بسمه تعالی

گزارش پروژه یادگیری ماشین

استاد :

دکتر ابوالفضل مطهری

نویسنده

عرفان صدرائیه 99101835

دانشگاه صنعتی شریف

بهار ۱۴۰۲

در این پروژه ما می‌خواهیم که برای هر عبارت یا جمله، یک برچسب پیش‌بینی کنیم که بین ۰ تا ۴ به آن عبارت از نظر توضیحی که داده است امتیاز می‌دهد. مثلا جمله 'آن فوق العاده بود' باید امتیاز ۳ یا ۴ بگیرد.

ما در این پروژه ابتدا داده های خود را به ۳ قسمت آموزش، اعتبارسنجی و آزمایشی تقسیم کردیم. ابتدا باید هرجمله را به صورت بردار عددی در بیاوریم، سپس آن بردارها و برچسب های مربوطه را به یک مدل آموزشی بدهیم و سعی کنیم با استفاده از داده های اعتبارسنجی، هایپرپارامتر ها را تنظیم کنیم و در آخر دقت مدل را روی داده تست بدست آوریم.
برای تبدیل هر جمله به بردار، از رویکرد ها و راه های متفاوتی می‌توانیم استفاده کنیم. ما در این پروژه از دو رویکرد متفاوت استفاده کردیم:

1. TF-IDF

TF-IDF یک تکنیک پرکاربرد برای تبدیل اسناد متنی به بردارهای عددی است. هدف اصلی آن، گرفتن اهمیت کلمات در یک سند نسبت به یک مجموعه از اسناد است. این روش به این صورت عمل می‌کند:

- TF (Term Frequency) : تعداد بارهایی را که یک ترم (کلمه) در یک سند ظاهر شده است، می‌سنجد و وزنی برای آن ترم درون سند تعیین می‌کند.
- IDF (Inverse Document Frequency) : اهمیت یک ترم در کلیه اسناد مجموعه را اندازه‌گیری می‌کند. به هر ترم وزنی اختصاص می‌دهد که بر اساس کمیت آن در کلیه مجموعه اسناد تعیین می‌شود. ترم‌های کمتر متداول وزن‌های بیشتری دریافت می‌کنند، در حالی که ترم‌های پرتکرار وزن‌های کمتری را دریافت می‌کنند.
- TF-IDF : مفهوم‌های TF و IDF را با یکدیگر ترکیب می‌کند تا هر سند را به یک بردار عددی تبدیل کند. این روش TF را در IDF ضرب می‌کند.

بردار TF-IDF حاصل، سند را نمایش می‌دهد و هر جزء از بردار نشان‌دهنده اهمیت یک ترم خاص در سند است. TF-IDF تکنیکی است که منحصر به فرد برتری‌های هر ترم را درون یک سند و همچنین ارتباط آن با سایر اسناد نشان می‌دهد.

2. FastText:

FastText یک توسعه‌دهنده از مدل word2vec است که به طور خاص برای رسم نمایش کلمات واژه به واژه و همچنین واحدهای زیر واژگانی (subword) طراحی شده است. این روش به این صورت عمل می‌کند:

- این مدل کلمات، زیرواژگان و جملات را می‌تواند به بردار با طولی ثابت تبدیل کند. جمله حاصل جمع یا میانگین یا عملگر های دیگر روی واژگان و زیرواژگان جمله است. چون این مدل از زیرواژگان استفاده می‌کند، پس می‌توانیم کلماتی که در داده های آموزش آن را ندیده ایم را نیز بصورتی نمایش دهیم.

ما در این پروژه از تبدیل جمله به بردار این مدل استفاده کرده ایم.

