به نام خدا

تکلیف سوم- یادگیری ماشین کاربردی

استاد مربوطه: دکتر ناظرفرد

نام: زهرا اخلاقی

شماره دانشجویی: ۴۰۱۱۳۱۰۶۴

ایمیل: zahra.akhlaghi@aut.ac.ir

بهار ۱۴۰۲

فهرست مطالب

Q1

(a

بله، امکان استفاده از آن وجود دارد. این کار را میتوان با استفاده از softmax انجام داد، درواقع softmax تعمیم رگرسیون لجستیک است که برای طبقه بندی K کلاس های مختلف استفاده می شود.

تابع softmax، به عنوان ورودی یک بردار z از k عدد حقیقی را می‌گیرد و آن را به یک توزیع احتمال تبدیل می‌کند که متشکل از k احتمال متناسب با نمای اعداد ورودی هستند. این امر به این معنی است که قبل از استفاده از تابع Softmax، بعضی از اجزای بردار z ممکن است منفی یا بیشتر از یک باشند. اما بعد از استفاده از آن، هر مولفه در بازه (0،1) قرار می‌گیرد بطوری که مجموع آن‌ها برابر با ۱ باشد. بنابراین می‌توان آنها را به عنوان مقدار احتمال تفسیر کرد. علاوه بر این، ورودی‌های با مقادیر بزرگتر، دارای احتمال بیشتر نیز خواهند بود.

در طول آموزش، مدل با به حداقل رساندن افت آنتروپی متقاطع بین توزیع احتمال پیش‌بینی‌شده و توزیع واقعی، وزن‌ها و سوگیری‌ها را برای هر کلاس یاد می‌گیرد. هنگامی که مدل آموزش داده شد، می توان از آن برای پیش بینی کلاس داده های ورودی جدید به سادگی با انتخاب کلاس با بالاترین احتمال استفاده کرد. به طور کلی، مدل رگرسیون لجستیک چند جمله ای یا رگرسیون softmax به ما اجازه می دهد تا به طور مستقیم چندین کلاس را بدون نیاز به آموزش و ترکیب چند طبقه بندی کننده باینری مدل سازی و پیش بینی کنیم.

(b

در رگرسیون لجستیک نسبت شانس، حاصل تقسیم احتمال وقوع رویداد بر احتمال رخ ندادن آن است. اگر نسبت شانس بزرگتر از 1 باشد، نشان می دهد که وجود متغیر پیش بینی کننده شانس نتیجه را افزایش می دهد، در حالی که اگر نسبت شانس کمتر از 1 باشد، نشان می دهد که وجود متغیر پیش بینی کننده شانس نتیجه را کاهش می دهد و اگر مقدار برابر با 1 باشد، نشان می دهد که هیچ تفاوتی در احتمال وقوع رویداد وجود ندارد.

نسبت شانس یک مفهوم کلیدی در رگرسیون لجستیک است و راهی برای کمی سازی ارتباط بین متغیرهای پیش بینی کننده و نتیجه ارائه می دهد. همچنین امکان تفسیر آسان ضرایب مدل را فراهم می‌کند که می‌تواند برای پیش‌بینی و درک عواملی که به نتیجه کمک می‌کنند مفید باشد.

(c

الگوریتم Naive Bayes یک طبقه‌بندی کننده احتمالی است که می‌تواند برای پیش‌بینی احتمال یک کلاس خاص با توجه به مجموعه‌ای از ویژگی‌های ورودی استفاده شود و این الگوریتم معمولاً برای ورودی هایی که به صورت دسته بندی هستند، استفاده می شود.

اگر ویژگی های ورودی عددی باشند، برای استفاده از الگوریتم Naive Bayes باید آنها را به تعداد محدودی از دسته های گسسته تبدیل کنیم. روش های زیر معمولا برای انجام این کار استفاده میشود:

- Binning: این روش شامل تقسیم محدوده ویژگی عددی به مجموعه ای از bin ها یا فواصل، و در نظر گرفتن هر bin به عنوان یک دسته جداگانه است.
- تقریب گاوسی: این روش شامل تقریب توزیع ویژگی عددی با استفاده از توزیع گاوسی، و سپس در نظر گرفتن میانگین و واریانس توزیع به عنوان دسته‌های جداگانه است.
- تخمین چگالی هسته: این روش شامل تخمین تابع چگالی احتمال مشخصه عددی با استفاده از تخمین چگالی هسته و سپس گسسته سازی تابع چگالی به تعداد محدودی از دسته ها است.

