به نام هستی بخش مهربان
هوش مصنوعی – پاییز ۹۸
علی شهیدی ۸۱۰۱۹۴۳۴۱

الگوریتم:

ابتدا کل داده ها را به نسبت ۸۰ به ۲۰ به صورت تصادفی به دو بخش train و test تقسیم کردم.

محاسبات تعداد استفاده از هر کلمه توسط هر شاعر

یک کلمه به شکل زیر نگهداری میشود. (هر مجموعه نشاندهنده تعداد آن کلمه در کل داده‌های train شده یک شاعر است.) علاوه بر این تعداد کل کلمه‌های استفاده شده توسط یک شاعر (چه تکراری و چه غیرتکراری) در تمام بیت‌های داده‌ی trainشده را به همراه تعداد بیت‌های بررسی شده آن را نگه داشتم.

```
words_stats[word] = {'saadi': 0, 'hafez': 0}
```

مثال:

```
print(words_stats["کشتی"])
>> {'saadi': 2, 'hafez': 3}
```

پیش بینی شاعر و محاسبه کسر احتمالاتی

فیچرها: تعداد استفاده از هر کلمه در شعر

برای محاسبه احتمال اینکه این شعر به حافظ تعلق دارد یا نه (همینطور برای سعدی) :

$$P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i|Y)$$

که W یک کلمه است و Y یا سعدی یا حافظ است و احتمال بخش دوم (صورت کسر همان **Likelihood**) به صورت زیر محاسبه میشود:

$P(W_i \mid Y) = n_k / (n\_y)$

بخش قرمز : Predictor Prior Probability

n_k = تعداد تکرار این کلمه در داده‌های ترین شده برای این شاعر

تعداد کل کلمات بکاربرده شده توسط این شاعر در داده های ترین شده= n_y

Class prior probability:

P(Y) = C_Y / (C_Y + C_V)

تعداد بیت های ترین شده شاعر Y = C_Y

تعداد بیت های ترین شده شاعر X = C_V

سپس احتمال هرکدام که بیشتر شد آن بیت به آن شاعر تعلق دارد. (**Posterior**)

```
saadi_p *= (words_stats[word]['saadi']) / (saadi_words_count)
hafez_p *= (words_stats[word]['hafez']) / (hafez_words_count)
```

معیار ارزیابی مدل:

۱- چنانچه برای ارزیابی خروجی یک مدل ماشین لرنینگ فقط به مقدار Precision توجه شود چه مشکلی پیش می آید؟ برای مثال یک مدل ماشین لرنینگ معرفی کنید که Precision بالایی دارد ولی خوب کار نمیکند.

جواب: در اینجا تعداد جواب های درست اصلی مهم نیست. یعنی امکان دارد مثلا در یک دسته داده ۱۰۰ بیت حافظ داشته باشیم و هیچ بیتی از بقیه شاعرها نداشته باشیم و precision صد در صد داشته باشیم در حالی که accuracy برابر با ۱ درصد است. مثال: از این ۱۰۰ بیت تابع ما فقط یک بیت را برای حافظ تشخیص میدهد : precision = 100

Accuracy = 1/100

۲- چرا مقدار Accuracy به تنهایی برای تشخیص خوبی مدل کافی نیست؟ برای مثال یک دیتاست معرفی کنید که یک مدل ضعیف ماشین لرنینگ روی آن به Accuracy بالایی برسد بدون آنکه عملا تشخیصی انجام دهد.

جواب: دسته بندی داده هایی که پراکندگی آنها خیلی متفاوت است ولی اشتباه در آن باعث آسیب بزرگی میشود. مثلا ۹۹ درصد داده ها در دسته ۱ قرار میگیرند و بقیه در دسته ۲. حالا اگر تابع ما همیشه همه داده های تست را در دسته ۱ بگیرد ما ۹۹ درصد Accuracy داریم. مسئله اینجاس که

ما هزینه محاسبه اشتباه با محاسبه درست را یکی فرض کردیم. حال اگر مهم باشد که آن ۱ درصد را حداقل نصفش را درست تشخیص دهیم در اینجا قابل تشخیص نیست که اینکار را کردیم یا نه.
مثال: ۹۹ درصد سیب – ۱ درصد موز
تابع: همه را سیب در نظر بگیر. این تابع Recallش برای موز صفر است!

