تمرین کامپیوتری پنجم هوش مصنوعی

امیرمرتضی رضائی – 810003004

پیش پردازش و استخراج ویژگی

پیاده‌سازی موارد ذکر شده انجام شد.

1- در روشStemming ، کلمه به ریشه یا پایه خود کاهش داده می‌شود، اما این کار اغلب با حذف پسوندها یا پیشوندها و بدون توجه به معنای دقیق انجام می‌گیرد. این فرآیند معمولا ساده و مبتنی بر قوانین است. به همین دلیل، ممکن است خروجی همیشه یک کلمه معتبر نباشد.

مثلا کلمه better ممکن است به bett تبدیل شود!

در روشLemmatization ، کلمه به شکل پایه (lemma) خود تبدیل می‌شود، اما این فرآیند با در نظر گرفتن معنی کلمه و نقش دستوری آن انجام می‌گیرد .

Lemmatizationمعمولا از یک فرهنگ لغت یا مدل زبان استفاده می‌کند تا شکل پایه‌ی درست کلمه را بیابد. این روش نسبت به stemming دقیق‌تر است ولی به زمان پردازش بیشتری نیاز دارد!

- **کاهش پیچیدگی متن :** متن خام ممکن است شامل نشانه‌گذاری‌ها، اعداد، یا کاراکترهای خاص باشد که برای تحلیل و مدل‌سازی ضروری نیستند.
- **هماهنگی و یکدستی در داده‌ها :** متن ممکن است شامل کلمات با حروف بزرگ و کوچک یا شکل‌های مختلف نوشتاری باشد. این موارد باعث می‌شود که کلمات یکسان به‌صورت اشتباه به‌عنوان کلمات متفاوت در نظر گرفته شوند. استفاده از روش‌هایی مثل تبدیل به حروف کوچک و حذف فاصله‌های اضافی به هماهنگی داده‌ها کمک می‌کند.
- **کاهش ابعاد داده‌ها :** کلمات زائد (Stop Words)مانند"the" ، "is"، "and"، و... معمولاً معنای خاصی ندارند و برای تحلیل مفید نیستند. حذف این کلمات باعث کاهش تعداد ویژگی‌ها (ابعاد داده) می‌شود و پردازش سریع‌تر خواهد بود.
- **تمرکز روی مفهوم اصلی :** استفاده از روش‌هایی مانند Stemming یا Lemmatization به استخراج ریشه یا شکل پایه کلمات کمک می‌کند. این کار به مدل اجازه می‌دهد که شکل‌های مختلف یک کلمه را به‌عنوان یک مفهوم یکسان در نظر بگیرد.
- **حذف نویز و داده‌های بی‌ربط :** متن ممکن است شامل لینک‌ها، ایموجی‌ها، یا کاراکترهای غیرمفید باشد که بر تحلیل تأثیری ندارند. حذف این عناصر نویز داده را کاهش می‌دهد.

3- از آنجا که داده‌های متنی خام به دلیل ماهیت غیرعددی، تنوع ساختاری، و نویزهای مختلف برای مدل‌های یادگیری ماشین مستقیما قابل استفاده نیستند استخراج ویژگی‌ها ضروری است. این فرآیند با تبدیل متن به قالب عددی، حذف اطلاعات زائد، و تمرکز بر مفاهیم کلیدی، داده‌ها را ساده‌تر و قابل‌پردازش می‌کند. استخراج ویژگی‌ها با کاهش ابعاد داده، نمایش روابط معنایی و ساختاری، و استانداردسازی متن، به بهبود دقت و عملکرد مدل‌ها کمک می‌کند. همچنین این مرحله امکان درک بهتر معنا و الگوهای متن را فراهم کرده و متن خام را به شکلی بهینه برای تحلیل و یادگیری تبدیل می‌کند.

فرآیند مسئله

4- در یادگیری تحت نظارت، مدل با استفاده از داده‌های برچسب‌گذاری شده آموزش می‌بیند تا الگوی رابطه بین ورودی‌ها و خروجی‌ها را یاد بگیرد و به پیش‌بینی یا طبقه‌بندی داده‌های جدید بپردازد، مانند تشخیص ایمیل‌های اسپم. بر خلاف آن، در یادگیری بدون نظارت، داده‌ها برچسب‌گذاری نشده‌اند و مدل باید به طور خودکار ساختارهای پنهان یا الگوهایی مانند خوشه‌ها را در داده‌ها کشف کند، مانند تقسیم‌بندی مشتریان بر اساس ویژگی‌های مشترک یا کاهش ابعاد داده‌ها. در حالی که یادگیری تحت نظارت نیازمند داده‌های برچسب‌گذاری شده است و برای مسائل پیش‌بینی یا طبقه‌بندی مناسب است، یادگیری بدون نظارت برای کشف الگوها و ساختارهای داده‌ای مفید است و کاربردهایی مانند تحلیل داده‌های بزرگ یا خوشه‌بندی دارد.

