آزمون کمی بیشتر از خطا برای مجموعه آموزش شده است.

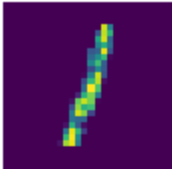







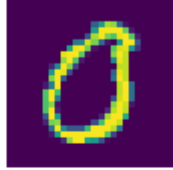



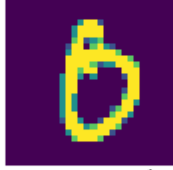
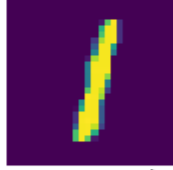
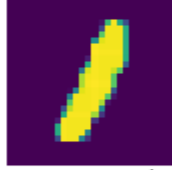


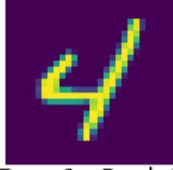






الف) دسته بند را آموزش دهید و خطای آموزش و آزمون را به همراه ماتریس درهم ریختگی گزارش کنید.

```
Confusion Matrix for Training Data:
[[5796  1  34  22  6  50  29  13  41  26]
 [  1 6590  42  20  24  18  10  19 107  20]
 [ 13  34 5424 146  28  35  33  62  71  23]
 [ 10  16  81 5508  7 188  1  14 143  91]
 [  6  8  51  7 5469  48  22  47  29 170]
 [ 18  14  18 148 11 4704  72  6 159  36]
 [ 28  6  59  22  35  97 5705  4  40  3]
 [  4  9  55  47  16  15  2 5875  26 171]
 [ 41  54 181 148  59 193  41  29 5153  64]
 [  6  10  13  63 187  73  3 196  82 5345]]
Error for Each Class, Training Data: [0.00581667 0.00688333 0.01631667 0.01956667 0.01268333 0.01998333
 0.00845  0.01225  0.02513333 0.02061667]
Mean of Error, Test Data: 0.01477
```

```
Confusion Matrix for Test Data:
[[ 957  0  8  3  1 11  7  3  9  9]
 [  0 1116 12  0  2  2  3  6 14  6]
 [  0  3 905 19  4  1  7 24  7  2]
 [  4  1 18 915  3 34  2  4 22 13]
 [  0  0  9  2 910 10  4  7 11 30]
 [  3  1  5 22  0 762 17  1 27  4]
 [  6  4 10  5 12 16 909  1  7  0]
 [  2  1 11 11  2  7  1 945 12 24]
 [  6  8 51 25 10 40  8  5 854 16]
 [  2  1  3  8 38  9  0 32 11 905]]
Error for Each Class, Test Data: [0.0074 0.0064 0.0194 0.0196 0.0145 0.021 0.011 0.0154 0.0289 0.0208]
Mean of Error, Test Data: 0.01644
```

ب) ۲۵ داده از مجموعه ی تست به صورت تصادفی انتخاب کرده و برای هر داده، کلاس واقعی و کلاس پیش بینی شده توسط مدل آموزش داده شده را در تصویری گزارش کنید.

¹ http://rasbt.github.io/mlxtend/user_guide/data/loadlocal_mnist/

True: 1 , Pred: 1	True: 6 , Pred: 6	True: 8 , Pred: 8	True: 2 , Pred: 2	True: 3 , Pred: 3
				
True: 9 , Pred: 9	True: 2 , Pred: 8	True: 9 , Pred: 9	True: 0 , Pred: 0	True: 4 , Pred: 6
				
True: 1 , Pred: 1	True: 5 , Pred: 5	True: 0 , Pred: 0	True: 1 , Pred: 1	True: 1 , Pred: 1
				
True: 3 , Pred: 3	True: 9 , Pred: 9	True: 4 , Pred: 4	True: 3 , Pred: 3	True: 4 , Pred: 4
				
True: 3 , Pred: 3	True: 7 , Pred: 7	True: 1 , Pred: 1	True: 3 , Pred: 3	True: 0 , Pred: 0
				

ج) عملکرد این روش را با روش K-نزدیکترین همسایه مقایسه کرده و توضیح دهید هر کدام از این روش ها برای چه مجموعه داده ای مناسب تر است.

KNN یک مدل غیرپارامتریک و تنبل است، در حالیکه رگرسیون لاجستیک مدلی پارامتریک می باشد. KNN نسبت به Logistic Regression کندتر عمل می کند و همچنین در مسائل دسته بندی، تنها دسته را مشخص می کند و احتمال تعلق به هر یک از کلاس ها را به ما نمی دهد.

با توجه به این مطالب، بهتر است در صورت وجود تعداد زیاد داده، از KNN استفاده نشود.

در این تمرین، عملکرد KNN به خوبی Logistic Regression نبود و مقدار خطا حدود 0.0295 برای $k=3$ بر می گرداند. برای $k=1,5,7,9$ نتایج بدتر از این می شد. به ازای K های دیگر امتحان نکردم.