به نام خداوند بخشنده مهربان

عنوان تمرین

اسپم یا غیر اسپم

عنوان درس

یادگیری ماشین

استاد

دکتر الهام قصرالدشتی

دستیاران آموزشی

مهرداد قصابی

مریم صفوی

گردآورنده

سید حسین حسینی

بهار ۱۴۰۴

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

مستندسازی پروژه: تشخیص ایمیل‌های هرزنامه (Spam) با طبقه‌بند بیز ساده

مقدمه

این پروژه با هدف ساخت یک مدل یادگیری ماشین برای تشخیص ایمیل‌های هرزنامه (Spam) از ایمیل‌های عادی (Ham)طراحی شده است. تشخیص هرزنامه یکی از کاربردهای کلاسیک و موفق الگوریتم‌های پردازش زبان طبیعی و طبقه‌بندی متن است. مدل ساخته‌شده، با تحلیل محتوای یک ایمیل، آن را به یکی از دو دسته "Spam" یا "Ham" طبقه‌بندی می‌کند.

در این پروژه، همانند پروژه قبلی، از الگوریتم طبقه‌بند بیز ساده (Naive Bayes Classifier) استفاده شده است. این الگوریتم به دلیل سادگی، سرعت بالا و عملکرد بسیار خوب در وظایف طبقه‌بندی متن، انتخاب محبوبی برای این کاربرد است. در اینجا نیز الگوریتم به صورت کامل از پایه و بدون استفاده از کتابخانه‌های آماده یادگیری ماشین پیاده‌سازی شده تا مراحل محاسباتی آن به طور شفاف نمایش داده شود.

داده‌های مورد استفاده از فایل spam_ham_dataset.csvخوانده شده‌اند که شامل مجموعه‌ای از ایمیل‌ها و برچسب مربوط به آن‌ها (اسپم یا غیر اسپم) است.

مراحل اجرای پروژه

پروژه شامل مراحل مختلفی از آماده‌سازی داده تا ساخت، آموزش و ارزیابی مدل است که در ادامه به تفصیل شرح داده می‌شوند.

۱ .فراخوانی کتابخانه‌ها(Libraries)

در گام نخست، کتابخانه‌های ضروری برای انجام پروژه فراخوانی می‌شوند:

- matplotlib.pyplotو seaborn: برای مصورسازی داده‌ها و رسم نمودارها، به ویژه ماتریس درهم‌ریختگی به صورت نقشه حرارتی.(Heatmap)
- Pandas:برای مدیریت و تحلیل داده‌ها در قالب دیتافریم.
- Numpy: برای کار با آرایه‌های عددی، به خصوص در ساخت ماتریس درهم‌ریختگی.
- String: برای دسترسی به لیستی از علائم نگارشی جهت حذف آن‌ها از متن.
- Math: برای استفاده از تابع لگاریتم در محاسبات احتمالاتی مدل.

- collections.defaultdict: برای ساخت دیکشنری‌هایی با مقدار پیش‌فرض، که در شمارش کلمات بسیار کارآمد است.

۲ .بارگذاری مجموعه داده(Load Dataset)

مجموعه داده از فایل spam_ham_dataset.csvخوانده و در یک دیتافریم pandasبه نام dfبارگذاری می‌شود. این دیتافریم شامل ستون‌های اصلی زیر است:

- text: محتوای کامل ایمیل که به عنوان ورودی مدل استفاده می‌شود.
- Label: برچسب ایمیل که می‌تواند "spam" یا "ham" باشد.
- label_num: نسخه عددی برچسب‌ها (معمولاً ۱ برای spam و ۰ برایham) که برای ارزیابی مدل استفاده می‌شود.

۳ .پیش‌پردازش متن(Preprocessing Function)

برای آماده‌سازی متن ایمیل‌ها جهت تحلیل، یک تابع به نام preprocess_textتعریف شده است. این تابع وظایف زیر را بر عهده دارد:

1. تبدیل به حروف کوچک(Lowercase) : یکسان‌سازی حروف برای جلوگیری از تفاوت قائل شدن بین کلماتی مانند "Free" و."free"
2. حذف علائم نگارشی(Punctuation Removal) : حذف تمام علائم نگارشی (مانند نقطه، ویرگول، علامت تعجب و غیره) از متن.
3. توکن‌سازی(Tokenization) : تقسیم متن پاک‌سازی‌شده به لیستی از کلمات مجزا (توکن‌ها).

این تابع بر روی ستون textدیتافریم اعمال شده و نتایج آن در ستون جدیدی به نام clean_textذخیره می‌شود.

۴ .تقسیم داده‌ها به دو بخش آموزش و آزمون(Split Train / Test)

برای ارزیابی بی‌طرفانه مدل، داده‌ها به دو بخش تقسیم می‌شوند. در این پروژه، این تقسیم به صورت دستی انجام شده است:

- داده‌های آموزش(Train Data) : ٪۸۰ ابتدایی داده‌ها برای آموزش مدل اختصاص داده می‌شود.
- داده‌های آزمون(Test Data) : ٪۲۰ انتهایی داده‌ها برای سنجش عملکرد مدل بر روی داده‌های جدید کنار گذاشته می‌شود.

۵ .آموزش مدل (ساخت واژگان و شمارش فراوانی‌ها)

در این مرحله، مدل Naive Bayes با استفاده از داده‌های آموزشی، پارامترهای آماری خود را یاد می‌گیرد. این فرآیند شامل مراحل زیر است:

1. ساخت واژگان(Vocabulary) : مجموعه‌ای از تمام کلمات منحصر به فرد موجود در ایمیل‌های آموزشی ایجاد می‌شود.
2. شمارش فراوانی کلمات:
   - word_counts_spam: تعداد تکرار هر کلمه در ایمیل‌های اسپم.
   - word_counts_ham: تعداد تکرار هر کلمه در ایمیل‌های عادی.
3. شمارش اسناد و کلمات:
   - spam_docsو ham_docs: تعداد کل ایمیل‌های اسپم و عادی.
   - total_words_spamو total_words_ham: تعداد کل کلمات در تمام ایمیل‌های اسپم و عادی.

