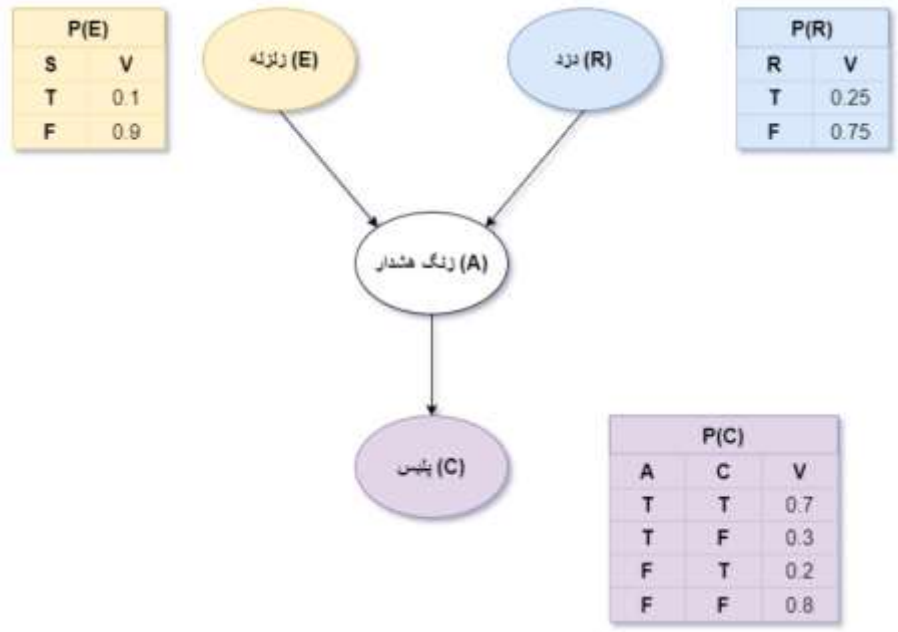
سوال ۱)



$$P(C=T, A=T, E=T, R=T) = p(c|A)\ p(A|E, R)\ P(E)\ P(R)$$

با توجه به رابطه بالا باید جدول احتمال شرطی $P(A|E,R)$ را بدست آوریم.همانطور که در صورت سوال آمده است احتمال خطا صفر است پس با اتفاق افتادن هر یک از زلزله و دزد ویا هر دو زنگ هشدار به صدا در می آید و فقط زمانی زنگ هشدار به صدا در نمیاد که هیچ کدام از دزد و زلزله رخ ندهد. پس داریم :

E	R	A	P(A|E,R)
T	T	T	1
T	T	F	0
T	F	T	1
T	F	F	0
F	T	T	1
F	T	F	0
F	F	T	0
F	F	F	1

i) P(C=T, A=T, E=T, R=T)

P(C=T, A=T, E=T, R=T) = p(c|A) p(A|E , R) P(E) P(R) =
0.7*1*0.1*0.25=0.0175

ii) P(C=F, A=T, E=F, R=T)

P(C=F, A=T, E=F, R=T) = p(c|A) p(A|E , R) P(E) P(R) =
0.3*1*0.9*0.25=0.0675

iii) P(C=T, A=T, R=T)

دراینجا باید ۲ حالت رو بررسی و جمع کنیم این حالت ها میشه E=T و E = F .

if E = T :

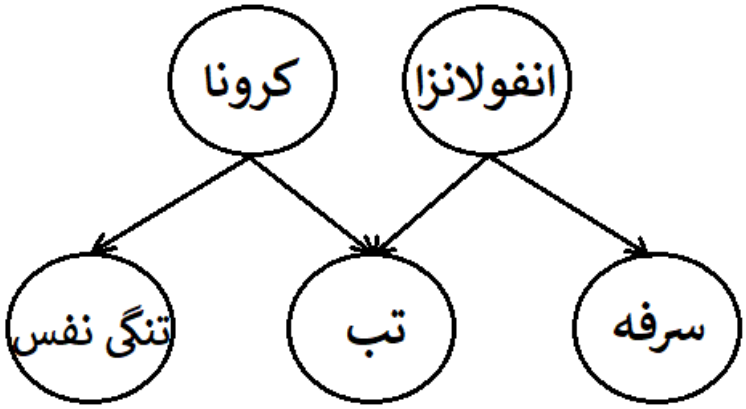
P(C=T, A=T, R=T) = p(c|A) p(A|E , R) P(E) P(R) = 0.0175

if E = F :