(d

الگوریتم Naive Bayes به دلیل سادگی و سرعت خود شناخته شده است، که آن را به گزینه ای جذاب برای مسائل طبقه بندی در جایی که داده های آموزشی کوچک یا مجموعه داده دارای تعداد زیادی ویژگی است، تبدیل میکند.
یکی از دلایلی که Naive Bayes می تواند با داده های آموزشی کوچک به خوبی کار کند این است که به داده های زیادی برای تخمین توزیع احتمال ویژگی های ورودی و برچسب های کلاس نیاز ندارد. زیرا Naive Bayes فرض می‌کند که ویژگی‌های ورودی با توجه به برچسب کلاس، به صورت شرطی مستقل هستند، که مقدار داده‌های مورد نیاز برای تخمین احتمالات شرطی را کاهش می‌دهد.
Naive Bayes همچنین می تواند با مجموعه داده های با ابعاد بالا به خوبی کار کند، زیرا از یک مدل احتمالی ساده برای نمایش توزیع مشترک ویژگی های ورودی و برچسب های کلاس استفاده می کند. که باعث می شود محاسبه احتمالات پسین برچسب های کلاس، حتی با تعداد زیادی ویژگی ورودی، نسبتا ساده و کارآمد باشد.
با این حال، توجه به این نکته مهم است که Naive Bayes می‌تواند به ویژگی‌های ورودی نامربوط یا بسیار مرتبط حساس باشد، که می‌تواند منجر به بیش از حد برازش یا عدم تناسب مدل شود. بنابراین، انتخاب دقیق و پیش پردازش ویژگی های ورودی برای جلوگیری از این مسائل مهم است.

(e

Logistic regression مرز تصمیم خطی تولید می‌کند که فضای ویژگی را به دو ناحیه مربوط به دو کلاس در مسئله طبقه‌بندی باینری جدا می‌کند. ولی، می توان آن را برای به دست آوردن یک مرز تصمیم غیر خطی با استفاده از تبدیل غیرخطی ویژگی های ورودی گسترش داد. دو راه برای انجام این کار وجود دارد :

- مهندسی ویژگی ها: در این روش، از تبدیل ویژگی های ورودی به فضایی با ابعاد بالاتر استفاده می شود، به طوری که یک مرز تصمیم گیری خطی در این فضای جدید می تواند با یک مرز تصمیم گیری غیر خطی در فضای ویژگی اصلی مطابقت داشته باشد. این کار را می توان با استفاده از ویژگی های چند جمله ای یا سایر تبدیل های غیر خطی انجام داد.

- روش هسته: شامل نگاشت ویژگی‌های ورودی در فضایی با ابعاد بالاتر به طور ضمنی، با استفاده از یک تابع هسته است که شباهت بین جفت بردارهای ویژگی ورودی را محاسبه می‌کند. سپس مرز تصمیم را می توان با استفاده از رگرسیون لجستیک در فضای ضمنی با ابعاد بالاتر، بدون نیاز به محاسبه صریح ویژگی های تبدیل شده، آموخت.

Q2

(a

هدف از این سوال استفاد از رگرسیون لجستیک برای طبقه بندی تصاویر در احتمال آتش در یک عکس می باشد، برای پیش پردازش تصویر ابتدا آنها را از خوانده و همه آنها را به اندازه مشخصی تغییر اندازه داده و برچسب مناسب را به آن اضافه کردم

(b

داده ها را به دو دسته train , test با در نظر گرفتن 0.2 داده ها برای تست دسته بندی کردم.

(d

```
Accuracy: 0.92
Precision: 0.9113924050632911
Recall: 0.9863013698630136
F1 score: 0.9473684210526315
confusion_matrix:
 [[ 40  14]
 [  2 144]]
```

(e

```
Best threshold: 0.92
Accuracy: 0.95
Precision: 0.9594594594594594
Recall: 0.9726027397260274
confusion_matrix:
 [[ 48   6]
 [  4 142]]
```

نتیجه تنها چند درصد نسبت به حالت قبل تفاوت دارد و کمی عملکرد بهتری دارد. ولی با توجه به اینکه Accuracy و Presision , Recall بالای ۹۰ درصد می باشد و مشاهده Confusion_matrix، مدل کارایی خوبی دارد.

