وکنایز کردن (Tokenization) یکی از مراحل اصلی پردازش زبان طبیعی (NLP) است که متن را به واحدهای کوچک‌تر به نام توکن‌ها تقسیم می‌کند. این توکن‌ها معمولاً کلمات، عبارات، علائم نشانه‌گذاری یا حتی کاراکترهای منفرد هستند

هدف توکنایز کردن

هدف :توکنایز کردن به منظور تقسیم متن به اجزای کوچکتر و قابل تحلیل انجام می‌شود. این فرآیند پایه‌ای برای مراحل بعدی مانند ریشه‌یابی، حذف کلمات توقف، و استخراج ویژگی‌ها است. توکنایز کردن به مدل‌های یادگیری ماشین و الگوریتم‌های NLP کمک می‌کند تا بتوانند متن را بهتر درک و پردازش کنند.

مزایای توکنایز کردن

- **سهولت پردازش :**با شکستن متن به اجزای کوچکتر، پردازش متن ساده‌تر و کارآمدتر می‌شود.
- **افزایش دقت مدل :**مدل‌های یادگیری ماشین و الگوریتم‌های NLP با داده‌های توکنایز شده دقیق‌تر عمل می‌کنند.
- **قابلیت تحلیل دقیق‌تر :**توکنایز کردن امکان تحلیل دقیق‌تر و عمقی‌تر متن را فراهم می‌کند.

ریشه‌یابی(Lemmatization)

هدف :تبدیل کلمات به شکل قاموسی یا ریشه‌ای خود. این فرآیند شامل تحلیل معنایی کلمات است تا شکل صحیح و قاموسی آن‌ها را تعیین کند.

عملکرد :ریشه‌یابی با استفاده از دیکشنری‌های زبانی و قوانین گرامری کلمات را به ریشه قاموسی‌شان تبدیل می‌کند. به عنوان مثال، runningبه runو betterبه goodتبدیل می‌شود.

استمینگ(Stemming)

هدف :حذف پیشوندها و پسوندها از کلمات برای رسیدن به ریشه کلمه. استمینگ معمولاً بدون در نظر گرفتن معنای کلمه انجام می‌شود و به شکل ساده‌تری نسبت به ریشه‌یابی عمل می‌کند.

عملکرد :استمینگ با استفاده از الگوریتم‌های ساده‌تر مانند الگوریتم Porter یاSnowball ، کلمات را به شکل پایه‌ای‌تر و گاهی نادرست کاهش می‌دهد. به عنوان مثال، runningبه runو betterبه betterتبدیل می‌شود.

تفاوت‌های کلیدی بین ریشه‌یابی و استمینگ

- **دقت :**ریشه‌یابی دقت بالاتری دارد زیرا از دیکشنری و قوانین گرامری استفاده می‌کند. استمینگ ساده‌تر و سریع‌تر است اما دقت کمتری دارد.
- **پیچیدگی :**ریشه‌یابی پیچیده‌تر و زمان‌برتر است زیرا نیاز به تحلیل معنایی دارد. استمینگ سریع‌تر و با الگوریتم‌های ساده‌تری انجام می‌شود.
- **خروجی :**خروجی ریشه‌یابی معمولاً کلمات معنی‌دار هستند، در حالی که خروجی استمینگ ممکن است همیشه کلمه‌ای معنی‌دار نباشد.

هدف حذفStopwords

هدف:

- کاهش حجم داده‌ها و افزایش کارایی مدل‌های یادگیری ماشین

- تمرکز بر روی کلمات با بار معنایی بالا و حذف کلمات بی‌اهمیت
- بهبود دقت و سرعت الگوریتم‌های پردازش متن

عملکرد حذفStopwords

عملکرد :حذف Stopwords با استفاده از یک لیست از پیش تعریف‌شده از کلمات پرتکرار و کم‌اهمیت انجام می‌شود. این لیست‌ها بسته به زبان مورد استفاده متفاوت هستند و می‌توانند به صورت دستی و یا از منابع معتبر به‌دست‌آیند.

