به نام خدا

گزارش فاز دوم پروژه درس بازیابی اطلاعات

مدرس : دکتر سلیمانی

حسین ابراهیمی کندری ـ ۹۵۱۰۵۳۰۲

۱ پیش پردازش داده

پیش پردازش‌هایی که برای سندهای خام برای قسمت‌های اولیه انجام شد، شامل lowercase کردن حروف، جایگذاری فاصله به جای punctuation ها و جداکردن کلمات از هم بر اساس فاصله یا space بود. متن هر سند و عنوانش با هم ادغام شده و به عنوان یک متن در نظر گرفته شدند.

۲ K Nearest Nighbours (KNN)

۱.۲ بردن به فضای tf-idf

ابتدا بعد از پیش‌پردازش بر روی سندها، کلمات متمایز و تعداد آن‌ها به دست آورده شد که مقدار tf را به ما می‌دهد. برای سرعت بخشیدن به فرآیند بدست آوردن df برای هر کلمه، از داده ساختار trie استفاده شد و با هر insert به آن، داک موردنظر کلمه دخیره شد. در انتهای این فرآیند مقدار $tf \times \log(N/df)$ محاسبه شد و در سطر سند و ستون کلمه ماتریس X_{train} ذخیره شد. همین کار برای سندهای validation نیز انجام شد.

۲.۲ Vectorized Implemention

در این قسمت با انجام محاسباتی بر روی ماتریس‌ها فاصله‌ی هر دو سند در مجموعه داده‌های train و validation را در حالت‌های فاصله اقلیدسی و شباهت کسینوسی محاسبه می‌کنیم.

- **Euclidean Distance**

$$dists = \frac{X_{val}}{||X_{val}||} \times (\frac{X_{train}}{||X_{train}||})^T$$

- **Cosine Similarity**

$$dists = N_{train}^2 + (N_{val}^2)^T - 2 \times X_{val} X_{train}^T$$

در رابطه‌ی بالا N_{train} و N_{val} ماتریس‌های نرم هستند بدین صورت که هر درایه سطر iام این ماتریس‌ها برابر با نرم سطر iام ماتریس‌های train و val هستند.

۳.۲ Best KNN Model

مدل KNN با پارامترهای مختلف k که ۱، ۳ و یا ۵ است و پارامتر معیار فاصله که حالت‌های فاصله اقلیدسی و شباهت اقلیدسی را داراست، آموزش داده شد که از بین آن‌ها مدل با $k = 5$ و شباهت کسینوسی بیشتر مقدار accuracy برابر با **87%** را روی مجموعه داده validation داشت.

K	Distance Measure	Accuracy
1	Cosine	83.76%
1	Euclidean	68.86%
3	Cosine	85.53%
3	Euclidean	64.16%
5	Cosine	**87%**
5	Euclidean	63.4%

۱.۳.۲ معیارهای ارزیابی بهترین مدل KNN

- **Precision-Recall**

Class	Precision	Recall
1	87.5%	86.8%
2	91%	95.2%
3	83%	82%
4	85%	84%

- **Confusion Matrix**

$$\begin{pmatrix} 651. & 25. & 48. & 26. \\ 14. & 714. & 12. & 10. \\ 44. & 24. & 616. & 66. \\ 42. & 19. & 66. & 623. \end{pmatrix}$$

- **macro F1 with $\beta = 1$**

F۱ Averaged Macro $= 86.94\%$

۳ Naive Bayes (NB)

این مدل در کد با استفاده از دو تابع train و apply پیاده سازی شده است. در تابع train با استفاده از tf مربوط به هر کلمه در داک‌های کلاس‌های متفاوت مقدار احتمال $\hat{P}(t, c)$ تخمین زده می‌شود و در ماتریس $condProb$ که سطرهای آن کلمات corpus و ستون‌های کلاس‌ها هستند و در همین حال مقدار prior هر کلاس نیز محاسبه می‌شود. تابع apply نیز مدل را بر روی داده validation اجرا می‌کند و بردار پیش‌بینی را خروجی می‌دهد.

۱.۳ Best NB Model

در این قسمت مدل را با پارامتر smoothing (α) های متفاوت بر روی داده‌ی validation ارزیابی می‌کنیم. ابتدا مقدار α به صورت یکنواخت در بازه $\alpha \sim \mathcal{U}(0, 100)$ نمونه گرفته شد و مشاهده شد که با کاهش مقدار α مقدار accuracy بر روی داده‌ی ارزیابی افزایش پیدا می‌کند به صورتی که مقدار کمتر از ۱ α بهترین دقت را داشت. در مرحله بعد α از بازه $\alpha \sim \mathcal{U}(0, 1)$ به صورت یکنواخت گرفته شد که در این بین مقدار $\boldsymbol{\alpha = 0.419}$ با دقت **89.23%** بیشترین دقت را داشت.

