به نام خدا

گزارش تمرین شماره ۲ داده کاوی

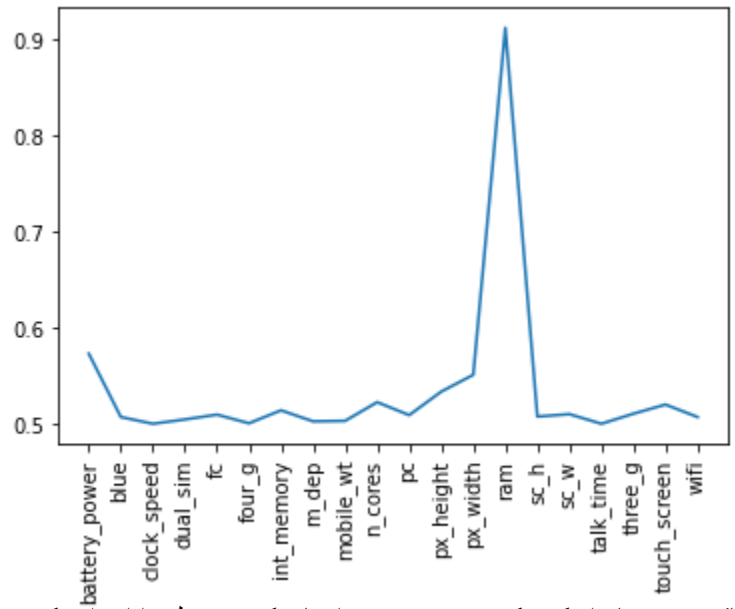
یگانه بهاری ۴۰۰۴۲۲۰۴۷

سوال ۱

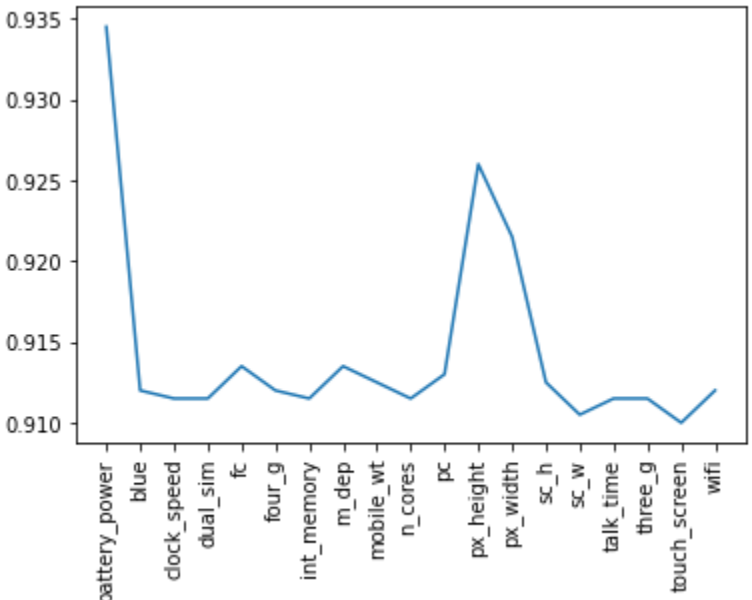
ابتدا داده را به وسیله کتابخانه pandas باز میکنیم سپس در قدم اول وجود مقادیر null را چک میکنیم سپس به تسک ها میپردازیم

۱ برای اعمال forward selection ابتدا تعداد کلاس های ستون y که همان دسته بندی قیمت گوشی می باشد را از ۴ کلاس به ۲ کلاس تبدیل میکنیم .سپس این ستون از داده را به عنوان y جداسازی میکنیم. در ادامه مدل رگرسیون لاجستیک را روی باقی ستون ها اول تک تک اجرا کرده و معیار AUC را برای هر یک میسنجیم و در مرحله با ثابت نگه داشتن ستونی که بیشترین AUC را دارد باقی ستون ها را اضافه کرده و با سنجش AUC در هر مرحله یک ستون را اضافه کرده و ثابت نگه میداریم تا به ۵ فیچر با بیشترین اثر برسیم. نتایج را در نمودار های زیر مشاهده میکنیم:

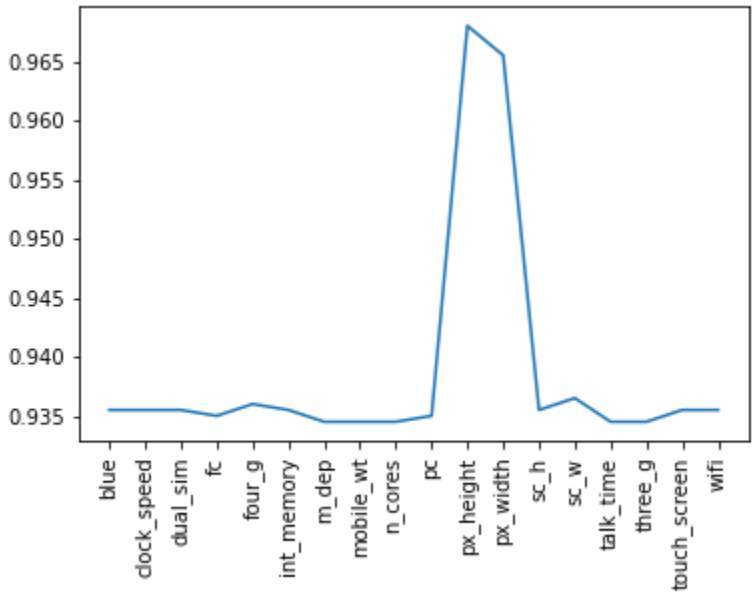
در مرحله اول ستون ها را یکی یکی میسنجیم:

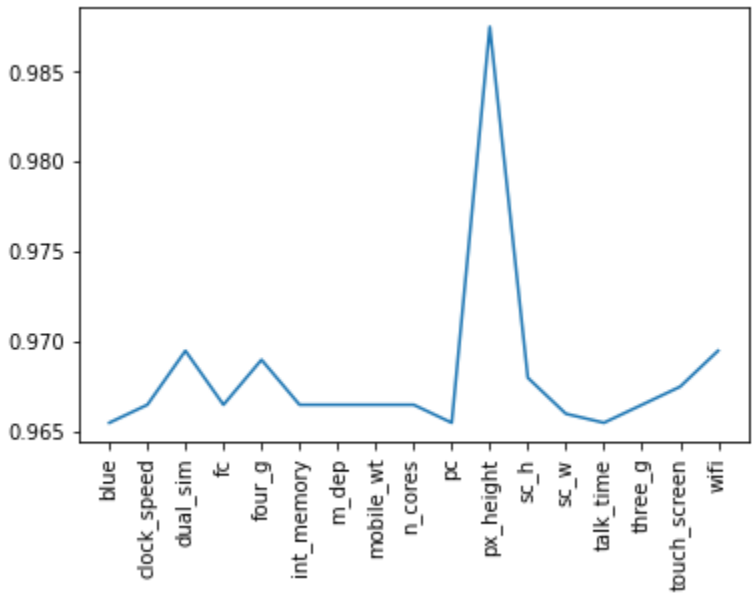


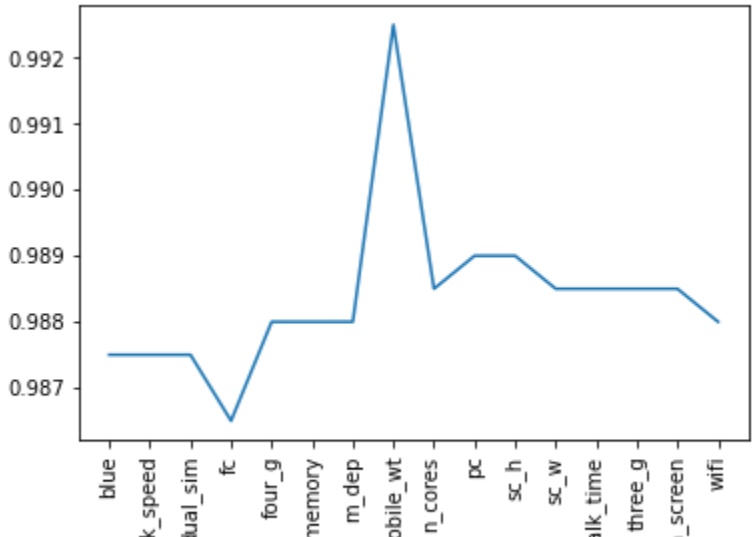
ستون ram را نگه داشته و باقی ستون ها را یکی یکی به ستون ram اضافه کرده و مدل را اجرا میکنیم:



ستون های battery power , ram را نگه داشته و باقی ستون ها را اضافه کرده و مدل را اجرا میکنیم. همین مراحل را تکرار کرده تا به ۵ ویژگی مهم دست میابیم:





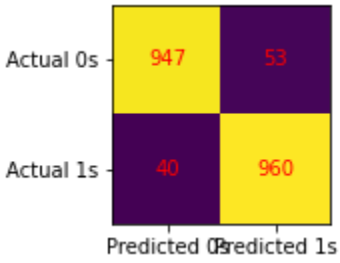


طبق نمودار های مشاهده شده ۵ ستون اصلی برابر است با:

{'ram':x1 ,'battery_power':x2 , 'px_width':x3 , 'px_height':x4, 'mobile_wt':x5}

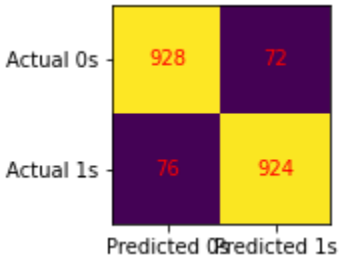
۲ در این مرحله با ستون های بدست آمده مدل رگرسیون لاجستیک را اجرا کرده و نتایج را مشاهده میکنیم:

precision_score: 0.947680157946693
recall_scor: 0.96
f1_score: 0.9538002980625931



۳و ۴ به وسیله الگوریتم PCA دیتاست را تغییر دادیم و نتایج اجرای مدل رگرسیون لاجستیک را بر روی دیتای جدید مشاهده میکنیم:

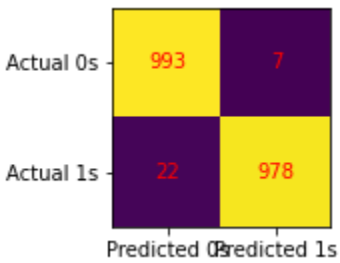
precision_score: 0.9919354838709677
recall_scor: 0.984
f1_score: 0.9879518072289156



۵

۶ مدل svm را بر روی داده اجرا کرده و نتایج را مشاهده میکنیم:

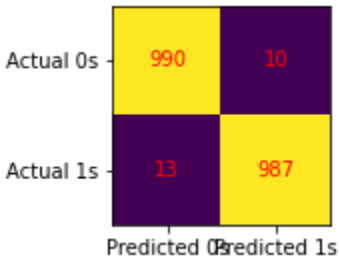
precision_score: 0.9928934010152284
recall_scor: 0.978
f1_score: 0.9853904282115868



۷ کرنل های مختلف را بر روی svm پیاده سازی کرده و نتایج هریک را مشاهده میکنیم:

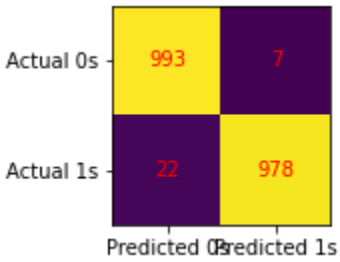
kernel='linear'

precision_score: 0.9899699097291875
recall_scor: 0.987
f1_score: 0.9884827240861291



kernel='rbf'

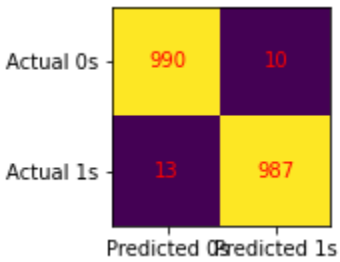
precision_score: 0.9928934010152284
recall_scor: 0.978
f1_score: 0.9853904282115868



kernel=my_kernel ساخت کرنل دلخواه

```
def my_kernel(X, Y):
    return np.dot(X, Y.T)
```

precision_score: 0.9899699097291875
recall_scor: 0.987
f1_score: 0.9884827240861291



سپس برای کرنل rbf پارامترهای gamma , c را تغییر داده و برای هر یک عملکرد مدل را مشاهده میکنیم:

((()SVC ,0.1 ,0.01)
precision_score: 0.99289340101098706
recall_scor: 0.978
f1_score: 0.9853904282345898

((()SVC ,1 ,0.01)
precision_score: 0.9899695697291875
recall_scor: 0.986

f1_score: 0.988627240861291

۸

این قسمت را چند بار اجرا کرده اما متاسفانه به علت محاسبه زیاد دچار dead kernel شدم