مزایای حذفStopwords

- **کاهش حجم داده‌ها :**با حذف کلمات بی‌اهمیت، حجم داده‌ها کاهش می‌یابد که باعث افزایش سرعت پردازش می‌شود.
- **تمرکز بر کلمات مهم :**حذف Stopwords کمک می‌کند تا مدل‌ها بر روی کلمات با بار معنایی بیشتر تمرکز کنند.
- **بهبود دقت مدل :**با کاهش نویز در داده‌ها، دقت مدل‌های یادگیری ماشین بهبود می‌یابد.

هدف حذف علائم نشانه‌گذاری

هدف:

- حذف نویزهای غیرضروری از متن
- ساده‌سازی و تمیز کردن داده‌ها برای تحلیل بهتر
- بهبود دقت مدل‌های یادگیری ماشین و پردازش متن

عملکرد حذف علائم نشانه‌گذاری

عملکرد :حذف علائم نشانه‌گذاری با استفاده از روش‌های مختلفی مانند استفاده از توابع پیش‌فرض در کتابخانه‌های NLP یا استفاده از الگوهای منظم (Regular Expressions) انجام می‌شود

مزایای حذف علائم نشانه‌گذاری

- **کاهش نویز :**حذف علائم نشانه‌گذاری به کاهش نویز در داده‌ها کمک می‌کند و داده‌ها را تمیزتر و ساده‌تر می‌کند.
- **افزایش دقت مدل‌ها :**با حذف نویزها، مدل‌های یادگیری ماشین می‌توانند بهتر تمرکز کنند و دقت بالاتری داشته باشند.
- **پیش‌پردازش موثر :**این مرحله باعث می‌شود که داده‌ها برای تحلیل‌های بعدی آماده‌تر و مناسب‌تر شوند.

تشخیص زبان:

برای تشخیص زبان می توان از تابع اماده کتاب خانه langdetect به نام detect استفاده کرد. برای استفاده از این تابع body text مقاله را به ان تابع داده خروجی یک زبان است و خروجی را درون یک ستون جدید ذخیره می کنیم. تابع detect عملکرد ان به صورت زیر است:

متن ورودی به قطعات کوچکتری (n-grams) تقسیم می‌شود. این قطعات می‌توانند شامل تک‌واژه‌ها، دوواژه‌ای‌ها و غیره باشند.

کتابخانه دارای پروفایل‌های زبانی است که بر اساس داده‌های آموزشی از قبل ساخته شده‌اند. هر پروفایل شامل فراوانی n-grams برای یک زبان خاص است.

پروفایل n-grams متن ورودی با پروفایل‌های موجود برای هر زبان مقایسه می‌شود. این مقایسه برای پیدا کردن نزدیک‌ترین پروفایل زبانی انجام می‌شود.

زبانی که پروفایل آن بیشترین شباهت را به پروفایل متن ورودی داشته باشد به عنوان زبان تشخیص داده می‌شود.

عملکرد روشTF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency)یک روش بسیار رایج در پردازش زبان طبیعی (NLP)برای تبدیل متن به نمایش برداری است. این روش به منظور ارزیابی اهمیت یک کلمه در یک سند نسبت به کل مجموعه اسناد (مجموعه داده) استفاده می‌شود TF-IDF .ترکیبی از دو معیار است: فرکانس کلمه (TF) و فرکانس معکوس سند.(IDF)

1. **TF (Term Frequency):**تعداد دفعات تکرار یک کلمه در یک سند را نشان می‌دهد. فرمول آن به صورت زیر است:

$$\frac{how\ many\ time\ word\ t\ has\ been\ repeated\ in\ document\ d}{all\ words\ in\ document\ d} = TF(t,d)$$

2. **IDF (Inverse Document Frequency):**معیاری برای ارزیابی اهمیت یک کلمه در کل مجموعه اسناد است. فرمول آن به صورت زیر است:

$$log(\frac{N}{|\{d\ \epsilon\ D : t\epsilon\ d\}|}) = IDF(t)$$

در فرمول بالا N برابر تعداد کل اسناد و مخرج برابر با تعداد سند هایی که کلمه t در ان وجود دارد

