دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

پایان‌نامه کارشناسی‌ارشد

درس یادگیری ماشین

پروژه نهایی

نگارش

عارف عزیزیان

استاد راهنما

دکتر قطعی

دی ۱۴۰۲

فهرست مطالب

عنوان صفحه

۱ مقدمه ۲
۱-۱ مقدمه نویسنده ۳
۱-۲ مقدمه مقاله ۳

۲ آماده‌سازی داده ۵
۲-۱ توزیع برچسب‌ها ۶
۲-۲ پاکسازی متن ۶
۲-۲-۱ نمونه پاکسازی متن ۶
۲-۳ تقسیم داده ۷

۳ روش‌ها ۸
۳-۱ روش‌های استخراج ویژگی ۹
۳-۱-۱ بسته کلمات ۹
۳-۱-۲ $TFIDF$ ۹
۳-۱-۳ $BERT$ ۹
۳-۲ مدل‌های کلاسیک ۱۰
۳-۳ شبکه عصبی ۱۰
۳-۳-۱ آموزش شبکه عصبی ۱۰

۴ نتایج ۱۲
۴-۱ نحوه آزمایش ۱۳
۴-۲ آزمایش درون دامنه‌ای ۱۳
۴-۲-۱ دیتاست $Parenting$ ۱۴
۴-۲-۲ دیتاست Gun ۱۵
۴-۲-۳ دیتاست $MeToo$ ۱۷
۴-۳ آزمایش میان دامنه‌ای ۱۹
۴-۳-۱ مدل $Parenting$ ۱۹
۴-۳-۲ مدل Gun ۲۰
۴-۳-۳ مدل $MeToo$ ۲۱
۴-۴ نتیجه‌گیری ۲۲

