دسته بندی اخبار فارسی با استفاده از مدل های یادگیری نظارتی

محمدحسین مازندرانیان
دانشجو ارشد مهندسی فناوری اطلاعات پزشکی – دانشگاه تهران
mh.mazandaranian@ut.ac.ir

چکیده
با استفاده از داده های جمع آوری شده از اخبار وبسایت بی بی سی فارسی سه مدل دسته بندی با نظارت Naïve Bayes، ماشین بردار پشتیبان و شبکه ی عصبی (Ann) ساخته شد. دقت مدل Naïve Bayes برابر با 75، دقت ماشین بردار ماشین 96 و دقت مدل شبکه ی عصبی برابر با 97 درصد شد. اما با تست مدل های ساخته شده روی یک مجموعه داده جدید که از وبسایت های خبرگزاری های تسنیم و تجارت نیوز که مربوط سال جاری است دقت پیش بینی در نهایت از 85 درصد بیشتر نشد که نشان دهنده ی این است که در حوزه ی پردازش متن با چالش هایی از قبیل دایره لغات جدید و به مقدار زیاد نیاز داریم.

واژگان کلیدی: دسته بندی اخبار، اخبار فارسی، یادگیری با نظارت، شبکه های عصبی

مقدمه

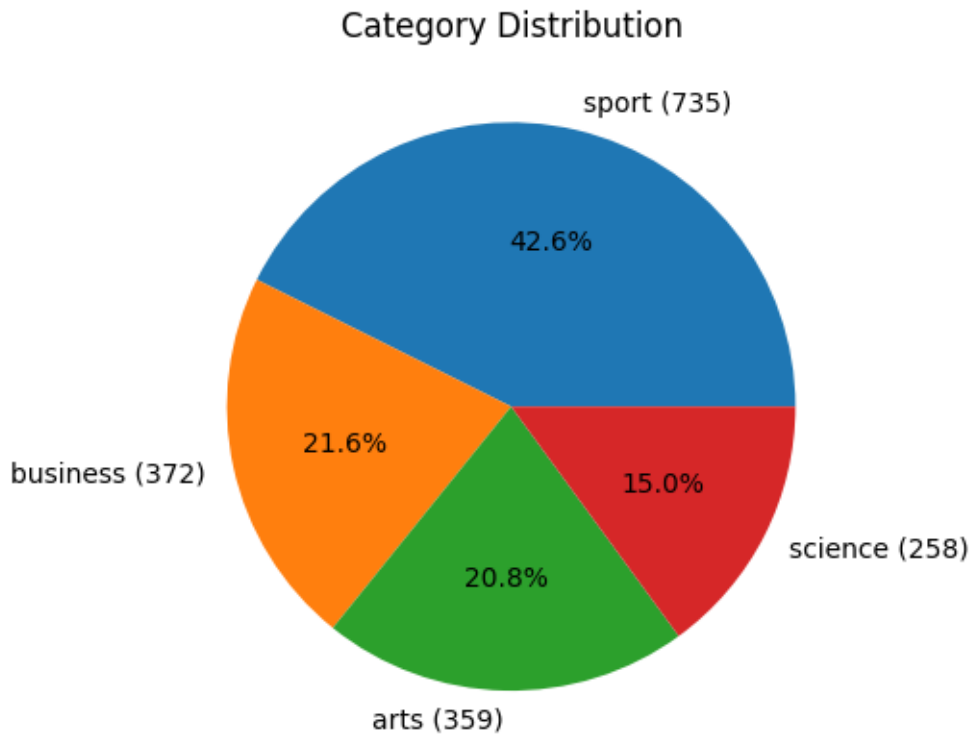
دسته بندی متن یکی از مسائل کلاسیک در حوزه ی پرداش متن طبیعی است، یکی از چالش های این حوزه جمع آوری داده ی مناسب است چون دایره لغات استفاده شده در متن ها با گذشت زمان دچار تغییر می شود. در مسائلی مثل درک مفهوم و احساسات در جمله و پیدا کردن موجودیت های نامدار با چالش های بیشتری نسبت به یک دسته بندی ساده طرف هستیم. هدف نویسنده، پیاده سازی یک پروژه روی یک دیتای واقعی در زبان فارسی بوده است، با توجه به جستجوی های انجام شده تا زمان نوشتن این گزارش پروژه ای که روی این دیتاست انجام شده باشد در بستر اینترنت یافت نشد.

