به نام خدا

گزارش کار تمرین 2 الگوریتم های یادگیری ماشین

امیرحسین مرتضی 810895050

سوال اول

1- طبقه بند با مدل مولد GDA:
برای توزیع p(x|y d) یک گوسی چندمتغیره فرض می کنیم

Y ~ Bernoulli(Q)
p(x| y = 0) ~ N(u0, [ )
p(x| y = 1) ~ N(u1, [ )
p(y) = Q^y(1-Q)^(1-y)

مقادیر U1 .Q, u0,] با maximum liklihood به دست می آیند

Arg max p(Y = y | X =x)
y = {0, 1}

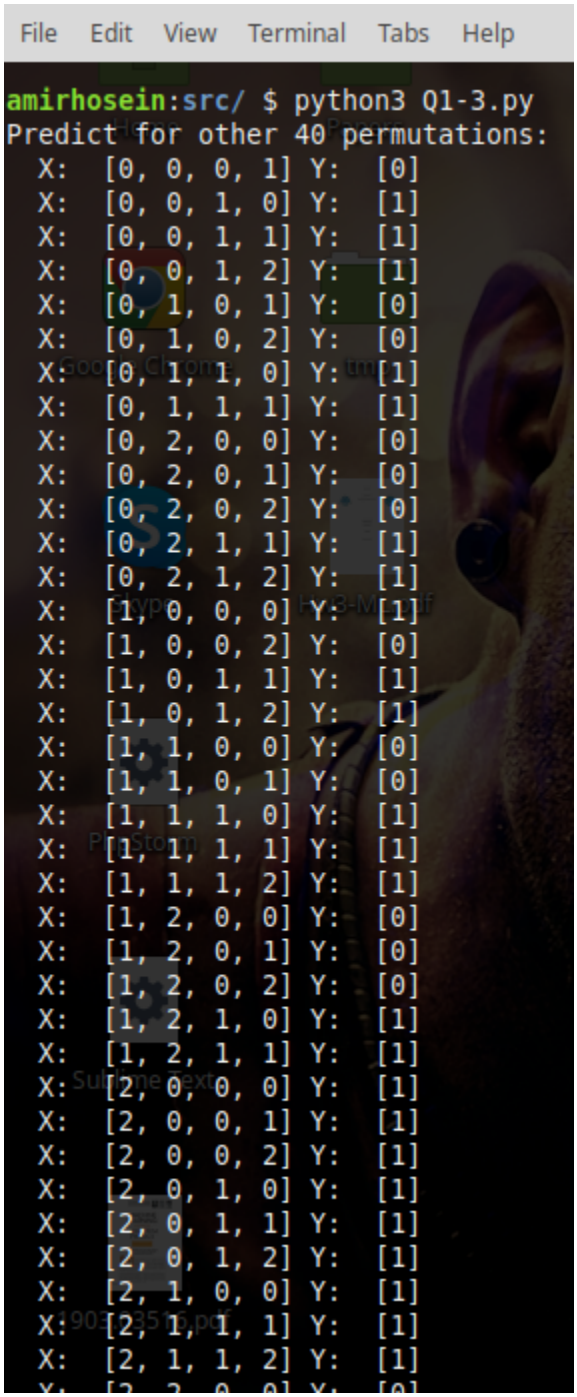
هنگامی که توزیع عادی را برای هر کلاس محاسبه کردید ، برای طبقه بندی داده ای که باید محاسبه کنید ، برای هر یک احتمال این که متعلق به آن باشد را طبقه بندی می کنید. کلاس با بیشترین احتمال به عنوان کلاس وابستگي انتخاب خواهد شد پس در پایان داریم:

p(y =1| Xtest) = p(x|y)p(y)/p(x)

خروجی y ها play و داده ها بردار Outlook, Temp, Humidity, Wind هستند
به تعداد m = 14 نمونه آموزشی و n = 4 ویژگی داریم همچنین X ها لزوما باینری نیستند و در دیکشنری میتوانند مقادیر گسسته { 2, 1, 0} را اختیار کنند لذا توزیع مالتی نومیال به جای برنولی استفاده می شود یعنی به جای دوتا Qj چندین Qj خواهیم داشت برای دیتای زیاد میتپانیم مشابه سوال طبقه بندی ایمیل ها از Naive Bayesian استفاده می کنیم تا تعداد نمونه های مورد نیاز به صورت نمایی با تعداد ویژگی ها افزایش پیدا نکند

-2
مدل discriminative مرز بین دو کلاس را یاد میگیرد اما مدل generative توزیع هر کلاس ذا مدل سازی می کند برتری آن این است که داده های آزمون ممکن است توسط توزیع های اساسی مختلف نسبت به داده های آموزش ایجاد شود. به طور معمول ، تشخیص صحیح تغییرات توزیع و به روز کردن یک مدل مولد ساده تر از انجام این کار برای یک مرز تصمیم گیری در SVM است ، به خصوص اگر نیاز باشد که به روزرسانی های آنلاین نیاز به نظارت نداشته باشند ، ساده تر است. همچنین مدل های discriminative برای تشخیص نقاط دور از دسترس کار نمی کنند در حالی که مدل های مولد بطور کلی انجام می دهند
در GDA داده ها باید با یک گاوسی فیت شده باشند تا مدل خوب عمل کند چون از فرضیات اطمینان داریم یعنی داده ها را میتوان با یک توزیع عادی تقریب زد از GDA استفاده می شود