هر دو TF-IDF و FastText روش‌های محبوبی برای تبدیل جملات به بردارها هستند، اما در اصول و نمایش‌های اصلی آنها تفاوت‌هایی وجود دارد. TF-IDF بر تأثیر کلمات در یک سند تمرکز می‌کند، در حالی که FastText با استفاده از نمایش‌های کلمه و واحدهای زیر واژگانی، به درستی کنترل واژه‌های دیده نشده یا کمتر دیده شده را مدیریت می‌کند و ساختار کلمات درون جملات را به طور موثری ضبط می‌کند. ما هر جمله را با دو روش به بردار تبدیل می‌کنیم و هر دو روش را بررسی می‌کنیم.

لیست پکیج های استفاده شده:

- Fasttext:
  - این پکیج برای آموزش دادن مدل برای یادگیری زبان داده های ما و تبدیل جملات به بردار های عددی است.
- Nltk:
  - در پیش‌پردازش داده‌ها همانطور که در داک پروژه گفته شد، نیاز به حذف stopword ها و حذف علائم نگارشی و پیدا کردن ریشه کلمات بود و ما ازین پکیج برای حذف این موارد استفاده کردیم.
- Torch:
  - با استفاده از این پکیج شبکه عصبی دلخواه خود را ساختیم و آن را آموزش دادیم و برای پیش‌بینی برچسب های داده آموزش استفاده کردیم.
- Sklearn:
  - ازین پکیج برای ارزیابی داده های پیش‌بینی شده و همچنین مدل های آماده برای آموزش روی داده ها استفاده کردیم.

مشکل نامتعادل بودن دسته ها

هنگامی که داده ها را بررسی کردیم مشاهده کردیم که اغلب برچسب ها به کلاس ۲ تعلق دارند و کلاس ها بالانس نیستند. از روش هایی مثل downsampling, upsampling نیز استفاده کردم ولی این کار باعث شد که دقت مدل کاهش یابد. برای همین این روش در مدل نهایی آورده نشده است.

پیش‌پردازش داده ها

در این مرحله ما ابتدا جملات را توکنایز کردیم و سپس تمام حروف را به حروف کوچک تبدیل کردیم. سپس stopword ها را حذف کردیم و بعد از آن علائم نگارشی را حذف کردم. در آخر نیز هر کلمه را با stemming به ریشه‌اش تبدیل کردیم.

نحوه استخراج ویژگی از داده‌ها در رویکرد ۲

در این رویکرد من از مدل fasttext استفاده کردم. به صورتی که کل جملات داده های آموزش را در یک فایل ذخیره کردم و مدل fasttext را روی آن به صورت unsupervised آموزش دادم تا زبان ما و رابطه کلمات را یاد بگیرد. سپس هر جمله را با استفاده از این مدل به بردار تبدیل کردم.

لیستی از مدلهای به کار گرفته‌شده، هدف از به‌کارگیری آنها و تعداد پارامترهای هر مدل

در این پروژه من مدل های SVM, Random Forest را روی داده ها اجرا کردم ولی مدت زمان اجرا طولانی بود یا با کمبود حافظه روبرو می‌شدم. در اینجا من بردارهای حاصل دو رویکرد را با مدل های شبکه عصبی، لاجستیک رگرشن و Naive Bayes تست کردیم.

- شبکه عصبی: در اینجا من از شبکه عصبی MLP با ۲ لایه پنهان استفاده کردم و مدلی را انتخاب کردم که پارامتر های آن روی داده های اعتبارسنجی بهترین دقت را داشته باشد. شبکه عصبی با استفاده از توابع غیرخطی توانایی بالایی در یادگیری روابط غیرخطی دارد و می‌توان از آن به عنوان یکی از قدرتمندترین مدل ها نام برد.
- Naive Bayes: این مدل تقریبا هایپر پارامتری ندارد. این مدل فرض می‌کند که ویژگی ها از یکدیگر مستقل هستند که در اینجا ویژگی های ما کلمات هستند(رویکرد اول). این فرض با کار ما می‌سازد و محاسبه این مدل نیز ساده است. پس استفاده از این در اینجا می‌تواند نتایج خوبی بدهد. بردار های حاصل از TF-IDF بعد زیادی پیدا می‌کنند برای همین فیت کردن مدل پیچیده روی آن ها بعضی اوقات امکان‌پذیر نیست.
- لاجستیک رگرشن: این مدل نیز بر روی داده های متنی عملکرد خوبی دارد. این مدل در مقابل داده های نامتعادل خوب عمل می‌کند و در اینجا داده ما نیز نامتعادل است. همچنین این مدل توانایی کار با داده های بزرگ را نیز دارد/