۹

الف بر روی ستون battery power روش binnig را اعمال کرده و نتایج را مشاهده میکنیم:

q=3
(500.999, 977.667] 667
(977.667, 1496.0] 667
(1496.0, 1998.0] 666

q=5
(500.999, 781.0] 401
(781.0, 1076.0] 400
(1395.4, 1698.2] 400
(1698.2, 1998.0] 400
(1076.0, 1395.4] 399

q=10
(1395.4, 1549.0] 202
(634.9, 781.0] 201
(920.7, 1076.0] 201
(1698.2, 1851.0] 201
(500.999, 634.9] 200
(1226.0, 1395.4] 200
(781.0, 920.7] 199
(1076.0, 1226.0] 199
(1851.0, 1998.0] 199
(1549.0, 1698.2] 198

ب استفاده از one hot encoding در واقع یک روش برای دسته بندی ستون ها و تغییر شکل داده به صورتی که برای ماشین قابل فهم باشد است. بیشتر برای دسته بندی ویژگی هایی که به صورت متن هستند به کار میرود.
مثال این روش را در زیر مشاهده میکنیم:

Type
AA
AB
CD
AA

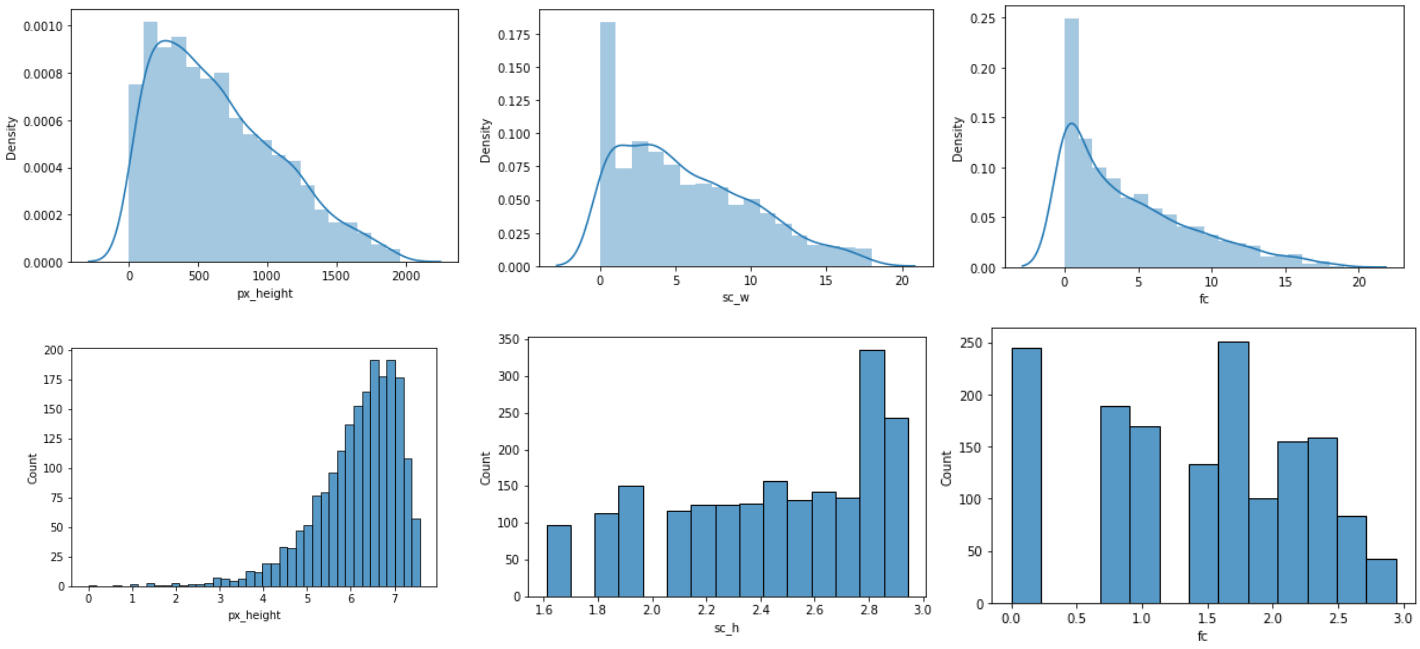
Onehot encoding →

Type	AA_Onehot	AB_Onehot	CD_Onehot
AA	1	0	0
AB	0	1	0
CD	0	0	1
AA	0	0	0

در داده ای که در دست داریم میتوان روش one hot را برای مثال روی ستون های four_g,wifi , dual_sim , blue که ستون های categorical هستند اعمال کرد. نتیجه به صورت زیر میشود:
میتوان از get dummies نیز برای اینکار استفاده کرد.