۶ .محاسبه احتمالات(Probabilities)

پس از شمارش فراوانی‌ها، احتمالات مورد نیاز برای فرمول Naive Bayes محاسبه می‌شوند:

1. احتمالات پیشین(Prior Probabilities) : احتمال اینکه یک ایمیل به صورت تصادفی اسپم یا عادی باشد.
   - P_spam = spam_docs / len(train_df)
   - P_ham = ham_docs / len(train_df)
2. احتمالات شرطی(Conditional Probabilities) : احتمال مشاهده یک کلمه خاص، به شرط آنکه ایمیل از نوع اسپم یا عادی باشد. این محاسبات با استفاده از هموارسازی لاپلاس (Laplace Smoothing)انجام می‌شود تا از بروز احتمال صفر جلوگیری گردد.

۷ .پیاده‌سازی طبقه‌بند(Naive Bayes Classifier)

تابع predictبر اساس این احتمالات، یک ایمیل جدید را طبقه‌بندی می‌کند:

1. برای هر ایمیل، امتیاز لگاریتمی تعلق به دسته "spam" و "ham" محاسبه می‌شود.

2. این امتیاز با جمع لگاریتم احتمال پیشین و لگاریتم احتمالات شرطی تمام کلمات موجود در ایمیل به دست می‌آید.
3. در نهایت، دسته‌ای که امتیاز لگاریتمی بالاتری دارد، به عنوان پیش‌بینی نهایی انتخاب می‌شود. (۱ برای اسپم و ۰ برای عادی).

۸ .ارزیابی مدل و نمایش نتایج(Evaluation)

عملکرد مدل بر روی مجموعه داده آزمون سنجیده می‌شود:

1. پیش‌بینی بر روی داده‌های آزمون :تابع predictبر روی تمام ایمیل‌های مجموعه آزمون اجرا می‌شود و نتایج در لیست y_predذخیره می‌شود.
2. محاسبه دقت(Accuracy) : با مقایسه برچسب‌های واقعی (y_true) و پیش‌بینی‌شده(y_pred) ، دقت مدل محاسبه می‌شود. مدل به دقت فوق‌العاده ٪۹۸.۲۶ دست یافته است.
3. ماتریس درهم‌ریختگی(Confusion Matrix) : این ماتریس به صورت دستی و سپس با استفاده از Seabornبرای نمایش بهتر، ساخته می‌شود. مقادیر آن عبارتند از:
   - مثبت صحیح(TP) : تعداد ایمیل‌های اسپم که به درستی اسپم تشخیص داده شده‌اند (۳۰۸ مورد).
   - منفی صحیح(TN) : تعداد ایمیل‌های عادی که به درستی عادی تشخیص داده شده‌اند (۷۰۹ مورد).
   - مثبت کاذب(FP) : تعداد ایمیل‌های عادی که به اشتباه اسپم تشخیص داده شده‌اند (۶ مورد).
   - منفی کاذب(FN) : تعداد ایمیل‌های اسپم که به اشتباه عادی تشخیص داده شده‌اند (۱۲ مورد).

این نتایج نشان می‌دهد که مدل در هر دو زمینه تشخیص اسپم و عادی عملکرد بسیار دقیقی دارد و تعداد خطاهای آن بسیار کم است.

۹ .ذخیره نتایج(Save Predictions to CSV)

در انتها، پیش‌بینی‌های مدل برای داده‌های آزمون، در یک فایل CSV به نام spam_predictions_from_scratch.csvذخیره می‌شود. این فایل شامل یک ستون به نام Spamاست که مقادیر ۰ یا ۱ را در خود دارد.

جمع‌بندی و نتیجه‌گیری

این پروژه با موفقیت یک سیستم تشخیص هرزنامه را با استفاده از الگوریتم Naive Bayes پیاده‌سازی کرد. دقت بالای ٪۹۸ نشان می‌دهد که این الگوریتم، علی‌رغم سادگی مفروضاتش، برای کاربردهای طبقه‌بندی متن بسیار قدرتمند است. پیاده‌سازی مدل از پایه نیز درک عمیقی از نحوه کارکرد داخلی این الگوریتم فراهم می‌کند.

منابع و مراجع (References)

1. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
   - توضیح :منبعی کلاسیک برای یادگیری مفاهیم Naive Bayes در طبقه‌بندی متن، شامل توضیحات کامل درباره هموارسازی لاپلاس.
2. Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing*. 3rd Edition.
   - توضیح :مرجعی جامع در حوزه پردازش زبان طبیعی که به تفصیل به الگوریتم‌های طبقه‌بندی متن می‌پردازد.
3. Scikit-learn Documentation: Naive Bayes.
   - توضیح :برای مقایسه و درک عمیق‌تر انواع مدل‌های Naive Bayes ، مستندات کتابخانه Scikit-learn بسیار مفید است.
   - لینک: https://scikit-learn.org/stable/modules/naive_bayes.html
4. Spam/Ham Email Dataset (Kaggle)
   - توضیح :مجموعه داده استفاده شده در این پروژه، یکی از دیتاست‌های استاندارد برای وظایف تشخیص هرزنامه است که در پلتفرم‌هایی مانند Kaggle در دسترس عموم قرار دارد.

موفق باشید 😊😁