۵ جمع‌بندی ۲۳
۵-۱ اقدامات ۲۴

۵-۲ مقایسه نتایج بدست آمده و نتایج مقاله ۲۴

منابع و مراجع ۲۵

پیوست ۲۶

فهرست اشکال

شکل صفحه

۱-۱ اسپم محتوا محور ۳

۱-۲ توزیع برچسب‌ها ۶

۱-۳ ساختار شبکه عصبی ۱۱

۴-۱ $Parenting - WithinDomain - NN$ ۱۴

۴-۲ $Parenting - WithinDomain - RF$ ۱۴

۴-۳ $Parenting - WithinDomain - kNN$ ۱۵

۴-۴ $Gun - WithinDomain - NN$ ۱۵

۴-۵ $Gun - WithinDomain - RF$ ۱۶

۴-۶ $Gun - WithinDomain - kNN$ ۱۶

۴-۷ $MeToo - WithinDomain - NN$ ۱۷

۴-۸ $MeToo - WithinDomain - RF$ ۱۷

۴-۹ $MeToo - WithinDomain - kNN$ ۱۸

۴-۱۰ $Parenting - CrossDomain - GunData$ ۱۹

۴-۱۱ $Parenting - CrossDomain - MeTooData$ ۲۰

۴-۱۲ $Gun - CrossDomain - ParentingData$ ۲۰

۴-۱۳ $Gun - CrossDomain - MeTooData$ ۲۱

۴-۱۴ $MeToo - CrossDomain - ParentingData$ ۲۱

۴-۱۵ $MeToo - CrossDomain - GunData$ ۲۲

فصل اول

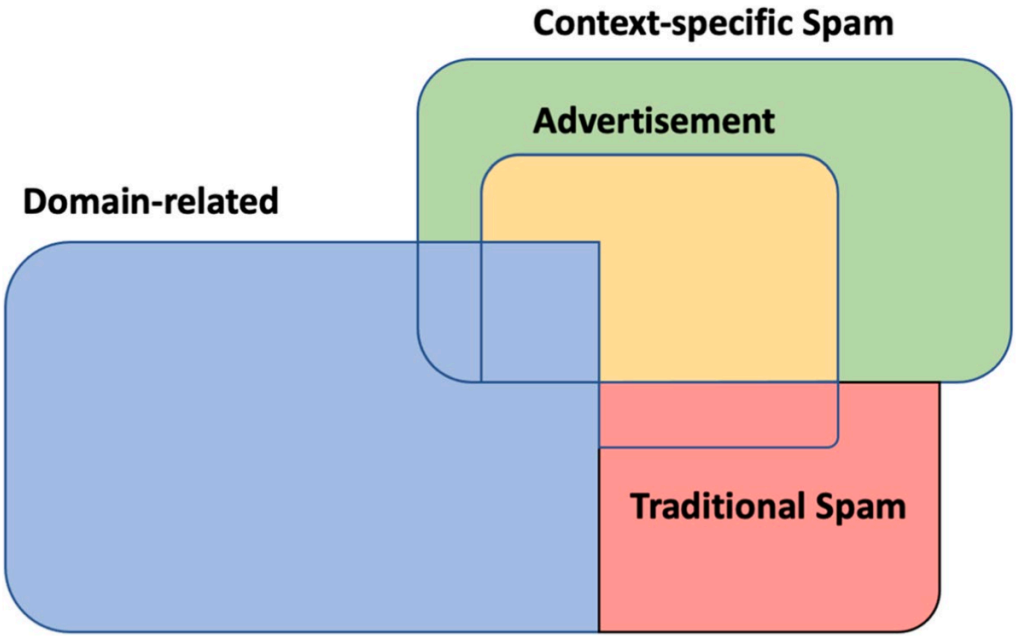
مقدمه

۱-۱ مقدمه نویسنده

تلاش من بر این است تا ضمن حفظ محتوای اصلی و روش‌های مقاله[۱]، خلاصه‌ای ارائه کنم و نتایج جدید که بعضا ممکن است تا حدی نیز با نتایج اعلام شده توسط مقاله متفاوت باشند را بررسی و اعلام کنم و در ادامه از جایگاه نویسنده اصلی خواهم نوشت.

۱-۲ مقدمه مقاله

امروزه به دلیل گسترش بازاریابی مستقیم آنلاین، پیام‌های اسپم و محتوای غیر ضروری در اینترنت و به ویژه در پلتفرم‌های اجتماعی مانند توییتر به سرعت در حال افزایش است. طبق تعریف، اسپم شامل محتوای ناخواسته و غیرقابل جستجوی آنلاین مانند پورنوگرافی، محتوای نامناسب یا بی‌معنی و تبلیغات تجاری است. این انواع مختلف اسپم در محیط رسانه‌های اجتماعی، به ویژه در پلتفرم‌هایی مانند توییتر، معنای جدیدی پیدا می‌کنند. به عنوان مثال، همه تبلیغات اسپم نیستند. هنگامی که تحلیل محتوایی روی توییت‌هایی درباره انتخابات انجام می‌شود، تبلیغات مربوط به پوشک، مربوط به بحث انتخابات نیست و اسپم تلقی می‌شود. اگر به جای آن، محتوای مربوط به فرزندپروری را تجزیه و تحلیل کنیم، تبلیغات پوشک برای تجزیه و تحلیل محتوا مرتبط هستند و ممکن است به عنوان اسپم دیده نشوند. به دلیل این ناهمخوانی‌ها، مفهوم اسپم محتوا محور را معرفی می‌کنیم و سعی می‌کنیم درک کنیم که چگونه می‌توان پست‌های حاوی این نوع اسپم و همچنین اشکال سنتی‌تر اسپم در توییتر را به طور دقیق شناسایی کرد.



شکل ۱-۱: اسپم محتوا محور

دیتاست ما به صورت پابلیک موجود است و از طریق گیتهاب پروژه میتوان به آن دسترسی داشت. این دیتاست از توییتر جمع‌آوری شده و شامل توییت‌هایی در ۳ هشتگ به زبان انگلیسی است که شامل موارد زیر است:

۱. $GunViolence$

۲. $MeToo$

۳. $Parenting$

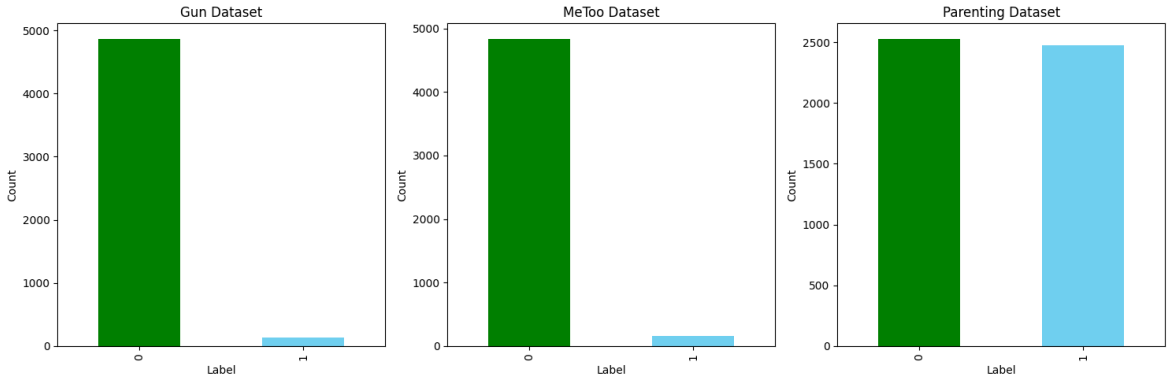
در ادامه به روشهای پاکسازی داده‌ها و نحوه طراحی مدل و اعلام نتایج میپردازیم.