(f


Fire: 1.00


Fire: 0.98


Fire: 1.00


No Fire: 1.00


Fire: 1.00


No Fire: 0.95


No Fire: 0.98


No Fire: 1.00

Q3

(a

برای پیش پردازش داده کارهای زیر انجام شده:

۱ - با بررسی داده های null، ستون Unnamed: 0 به دلیل اینکه همه ی مقادیر آن null بودند حذف شده است.

۲- ستون label به دلیل اینکه برای پیش بینی فقط label_id هدف است و تنها label توضیحات بیشتری درباره label_id ارائه میکند، حذف شده است.

۳- مقادیر null باقی مانده در جدول حذف شده اند.

۴- با استفاده از تابع remove_tag تگ های html , url اگر در متن وجود داشته باشد بررسی شده و با " جایگزین میگردد (عملا حذف شده).

۵- در فایل persion حروف اضافی، کاراکتر ها، حروف اشاره و ... قرار دارد. که در هر یک از comment ها بررسی شده و در صورت وجود این حروف شده شده است.

۶- ابتدا با استفاده از word_tokenize کلمات موجود در هر کامنت از یکدیگر جدا میشوند و سپس هر کلمه با ریشه آن جایگزین میشود.

داده ها را به دو دسته train , test با در نظر گرفتن test_size = 0.2 تقسیم شده اند.

(b

Naive Bayes classifier به صورت زیر نوشته شده؛

۱- ایجاد یک دایره لغات از تمام کلمات منحصر به فرد در مجموعه آموزشی ایجاد شده

۲- شمارش تعداد دفعاتی که هر کلمه در بررسی های مثبت و منفی ظاهر شده

۳- محاسبه احتمالات ظاهر شدن هر کلمه در بررسی مثبت یا منفی

۴- محاسبه احتمالات قبلی مثبت یا منفی بودن یک بررسی.

۵- استفاده از قضیه بیز برای محاسبه احتمال مثبت یا منفی بودن یک بررسی با توجه به کلمات آن.

در متغیر vocab کلمات و word_counts تعداد رخداد هر کلمه ذخیره شده است و در تابع fit احتمال مثبت یا منفی بودن کلمه در متن بررسی میشود.

از لگاریتم احتمالات در این پیاده سازی استفاده شده است.

از رخ داد کلمه ها و تاثیر آنها بر مثبت یا منفی بودن جمله در داده آموزش برای پیش بینی در این مدل استفاده میشود.

(c

```
Accuracy: 0.7784254461715602
Precision: 0.7024714828897338
Recall: 0.9621003905684942
F1 score: 0.8120383370978572
```

Q4

(a

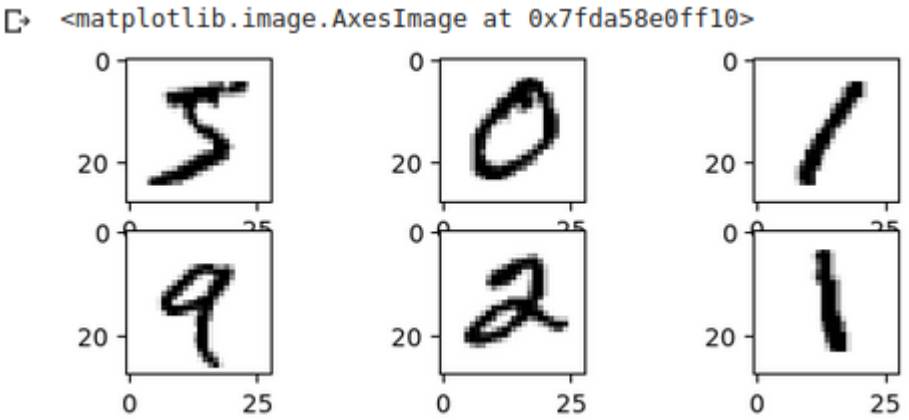
[14] mnist.data

	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	pixel10	...	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783	pixel784
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
69995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

