به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

پردازش زبان‌های طبیعی

تمرین دوم

نام و نام خانوادگی : حسین سیفی

شماره دانشجویی : ۸۱۰۱۰۰۳۸۶

فروردین ۱۴۰۱

فهرست

سیستم تشخیص پیام اسپم: ۳
پیش پردازش: ۳
استخراج ویژگی: ٤
آموزش طبقه بند: ٥
ارزیابی: ٥

سیستم تشخیص پیام اسپم

در طراحی این سیستم از یک طبقه بند Naïve Bayes و مجموعه داده UCIML استفاده می‌کنیم. در ادامه مراحل پیش پردازش تا آموزش و تست طبقه بند را به تفکیک توضیح خواهیم داد.

پیش پردازش

در ابتدا برای شروع پردازش مجموعه داده و استفاده از آن نیاز به بارگزاری فایل شامل داده‌ها که در قالب جدول با فرمت CSV هستند داریم. برای انجام این کار از کتابخانه Pandas و تابع read_excel استفاده می‌کنیم تا مجموعه داده به شکل یک دیتافریم در محیط پایتون بارگزاری شود. کد استفاده شده برای این بخش به شکل زیر می‌باشد:

```
import pandas as pd
mails = pd.read_excel("spam.xlsx")
```

مجموعه داده مورد نظر که برای آموزش و تست مدل مورد نظر استفاده خواهد شد شامل متن پیام به شکل خام می‌باشد که با این شکل قابل پردازش نیست و نمی‌توان از آن برای آموزش مدل استفاده کرد. با این منظور ابتدا باید روی این داده‌ها عمل پیش پردازش را انجام دهیم. عمل پیش پردازش انجام شده شامل توکن‌سازی متن پیام‌ها می‌باشد. با توجه به شرایط مجموعه داده، کاربرد آن و مدل مورد نظر برای آموزش(Naïve Bayes) نیازی به حذف علائم نگارشی(Punctuationها) وجود ندارد و تنها با اعمال توکن سازی می‌توان اطلاعات مورد نیاز را استخراج کرد و ویژگی‌های(Featureها) مورد نیاز برای آموزش یک مدل مناسب با دقت و فراخوانی قابل قبول را ایجاد کرد. این توکن سازی با استفاده از کتابخانه NLTK و اعمال تابع word_tokenize روی متون قابل انجام است. کد استفاده شده بدین منظور به شرح زیر می‌باشد:

```
text = np.array([word_tokenize(str(s).lower()) for s in list(mails['text'])])
```

در این قطعه کد به ازای هر یک از پیام‌های موجود در مجموعه داده آنرا به حروف کوچک تبدیل می‌کنیم و سپس با استفاده از تابع و کتابخانه ذکر شده توکن سازی می‌کنیم.

همچنین در این مرحله برچسب‌های موجود در مجموعه داده را در لیستی جدا ذخیره می‌کنیم تا برای آموزش مدل مورد نظر قابل استفاده باشند. کد این بخش به شرح زیر می‌باشد:

```
label = np.array(mails['label'])
```

پیش پردازش در این پروژه (و هر پروژه مشابه دیگری که مجموعه داده‌ای به شکل فوق داشته باشد) امری حیاتی محسوب می‌شود چرا که استفاده از مجموعه داده خام به شکلی که متن خام و برچسب ها در کنار هم در یک دیتافریم قرار گرفته باشند برای کاربردی مثل یادگیری ماشین و به خصوص الگوریتم Naïve Bayes غیر قابل استفاده است. همچنین پیش پردازش می‌تواند باعث حذف مقداری از داده‌های اضافی و بلااستفاده شود و حجم داده‌های آموزش را کم کند.

قابل ذکر است که عمل نرمال‌سازی و حذف علائم نگارشی نیز به کمک کتابخانه nltk و تابع isalpha روی مجموعه داده اعمال شده که نتیجه آن کاهش حدود ۴ درصدی دقت و فراخوانی بود. این کار به کمک دستور زیر در جایگزینی دستور دوم بخش پیش پردازش انجام پذیرفت:

```
text = np.array([[w for w in word_tokenize(str(s).lower()) if w.isalpha() or w in ['!', '£', '$', '€']] for s in list(mails['text'])])
```

همانطور که مشخص است این کد به صورتی نوشته شده است که تمامی علائم نگارشی(به غیر از علامت تعجب و علائم واحدهای پولی) و کلمات نادرست را حذف کند اما در نهایت باعث کاهش کیفیت و ضعیف شدن نتیجه نهایی پروژه شد. این ضعف می‌تواند به دلیل استفاده از کلمات غیر ملموس و علائم نگارشی فراوان در پیام‌های اسپم باشد و با نرمال‌سازی طول پیام‌های اسپم را به شدت کاهش می‌دهد. از آن‌جایی که(همانطور که گفته خواهد شد) طول پیام یکی از ویژگی‌های مورد نظر ما برای این پروژه است، این کاهش طول به مقدار معیارهای ارزیابی صدمه وارد خواهد کرد.