P(C=T, A=T, R=T) = p(c|A) p(A|E , R) P(E) P(R) = 0.7*1*0.9*0.25 = 0.1575

⇒ P(C=T, A=T, R=T) = 0.0175 + 0.1575 = 0.175

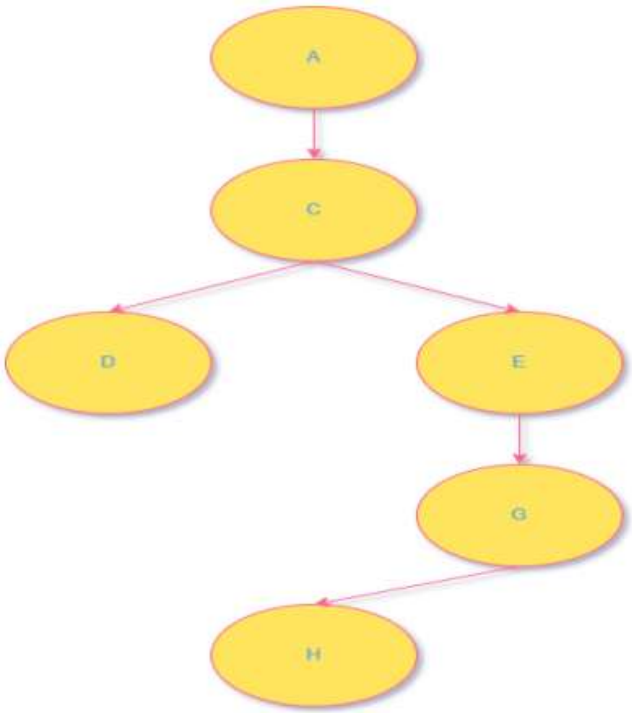
سوال ۲)



P(سرفه , تب ,تنگی نفس ,سرفه , انفولانزا , کرونا) =

P(انفولانزا) P (کرونا) P (انفولانزا|سرفه) P (انفولانزا ,کرونا |تب) P (کرونا|تنگی نفس)

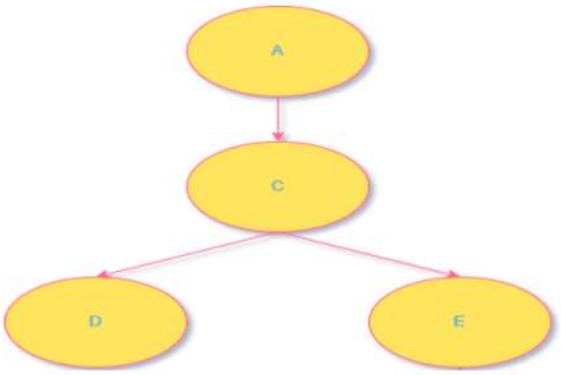
سوال ۳)



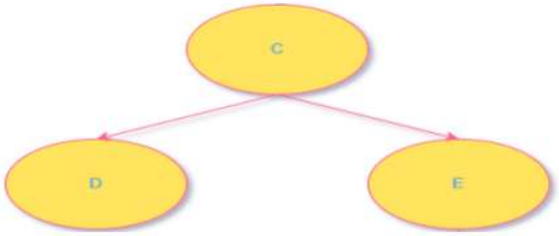
یکی از روش های تشخیص استقلال این است که آن قسمت از درخت که داریم آن را مورد بررسی قرار میدهیم رو جدا میکنیم و بدون در نظر گرفتن جهت فلش ها راسی که داده شده است را حذف میکنیم حال اگر دو راسی که مورد بررسی قرار داده ایم را میبینیم اگر که در بخشی همبند از گراف قرار دارند مستقل نیستند اما اگر در دو بخش نا همبند اند مستقل اند پس داریم:

۱)آیا E و D لزوما از هم مستقل اند با توجه به اینکه A داده بشود؟

قسمتی از درخت که باید آن را بررسی کنیم :



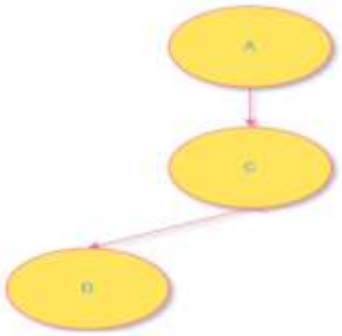
حال میایم راس A که داده شده است را حذف میکنیم حال داریم :



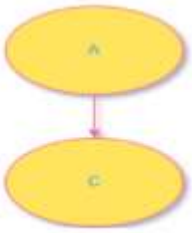
راس های E و D را میخواهیم ببینیم مستقل هستند یا خیر. در گراف بالا میبینیم که هر دو این راس ها در یک بخش همبند قرار دارند و یک مسیر بین آن ها است پس لزوما مستقل نیستند.