m_dep	mobile_wt	n_cores	pc	px_height	px_width	...	price_range	two_class_price	blue_0	blue_1	four_g_0	four_g_1	dual_sim_0	dual_sim_1	wifi_0	wifi_1
0.6	188	2	2	20	756	...	1	0	1	0	1	0	1	0	0	1
0.7	136	3	6	905	1988	...	2	1	0	1	0	1	0	1	1	0
0.9	145	5	6	1263	1716	...	2	1	0	1	0	1	0	1	1	0
0.8	131	6	9	1216	1786	...	2	1	0	1	1	0	1	0	1	0
0.6	141	2	14	1208	1212	...	1	0	0	1	0	1	1	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.8	106	6	14	1222	1890	...	0	0	0	1	0	1	0	1	1	0
0.2	187	4	3	915	1965	...	2	1	0	1	1	0	0	1	0	1
0.7	108	8	3	868	1632	...	3	1	1	0	0	1	0	1	1	0
0.1	145	5	5	336	670	...	0	0	1	0	0	1	1	0	0	1
0.9	168	6	16	483	754	...	3	1	0	1	0	1	0	1	0	1

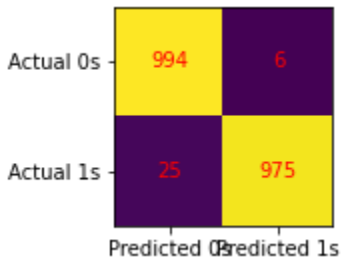
ج از تبدیل لگاریتمی معمولا برای مشاهده بهتر توزیع مقادیر در داده و بررسی و تخمین رفتار آن ها استفاده میشود برای مثال در داده مورد نظر ما توزیع سه پارامتر را مشاهده میکنیم و در ادامه توزیع لگاریتمی آنها را مشاهده میکنیم:



۱۰

برای حالتی که داده را one hot کرده ایم یک مدل svm اجرا کرده و نتیجه آن را مشاهده میکنیم:

precision_score: 0.9938837920489296
recall_scor: 0.975
f1_score: 0.9843513377082281



۱۱ و ۱۴

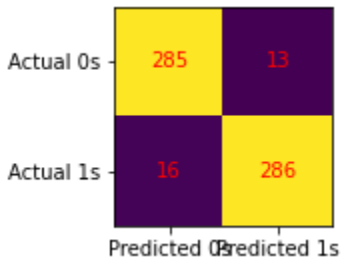
در واقع الگوریتم های مختلف درخت ها در تعداد گره ها و پارامتر ها و گره ها تفاوت دارند و در ادامه روی هریک از این الگوریتم ها هرس کردن اتفاق می افتد. بعضی الگوریتم ها ساختار سلسله مراتبی دارند و لایه به لایه پیش می روند و برخی محدودیت دارند در تعداد پارامترهایی که به کار میبرند.و برخی الگوریتم های درخت برای کلاس بندی و برخی برای مسایل رگرسیون به کار میروند.

از هرس کردن درخت برای این استفاده میشود که درخت ها ممکن است برروی نمونه های آموزشی نتایج خوبی بدهند اما بر روی نمونه های تست کارایی پایینی دارند و در واقع ممکن است overfit شوند. اما درخت با انشعاب کمتر ممکن است کمی بایاس داشته باشد اما به شدت واریانس کمتری دارد

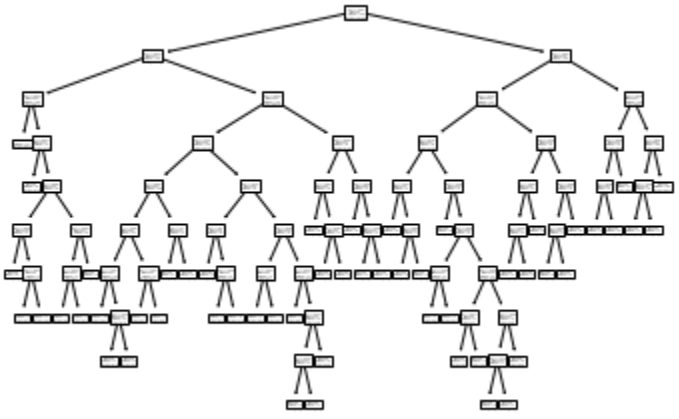
۱۲

یک مدل درخت تصمیم بر روی داده اجرا کرده و نتایج را مشاهده میکنیم:

precision_score: 0.9565217391304348
recall_scor: 0.9470198675496688
f1_score: 0.951747088186356



