به نام خدا

تمرین دوم مبانی یادگیری ماشین (بخش 3)

محمدرضا ضیالاری (97222057)

پیش پردازش داده ها :

ابتدا بررسی میکنیم که داده های تهی نداشته باشند و تایپ داده ها را بررسی میکنیم و مشاهده می کنیم که همه ی داده ها عددی و ناتهی هستند .

بخش 1:

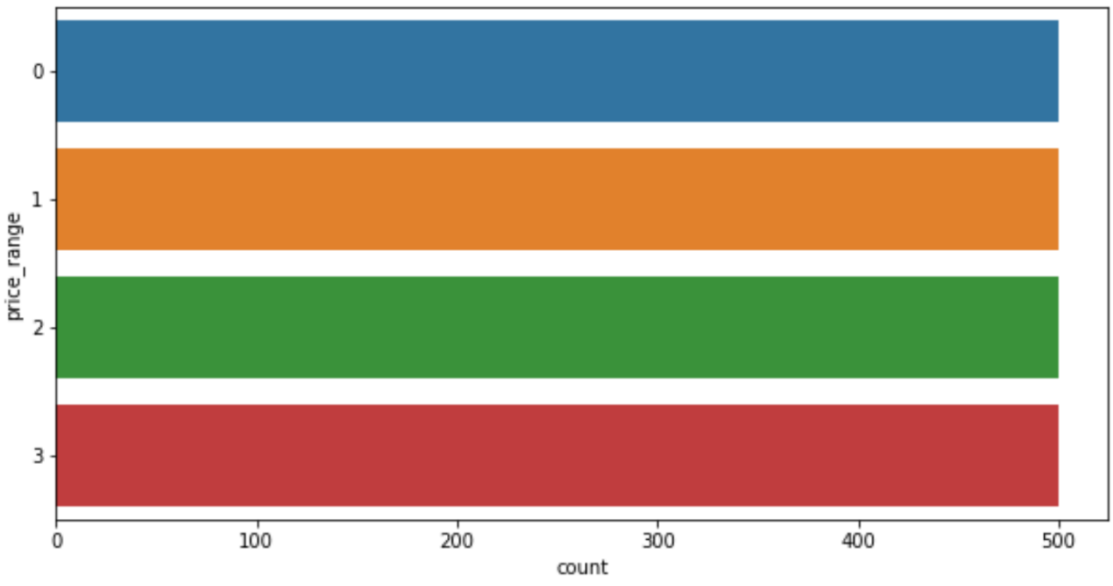
از طریق پکیج sklearn یک رگرسیون لاجستیک می سازیم که تارگت آن محدوده قیمت و ورودی آن دیگر ویژگی ها می باشد و داده ها را به دو دسته ترین و ولیدیشن تقسیم میکنیم و مدل را فیت میکنیم .

نتایج حاصل یصورت رو به رو بود :

```
score 0.6766666666666666
f1_score 0.6664075836297976
precision_score 0.6672084314848754
recall_score 0.6683210784313725
```

بخش 2:

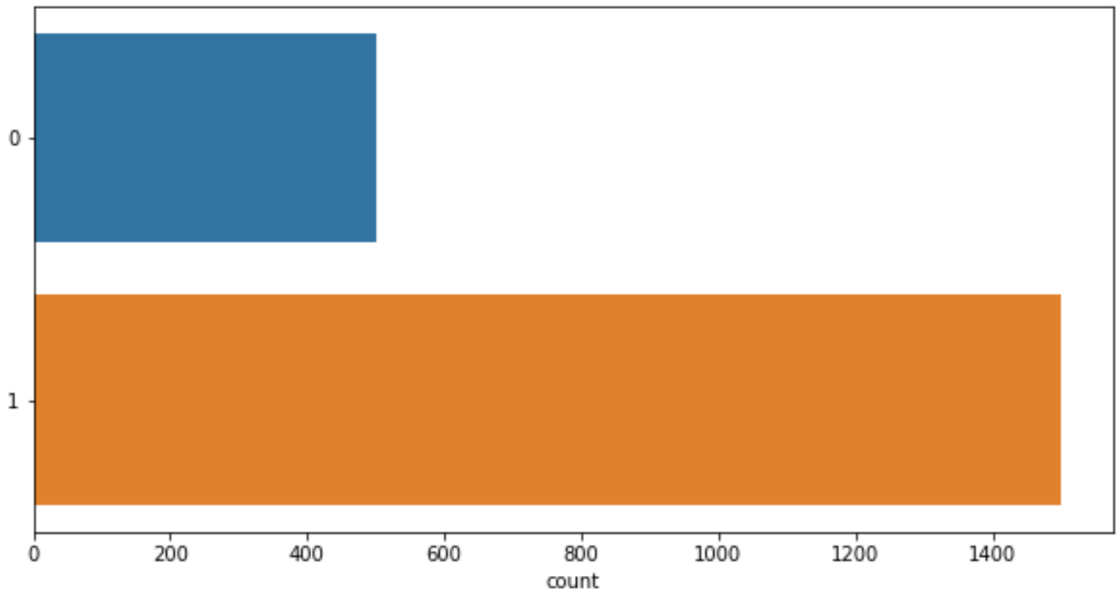
در این بخش تعداد نمونه های هر کلاس برای محدوده قیمتی را بررسی می کنیم



