گزارش فاز دوم پروژه بازیابی پیشرفته اطلاعات- دکتر بیگی

پرند علیزاده علمداری 94100024
شبنم قاسمی راد 94105803

تمام قسمت‌ها مطابق مطالب درس پیاده سازی شده است. تقسیم کار در این پروژه به صورت مساوی انجام شده است. زبان برنامه نویسی پایتون است.

*** به دلیل یک اشکال فنی در ران کردن بخش svm که زمان زیادی می‌برد، پروژه را با 15 دقیقه تاخیر آپلود کردیم، اگر امکان دارد لطفا 1 روز اضافی لحاظ نکنید.

پیش‌پردازش:

فایل utils یک فایل کمکی برای پارس کردن فایل‌های csv در شروع برنامه است. برای csv از کتابخانه‌ی csv پایتون استفاده شده است. در فایل tf-idf، با کمک اطلاعات استخراج شده از فایل csv مربوط به داده های train، تگ ها را ذخیره‌سازی کرده‌ایم و برای ساختن فضای برداری، به ازای تمام کلمات ایندکس شده tf و idf را در این مجموعه محاسبه کرده و نهایتا این بردارها را نیز ذخیره‌سازی کرده‌ایم. مشابه همین کار برای داده‌های تست در فایل test_vectors انجام شده است.

Precision و Recall و F1 برای هر کلاس جدا حساب شدند.

random_forest:

دقت random forest با ۱۰۰ درخت و عمق نامحدود برای داده train مقادیر زیر بود:

train:

Accuracy: 0.9933333333333333

Precision: [0.99333629 0.99204596 0.9915518 0.99284756]

Recall: [0.99377778 0.99777778 0.99111111 0.98711111]

برای داده test :

test:

Accuracy: 0.749

Precision: [0.76422764 0.81617647 0.692 0.71551724]

Recall: [0.752 0.888 0.692 0.664]

F1: [0.7580645161290323, 0.8505747126436781, 0.692, 0.6887966804979253]

Svm:

90 درصد داده ها ترین شدند و 10 درصد برای ولیدیشن استفاده شد که نتایج برای c = 0.5, 1, 1.5 ,2 به صورت زیر بود:

```
c = 0.5
Accuracy:  0.8355555555555556
Precision:  [0.91346154 0.87398374 0.9273743  0.67790262]
Recall:  [0.85972851 0.95132743 0.664      0.89162562]

c = 1
Precision:  [0.91549296 0.91561181 0.90990991 0.78070175]
Recall:  [0.88235294 0.96017699 0.808      0.87684729]

c= 1.5
Precision:  [0.92344498 0.93886463 0.9086758  0.74485597]
Recall:  [0.87330317 0.95132743 0.796      0.89162562]

c = 2
Accuracy:  0.8777777777777778
Precision:  [0.91981132 0.93859649 0.90178571 0.75847458]
Recall:  [0.88235294 0.94690265 0.808      0.8817734 ]
```

همچنین برای داده‌های تست با در نظر گرفتن بهترین پارامتر (c = 2) ، نتایج یاد گرفتن کل داده‌ی train و سپس اجرا روی test به صورت زیر بود:

```
Accuracy:  0.858
Precision:  [0.90909091 0.92125984 0.80784314 0.8       ]
Recall:  [0.84  0.936 0.824 0.832]
F1:  [0.8731808731808731, 0.9285714285714286, 0.8158415841584158,
0.8156862745098038]
```

Knn:

برای افزایش سرعت به صورت ماتریسی پیاده سازی شد.

به ازای k = 1, 5, 9 مقادیر زیر روی داده validation محاسبه شد: (۱۰٪ از کل داده train به داده validationاختصاص یافت و با ۹۰٪ باقیمانده یادگیری انجام شد)

validation:

k = 1

Accuracy: 0.8222222222222222

Precision: [0.8173913 0.91363636 0.80578512 0.75   ]

Recall: [0.85067873 0.88938053 0.78   0.76847291]

k = 5

Accuracy: 0.8588888888888889

Precision: [0.84347826 0.93362832 0.84016393 0.815   ]

Recall: [0.87782805 0.93362832 0.82   0.80295567]

k = 9

Accuracy: 0.86

Precision: [0.86936937 0.93362832 0.81568627 0.82233503]

Recall: [0.87330317 0.93362832 0.832   0.79802956]

و در نهایت k = 9 برای تست انتخاب شد:

test: k = 9

Accuracy: 0.84

Precision: [0.90948276 0.88582677 0.76653696 0.80544747]

Recall: [0.844 0.9   0.788 0.828]

F1: [0.8755186721991701, 0.8928571428571428, 0.777120315581854,

0.8165680473372781]

:Naive Bayes

مقادیر دقت روی داده train:

#train

 # Accuracy: 0.9257777777777778

 # Precision: [0.94922232 0.96336677 0.89238264 0.898365]

 # Recall: [0.92222222 0.98177778 0.89555556 0.90355556]

دقت روی داده test:

#test:

 # Accuracy: 0.858

 # Precision: [0.91101695 0.91119691 0.78461538 0.82857143]

 # Recall: [0.86 0.944 0.816 0.812]

 # F1: [0.8847736625514404, 0.9273084479371315, 0.7999999999999999,

0.8202020202020203]

برای بخش دوم پروژه، ابتدا در فایل tagging_untagged_files با گرفتن ایندکس فایل‌های انگلیسی فاز1 که قبلا آن ها را ذخیره کرده بودیم، وکتورهای این فایل‌ها را با استفاده از همان لیست کلمات فایل‌های train ساختیم. سپس با استفاده از یکی از سریع‌ترین و دقیق ترین پیش بینی کننده‌مان (random forrest)، تگ هایشان را پیش بینی کردیم (فایل ها دسته بندی شدند) و سپس این تگ‌ها را در phase1_tags.data ذخیره کردیم.

سپس در بخش search فاز قبل امکان وارد کردن سابجکت برای سرچ انگلیسی را به صورت عددی را قرار دادیم. به این ترتیب هنگام بازیابی، شماره داک هایی که بازیابی شده‌اند را بر اساس تگ‌های phase1_tags.data فیلتر می‌کنیم و فقط داک‌هایی را که موضوع مورد نظر کاربر را دارند برمی‌گردانیم.