3- در کد Q1-3.py با دیکشنری ساختیم سپس مدل GDA یادگیری و برای 40 حالت دیگر خروجی پیشبینی شد.
طبقه بندی کننده ای با مرز تصمیم گیری خطی ، با تناسب چگالی شرطی کلاس به داده ها و با استفاده از قانون Bayes. مدل برای هر کلاس توزیع گاوسی در نظر میگیرد با فرض اینکه همه کلاس ها یک ماتریس کوواریانس یکسان دارند.



4- به طور کلی وقتی از داده ها اطمینان نداریم بهتر است از logistic regression استفاده شود اما در صورت استفاده از مدل GDA قبلی آن دو ویژگی که نمیدانیم را از داده حذف کردیم با اجری Q1-4.py امکان پیش بینی وجود نداشت زیرا در این مدل به هر 4 ویژگی نیاز است

```
amirhosein:src/ $ python3 Q1-4.py
Predict for other 10 couple permutations:
Traceback (most recent call last):
  File "Q1-4.py", line 39, in <module>
    print("  X: ", data, "Y: ", clf.predict([data]))
  File "/home/amirhosein/.local/lib/python3.6/site-packages/
    scores = self.decision_function(X)
  File "/home/amirhosein/.local/lib/python3.6/site-packages/
    % (X.shape[1], n_features))
ValueError: X has 2 features per sample; expecting 4
amirhosein:src/ $
```

لذا در training مدل دو ستون از ویژگی ها را ignore کرده و مجددا یادگیری انجام دادیم

```
File  Edit  View  Terminal  Tabs  Help
amirhosein:src/ $ python3 Q1-4.py
Predict for other 10 couple permutations:
  X:  [0, 0] Y:  [0]
  X:  [0, 1] Y:  [1]
  X:  [0, 2] Y:  [1]
  X:  [1, 0] Y:  [1]
  X:  [1, 1] Y:  [1]
  X:  [1, 2] Y:  [1]
  X:  [2, 0] Y:  [1]
  X:  [2, 1] Y:  [1]
  X:  [2, 2] Y:  [1]
amirhosein:src/ $
```

سوال دوم

۱- بدون استفاده از داده ها احتمال بیمار بودن برای همه 50 درصد است اما اگر label های داده های آموزشی را داشته باشیم احتمال بیمار بودن تعداد بیماران تقسیم بر تعداد کل داده ها یعنی 10000 است

۲-در Q2-2.py با GaussianNB مدل fit و پیش بینی انجام شد

```
File  Edit  View  Terminal  Tabs  Help
amirhosein:src/ $ python3 Q2-2.py
Accuracy: % 72.9
```

۳- با فرض مثبت بودن آزمایش یک فرد با توجه به ویژگی های Xtest به دست می آید که دقت 72 % دارد حال که مقدار label بیمار می باشد لذا با احتمال 72 % بیمار است

سوال سوم

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Attribute Information:

1. sepal length in cm

2. sepal width in cm

3. petal length in cm

4. petal width in cm

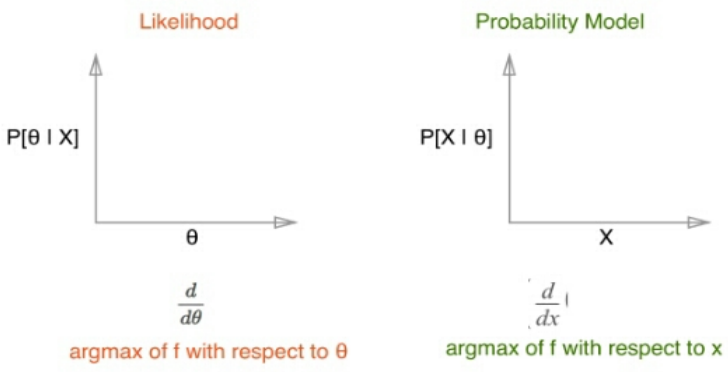
5. class:

-- Iris Setosa

-- Iris Versicolour

-- Iris Virginica

۱- در مدل Bayesian به جایی محاسبه یک نقطه بهینه با استفاده از Max Likelihood برای theta یک متغیر تصادفی دارای توزیع در نظر گرفته می شود اما در محاسبه احتمال چون محاسبه انتگرال روی بردار theta به دلیل ابعاد آن هزینه بر است از Max p(theta|x,y) MAP استفاده می کنیم. به طور کلی برآورد ML تلاش می کند مقدار x را که حداکثر می کند p (y | x) را پیدا کند ، در حالی که برآورد MAP تلاش می کند مقدار x را که حداکثر می کند p (x | y) را از قانون بیز پیدا کند.