روش جمع آوری داده

برای داده های مربوط به اخبار بی بی سی فارسی از یک دیتاست آماده از وبسایت Kaggle مربوط به سال 2020 میلادی استفاده شده است و داده های مربوط به خبرگزاری های تسنیم و تجارت نیوز در سال 2024 توسط نویسنده ی گزارش با استفاده از ابزارهای خزش مهیا شده است.

بررسی داده

چهار دسته بندی اخبار علمی، هنری، هنری و اقصادی-تجاری از مجموعه داده های اخبار بی بی فارسی برای انجام این پروژه انتخاب شده است. در مجموع داده های انتخاب شده شامل 1724 سطر می باشد. توزیع داده ها رو میتوانیم در شکل 1 ببینیم. تعداد اخبار ورزشی بیشتر از بقیه ی دسته بندی ها است ولی در مجموع این عدم توازن و تعداد کلاس های کم تاثیری محسوسی در روند پروژه ندارد.



شکل 1 - توزیع دسته بندی داده های اخبار بی بی سی فارسی

پیش پردازش

قبل از ساخت مدل، ابتدا باید کارکتر های اضافی یا به اصطلاح stop words را از متن اخبار حذف کنیم، دلیل این کار این است که بصورت کلی برای دسته بندی موضوعی اخبار، کلمات استفاده شده در هر موضوع را پیدا میکنیم و با توجه به وزن آن کلمات در جمله میتوانیم موضوع خبر رو پیش بینی کنیم. به عنوان مثال اگر در جمله کلمه ی "پزشکیان" و "دولت" داشته باشد به احتمال زیاد این جمله یک خبر سیاسی است. قاعدتا در چنین شرایطی کارکتر هایی از قبیل، ویرگول، اعداد، پرانتز اهمیتی برای ندارد و وجود آن ها در متن حتی ممکن است روی پیش بینی ما تاثیر منفی بگذارد. توی این مورد خاص چون ما اخبار فارسی را دسته بندی می کنیم، کلمات انگلیسی هم از متن اخبار حذف شده است. هر چند که تاثیر قابل ملاحظه ای در میزان دقت مدل ها نداشت ولی حجم نهایی مدل کمتر شده است. در مرحله ی بعد نیاز به Label Encoding دسته بندی ها داریم، به این معنا که مقادیر دسته بندی را به یک عدد متناظر می کنیم. در الگوریتم های یادگیری ماشین معمولا مقادیر رشته را به عدد تبدیل میکنیم چون هم محاسبات ساده تر و دقیق تر می شود و هم این که بعضی از الگوریتم ها با داده های غیر عددی قابل اجرا نیستند. کتابخانه هایی از جمله nltk کلمات ممنوعه ی فارسی را ندارند بنابراین از یک مجموعه ی منتشر شده در گیتهاب استفاده شد، البته کلمات و افعالی هم بصورت دستی به آن اضافه شده است. نکته ی مهمی که باید در نظر بگیریم در این پروژه به عنوان مثال فعل "است" از جملات حذف شده است ولی در "استان تهران" کلمه ی "استان" نباید تغییر کند.

پیاده سازی مدل 1: Naïve Bayes

معمولا برای تبدیل کلمات به vector از دو روش Bag of Words و TFIDF استفاده می شود. در این پروژه برای پیاده سازی هر سه مدل از روش TFIDF استفاده شده است به دلیل این که در روش اول کلمات رایج در متن وزن بیشتری پیدا میکنند و به وزن کلمات در کل مجموعه اهمیتی نمی دهد، در مقابل روش Bag of Words سرعت و کارایی بهتری در بعضی مسائل مثل تشخیص ایمیل های اسپم و تحلیل مثبت و منفی بودن متن از لحاظ احساسات دارد. بعد از این که داده ها را به دو قسمت train و تست تقسیم کردیم مدل Multinomial Naïve Bayes خودمون رو fit و روی داده های تست Predict میکنیم. در الگوریتم Naïve Bayes از روش Multinomial برای داده های گسسته و از Gaussian برای داده های پیوسته استفاده می شود. البته از توزیع برنولی هم میتوان برای نوع داده های بیتی یا خام هم استفاده کرد.