استخراج ویژگی

برای آموزش مدل Naïve Bayes نیاز به ویژگی‌هایی از جنس عدد داریم. همچنین این مدل‌ها همانند الگوریتم‌های شبکه عصبی قابلیت استخراج خودکار ویژگی‌ها را ندارند اما خوشبختانه طبق تعریف موجود برای این مدل خاص، برای آموزش آن‌ها و به دست آوردن دقت مطلوب نیاز به تعداد زیادی ویژگی وجود ندارد و با تعداد ویژگی کم می‌توان به هدف مورد نظر دست یافت. در ادامه به چند مورد از ویژگی‌های استخراج شده از مجموعه داده UCIML می‌پردازیم و بررسی می‌کنیم که آیا این ویژگی‌ها نیاز ما را برطرف می‌کنند یا خیر. این ویژگی‌ها به صورت دستی و با بررسی الگوهای موجود در پیام‌های اسپم و غیراسپم موجود در مجموعه داده به دست آمده‌اند.

۱. **علامت تعجب**: در بسیاری از پیام‌های اسپم علامت تعجب به عنوان عنصری برای جلب توجه مورد استفاده قرار می‌گیرد. در بررسی پیام‌های مجموعه داده که انجام گرفت مشخص شد که نسبت بسیار بالایی از پیام‌هایی که شامل علامت تعجب هستند در دسته پیام‌های اسپم قرار می‌گیرند.
۲. **علائم واحدهای پول(£،$و€)**: تعداد زیادی از پیام‌های اسپم با وعده‌های مالی برای ترغیب دریافت کننده به کلیک روی لینک یا شرکت در فرآیند گفته شده ارسال می‌شوند و طبیعتا برای نشان دادن این مبالغ نیاز به علائم واحدهای پول مانند یورو، پوند و دلار وجود دارد (در این مجموعه داده بیشتر علامت واحد پولی یورو دیده می‌شود) در نتیجه این علائم نیز می‌توانند دلیلی برای اسپم بودن یک پیام باشند. همچنین درصد بسیار بالایی از پیام‌های شامل علامت یورو در این مجموعه داده در دسته اسپم قرار گرفته‌اند.
۳. **کلمه Urgent**: کلمه Urgent نیز یکی دیگر از کلماتی است که با استدلال مشابه با واحدهای نقدینگی اما در بعضی موارد مجزا نشانگر پیام‌های اسپم است.
۴. **وجود لینک**: یکی از ویژگی‌های مشخص و بارز پیام‌های اسپم این است که این دسته از پیام‌ها شامل لینک هستند. اگرچه لینک به تنهایی نشانه اسپم بودن نیست و در پیام‌های عادی نیز ممکن است شاهد وجود لینک باشیم اما لینک ها در درصد قابل توجهی از پیام‌های اسپم وجود دارد. دسته‌های زیادی از پیام‌های اسپم که با کاربردهای متفاوتی از جمله phishing و skamming ارسال می‌شوند شامل لینک می‌باشند و این موارد لینک را به گزینه مناسبی جهت حضور در ویژگی‌های سیستم تشخیص اسپم تبدیل می‌کند. به کمک قطعه کد زیر و عبارات منظم می‌توان تعداد لینک را در یک پیام تشخیص داد.

```
def Find(string):
    regex = r"(?i)\b((?:https?://|www\d{0,3}[.]|[a-z0-9.\-]+[.][a-
z]{2,4}/)(?:[^\s()<>]+|\(([^\s()<>]+|(\([^\s()<>]+\)))*\))+(?:\(([^\s()<>]+|(\([^\s()<>]+\)))*\)|[^\s`!()\[\]
{};:'\".,<>?«»“”‘’]))"
    url = re.findall(regex,string)
    return [x[0] for x in url]
```

۵. **طول پیام**: در مجموعه داده UCIML دیده می‌شود که پیام‌های اسپم به دلیل شامل شدن لینک و کلمات اغواگر مقداری طولانی‌تر از پیام‌های عادی هستند به همین دلیل می‌توان از طول پیام به عنوان ویژگی مناسبی برای تشخیص پیام‌های اسپم استفاده کرد.

۶. **کلمه Free**: برای کلمه Free نیز استدلال مشابهی با کلمه Urgent می‌تواند برای طبقه بند ما مورد استفاده قرار بگیرد.