5- بردار ویژگی یک نمود عددی از ویژگی‌های مختلف یک داده است که به مدل‌ها کمک می‌کند تا داده‌های را به فرم قابل پردازش تبدیل کنند. این بردارها ویژگی‌های کلیدی داده را به صورت عددی در می‌آورند و اصولا باعث کاهش ابعاد داده‌ها می‌شوند، به طوری که اطلاعات ضروری حفظ شده و حجم محاسبات کاهش می‌یابد. استفاده از بردار ویژگی امکان پردازش خودکار داده‌ها و شبیه‌سازی روابط میان ویژگی‌ها را برای مدل‌ها فراهم می‌کند، بدون نیاز به پردازش دستی پیچیده. ویژگی‌های بردار شامل ابعاد مختلفی هستند که نمایانگر صفات خاص داده‌ها می‌باشند، مانند اندازه، رنگ، یا کلمات کلیدی در داده‌های متنی.

6- مجموعه مدل‌های Sentence Transformer مجموعه‌ای از مدل‌های یادگیری عمیق است که به‌طور خاص برای تبدیل جملات به نمایش‌های عددی (بردارهای ویژگی) طراحی شده است تا امکان مقایسه، جستجو و دسته‌بندی جملات را فراهم کند. این مدل‌ها برای وظایفی مانند تشابه معنایی جملات، پاسخ‌دهی به سوالات و ترجمه متنی بسیار مفید هستند. یکی از مدل‌های محبوب این مجموعه، all-MiniLM-L6-v2 است که یک مدل مبتنی بر معماری MiniLM است و به‌طور خاص برای سرعت و کارایی بالاتر با دقت خوب طراحی شده است. این مدل با استفاده از 6 لایه از شبکه‌های ترنسفورمر و آموزش بر روی داده‌های متنی متنوع، قادر است جملات را به بردارهای فشرده و باکیفیتی تبدیل کند که برای مقایسه و پردازش‌های مشابهت متنی مناسب هستند.

7- روش‌های K-means، DBSCAN و Hierarchical Clustering هر کدام الگوریتم‌هایی برای خوشه‌بندی داده‌ها هستند، اما با رویکردها و ویژگی‌های متفاوت. K-means ابتدا تعداد خوشه‌ها را مشخص کرده و سپس داده‌ها را بر اساس کمترین فاصله به مراکز خوشه‌ها تقسیم می‌کند، که سریع و کارا است، اما نیاز به تعیین تعداد خوشه‌ها از پیش دارد و در مواجهه با داده‌های دارای ساختار پیچیده یا نویز عملکرد ضعیفی دارد.
DBSCAN خوشه‌ها را بر اساس تراکم نقاط داده ایجاد می‌کند و قادر به شناسایی خوشه‌ها با اشکال پیچیده و همچنین شناسایی نقاط نویزی است، اما نیاز به تنظیم دو پارامتر (حداقل تعداد نقاط و فاصله) دارد که می‌تواند حساسیت به داده‌ها را افزایش دهد. Hierarchical Clustering داده‌ها را به‌طور تدریجی خوشه‌بندی کرده و درختی از روابط خوشه‌ها ایجاد می‌کند، که برای تحلیل روابط در مقیاس‌های مختلف مفید است، ولی به دلیل پیچیدگی محاسباتی، برای داده‌های بزرگ کارایی کمتری دارد. در مجموع، K-means برای داده‌های با ساختار مشخص مناسب است، DBSCAN برای داده‌های نویزی و پیچیده کاربرد دارد و Hierarchical Clustering برای تحلیل‌های دقیق‌تر و داده‌های کوچک یا متوسط مناسب است.

8- در این روش، الگوریتم K-means برای مقادیر مختلف K (تعداد خوشه‌ها) اجرا می‌شود و میزان اختلاف میان داده‌ها و مراکز خوشه‌ها محاسبه می‌شود. سپس نموداری از این مقادیر رسم می‌شود که در آن محور افقی تعداد خوشه‌ها و محور عمودی میزان Inertia است. نقطه‌ای که در نمودار که شکلی شبیه به آرنجدرا می‌سازد، بهترین تعداد خوشه‌ها را نشان می‌دهد، زیرا پس از این نقطه کاهش Inertia به‌طور چشمگیری کند می‌شود و افزودن تعداد بیشتر خوشه‌ها تغییرات زیادی در کیفیت خوشه‌بندی ایجاد نمی‌کند.