70000 rows × 784 columns

[15] mnist.target.shape

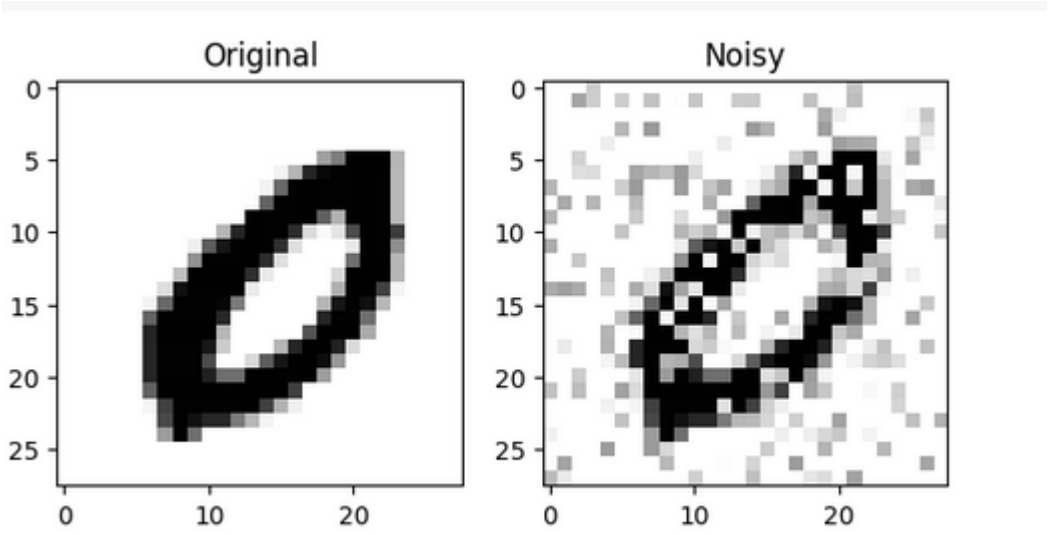
نمونه ای از داده ها در دیتاست:



دیتاست شامل ۷۸۴ ستون و ۷۰۰۰۰ ردیف میباشد، برای پیاده سازی این سوال ۱۰۰۰ داده به صورت رندوم از دیتاست انتخاب شده اند.

(b

برای افزودن نویز به تصویر ابتدا مقادیر همه پیکسل ها به ۲۵۵ تقسیم شده(همه پیکسل ها در بازه [0,1] قرار بگیرند)، سپس ۲۰۰ پیکسل به صورت رندوم از هر تصویر انتخاب شده و مقدار آن با مقدار جدیدی میان [0,0,4] جایگزین میشود.



(c

هدف پیش بینی تصویر بدون نویز می باشد، بنابراین تصویر بدون نویز به عنوان y استفاده میشود و داده ای که به آن نویز اضافه شده به عنوان x.

با توجه به اینکه x , y تصاویر هستند و هدف از این سوال تنها پیش بینی مقدار خاصی به عنوان label نیست و باید آرایه پیش بینی شود باید از مدل هایی با ویژگی multi- output classification استفاده شود.

با استفاده از روش های classification و ویژگی تصاویری که در یک کلاس قرار دارند، امکان حذف نویز وجود دارد.

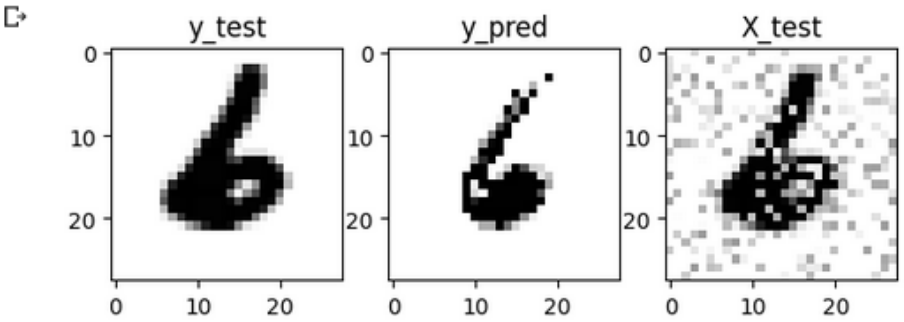
(d

در این مسئله امکان استفاده از logistic regression, knn, decision tree, .. وجود دارد

برای پیاده سازی با توجه به اینکه در این مسئله تنها پیش بینی یک تصویر باید پیش بینی شود، در روش پیشنهادی از یک مدل multi-output classification با استفاده از KNN پیاده سازی شده و یک MultiOutputClassifier ایجاد کرده و مدل را روی مجموعه آموزشی با استفاده از fit آموزش می‌دهیم.

تعداد همسایه ها در KNN برابر با ۵ در نظر گرفته شده.

(e



در تصویر بالا X_test به عنوان Before

y_pred به عنوان After