جدول 1- دقت مدل Naive Bayes

Accuracy	Precision	Recall	F1-Score
0.7507246376811594	0.8439397860593513	0.7507246376811594	0.7289311127816501

همانطور که در جدول شماره 1 مشاهده می کنیم. دقت این مدل حدود 75 درصد است که عدد خوبی نیست. از دلایل این که این الگوریتم مناسب مسئله ی ما نیست میتونیم به 2 مورد اشاره کنیم:

1) استقلال نداشتن کلمات: الگوریتم Naïve Bayes فرض میکند که کلمات استقلال دارند. به عنوان مثال در مجوعه ی "تیم استقلال تهران" هر سه کلمه ی "تیم"، "استقلال" و "تهران" به عنوان یک کلمه ی مستقل دیده می شود پس اگر مثلا کلمه ی "تهران" چند بار در جمله تکرار شود این الگوریتم پیش بینی میکند که جمله ی ما جزو دسته بندی اخبار ورزشی است. این ضعف بیشتر مربوط به داده های train ما می شود که با انتخاب داده ی مناسب تر و پیش پردازش بهتر میتوان تا حدودی دقت این الگوریتم را بالا برد.
2) در الگوریتم ما بصورت پیش فرض از هموار سازی Laplace استفاده شده است. بطور خلاصه از هموارسازی یا smoothing برای جلوگیری از صفر شدن احتمال کلمات دیده نشده هنگام train استفاده می شود. شاید روش های دیگه ی هموارسازی مثل Add-k و Good-Turing دقت مدل ما را بیشتر کند.

از آنجایی که دو مدل بعدی ای که پیاده سازی شده است دقت خوبی را به ما می دهد از بررسی بیشتر این الگوریتم صرف نظر

شده است. البته مقدار precision این مدل تقریبا معادل 84% است که این نشان میدهد در پیش بینی های True Positive نسبتا دقت بهتری دارد.