9-

```
Cluster Distribution:
K-Means: (array([0, 1, 2, 3, 4], dtype=int32), array([841, 670, 896, 602, 888]))
DBSCAN: (array([-1,  0,  1,  2,  3]), array([3876,    4,    5,    5,    7]))
Hierarchical: (array([1, 2, 3, 4, 5], dtype=int32), array([ 950,  569,  298,  881, 1199]))
```

```
Silhouette Scores:
K-Means: 0.07769274711608887
DBSCAN: -0.0948309674859047
Hierarchical: 0.08096538484096527
```

کاهش بعد

10- PCA یک تکنیک آماری است که برای کاهش ابعاد داده‌های پیچیده و کشف الگوهای اصلی در داده‌ها استفاده می‌شود. هدف از PCA این است که داده‌های با ابعاد زیاد را به فضایی با ابعاد کمتر تبدیل کند، به طوری که بیشترین اطلاعات (واریانس) داده‌ها حفظ شود. در این روش، ابتدا ماتریس کوواریانس داده‌ها محاسبه می‌شود که نشان‌دهنده روابط و همبستگی‌ها بین ویژگی‌های مختلف داده‌ها است. سپس مؤلفه‌های اصلی از طریق محاسبه مقادیر ویژه و بردارهای ویژه ماتریس کوواریانس به‌دست می‌آیند. هر مؤلفه اصلی به‌طور موفقیت‌آمیز بیشترین واریانس داده‌ها را در خود جای می‌دهد.

ارزیابی و تحلیل

11- معیار Silhouette برای ارزیابی کیفیت خوشه‌بندی‌ها و جداسازی خوشه‌ها به کار می‌رود. این معیار نشان می‌دهد که هر داده چقدر به خوشه خودش نزدیک است و چقدر از خوشه‌های دیگر دور است. برای هر نمونه، مقدار Silhouette از دو پارامتر محاسبه می‌شود:

a(i) میانگین فاصله نمونه i از سایر نمونه‌های همان خوشه

b(i) فاصله متوسط تا اعضای خوشه‌های دیگر

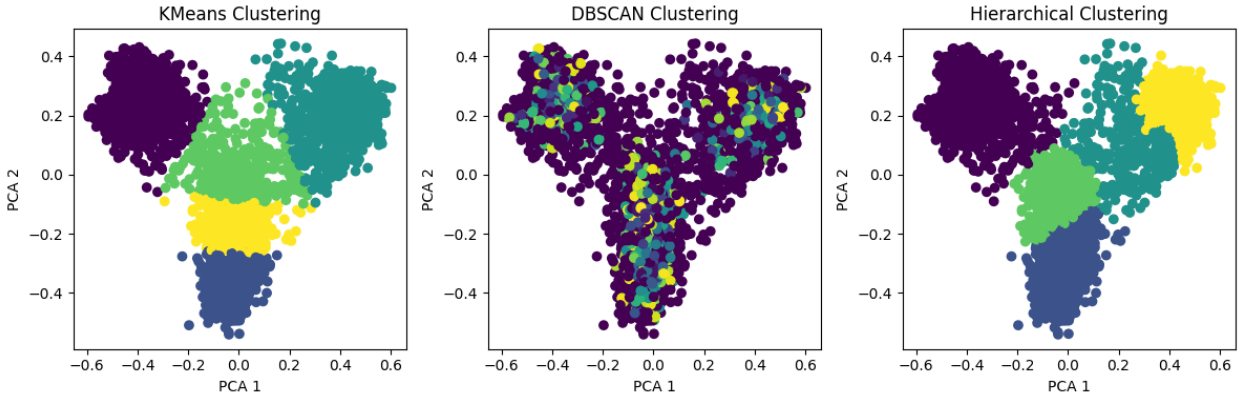
$$s = \frac{b - a}{max(a, b)}$$

اگر s(i) نزدیک به 1 باشد، نشان‌دهنده این است که نمونه به خوبی در خوشه خود قرار گرفته است.

اگر s(i) نزدیک به 0 باشد، یعنی نمونه در مرز بین دو خوشه قرار دارد.

اگر s(i) نزدیک به -1 باشد، نشان‌دهنده این است که نمونه احتمالاً در خوشه اشتباهی قرار دارد.

معیار Homogeneity کیفیت خوشه‌بندی را از نظر همگنی یا یکدستی خوشه‌ها ارزیابی می‌کند. به عبارت دیگر، این معیار بررسی می‌کند که آیا نمونه‌های یک خوشه به یک دسته واقعی (برچسب‌گذاری‌شده) تعلق دارند یا خیر. برای محاسبه این معیار، از تقسیم تعداد نمونه‌های درست خوشه‌بندی‌شده بر تعداد کل نمونه‌ها استفاده می‌شود.



-12

```
KMeans Silhouette Score: 0.4897468388080597
DBSCAN Silhouette Score: -0.07823939621448517
Hierarchical Clustering Silhouette Score: 0.466813921928405 76
```