۱.۱.۳ معیارهای ارزیابی بهترین مدل KNN

- **Precision-Recall**

class	Precision	Recall
1	90.17%	88.13%
2	94.55%	97.33%
3	87.22%	83.73%
4	84.9%	87.73%

- **Confusion Matrix**

$$\begin{pmatrix} 661. & 26. & 32. & 31. \\ 11. & 730. & 4. & 5. \\ 31. & 10. & 628. & 81. \\ 30. & 6. & 56. & 658. \end{pmatrix}$$

- **macro F1 with $\beta = 1$**

F۱ Averaged Macro = 89.2%

۴ تاثیر روش‌های پردازش متن بر روی دسته‌بندی

تاثیر این روش‌ها بر روی بهترین مدل‌های KNN و NB قسمت قبل مورد ارزیابی قرار گرفت. دو معیار Accuracy و Avg F۱ به عنوان دو معیار کلی در جدول زیر برای مدل‌های این قسمت و قسمت قبل انتخاب شده‌اند. سایر معیارهای جزئی‌تر در کد قابل مشاهده هستند.

Model	NLTK Method	Difference Macro Avg F1	Difference Accuracy
KNN	Stemming	**+0.04%**	**+0.03%**
NB	Stemming	-0.01%	0%
NB	Lemmatization	**-0.11%**	**-0.1%**
NB	Stopword Removal	-0.04%	-0.03%

همانطور که در جدول بالا مشاهده می‌شود، تنها روشی که باعث بهبود کمی در عملکر مدلی شده است، stemming در مدل KNN است. اما روش‌های دیگر در مدل NB یا باعث کاهش مقدار دقت شده و یا تغییری در آن ایجاد نکرده‌اند. این تغییرات در معیارهای ارزیابی بسیار اندک بوده و نمی‌توان دلیل جامعی برای این رفتار بیان کرد. این تغییرات در سایر معیارهای دیگر یعنی Precision-Recall و Confusion Matrix بسیار ناچیز هستند.

۵ Support Vector Machine (SVM)

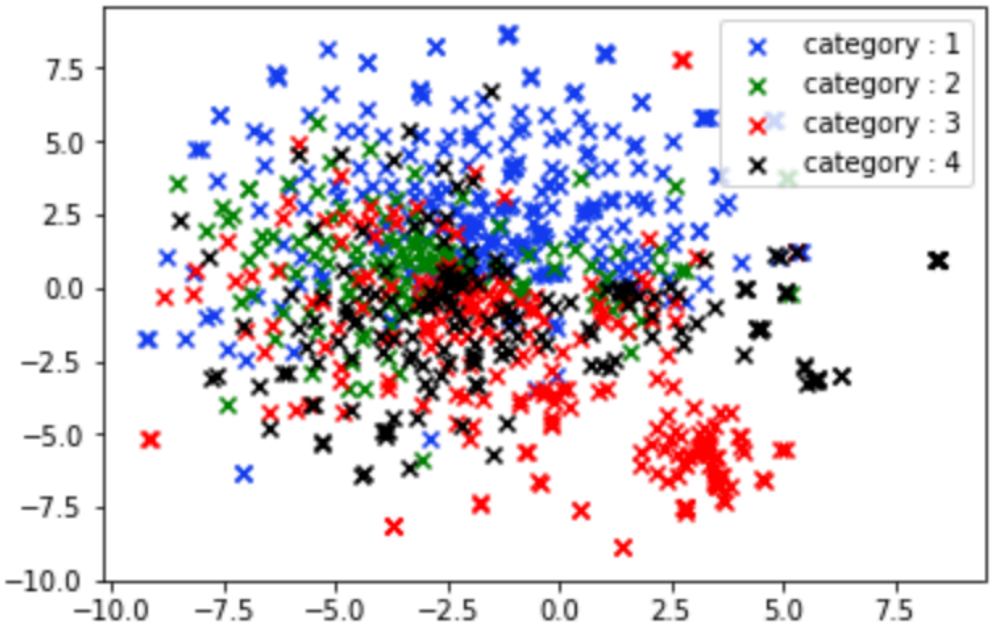
با توجه به آهسته بودن فرآیند یادگیری در svm.SVC تعداد داده آموزشی نصف شده و به صورت تصادفی از بین داده‌ها انتخاب شدند. همچنین با توجه به ابعاد داده، همگرا شدن این روش بسیار کند صورت می‌گرفت (> ۱ ساعت) به همین منظور حد بالایی از iteration در پیاده سازی لحاظ شد تا بتوان نتیجه را بررسی کرد ($iter_{max} = 600$). مشاهدات از دفعات اجرا شدن متوالی نشان داد که با بیشتر کردن حد بالای iteration وزن‌های بهتری همگرا شده و مقدار دقت افزایش می‌یابد. بهترین ضریب منظم‌ساز نیز با استفاده از نمونگیری یکنواخت $C \sim \mathcal{U}(0, 10)$ بر روی داده‌ی کمتر و iteration کمتر انتخاب شد. بهترین مدل بدست آمده با ۶۰۰ مرتبه iteration و ضریب منظم‌ساز $\boldsymbol{\lambda = 0.5}$ به مقدار دقت **85.46%** بر روی داده‌ی ارزیابی رسید.