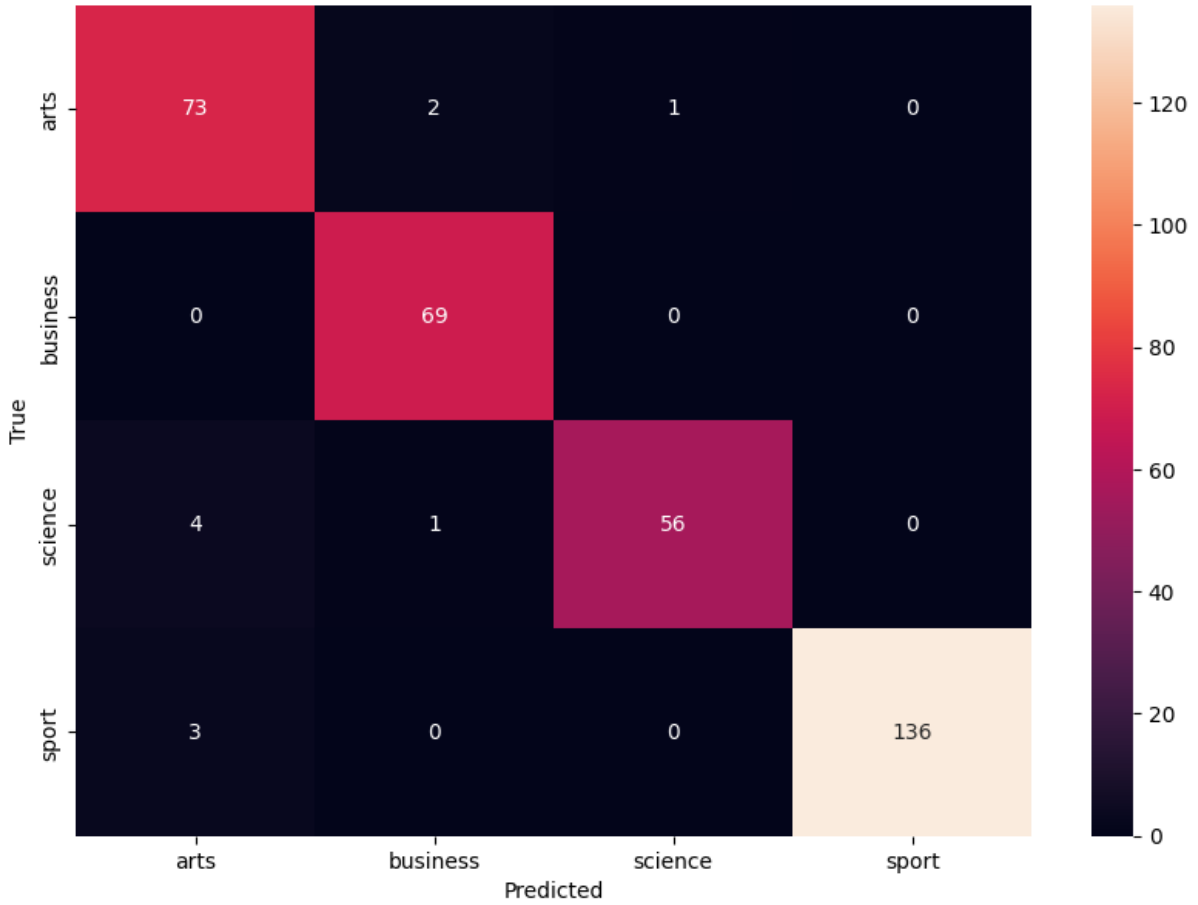
پیاده سازی مدل 2: ماشین بردار پشتیبان

از آنجایی که داده ی ما خطی نیست، کرنل SVC (Support Vector Classifier) را RBF یا Radial Basis Function انتخاب میکنیم. بطور خلاصه در مرحله ی Train، الگوریتم ما صفحات Hyperplane ای را شناسایی میکند که فاصله ی بین هر دو کلاس بیشترین مقدار ممکن باشد. از آنجایی که این الگوریتم استقلال ویژگی ها را در نظر ندارد، همانطور که انتظار داشتیم دقت این مدل به مراتب از Naïve Bayes بهتر است.

جدول 2 - دقت مدل بردار ماشین

Accuracy	Precision	Recall	F1-Score
0.9681159420289855	0.9692893465547928	0.9681159420289855	0.9682378593339256

ماتریس درهم ریختگی (Confusion Matrix) این مدل را روی داده هایی که برای تست در نظر گرفتیم در شکل 2 مشاهده میکنید.



شکل 2 - جدول درهم ریختگی مدل ماشین بردار پشتیبان

برای مثال میتوان از ماتریس متوجه شد که 136 مورد از اخبار ورزشی درست پیش بینی شده اند و 3 مورد هم به اشتباه در دسته ی اخبار هنری قرار دارند،