تحلیل و نتایج ارزیابی مدل

در اینجا ما به ازای هر رویکرد مدل های مختلفی را اجرا کردیم ولی بهترین عملکرد را شبکه عصبی روی داده های TF-IDF دارد. مقادیر ارزیابی این مدل به این صورت است:

- Weighted average precision: 0.62
- Weighted average recall: 0.70
- Weighted average f1-score: 0.66
- Accuracy = 70%

مدل شبکه عصبی ما قدرت زیادی دارد و توانسته نسبت به مدل های دیگر دقت بهتری را کسب کند. ولی یکی از ایراد های این مدل این است که در روش TF-IDF ما تعداد ویژگی هایمان خیلی زیاد است و در اردر ۱۰هزار است و اموزش دادن این مدل خیلی زمان بر می‌شود. من سعی کردم که با PCA ابعاد را کاهش دهم ولی با کرش کردن بدلیل پر شدن رم روبرو شدم.

دقت مدل های دیگر بر روی دو رویکرد را فقط اشاره می‌کنیم و متریک های دقیق در جوپیتر آورده شده است:

- رویکرد اول:
  - MultinomialNB = 63%
  - Logistic Regression = 63%
- رویکرد دوم:
  - Logistic Regression = 51%
  - Random Forest = 58%
    - در این روش چون داده ها زیاد بود، نتوانستم روی کل داده ها ازمایش کنم و تقریبا روی نیمی از داده ها ازمایش کردم و چون زمان بر بود نتوانستم تعداد درخت که روی ولیدیشن بهینه باشد را پیدا کنم.
  - Neural Network = 58%

مقایسه رویکردهای اولیه و نوین

در داده های ما، روش اولیه بهتر از روش نوین یا همان fasttext عمل می‌کند و مدل ها روی داده های روش اول دقت بهتری کسب می‌کنند. این به این معنی است که در داده هایی که ما داریم، وزن دهی به کلماتی که داریم و استفاده از کلمات دیده‌شده بر عملکرد ما تاثیر بهتری دارد نسبت به کاری که fasttext می‌کند و با زیرواژگان کار می‌کند.پس به عبارتی وزن دهی به کلمات با توجه به تکرارشان برای ما مفیدتر است. همچنین کیفیت fasttext به داده هایی است که روی آن آموزش داده می‌شود. وقتی حجم

داده ها کم است این مدل دقت خوبی نخواهد داشت و روی داده های کم عملکرد خوبی ندارد زیرا نمی‌تواند به خوبی با کلمه ها آشنا شود.

نتیجه گیری

شبکه عصبی با توجه به توانایی در درک روابط غیرخطی و قدرت بالای آن،با استفاده از TF-IDF که با وزن دهی به تکرار کلمات عمل می‌کند بهترین نتیجه را به ما می‌دهد که دور از ذهن هم نیست که این مدل بهترین عملکرد را داشته باشد. بعد از آن Logistic regression بهترین عملکرد را دارد که می‌دانیم این مدل نزدیک ترین مدل خطی به SVM است.

همچنین یکی از دلایل بالا نبودن کلی دقت ها داده های ما است. در داده ها مثال هایی دیده می‌شود که نمی‌توان آن ها را پیش‌بینی کرد. برای مثال ما برای ۳ جمله زیر این برچسب ها را داریم:

- ' is wondrously creative .2
- is wondrously creative .,3
- is wondrously creative,4

که مدل ما برای این جملات یک لیبل می‌دهد که منطقی است. همچنین بالانس نبودن کلاس ها مورد دیگر برای این اتفاق است.