۲)آیا A و C لزوما از هم مستقل اند با توجه به اینکه D داده بشود ؟

قسمتی از درخت که باید آن را بررسی کنیم :



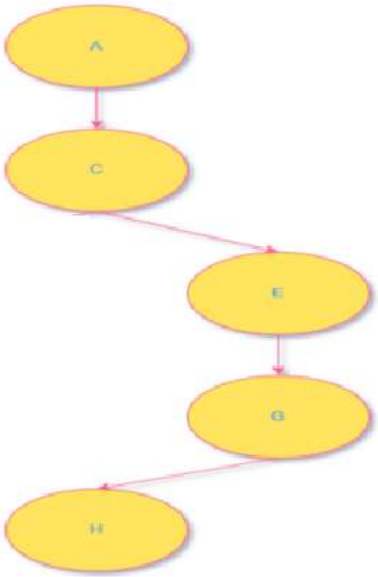
میخواهیم راس های A و C رو مستقل بودن و یا نبودنشون رو بررسی کنیم. راس D داده شده است پس آن را حذف میکنیم:



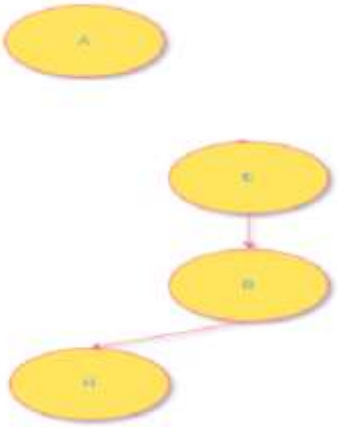
A و C در یک بخش همبند قرار دارند پس لزوما مستقل نیستند

۳) آیا A و H لزوما از هم مستقل اند با توجه به اینکه C داده بشود ؟

قسمتی از درخت که باید آن را بررسی کنیم :



چون C داده شده است آن راس را حذف میکنیم و داریم :



راس های A و H را که میخواهیم بررسی کنیم در دو بخش نا همند قرار دارند پس میتوان نتیجه گرفت که این دو مستقل از هم هستند با داده شدن C.

سوال۴)

	Doc	Words	Class
Training	1	میلیارد فوری فرصت برنده	spam
	2	فوری فوری تخفیف برنده	spam
	3	جلسه فردا تنظیم	ham
	4	فوری جلسه فردا خبر	ham
Test	5	تخفیف آموزش میلیارد برنده	؟

الف) جدول احتماالت را برای لیبلهای spam, ham محاسبه کنید.

ابتدا کلمات بکار رفته را میشماریم. ممکن است کلماتی که در spam بکار رفته است در ham بکار نرفته باشند که در اینصورت برابر صفر میشه اما این صفر برای ما مشکل ایجاد میکند. برای رفع این مشکل کافیست به تعداد تمامی کلمات چه در ایمیل های spam و چه از نوع ham. پس داریم :

کلمات	تعداد در spam+۱	تعداد در ham+۱	P(کلمه|spam)	P(کلمه|ham)
میلیارد	۲	۱	۲/۱۳	**۱/۱۲**
فوری	۴	۲	۴/۱۳	**۲/۱۲**
فرصت	۲	۱	۲/۱۳	**۱/۱۲**
برنده	۳	۱	۳/۱۳	**۱/۱۲**
تخفیف	۲	۱	۲/۱۳	**۱/۱۲**
جلسه	۱	۳	۱/۱۳	**۳/۱۲**
فردا	۱	۳	۱/۱۳	**۳/۱۲**
تنظیم	۱	۲	۱/۱۳	**۲/۱۲**
خبر	۱	۲	۱/۱۳	**۲/۱۲**

جمع کلمات به این صورت است که تعداد جمع کلمات بکار رفته در هر خط (مثلا برای spam برابر با ۸ است) و به ازای هر نوع کلمه بعلاوه ۱ میکنیم که در اینجا ۵ است پس میشه ۱۳. و برای ham هم به همین صورت.