پیاده سازی مدل 3: ANN

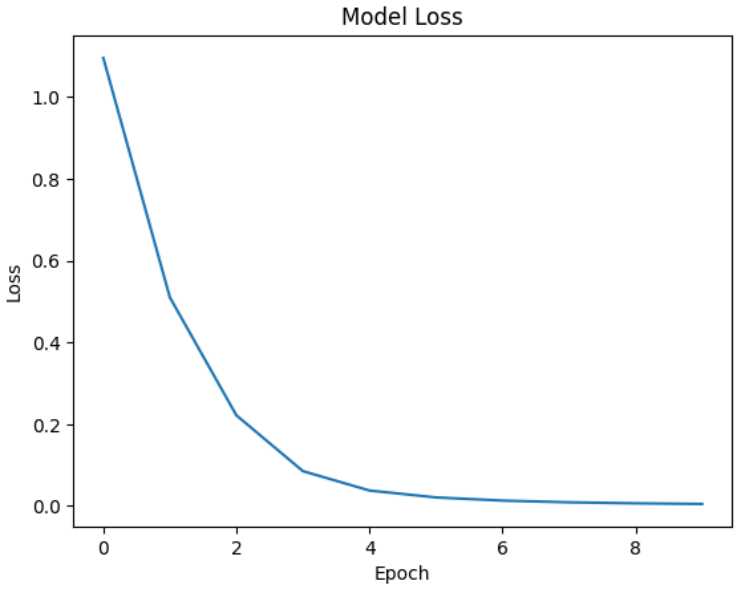
به عنوان آخرین مدل، یک شبکه ی عصبی مصنوعی ساخته شد، ابتدا از سه لایه ی dense 128، 64 و 16 به همراه یک لایه ی SoftMax استفاده شده که دقت 97 درصد روی داده های تست داشت، با حذف لایه 128 تایی dense تفاوت آشکاری در دقت مدل رو شاهد نبودیم ولی حجم نهایی مدل تقریبا به نصف کاهش پیدا کرد. لایه ی SoftMax پیش بینی نهایی هر کلاس را بین اعداد 0 تا 1 نرمال میکند و میتوانیم با argmax بیشترین تقریب را به عنوان دسته بندی پیش بینی شده برای خبر انتخاب کنیم. برای تابع loss از categorical_crossentropy که برای معمولا برای دسته بندی های multi-class استفاده می شود. برای optimizer تابع gradient descent از الگوریتم Adam استفاده شد. معمولا در شبکه های عصبی از این optimizer استفاده می شود به این دلیل که در یافتن نقطه مینیمم سراسری عملکرد خوبی دارد و نیاز به تنظیم پارامتر ها را کم میکند. از جانشین های این الگوریتم میتوان به stochastic gradient descent اشاره کرد.

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 64)	2,928,320
dense_4 (Dense)	(None, 16)	1,040
dense_5 (Dense)	(None, 4)	68

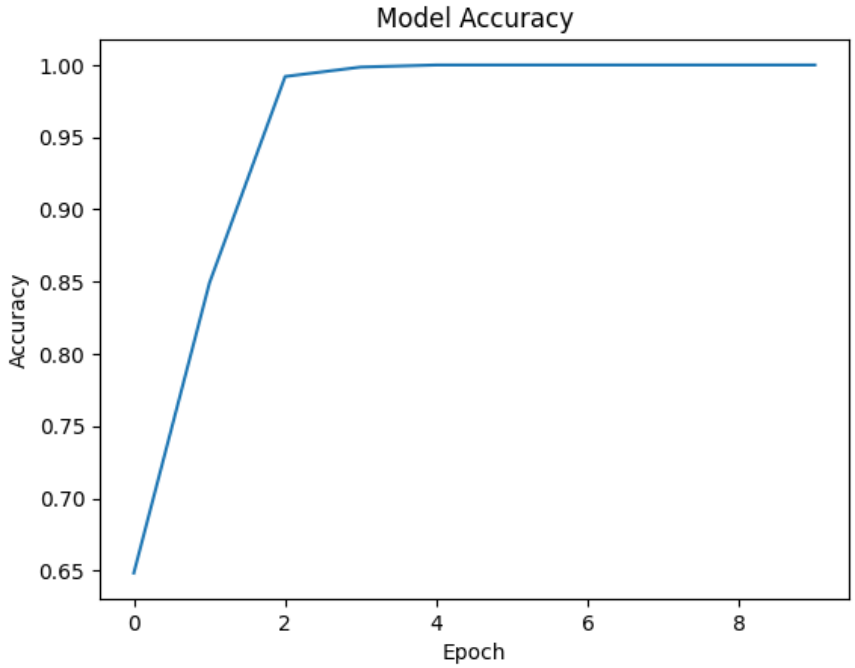
Total params: 8,788,286 (33.52 MB)
Trainable params: 2,929,428 (11.17 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 5,858,858 (22.35 MB)

شکل 3- پارامتر های لایه های مختلف مدل شبکه ی عصبی

تعداد epoch های train را برابر 10 و روی runtime gpu ابزار google colab اجرا شده است.



