به نام او

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده علوم کامپیوتر

روزبه بازرگانی

9513703

مباحثی در علوم کامپیوتر

تمرین اول

استاد درس: دکتر اکبری

پاییز 1399

هدف:

در این پروژه هدف پیاده سازی کد پایتون برای تحلیل احساسات جریان داده‌های میکروبلاگ است.

به این منظور توییت‌هایی در مورد تجربه مسافران از پرواز با خطوط هوایی آمریکا مورد تحلیل قرار گرفتند.

فایل‌ها:

علاوه بر این گزارش‌کار، کد مربوطه (main.ipynb) نیز وجود دارد. برای اجرا شدن فایل، باید در کنار کد، پوشه شامل هر سه اکسل را با نام همان فایل زیپ شده قرار دهید. همچنین می‌توان در قسمت extracting data مسیر فایل‌ها را عوض نمود. سپس برنامه قابل اجرا خواهد بود.

توجه: در صورت اجرا نشدن فایل به بودن کتابخانه‌ها و همچنین دانلود stopwords در تابع preprocess توجه نمایید. در صورت نبود می‌توان کد زیر را از کامنت خارج و اجرا کرد.

```
nl.download('stopwords')
```

شرح کار:

با وجود اینکه شرح کار در کد به صورت تیتروار در هر قسمت آمده و تحلیل اتفاقات افتاده در کد نیز با توجه به اسم تابع‌ها و کامنت‌ها قابل برداشت است، به صورت خلاصه در اینجا توضیح داده خواهد شد.

در گام اول، کتابخانه‌های مورد نیاز فراخوانی شده اند. سپس به استخراج داده‌ها پرداختیم. خروجی به صورت زیر مپ شد هر چند در انتها نیازی نبود و می‌توانستیم از خود positive، negative و neutral استفاده کنیم. در هر صورت:

positive -> 1 negative -> -1 neutral -> 0

همچنین متن‌های توییت‌ها را استخراج نمودیم. در ابتدا preprocessing نمودیم که شامل فرایندهای زیر بود:

1) removing user names

2) removing numbers

3) removing URLs

4) removing punctuations

5) removing stopwords

6) removing words with length less than 3

7) transform to the lower case

8) stemming

نیاز به توضیح خاصی نیست تنها شاید هدف ریشه‌یابی مهم باشد که دلیل آن مثلا تبدیل کلماتی همچون appreciate, appreciated, appreciating به appreci است زیرا مفهوم آن‌ها همگی یکی است. در این گام تنها مشکل وجود کلمه تهی بود که در نهایت به طور دستی آن را در تابع word_counter با یک if برطرف کردم.

در گام بعد از شمردن تعداد کلمات در هر اظهار نظر مثبت، منفی و خنثی، از chi-square استفاده کردیم. پیاده سازی آن در قالب دیکشنری برای هر کلمه بود و از روابط لکچر بهره برده شد. فرض ما این بود که کلمات دارای مقداری بیشتر از 0.1 دارای اهمیت هستند و آن‌ها را نگه داشتیم.

سپس از طبقه‌بند naive bayes استفاده شد. در این گام، بدون استفاده از کتابخانه و با کد پیاده‌سازی شد. ابتدا احتمال هر کلمه برای هر اظهار نظر و در انتها رابطه برای naive bayes استفاده شد. در خود تابع naive bayes تنها بقیه توابع صدا زده شدند. تابع طبقه بند به نحوی نوشته شد که یک لیست از جملات را بگیرد و در مورد آن‌ها اظهار نظر کند. همچنین از رابطه زیر استفاده شد تا به احتمال صفر برای کلمات جدید که در train set نبودند، نرسیم.

temp_dic[word]= (frequency + 1) / (freq + len(dic))

گرفتن ورودی آنلاین:

در قسمت Online data processing می‌توانید جملات مد نظر خود را در لیست وارد نمایید و سپس پاسخ تک تک آن‌ها را با توجه به قرارداد خروجی که گفتیم و آنجا هم نوشته شده، دریافت کنید.

نتایج:

در قسمت Accuracy and confusion matrix می‌توان نتایج و دقت کلی را در قالب جدول و خروجی مشاهده کرد. برای هر سه دیتا ست (train, dev, and test) ماتریس محاسبه شده است. نحوه محاسبه به این صورت است قرارداد کردم مقدار واقعی اول و مقدار حدس زده شده دوم بیاید. به طور مثال pos_neg تعداد دیتاهایی است که در واقعیت positive بودند ولی negative حدس زده شده‌اند. بقیه نیز به همین منوال. سپس همانطور که از روابط هم مشخص است 4 پارامتر tp، fp، fn و tn محاسبه شده‌اند که به منظور از هر کدام در زیر آمده است.

tn -> true negative, fn -> false negative

fp -> false positive, tp -> true positive

روابط محاسبه accuracy، recall، precision و F_1 نیز در تابع param_calculator آمده است.

خروجی train:

```
Accuracy of train set =  75.911
```

Confusion Matrix (positive)	correct	not-correct
selected	1355	987
not-selected	92	6350

Confusion Matrix (negative)	correct	not-correct
selected	3944	246
not-selected	1603	2991

Confusion Matrix (neutral)	correct	not-correct
selected	1369	883
not-selected	421	6111

	accuracy	precision	recall	F_1
positive	0.87716	0.57857	0.93642	0.71523
negative	0.7895	0.94129	0.71101	0.81011
neutral	0.85155	0.6079	0.7648	0.67739

خروجی dev:

```
Accuracy of dev set =  67.964
```

Confusion Matrix (positive)	correct	not-correct
selected	372	433
not-selected	78	2045

Confusion Matrix (negative)	correct	not-correct
selected	1221	134
not-selected	584	989

Confusion Matrix (neutral)	correct	not-correct
selected	397	371
not-selected	276	1884

	accuracy	precision	recall	F_1
positive	0.82548	0.46211	0.82667	0.59283
negative	0.75478	0.90111	0.67645	0.77278
neutral	0.77903	0.51693	0.5899	0.55101

خروجی test:

```
Accuracy of test set =  69.501
```

Confusion Matrix (positive)	correct	not-correct
selected	396	398
not-selected	70	2064

Confusion Matrix (negative)	correct	not-correct
selected	1264	126
not-selected	562	976

Confusion Matrix (neutral)	correct	not-correct
selected	375	369
not-selected	261	1923

	accuracy	precision	recall	F_1
positive	0.84016	0.49874	0.84979	0.62857
negative	0.76503	0.90935	0.69222	0.78607
neutral	0.78484	0.50403	0.58962	0.54348

Future work:

شاید برای این روش بتوان روی قسمت preprocessing و همینطور تاثیر تغییر مقدار 0.1 که برای تاثیرپذیری کلمات و حذف کلمات با chi-square کمتر بود، وقت گذاشته و بررسی نمود.

به دلیل تنگی وقت و شلوغی برنامه‌ها بنده قادر به بررسی نبودم.

با سپاس