3. **TF-IDF:**ترکیب TF و IDF است که اهمیت یک کلمه را در یک سند نسبت به کل مجموعه اسناد نشان می‌دهد. فرمول آن به صورت زیر است:

$$IDF(t,d) * TF(t,d) = TF - IDF(t,d,D)$$

pca:

pca یک روش قدرتمند برای تحلیل و ساده‌سازی داده‌هاست که در بسیاری از حوزه‌ها از جمله یادگیری ماشین، شناسایی الگو و تحلیل داده‌ها کاربرد دارد. این روش با کاهش ابعاد داده‌ها و حفظ بیشترین اطلاعات ممکن، باعث بهبود عملکرد مدل‌ها و ساده‌سازی تحلیل‌ها می‌شود. برای استفاده از این روش در پایتون می توان از کتاب خانه sklearn.decomposition استفاده

کرد. در پروژه ما اول 4096 ویژگی برای داده های خود استخراج کردیم سپس با استفاده از روش pca این مقدار ویژگی را کاهش دادیم. معیار توقف را 95 درصد واریانس در نظر گرفتیم و در اخر 949 ویژگی کاهش یافته ایجاد شد.

خوشه بندی:

در این پروژه 3 روش خوشه بندی DBscan , Kmeans و سلسله مراتبی پیاده سازی شده است. برای به دست اوردن روش بهترین عدد برای کلاسترینگ از روش elbow استفاده کرده ایم. این روش به این صورت کار می کند که برای هر تعداد کلاستر میزان انحراف را اندازه گیری می کنیم و در جایی که شکستگی رخ دهد(مانند ارنج) ان نقطه را به عنوان نقطه مورد نظر در نظر میگیریم.

خوشه‌بندی سلسله مراتبی :در این روش، نقاط داده به طور پیوسته به خوشه‌های کوچکتر ادغام یا جدا می‌شوند تا زمانی که به ساختار خوشه‌ای دلخواه برسیم . نحوه قرار گیری داده ها در یک خوشه میزان شباهت این داده ها نسبت به هم است برای و داده هایی که در یک خوشه قرار دارند دارای شباهت بیشتری نسبت بهم هستند.

DBSCAN یا خوشه‌بندی مبتنی بر چگالی فضایی :این الگوریتم نقاط داده را بر اساس چگالی فضایی آنها به خوشه‌هایی تبدیل می کند. نقاطی که در یک محله با چگالی بالا قرار دارند، به عنوان یک خوشه در نظر گرفته می‌شوند، در حالی که نقاط جدا شده به عنوان نویز در نظر گرفته می‌شوند.

K-Means: این الگوریتم نقاط داده را به K خوشه اختصاص می‌دهد، به طوری که مجموع فاصله هر نقطه تا مرکز خوشه خود به حداقل برسد K-Means .یک الگوریتم پارامتری است، به این معنی که کاربر باید تعداد خوشه‌ها (K) را از قبل مشخص کند.

برای پیاده سازی خوشه بندی در پایتون کار به شدت اسان است. فقط کافیست کتاب خانه های مربوطه را اضافه کنیم و از تابع های اماده برای خوشه بندی استفاده کنیم.

در روش DBscan دو مقدار برای ساختن کلاسترینگ می دهیم. Eps که نشان دهنده ماکسیسم فاصله برای دو نقطه که همسایه در نظر گرفته شود و min_sample که نشان می دهد مینیمم تعداد توی هر خوشه چه مقدار باشد.

برای مقایسه خوشه بندی از سیلوهت(silhouette) استفاده می کنیم به گونه که برای همه داده ها قابل محاسبه است. معیار سیلوهت به این گونه عمل می کند که یک نمره به هر نقطه داده در مجموعه داده اختصاص داده می‌شود و نشان می‌دهد که آن نقطه چقدر به خوشه خود تعلق دارد و از خوشه‌های دیگر جدا است. تفسیر این نمره به صورت زیر است.

اگر بزرگ تر از یک باشد یعنی به این خوشه تعلق دارد.

اگر برابر صفر باشد یعنی این داده دقیق بین دو خوشه قرار دارد.

اگر کوچک تر از صفر باید یعنی به این خوشه تعلق ندارد.