در MAP معادل عمل اضافه شدن ترم پایدار کننده به صورت غیرمستقیم اتفاق می افتد

MAP:

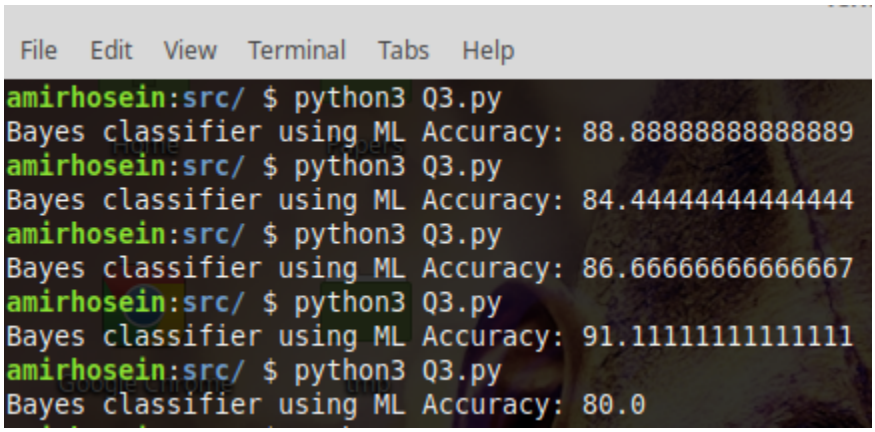
Max arg p(theta|X) =max arg log(p(X| theta)p(theta)) = arg max log(p(X|theta) + log(P(theta)) -> معادل ترم پایداری مدل است

در روش ML صرفا یک بردار theta بهینه داریم که likelihood را بیشینه کرده اما MAP مقدار توزیع پسین را بیشینه میکند, همچنین در MAP میتوانیم از همه بردار های theta در محاسبه احتمال استفاده کنیم و ماکزیمم بگیریم که در ML ممکن نیست

اگر توزیع پیشین فرضی یکنواخت باشد theta در ML و MAP یکسان خواهد شد

۲- کد Q3.py تخمین با روش ML انجام شد

۳- چون هر بار اجرا shuffle میکنیم دقت تغییر می کند



-۴

در کد Q3.py دقت برابر جمع قطر اصلی ماتریس بر تمام درایه ها محاسبه می شود

Confusion Matrix		Predicted		
		Class 1	Class 2	Class 3
Actual	Class 1	A	B	C
	Class 2	D	E	F
	Class 3	G	H	I

True positives True Negatives Misclassified cases.

مشاهده میکنیم دقت ها یکسان اند:

```
File  Edit  View  Terminal  Tabs  Help
amirhosein:src/ $ python3 Q3.py
Bayes classifier using ML Accuracy: % 86.6666666
Confusion Matrix:
 [[14  0  0]
 [ 0 12  4]
 [ 0  2 13]]
Accuracy: 0.8666666666666667
amirhosein:src/ $
```

۵- اگر داده اولیه به اندازه ی کافی نداشته باشیم مدل آموزشی دچار underfitting می شود اینجا با استفاده از 0.1 داده ها دقت کمتر شد اگر این مقدار را در حد 0.03 کم کنیم به دقت خیلی پایین یا ارور برخوردیم

```
Bayes classifier using ML Accuracy: % 81.48148148148148
amirhosein:src/ $ python3 Q3-5.py
Bayes classifier using ML Accuracy: % 78.51851851851852
amirhosein:src/ $
```

```
amirhosein:src/ $ python3 Q3-5.py
/home/amirhosein/.local/lib/python3.6/site-packages/numpy/core/fro
  out=out, **kwargs)
/home/amirhosein/.local/lib/python3.6/site-packages/numpy/core/_me
  ret = ret.dtype.type(ret / rcount)
/home/amirhosein/.local/lib/python3.6/site-packages/numpy/core/fro
  **kwargs)
/home/amirhosein/.local/lib/python3.6/site-packages/numpy/core/_me
  arrmean, rcount, out=arrmean, casting='unsafe', subok=False)
/home/amirhosein/.local/lib/python3.6/site-packages/numpy/core/_me
  ret = ret.dtype.type(ret / rcount)
/home/amirhosein/.local/lib/python3.6/site-packages/numpy/linalg/li
  r = _umath_linalg.det(a, signature=signature)
Bayes classifier using ML Accuracy: % 47.407407407407405
```

۶- برای دقت بیشتر باید مدل را بهتر کنیم با فرض clean بودن داده ها بعد از feature selection و normalization می توانیم کار های زیر را انجام دهیم:

1. تنظیم پارامترهای قابل تنظیم در تخمین
2. نوعی تکنیک ترکیب طبقه بند مانند , bagging , boosting ,ensembling اضافه کنیم
3. می توان به جای 3 کلاس از روش one-against-many استفاده کرد یعنی ابتدا بین کلاس A و بقیه طبقه بندی و در مرحله بعد بقیه را به B , C تقسیم کنیم
4. می توانیم به جای تخمین ML از MAP استفاده کنیم تا احتمال برای تمام بردار های theta را بتوانیم محاسبه و ماکزیمم را چک کنیم