



* کد و فایل های خود را به صورت زیپ با فرمت `fipr_[Fullname]_[Studentcode].zip` در سایت بارگذاری کنید.

هدف از این پروژه به کارگیری مطالب آموخته شده در طول ترم، روی یک مسئله واقعی و بررسی روش های مختلف در یادگیری ماشین روی مجموعه دادگانی خاص است.

زبان برنامه نویسی قابل استفاده در این پروژه python است و استفاده از زبان های برنامه نویسی دیگر مجاز نیست. تمامی کد های خود را باید در فضای Jupyter notebook بنویسید. با توجه به اینکه در بخش هایی از پروژه ممکن است حجم محاسباتی زیادی احتیاج داشته باشید، پیشنهاد می شود از سرویس google Collab استفاده کنید. برای الگوریتم های یادگیری ماشین فقط مجاز به استفاده از پکیج scikit learn هستید. در استفاده از پکیج های پردازش داده های متنی پیشنهاد می شود از پکیج NLTK استفاده کنید. استفاده از پکیج های یادگیری ژرف مانند TensorFlow و Pytorch مجاز نیست.

(۱) معرفی مسئله و مجموعه دادگان

مسئله تعریف شده در این پروژه در رابطه با تجزیه و تحلیل احساسات در متن های واقعی است. هدف این مسئله دسته بندی، که جزو مسائل حوزه پردازش زبان طبیعی قرار می گیرد، استخراج مثبت یا منفی بودن بار معنایی و احساسی یک متن می باشد. هدف این پروژه، ایجاد مدل هایی است که یک متن را به عنوان ورودی گرفته و مثبت یا منفی بودن بار احساسی آن را تشخیص می دهند. مجموعه دادگانی که در اختیار شما قرار می گیرد، متشکل از ۴۵ هزارتا از نظرات کاربران یک شبکه اجتماعی است که دارای برچسب های مثبت و منفی هستند. شما باید از این مجموعه داده برای آموزش مدل هایتان استفاده کنید و مجاز به استفاده از هیچ مجموعه داده کمکی برای بهبود نتیجه هایتان نیستید. استفاده از هر مجموعه داده کمکی تخلف در نظر گرفته می شود. این داده ها دارای دو ستون comment و sentiment هستند. ستون sentiment دارای مقادیر negative و positive است که منفی یا مثبت بودن بار معنایی comment را مشخص می کند. در تمامی مدل سازی ها، کلاس منفی را کلاس ۰ و کلاس مثبت را کلاس ۱ در نظر بگیرید.

(۲) تولید داده مناسب و قابل پردازش

معمولاً داده های متنی به صورت خام، بی نظمی و یا حالات خاص مختلفی دارند که تحلیل آن ها را دشوار می کند. به همین علت برای استفاده از آن ها، ابتدا پیش پردازش هایی روی آنها انجام می شود. همچنین برای استفاده از بیشتر مدل های یادگیری ماشین نیاز داریم که داده هایمان در قالب عدد بیان شوند. به همین علت در چنین مسائلی که داده ی متنی داریم، ابتدا باید آن ها را در قالب عددی بیان کنیم. برای این تبدیل، روش های زیادی وجود دارد که در ادامه با برخی از آن ها بیشتر آشنا می شویم.

۱.۲ پیش پردازش داده ها

پیش پردازش داده های متنی وابسته به منبعی که داده از آن بدست آمده اند و میزان نویزی که در داده های خام وجود دارد، می تواند دارای مراحل مختلفی باشد. برای پیاده سازی این روش ها معمولاً می توان از کتابخانه های آماده ای نظیر NLTK در python کمک گرفت. روش های ابتدایی پیش پردازش شامل تبدیل حروف به حالت lower-case و حذف اعداد یا تبدیلیشان به حروف، حذف کاراکترهای اضافی و جداسازی کلمات از هم است. از پیش پردازش های سطح بالاتر می توان به حذف Stop-word ها و انجام Stemming و Lemmatization روی کلمات بدست آمده اشاره کرد.

چنین پردازش هایی به این خاطر انجام می گیرند که خیلی از کلمات و اجزای متن لزوماً ویژگی های خوبی نیستند و اطلاعات اضافی را به همراه ندارند. به عنوان مثال حروف بزرگ درون کلمات اطلاعات خاصی را به همراه ندارند و مشابه حالتی هستند که همه حروف کوچکند؛ به همین خاطر تمام حروف را lower-case می کنیم. یا مثلاً کلمات پرتکراری همانند is، am، the و ...، در بیشتر جملات وجود دارند و در نتیجه ویژگی خوبی محسوب نمی شوند.

در بخش سوم صورت پروژه که مدل سازی های مورد نظر نوشته شده است، باید از این روش های پیش پردازش برای بهبود عملکرد مدل ها استفاده کنید. برای مشاهده تاثیر عملکرد این روش ها باید مدل سازی های خواسته شده را برای سه حالت بدون پیش پردازش داده ها، پیش پردازش ابتدایی و پیش پردازش سطح بالا انجام داده و نتایج را مقایسه نمایید.

۲.۲ تبدیل به بردار

پس از مرحله پیش پردازش، نوبت به اختصاص برداری عددی به هر نمونه متنی می رسد. برای این منظور از روش هایی مانند Word2vec و Bag of Words استفاده می شود. این روش ها هر کلمه را به یک بردار عددی با طول ثابت تبدیل می کنند. سپس می توان با روش های مختلفی بردار مربوط به کل متن را از روی بردار کلماتش بدست آورد. برای مثال یکی از این روش ها، میانگین گرفتن از بردار کلمات است.