: T-SNE

یک الگوریتم یادگیری ماشین بدون نظارت است که برای کاهش ابعاد و تجسم داده‌ها به کار می‌رود .این الگوریتم برای کاهش تعداد ابعاد مجموعه داده‌های با ابعاد بالا (مانند صدها یا هزاران بعد) به دو یا سه بعد استفاده می‌شود، در حالی که ساختار و روابط بین نقاط داده را حفظ می‌کند.

t-SNEنقاط داده را در فضای با ابعاد بالا به گونه‌ای نمایش می‌دهد که نقاط داده‌های مشابه به یکدیگر نزدیک باشند و نقاط داده‌های نامشابه از یکدیگر دور باشند .این امر به تجسم بصری روابط بین نقاط داده و شناسایی خوشه‌ها یا ساختارهای پنهان در داده‌ها کمک می‌کند.

نحوه عملکرد:t-SNE

محاسبه شباهت t-SNE :با محاسبه شباهت بین هر جفت نقطه داده در فضای با ابعاد بالا شروع می‌شود .این شباهت‌ها معمولاً با استفاده از معیاری مانند فاصله اقلیدسی یا میزان همبستگی محاسبه می‌شوند.

نقشه‌برداری به فضای با ابعاد پایین :سپس t-SNE نقاط داده را به فضای با ابعاد پایین (معمولاً دو یا سه بعد) نقشه‌برداری می‌کند .این کار با بهینه‌سازی یک تابع هدف انجام می‌شود که سعی می‌کند توزیع شباهت بین نقاط داده در فضای با ابعاد بالا را تا حد امکان به توزیع شباهت در فضای با ابعاد پایین حفظ کند.

توزیع t-Student: t-SNE از یک توزیع آماری به نام توزیع t-Student برای مدل‌سازی شباهت بین نقاط داده در فضای با ابعاد پایین استفاده می‌کند .این توزیع به t-SNE اجازه می‌دهد تا نقاط داده‌های دور از یکدیگر را بیشتر از نقاط داده‌های نزدیک به یکدیگر از هم جدا کند.

مزایای:t-SNE

حفظ ساختار t-SNE :به طور خاص برای حفظ ساختارهای غیرخطی در داده‌ها طراحی شده است، که آن را برای تجسم داده‌های پیچیده مناسب می‌کند.

مقاومت در برابر نویز t-SNE :نسبتاً در برابر نویز در داده‌ها مقاوم است.

قابلیت بصری t-SNE :می‌تواند نتایج را به صورت بصری جذاب در فضای دو یا سه بعدی نمایش دهد.

معایب:t-SNE

محاسبات پرهزینه t-SNE :می‌تواند برای مجموعه داده‌های بزرگ پرهزینه از نظر محاسباتی باشد.

حساسیت به پارامترها t-SNE :به چندین پارامتر تنظیم نیاز دارد که می‌توانند بر کیفیت نتایج تأثیر بگذارند.

عدم تفسیر احتمالی :تفسیر نتایج t-SNE همیشه آسان نیست.

کاربردهای:t-SNE

تجسم داده‌های با ابعاد بالا t-SNE :برای تجسم مجموعه داده‌های با ابعاد بالا، مانند داده‌های ژنومیک، داده‌های متنی و داده‌های تصویری، استفاده می‌شود.

کشف خوشه t-SNE :می‌تواند برای شناسایی خوشه‌ها یا ساختارهای پنهان در داده‌ها استفاده شود.

کاهش نویز t-SNE :می‌تواند برای کاهش نویز در داده‌ها و بهبود وضوح ساختارهای زیربنایی استفاده شود.

حال برای پیاده سازی t-SNE فقط کافیست کتاب خانه های مربوطه را اضافه کنیم و از تابع اماده ان استفاده کنیم. خود TSNE دو پارامتر n_components دارد که نشان می دهد داده ها را به چند بعد می خواهیم کاهش بدیم و یک پارامتر random state مانند یک سید عمل می کند و نمی گذارد با هربار اجرا شدن کد نتیجه های مختلف خروجی دهد.