فصل دوم

آماده‌سازی داده

۲-۱ توزیع برچسب‌ها

همانطور که در شکل ۲-۱ میبینیم بین ۳ دیتاست ما تنها یکی تقریبا داده‌های متوازن داشته و دو دیتاست دیگر شدیدا نامتوازن هستند، اما هدف ما ایجاد داده‌های جدید و متوازن سازی داده نیست و با همین داده‌ها ادامه می‌دهیم.



شکل ۲-۱: توزیع برچسب‌ها

۲-۲ پاکسازی متن

در یک توییت ممکن است هشتگ یا یوزرنیم یا لینک یا اعداد وجود داشته باشند که در مراحل بررسی ما تداخل ایجاد کنند، پس در اولین مرحله اقدام به حذف اینها کرده و در ادامه کارهایی مانند یکی کردن ریشه کلمات را انجام می‌دهیم و کلمات توقف را حذف می‌کنیم. بخش پاکسازی را با کمک کتابخانه $NLTK$ انجام دادیم.

۲-۲-۱ نمونه پاکسازی متن

Before: Pura Stainless Drinking Bottles Bundle Prizepack Giveaway @USER02 @USER03 URL01 Removed URL02 Removed

After: pura stainless drink bottl bundl prizepack giveaway

۲-۳ تقسیم داده

در اینجا برای هر دیتاست مراحل زیر را انجام می‌دهیم:

۱. با حفظ توزیع برچسب داده، ۱۰ درصد از داده را به عنوان داده آزمایش جدا می‌کنیم.

۲. از داده باقی مانده از مرحله قبل که داده آموزش است، ۱۰ درصد را با حفظ توزیع به عنوان داده اعتبارسنجی جدا می‌کنیم.

فصل سوم

روش‌ها

۳-۱ روش‌های استخراج ویژگی

برای استخراج ویژگی از متن‌ها و قابل فهم شدن آنها برای مدل‌های یادگیری ماشین، نیازمند روشهایی هستیم که در ادامه ذکر می‌کنیم.

۳-۱-۱ بسته کلمات

یکی از روشها، روش $BagOfWords$ است که تعداد تکرار هر کلمه در کل متن را می‌شمارد و یک ماتریس به عنوان خروجی می‌دهد. خروجی این روش را با الگوریتم‌های کلاسیک یادگیری ماشین مثل درخت تصمیم و یا جنگل تصادفی بررسی می‌کنیم و از این روش در شبکه عصبی استفاده نمی‌کنیم.

۳-۱-۲ $TFIDF$

$TFIDF$ یا همان $TermFrequencyInverseDocumentFrequency$، یک ماتریس است که میزان اهمیت هر کلمه نسبت به یک سند در مجموعه‌ای از اسناد را نشان می‌دهد.

۳-۱-۳ $BERT$

$BERT$ یا $BidirectionalEncoderRepresentationsfromTransformers$ یک شبکه عصبی بر پایه $Transformer$ است که توسط گوگل توسعه داده شده و روی دیتاست‌های بسیار بزرگ متنی آموزش دیده که توانایی درک زبان‌های طبیعی را به آن می‌دهد. از مدل‌های $BERT$ از پیش آموزش دیده میتوان استفاده کرد و خروجی را به عنوان ورودی به یک مدل جدید داد.

۳-۲ مدل‌های کلاسیک

ما از مدل‌های مختلفی شامل موارد زیر استفاده می‌کنیم:

۱. kNN - با استفاده از $TFIDF$

۲. $RandomForest$ - با استفاده از $BagOfWords$

در همه مدل‌های ذکر شده ما بوسیله $GridSearch$ هایپرپارامترهای مدل را تعیین می‌کنیم تا بهترین نتیجه را بگیریم.