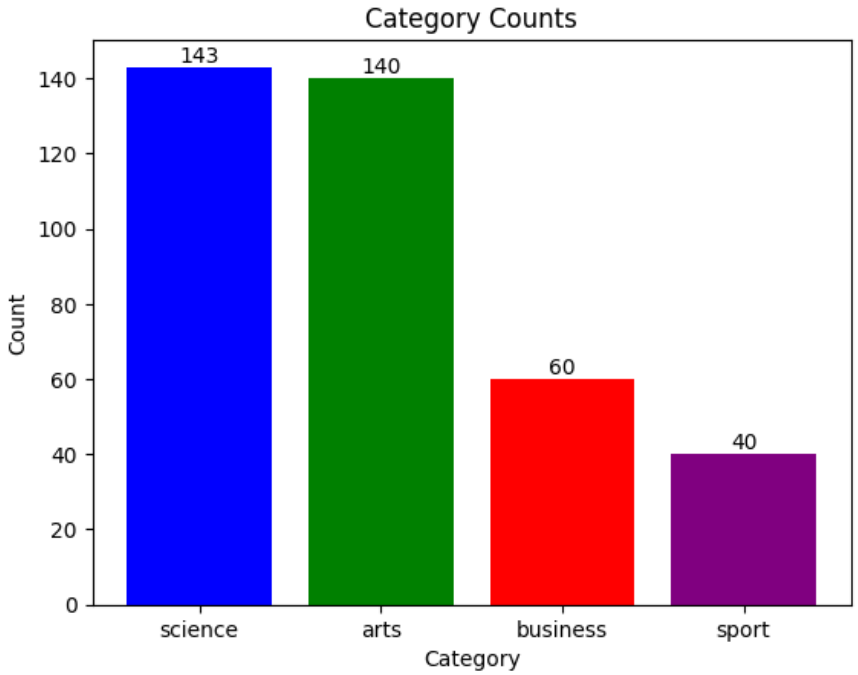
شکل 4- مقدار loss در epoch های مختلف



شکل 5 - میزان دقت مدل در epoch های مدل

تست روی داده های جدید

در 4 دسته بندی اخبار ورزشی، اقتصادی-تجاری، هنری و علمی تعدادی خبر از خبرگزاری های تسنیم و تجارت نیوز مربوط به تاریخ 10 مهر 1403 جمع آوری شد و این داده های جدید مدل های SVC و ANN را تست کردیم. دقت هر دو بیشتر از 85 درصد نشد. از مهمترین دلایل این که دقت رو این دسته کمتر از داده ی قبلی است میتوان به دایره لغات استفاده شده در این مجموعه اشاره کرد که کمی متفاوت است.



شکل 6 - توزیع داده های تست شده ی مجموعه اخبار تسنیم و تجارت نیوز

شکل 7- ابر واژگان داده ی اخبار تسنیم و تجارت نیوز

شکل 8- ابر واژگان اخبار بی بی سی فارسی

نتیجه گیری

برای مسائلی از این دست فراهم نمودن یک دیتاست مناسب که دایره لغات بالا و مناسبی را شامل شود یکی از چالش هایی است که با آن برخورد خواهیم کرد. چالش بعدی پیش پردازش داده است، حذف شکل های مختلف افعال، کلمات ربطی و... که نیازمند جمع آوری یک دیتاست مناسب است. در خصوص مدلسازی برای این دست مسائل معمولا از ماشین بردار پشتیان استفاده می شود ولی اگر مسئله خیلی پیچیده باشد میتوانیم از شبکه های عصبی هم استفاده کنیم. توی این پروژه به دلیل سادگی مدل شبکه ی عصبی ما پیچیدگی زیادی نیاز نداشت. در مسائلی که استقلال کلمات مورد نیاز از میتوانیم از الگوریتم Naïve Bayes استفاده کنیم، به عنوان مثال برای تشخیص زبان یک جمله از بین زبان های عربی، فارسی، انگلیسی و آلمانی این الگوریتم در عین سادگی دقت بسیاری خوبی خواهد داشت. اگر دیتاستی جمع آوری کنیم که دایره لغات آن در داده های Train ما نباشد دقت پیش بینی ما به مراتب کمتر خواهد بود، در چنین شرایطی استفاده از الگوریتم های unsupervised برای خوشه بندی و تخمین یک label از اطلاعاتی که مدل از قبل دارد میتواند یا رویکرد semi-supervised میتواند روش بهتر باشد.