۶ Random Forest

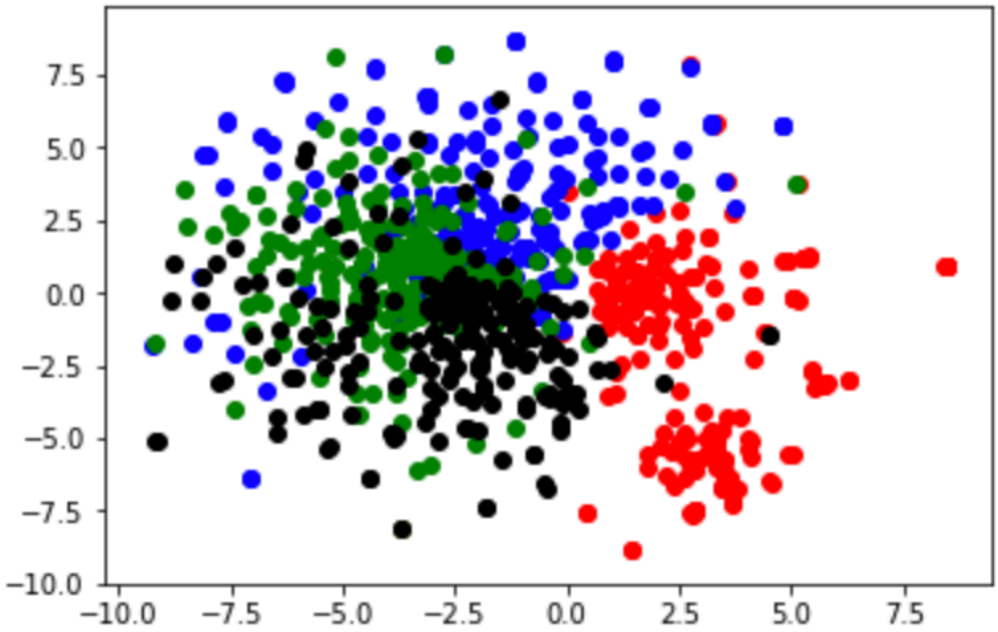
با تغییر تعداد درخت‌ها و حداکثر عمق آن‌ها این نتیجه حاصل شد که با افزایش این دو مقدار دقت مدل بدست آمده تا حدی افزایش می‌یابد و بعد از آن ثابت می‌شود. بهترن مقدار دقت برای مدل بر روی داده ارزیابی برابر با 84.6% بود که به ازای $\boldsymbol{num\ tree = 80}$ و $\boldsymbol{max\ depth = 50}$ بدست آمد.

۷ K-means + t-SNE

حاصل t-SNE embedding در ۲۰۰ داده از هر خوشه که با استفاده از الگوریتم K-means و برچسب‌های واقعی داده که امیدوار بودیم داک‌هایی با برچسب‌های یکسان در یک خوشه قرار گیرند.