لاپلاس:

مشکل این است که اگر مثلا کلمه "کشتی" در هیچکدام از بیت های train شده حافظ بکار نرفته باشد ولی در بیت های جدید که میخواهیم ببینیم برای کدام شاعر است بکار رفته باشد, چون این احتمال ضرب میشود در احتمال بقیه کلمه های بیت, این احتمال صفر میشود: یعنی اگر کلمه کشتی در یک بیت شعر حافظ در داده های تست باشد ولی در داده های ترین شده نباشد, بدون درنظر گرفتن بقیه کلمه ها این احتمال صفر میشود و بنابراین تشخیص داده میشود که این بیت برای سعدی است. (چون این احتمال ها در هم ضرب میشوند.)

راه حل:

صورت کسر را به علاوه ۱ کردم که هیچ کلمه ای اگر درشعر یک شاعر بکار رفته بود برای آنیکی شاعر برابر صفر نباشد که احتمال آن بیت را برای آن شاعر صفر کند. مخرج هم تقسیم بر دامنه کلمات کردم زیرا به اندازه کلمات دامنه یک یک, به تعداد همه اضافه میشود.

```
saadi_p *= (words_stats[word]['saadi'] + 1) / (saadi_words_count +
len(words_stats))
```

جواب ها در مجموع پس از ۳ بار ران کردن:

```
******************
train data:
       text
label
hafez  6723
saadi  9996
count  16719
******************
test data:
       text
label
hafez  1695
saadi  2475
count  4170
```

```
Report for Train Data
Recall: 0.8649412464673509
Precision: 0.9293591177880773
Accuracy: 0.9192535438722411
***********************************
Report for Test Data
Recall: 0.6873156342182891
Precision: 0.7482337829158638
Accuracy: 0.7788968824940048
***********************************
Report for Train Data + Laplas
Recall: 0.8078238881451733
Precision: 0.8692381562099872
Accuracy: 0.8738560918715235
***********************************
Report for Test Data + Laplas
Recall: 0.6896755162241888
Precision: 0.784037558685446
Accuracy: 0.7966426858513189
```

```
train data:
       text
label
hafez  6684
saadi  9939
count  16623
******************
test data:
       text
label
hafez  1734
saadi  2532
count  4266
```

```
Report for Train Data
Recall: 0.8653500897666068
Precision: 0.9371354504212573
Accuracy: 0.9225169945256573
***********************************
Report for Test Data
Recall: 0.6903114186851211
Precision: 0.7339055793991416
Accuracy: 0.7723863103609939
***********************************
Report for Train Data + Laplas
Recall: 0.8240574506283662
Precision: 0.8963384865744508
Accuracy: 0.8909342477290502
***********************************
Report for Test Data + Laplas
Recall: 0.68800461361015
Precision: 0.7807591623036649
Accuracy: 0.7946554149085795
```

```
 Train Data
       text
label
hafez  6671
saadi  9915
Total  16586
******************
 Test Data
       text
label
hafez  1747
saadi  2556
Total  4303
```

```
 Report for Train Data
Recall: 0.8964173287363214
Precision: 0.9128377346969928
Accuracy: 0.9239117327866876
********************
 Report for Test Data
Recall: 0.7080709788208357
Precision: 0.7171014492753623
Accuracy: 0.7680687892168254
********************
 Report for Train Data + Laplas
Recall: 0.8250637085894169
Precision: 0.8935064935064935
Accuracy: 0.890088026046063
********************
 Report for Test Data + Laplas
Recall: 0.6960503720663995
Precision: 0.8052980132450331
Accuracy: 0.8082732976992796
```