ب) با کمک جدول ها لیبل داده تست را تعیین کنید.

خب همانطور که در جدول داده شده مسئله داریم از ۴ تا داده داده شده ۲ تا ایمیل از نوع spam و ۲ تا ایمیل از نوعham داریم. پس اگر جدول احتمالاتی آن را بکشیم داریم:

نوع	احتمال
spam	2/4=1/2
ham	2/4=1/2

در ایمیل تست داریم : تخفیف آموزش میلیارد برنده.

میایم احتمال های (تخفیف , میلیارد, برنده, p(y=spam و (تخفیف , میلیارد, برنده, p(y=ham را حساب میکنیم. (آموزش چون کلمه ای جدید است آن را دیگر حساب نمیکنیم)

P(y=spam , برنده , میلیارد , تخفیف) =

p(y=spam) P(برنده|spam) P(میلیارد|spam) P(تخفیف|spam)=

(1/2)*(3/13)*(2/13)*(2/13) = (12/4395)

P(y=ham , برنده , میلیارد , تخفیف) =

p(y=ham) P(برنده|ham) P(میلیارد|ham) P(تخفیف|ham)=

(1/2)*(1/12)*(1/12)*(1/12) = (1/3456)

حال با مقایسه احتمال ها داریم که احتمال spam بودن بیشتر از ham بودن است.

گزارش قسمت پیاده سازی :

لینک کد:

https://colab.research.google.com/drive/13lKbAiC78iAb6Fn68opF00xziNqJ6MlG?usp=sharing

```
spam_sum = np.zeros(700)
ham_sum = np.zeros(700)
n = len(x_train)

for i in range(n):
    if y_train[i] == True:
        spam_sum += x_train[i]
    else:
        ham_sum += x_train[i]

spam_word = 0
ham_word = 0

for i in range (len(spam_sum)):
  spam_sum[i] += 1
  spam_word += spam_sum[i]

for i in range (len(ham_sum)):
  ham_sum[i] += 1
  ham_word += ham_sum[i]

conditional_prob_spam = spam_sum / spam_word
conditional_prob_ham = ham_sum / ham_word
```

در ابتدا برای هر قسمت spam و ham ارایه ای با ۷۰۰ مولفه ۰ میسازیم و اندازه داده های train را هم در یک متغیر نگه‌داری میکنیم.

در یک حلقه for که تکرار آن به اندازه داده های train است در هر قسمت تعداد تکرار کلمات هر n خانه رو که در spam هستند و چه در ham هستند رو به صورت جدا میشماریم و نگه میداریم.

حال دو متغیر spam_word و ham_word را با مقدار اولیه ۰ تعریف می کنیم. که این متغیر ها مجموع تمامی کلمات استفاده شده در هر کدام از این قسمت ها هستند.

اما همانطور که در صورت سوال ۴ داشتیم تمامی این اعداد را +۱ میکنیم و این اعداد را در متغیر های spam_word و یا ham_word نگه میداریم.

حال برای محاسبه احتمال امدن هر کلمه ارایه هایی که تعداد هر کلمه را مشخص میکند را تقسیم بر تعداد تمامی کلمات در هر بخشش میکنیم و در متغیر های جدید تعریف میکنیم.

```
def naive_bayes(email):
  spam_count = 0
  ham_count = 0

  for i in range (n):
    if y_train[i] == True:
      spam_count += 1
    else:
      ham_count += 1

  ham = ham_count/n
  spam =spam_count/n

  for i in range(700):
    if email[i]>0:
      ham = ham * conditional_prob_ham[i]
      spam = spam * conditional_prob_spam[i]

  if ham >= spam:
    return False
  else:
    return True
```

در این تابع میایم ایمیل های داده شده رو میگیریم. میایم تعداد ایمیل هایی که از قبل داشتیم رو به دو بخش تقسیم میکنیم و میشماریم که چه تعداد spam و چه تعداد ham بودند تا احتمال آن هارا بدست بیاوریم. حال ایمیل های test رو میگیریم و با حساب کردن احتمال spam و یا ham بودنشون میایم مقایسه میکنیم هر کدام احتمال بالاتری داشتند آن را انتخاب میکنیم.

```
y_predict = np.zeros(len(test_data))

for i in range(len(test_data)):
  y_predict[i] = naive_bayes(test_data[i])
```