(آ) برچسب‌های واقعی مربوط به داک‌ها

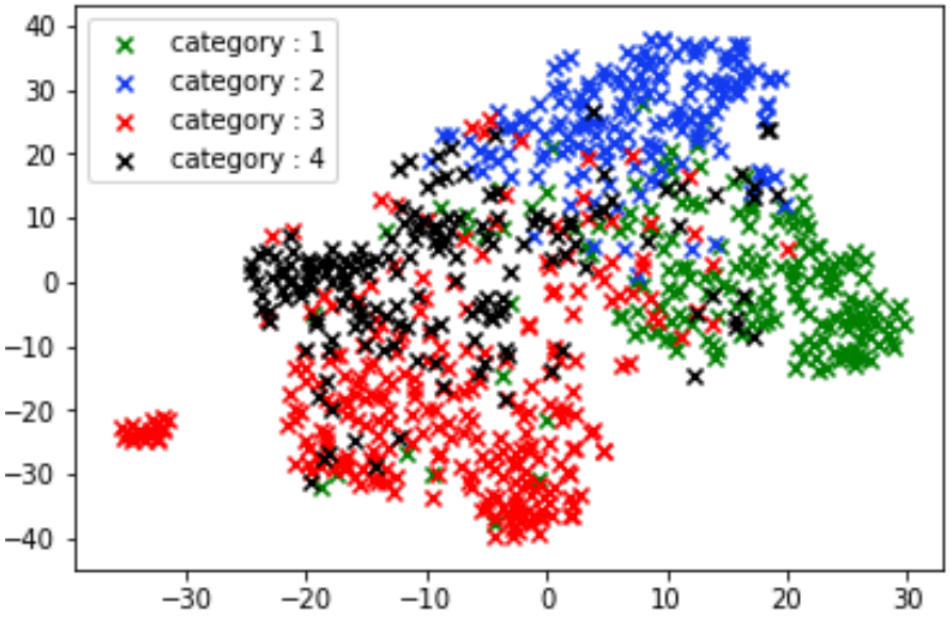


(ب) خوشه‌بندی انجام شده بعد از t-SNE

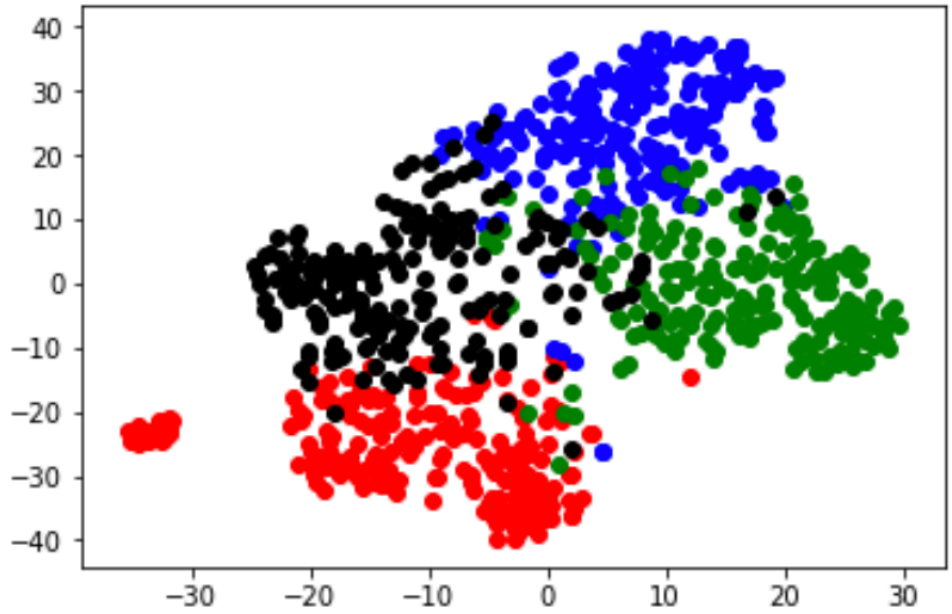
همانطور که در نمودارهای بالا دیده می‌شود، برخی از خوشه‌ها به طور تقریبی توانسته‌اند که جمعیت مربوط به یکی از category ها را پوشش دهند مانند خوشه‌های آبی و سبز و مشکی. اما مثلا خوشه قرمز دارای ۲ زیر جمعیت است که یکی از آن‌ها به درستی داک‌های موردنظر را پوشانده اما دیگری دارای مقداری خطا است. این خطاها حاصل از آن است که روش bag of wards وابستگی بین کلمات و ترتیب آن‌ها را در نظر نمی‌گیرد. در قسمت بعد خواهیم دید که با در نظر گرفتن این فاکتورها عملکرد به طور محسوسی بهبود می‌یابد.

۸ Word2Vec

این قسمت با استفاده از مدل CBOW کتابخانه gensim پیاده‌سازی شده است. ابتدا جملات سندها به عنوان ورودی به مدل داده می‌شود تا فرآیند یادگیری بر روی آن‌ها صورت گیرد. در این قسمت بهترین هایپر پارامترهای بدست آمده برابر با $\boldsymbol{window} = 8$ ، $\boldsymbol{min\ count} = 5$ و $\boldsymbol{size} = 300$ بود که بهترین خوشه‌بندی را حاصل می‌کرد.



(آ) برچسب‌های واقعی مربوط به داک‌ها



(ب) خوشه‌بندی انجام شده بعد از t-SNE

همانطور که شکل‌های بالا نشان می‌دهند، خوشه‌های به دست آمده حاصل از بردارهای کلمات مدل word۲vec توانسته به خوبی توزیع هر یک category ها را توضیح دهد.
در این قسمت برای بدست آوردن بردار مربوط به هر داک توسط بردار کلمات آن، روش‌های مختلفی شامل،

- Maximum Elementwise
- Minimum Elementwise
- Concatenate previous max and min vectors
- Average Elementwise

مورد ارزیابی قرار گرفت که در بین آن‌ها روش میانگین گیری درایه به درایه بهترین نتیجه را داشت.

۹ دقت دسته‌بندی

فایل‌های model.py و judge.py با تغییراتی در ورودی و خروجی تابع‌ها در فایل فرستاده شده پیوست شده است.