۳ مسئله دسته بندی نظرات کاربران

۱.۳ بررسی عملکرد روش های پیش پردازش

در این بخش ابتدا می خواهیم تاثیر پیش پردازش متن ها را روی عملکرد مدل ها بسنجیم. به این منظور دو مدل مبتنی بر روش های Logistic regression و k-NN را در سه حالت مختلف آموزش داده و نتایج را مقایسه کنید. همچنین برای بردارسازی، روش bag of words را بکار ببرید.

حالت اول: بدون پیش پردازش متن

حالت دوم: پیش پردازش ابتدایی شامل تبدیل حروف به حالت lower case، حذف اعداد، حذف کاراکترهای اضافی و جداسازی کلمات

حالت سوم: پیش پردازش سطح بالا شامل تمامی مراحل حالت دوم به اضافه حذف Stop-word ها، Stemming و Lemmatization. علاوه بر این روش ها، برای این بخش به مراجع موجود در اینترنت برای پیش پردازش داده های متنی مراجعه کنید و اگر روش های دیگری نیز یافتید که به نتیجه مدل سازی کمک کند، آن ها را در این

بخش پیاده سازی کنید. همچنین دلایل استفاده از این روش ها را در گزارش خود شرح دهید (مراجع استفاده شده در این بخش را در گزارش خود بیاورید).

در این بخش از پروژه زمان زیادی برای تنظیم دقیق ابرپارامترهای مسئله نگذارید. برای تعداد مناسبی حالت، این مدل سازی ها را انجام داده و بهترین نتیجه هر بخش را گزارش کنید. برای آشنایی بیشتر با شیوه پیش پردازش داده های متنی به لینک زیر مراجعه کنید:

- **Text Preprocessing for NLP, Beginners to Master**

۲.۳ بررسی عملکرد روش های بردار سازی و تنظیم دقیق ابرپارامترها

در این بخش تاثیر هر یک از روش های بردار سازی داده های متنی را بر کیفیت مدل سازی بررسی می کنیم. الگوریتم های Logistic regression و k-NN را با دو روش بردار سازی Word2vec و Bag of Words بکار گرفته و نتایج مدل سازی را در گزارش خود بیان کنید. برای پیش پردازش این بخش از پیش پردازش **حالت سوم** بخش قبل استفاده کنید. همچنین با تنظیم دقیق ابرپارامترهای مدل ها، برای هر یک از دو الگوریتم یادگیری، بهترین مدل بدست آمده را گزارش کنید. در گزارش خود مقدار این پارامتر ها را به همراه روش بردار سازی ای که به بهترین نتیجه می رسد، ارائه دهید.

مدل های پکیج scikit learn توسط ماژول pickle در python، قابل ذخیره سازی هستند. برای هر یک از دو الگوریتم بیان شده در بالا، مدل با بیشترین دقت دسته بندی را با نام های LR.pkl و kNN.pkl ذخیره کنید.

۳.۳ یافتن بهترین مدل (امتیازی)

در این بخش از الگوریتم MLP، برای مدل سازی استفاده کنید. پارامتر هایی که برای شبکه در نظر می گیرید (تعداد لایه و نورون های هر لایه، تابع های فعال سازی و ...) را به طور کامل در گزارش خود بنویسید. در این بخش، محدودیتی در پیش پردازش داده ها ندارید و برای بردار سازی نیز می توانید روش هایی غیر از Word2vec و Bag of Words را بکار ببرید. شرح کوتاهی از شیوه کارکرد روش هایی که در هر ماژول استفاده می کنید را با ذکر مرجع گزارش دهید. در پایان بهترین مدلی که یافته اید را با نام best.pkl ذخیره کنید.

۴) ثبت خروجی مدل ها

برای تمام مدل هایی که در این پروژه آموزش می دهید، خروجی تابع analysis (.,.) که تعریف آن در ادامه می آید را برای داده های آموزش و آزمون در کد و گزارش خود ثبت کنید. این تابع، تحلیلی از عملکرد دسته بندی که آموزش داده اید را نشان می دهد و معیار خوبی برای بررسی عملکرد مدل هاست.

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

def analysis(labels, predictions):
    print( Report: "Classification\n", classification_report(labels,
        predictions, target_names=["positive", "negative"]))
    print( Matrix: "Confusion\n", confusion_matrix(labels, predictions))
    print( "Accuracy:\n", accuracy_score(labels, predictions))
```

کد هر بخش از پروژه را در Cell متفاوتی از فضای Jupyter Notebook قرار دهید تا ارزیابی آن ها دقیق تر انجام شود. به منظور صرفه جویی در زمان خود و افزایش خوانایی، کد های خود را به صورت Modular بنویسید. گزارش خود را در قالب یک فایل pdf به همراه کدها و مدل های نهایی ای که ذخیره کرده اید، ارسال کنید. در نگارش گزارش، به موارد زیر توجه کنید:

- برای حالت سوم بخش ۱.۳، شرح کوتاهی از روش هایی که استفاده کرده اید را بیان کنید .
- در بخش ۲.۳، شرح کوتاهی از شیوه عملکرد روش های مختلف بردارسازی را بیان کرده و بررسی کنید که کدام روش با کدام الگوریتم به نتیجه بهتر رسیده است. همچنین نتایج بدست آمده از الگوریتم های مختلف را با هم مقایسه کنید.
- در بخش ۳.۳، اگر روشی غیر از روش های بکار رفته در بخش های ۱.۳ و ۲.۳ را استفاده کرده اید، آن را شرح دهید.
- در پایان به این نکته توجه کنید که نگارش دقیق و کامل گزارش، تاثیر مثبتی در نمره شما دارد.