نمونه درخت اجرا شده بر روی داده:



۱۵

از bootstrap برای برآورد خطای ضرایب رگرسیونی و همچنین انجام آزمون فرض استفاده میشود. در زمانی که اندازه نمونه کوچک و دقت برآوردگرها مطرح باشد،این روش، می‌تواند خطا را بوسیله روش نمونه‌گیری مجدد محاسبه کند و فاصله اطمینان یا انحراف استاندارد مناسب و پرتوانی، ارائه دهد.
روش cross validation یک روش ارزیابی مدل است که تعیین می‌نماید نتایج یک تحلیل آماری بر روی یک مجموعه‌داده تا چه اندازه قابل تعمیم و مستقل از داده‌های آموزشی است. این روش به‌طور ویژه در کاربردهای پیش‌بینی مورد استفاده قرار می‌گیرد تا مشخص شود مدل موردنظر تا چه اندازه در عمل مفید خواهد بود.

۱۶

5x2 cross validation به معنای این است که یک cross validation 5-fold را دو بار در حلقه تکرار کرده و یادگیری انجام میشود. در واقع عدد ۵ تعداد fold های لایه درونی مدل است و عدد ۲ مربوط به لایه بیرونی می باشد.معمولا از این روش برای تست اجرای مدل استفاده میشود برای مثال برای مقایسه عملکرد اجرایی آماری مدل ها با یک دیگر.

۱۷

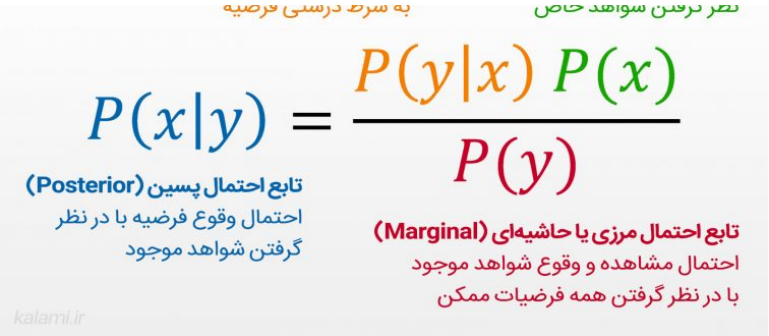
روش elbow یک روش برای تجزیه و تحلیل خوشه ای یا cluster analysis است که تغییر پارامترها را به عنوان تابعی از تعداد خوشه ها در قالب یک نمودار نشان میدهد از این روش میتوان برای یافتن دیگر پارامتر ها در مدل های مربوط به داده مانند تعداد اجزای اصلی یا principle component استفاده کرد.

سوال ۲

۱

قضیه بیز بیان میکند که می‌توان احتمال یک پیشامد را با مشروط کردن نسبت به وقوع یا عدم وقوع یک پیشامد دیگر محاسبه کرد. در بسیاری از حالت‌ها، محاسبهٔ احتمال یک پیشامد به صورت مستقیم کاری دشوار است. با استفاده از این قضیه و مشروط کردن پیشامد مورد نظر نسبت به پیشامد دیگر، می‌توان احتمال مورد نظر را محاسبه کرد.

به بیان ریاضی:



Guassian:
در مسایل classification استفاده میشود و بیان میکند ویژگی ها از توزیع نرمال پیروی میکنند.
multinomial
برای شمارش های گسسته استفاده میشود. میتوان از این روش برای شمارش تکرار های مقادیر مدل استفاده کرد.
Bernoulli
زمانی استفاده میشود که بردارها باینری باشند. برای کلاس بندی متنی استفاده میشود برای مثال یک کلمه آیا در متن وجود دارد یا ندارد.

۲

یک مدل gaussian naive bayes را بر روی داده اجرا کرده و پارامتر های زیر را بدست آوردیم:

[mean: [[134.39855072 251.08695652 139.10144928
[[158.46666667 242.23030303 129.3030303]
[var: [[350.8108008 2445.75880673 510.70496139
[[367.65284553 2867.91005174 261.4563932]
[prior probabilities: [0.45544554 0.54455446
[posterior probabilities: [8.82893481e-08 2.64854468e-16
[Conditional Probability of the classes given test-data: [9.99999996e-01 3.58677266e-09
final prediction: 0

حال یک مدل gaussian naive bayes را به وسیله کتابخانه اجرا میکنیم:

precision_score: 0.5128205128205128
recall_scor: 0.7142857142857143
f1_score: 0.5970149253731343