آموزش طبقه بند

برای آموزش این طبقه بند مطابق قطعه کد زیر از کتابخانه sklearn و کلاس Naive_bayes استفاده می‌کنیم و با فرخوانی Constructor این کلاس با نام GaussianNB مدل مورد نظر را ایجاد می‌کنیم.

```
from sklearn.naive_bayes import GaussianNB
NB = GaussianNB()
```

سپس با استفاده از کتابخانه sklearn و کلاس model_selection و تابع train_test_split مجموعه داده UCIML را با نسبت ۲۰/۸۰ به دو مجموعه داده آموزش و ارزیابی را همراه با برچسب متناظر آن‌ها به صورت تصادفی تقسیم بندی می‌کنیم. این تابع نه تنها تقسیم داده را به راحتی و کاملا تصادفی انجام می‌دهد بلکه از هرگونه اتفاق مشابه test in train set جلوگیری می‌کند.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, label, test_size=0.2, shuffle=True)
```

در نهایت می‌توانیم با استفاده از تابع fit مدل ایجاد شده را به کمک مجموعه داده تفکیک شده و برچسب‌های آن آموزش دهیم.

```
NB.fit(X_train,y_train)
```

همانطور که در قسمت قبل توضیح داده شد، لیست X_train شامل مجموعه داده آموزش و لیست y_train شامل برچسب‌های متناظر با آن‌هاست.

برای پیش بینی برچسب متناظر با داده‌های تست می‌توان از تابع predict مدل آموزش داده شده استفاده کرد که به شکل زیر انجام می‌گیرد.

```
y_pred = NB.predict(X_test)
```

برچسب‌های پیش بینی شده این مدل در لیست y_pred ذخیره می‌شود و برای مراحل ارزیابی مدل مورد استفاده قرار می‌گیرد.

ارزیابی

در آخرین مرحله آموزش طبقه بند، مدل آموزش داده شده برای داده‌های تست برچسب‌هایی را پیش بینی کرد. در این بخش به ارزیابی مدل با استفاده از برچسب‌های تولید شده و معیارهایی مانند Recall، Precision، F1 و Accuracy می‌پردازیم. این معیارها با کمک توابع مرتبط در کتابخانه sklearn و کلاس metrics قابل محاسبه هستند. در جدول زیر معیارهای ارزیابی محاسبه شده برای هر یکی از کلاس‌ها را به صورت جداگانه مشاهده می‌کنید:

	Non-Spam	**Spam**
Precision	۰.۹۵۳	۰.۸۵۶
Recall	۰.۹۷۸	۰.۷۲۵
F1	۰.۹۶۶	۰.۷۸۵

همانطور که در جدول فوق قابل مشاهده است، مدل ما برای کلاس غیر اسپم مقادیر دقت، فراخوانی و F1 بهتری را ارائه می‌کند که با توجه به کوچک تر بودن اندازه کلاس اسپم در مجموعه داده داده شده(UCIML) نسبت به کلاس غیر اسپم(۷۴۷ داده اسپم و ۴۸۲۵ داده غیر اسپم) این مقدار عادی به نظر می‌رسد. همچنین کم بودن مقدار بعضی معیارهای ارزیابی برای کلاس اسپم نسبت به کلاس دیگر اهمیت چندانی ندارند

برای مثال برای ما بسیار مهم‌تر است که recall کلاس غیر اسپم بسیار بالاتر باشد چرا که طبقه بندی پیام‌های غیر اسپم در کلاس صحیح و برچسب زدن چند اسپم به عنوان غیر اسپم اهمیت بسیار بالاتری نسبت به حالت برعکس این مورد دارد.

در جدول زیر نیز معیارهای ارزیابی کلی برای مدل آموزش داده شده ذکر شده‌اند:

Metric	Value
Accuracy	۰.۹۴۱
Micro Precision	۰.۹۴۱
Macro Precision	۰.۹۰۵
Micro Recall	۰.۹۴۱
Macro Recall	۰.۸۵۲
Micro F1	۰.۹۵۲
Macro F1	۰.۸۷۵

دلیل اختلاف قابل توجه بین مقادیر ماکرو و میکرو نیز همان چیزی است که قبلا ذکر شد، کوچک بودن اندازه داده‌های اسپم نسبت به غیر اسپم! در واقع مقادیر ماکرو میانگین بین دقت کلاس‌هایی است که در جدول قبل ذکر شده بودند اما مقادیر میکرو معیار را به صورت تجمیع شده روی تمامی داده‌های تست محاسبه می‌کند.

سیستم تشخیص دروغ