در این بخش spam بودن ویا ham بودن داده های test ما مشخص میشود

```
"""Show metrics and score!"""
print(classification_report(y_test, y_predict))
print('confusion matrix : ')
print(confusion_matrix(y_test, y_predict))
```

و در آخر هم خروجی ها را نشان میدهیم

خروجی کد به این شکل است :

```
              precision    recall  f1-score   support

       False       0.91      0.97      0.94       200
        True       0.97      0.91      0.94       200

    accuracy                           0.94       400
   macro avg       0.94      0.94      0.94       400
weighted avg       0.94      0.94      0.94       400

confusion matrix :
[[195   5]
 [ 19 181]]
```

به طور خلاصه میتوان گفت :

اطلاعاتی که این معیارها می‌دهند متفاوت است، هر چه تعداد تشخیص‌های نادرست برنامه بیشتر باشد Recall آن کمتر می‌شود و هر چه مواردی که باید بدست می‌آمدن ولی پیش بینی نشدن بیشتر باشد Precision کاهش پیدا می‌کند. معیار F1_score هم برابر میانگین هندسی این دو معیار برابر است.

معیار accuracy یا صحت :

شاید اولین و ساده‌ترین معیاری باشد که ما سراغ آن می‌رویم معیار accuracy یا همان صحت است که برابر است با تعداد مواردی که درست پیش بینی کردیم که آن را True Positive می‌نامیم تقسیم بر تعداد کل پیش بینی‌هایی که انجام شده‌است.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

معیار Recall یا یادآوری :

حداکثر مقدار این معیار یک ویا ۱۰۰ درصد و حداقل مقدار آن صفر است و هرچه مواردی که ما انتظار داشتیم پیش بینی شوند ولی برنامه پیش بینی نکرده‌است که به آن False Negative می‌گوییم نسبت به پیش بینی‌های درست یا True Positive بیشتر باشد مقدار Recall کمتر خواهد شد.

فرمول محاسبه‌یRecall :

در فرمول زیر TP مخفف True Positive و FN مخفف False Negative است.

$$Recall = \frac{TP}{TP + FN}$$

معیار Precision یا دقت :

حداکثر مقدار این معیار یک ویا ۱۰۰ درصد و حداقل مقدار آن صفر است و هرچه مواردی که برنامه به غلط پیش بینی کرده است که به آن False Positive می‌گوییم نسبت به پیش بینی‌های درست یا True Positiveبیشتر باشد مقدار Precision کمتر خواهد شد.

فرمول محاسبه‌یPrecision

در فرمول زیر TP مخفف True Positive و FP مخفف False Positive است.

$$Precision = \frac{TP}{TP + FP}$$

معیار f1-score :

زمانی که می‌خواهید معیار ارزیابی شما میانگینی از دو مورد قبلی باشد یعنی همان Recall یا Precision می‌توانید از میانگین هارمونیک این دو معیار استفاده کنید که به آن معیار f1-score می‌گویند.

$$F1 = \frac{2}{\frac{1}{Recall} \cdot \frac{1}{Precision}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

معیار Micro Avg وMacro Avg :

محاسبه‌ی Micro Avg برای Recall و Precision برابر است مقادیری که تا بحال برای محاسبه‌ی Recall و Precision بدست آوردیم. نکته‌ای که در این روش وجود داشت این بوده که هر یک از داده‌ها و نتایج مستقل از این که در کدام مجموعه داده هستن (در مثال ما دو دیتاست اینستاگرام و توییتر وجود دارد) روی نتیجه‌ی نهایی تاثیر می‌گذارند برای جلوگیری از آن می‌توان میزان Recall یا Precision هر دیتاست را جدا گانه حساب کرد و درنهایت میانگین آن را به دست آورد که به آن Macro Avg Recall یا Macro Avg Precision می‌گویند و با فرمول‌های زیر محاسبه می‌شوند.

$$\text{MacroAverage Recall} = \frac{\sum_{i=1}^{n} Recall_i}{n}$$

$$\text{MacroAverage Precision} = \frac{\sum_{i=1}^{n} Precision_i}{n}$$