مدل سازی موضوعی:

الگوریتم NMF :
NMF یک تکنیک تجزیه ماتریسی است که ماتریس متون-واژه‌ها را به دو ماتریس کوچکتر که تنها شامل مقادیر غیرمنفی هستند، تجزیه می‌کند. این ماتریس‌ها نمایانگر توزیع موضوعات در مستندات و توزیع واژه‌ها در موضوعات هستند.

مزایا:

سادگی و تفسیرپذیری: نتایج به‌دست‌آمده از NMF تفسیرپذیرتر هستند، زیرا همه مقادیر غیرمنفی هستند.

کارایی: در بسیاری از موارد، NMF سریعتر از الگوریتم‌های مشابه اجرا می‌شود.

معایب:

نیاز به پیش‌پردازش دقیق: نیاز به تنظیم دقیق مقادیر اولیه دارد تا به نتایج قابل قبول برسد.

حساس به پارامترها: حساس به پارامترهای اولیه و تنظیمات الگوریتم است.

الگوریتم LSA:

LSA یا SVD (Singular Value Decomposition) ماتریس متون-واژه‌ها را با استفاده از تجزیه مقدار تکین (SVD) تجزیه می‌کند تا ساختار پنهان معنایی واژه‌ها را کشف کند. این روش فضاهای با ابعاد کمتر ایجاد می‌کند که نمایانگر مفاهیم پنهان در داده‌ها هستند.

مزایا:

کاهش نویز: با کاهش ابعاد، نویز و نوفه‌های موجود در داده‌ها را کاهش می‌دهد.

کاربرد وسیع: به طور گسترده در تحلیل معنایی متون و جستجوی اطلاعات استفاده می‌شود.

معایب:

مقادیر منفی: نتایج شامل مقادیر منفی هستند که تفسیر معنایی آن‌ها مشکل است.

ارتباط خطی: فرض می‌کند که رابطه بین واژه‌ها و موضوعات خطی است که همیشه صحیح نیست.

الگوریتم LDA:

LDA یک مدل مولد احتمالی است که فرض می‌کند هر مستند مجموعه‌ای از موضوعات و هر موضوع مجموعه‌ای از واژه‌ها را دنبال می‌کند. این مدل با استفاده از توزیع دیریکله موضوعات را به مستندات و واژه‌ها را به موضوعات تخصیص می‌دهد.

مزایا:

مدل احتمالی: به عنوان یک مدل احتمالی، قادر به ارائه تفسیرهای دقیق‌تری از توزیع موضوعات و واژه‌ها است.

انعطاف‌پذیری: می‌تواند با داده‌های بزرگ و پیچیده به خوبی کار کند.

معایب:

پیچیدگی محاسباتی: نسبت به NMF و LSA پیچیدگی محاسباتی بیشتری دارد.

حساس به تنظیمات اولیه: نیاز به تنظیمات اولیه دقیق و تعداد مناسب موضوعات دارد.

NMF و LDA به دلیل غیرمنفی بودن مقادیر و مدل احتمالی بودن تفسیرپذیرتر هستند.

NMF معمولا سریعتر است، در حالی که LDA به دلیل پیچیدگی محاسباتی ممکن است زمان بیشتری ببرد.

LDA به عنوان یک مدل احتمالی می‌تواند توزیع‌های دقیق‌تری از موضوعات ارائه دهد، در حالی که LSA و NMF به مدل‌های خطی و غیرمنفی متکی هستند.

LSA نیاز به پیش‌پردازش کمتری دارد اما مقادیر منفی دارد که تفسیر را مشکل می‌کند. NMF و LDA نیاز به تنظیمات اولیه دقیقتری دارند.

برای استفاده از LDA و NMF در پایتون ابتدا کتاب خانه های مربوطه را اضافه می کنیم.سپس برای استفاده از این توابع نیاز است که متون خود را به ماتریس های بردار ویژه تبدیل کنیم تا بتوانیم تعداد دفعات وجود ی کلمه را پیدا کنیم. سپس توسط کتابخانه genism انرا تبدیل به ساختار داده ای می کنیم که تابع LDA بتواند از ان استفاده کند.