مشاهده می شود که تعداد همه با هم برابر و 500 عدد می باشد .

بخش 3:

در این بخش از ما خواسته شده که داده های کلاس های 1و2و3 را در کلاسی به نام 1 و داده های کلاس 0 را دست نخورده باقی بگذاریم .



در این حالت تعداد داده های کلاس 1 برابر 1500 عدد و داده های کلاس 0 برابر 500 عدد می شود .

بخش 4:

حال در حالتی که دو کلاس محدوده قیمتی داریم رگرسیون لاجستیک را پیاده سازی می کنیم و نتایج را بررسی میکنیم .

نتایج مطابق رو به رو می باشد :

```
score 0.94
f1_score 0.9207001879699248
precision_score 0.9367690058479532
recall_score 0.9073863636363637
```

بخش 5:

برای رفع مشکل نامتوازن بودن داده ها راه های مختلفی وجود دارد که در اینجا به سه مورد آن اشاره میکنیم :

1-کم کردن تعداد داده های کلاس داده ای که مقدار آنها بسیار زیاد است (down sampling)

2-زیاد کردن تعداد داده های کلاسی که مقدار آنها کم است (up sampling)

3-جنریت کردن داده ها ی جدید از روی داده های موجود از طریق شبکات GAN

ما در اینجا برای رفع عدم توازن دیتا از روش اول استفاده کردیم و محجدا مدل رگرسیون لاجستیک را اعمال کردیم .

نتایج حاصل به صورت روبه رو بود :

```
score 0.8866666666666667
f1_score 0.8865406006674083
precision_score 0.8867521367521367
recall_score 0.8864081124355097
```

بخش 6:

در این بخش از طریق انتخاب پیشرو ویژگی های مهم تر و تاثیر گذار تر را استخراج کردیم که ویژگی های زیر بودند :

1- Battery_power
2- Blue
3- Clock_speed
4- Dual_sim
5- Fc
6- Four_g
7- Int_memory
8- N_dep

9- Mobile_wt

10-N_cores

بخش 7:

در این بخش با استفاده از ویژگی های استخراج شده بخش 6 مدل رگرسیون لاجستیک خود را اجرا کردیم و نتایج زیر حاصل شد :

```
score 0.8866666666666667
f1_score 0.8865406006674083
precision_score 0.8867521367521367
recall_score 0.8864081124355097
```

بخش 8و9:

در این قسمت از طریق اجرای الگوریتم PCA بر روی ویژگی های استخراج شده از بخش 6 تعداد ویژگی ها را کم میکنیم و به دو ویژگی می رسانیم و سپس رگرسیون لاجستیک را پیاده سازی می کنیم . نتایج زیر بدست می آید :

```
score 0.8016666666666666
f1_score 0.8005721872510466
precision_score 0.8006894624153351
recall_score 0.800498146025972
```

بخش 10:

در این بخش ویژگی های مهم را از طریق روش انتخاب پسرو استخراج کردیم و رگرسیون لاجستیک را بر روی آن اعمال نمودیم . که نتایج زیر بدست آمد که مقادیر پایین تری نسبت به حالت پیشرو داشت :

```
score 0.3016666666666667
f1_score 0.2795234478899147
precision_score 0.2897427520069029
recall_score 0.29820086593135176
```

بخش 11:

در این بخش از 5fold-cross validation , 10fold-cross validation استفاده کردیم که مقدار میانگین خطا برای 5fold-cross validation برابر 0.42 و میانگین دقت برابر 63.55٪ می بود و برای حالت 10fold-cross validation میانگین خطا برابر 0.417 و میانگین دقت برابر 64 درصد بود .