*** قابل ذکر است که هرکدام از مدل‌ها با یکی از روش‌های استخراج ویژگی لرن و آزمایش می‌شود که توسط مقاله به عنوان بهترین نتایج منتشر شده‌اند و در بالا ذکر شده و شبکه عصبی نیز تنها با $BERT$ کار می‌کند.**

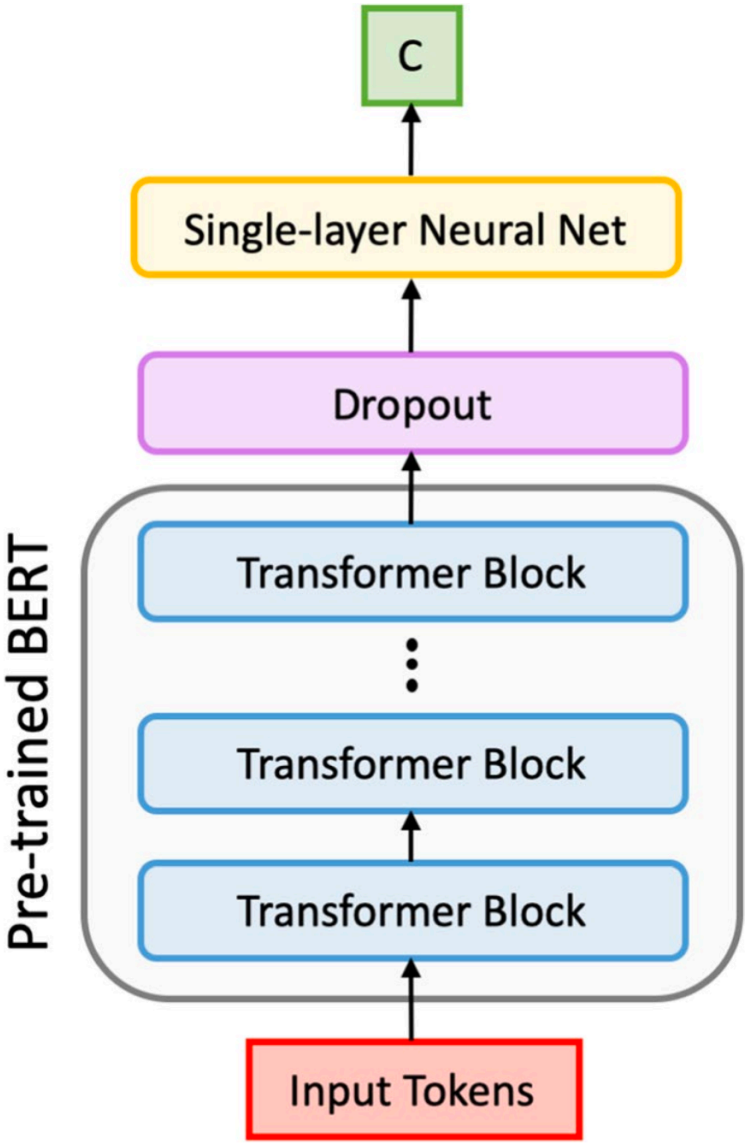
۳-۳ شبکه عصبی

ما مدل شبکه عصبی خود را با استفاده از کتابخانه $PyTorch$ می‌سازیم که ساختار آن را در ادامه می‌بینیم.

مطابق با شکل ۳-۱، در اولین مرحله ما با استفاده از $BERT-Tokenizer$ ورودی را توکنایز کرده و به عنوان ورودی به مدل $Pre-TrainedBERT$ می‌دهیم. در ادامه خروجی $BERT$ را به یک لایه $Dropout$ با $rate$ ۰/۱ می‌دهیم تا از $Overfit$ شدن جلوگیری کنیم، سپس خروجی را به یک شبکه تک لایه ($Linear$) میدهیم و بعد از دریافت خروجی از شبکه برای مشخص شدن جواب یک $Softmax$ روی خروجی اعمال می‌کنیم که C در واقع برچسب مشخص شده توسط خروجی $Softmax$ است.

۳-۳-۱ آموزش شبکه عصبی

این شبکه بوسیله بوسیله اپتیمایزر $Adam$ آموزش می‌بیند و از $CrossEntropyLoss$ به عنوان $LossFunction$ خود استفاده می‌کند. آموزش در ۲۰ $Epoch$ صورت می‌گیرد که البته شامل $EarlyStopping$ هم میشود که از داده اعتبار سنجی برای فهمیدن معیار توقف استفاده می‌کند.



شکل ۳-۱: ساختار شبکه عصبی

فصل چهارم

نتایج

۴-۱ نحوه آزمایش

آزمایش ما به دو دسته کلی تقسیم می‌شود:

۱. تشخیص اسپم بودن یک دسته از ۳ دسته به وسیله مدل لرن شده روی همان دسته

۲. تشخیص اسپم بودن یک دسته از ۳ دسته به وسیله مدل‌های لرن شده روی یکی از ۲ دسته دیگر

به دسته اول **آزمایش درون دامنه‌ای** و به دسته دوم **آزمایش میان دامنه‌ای** می‌گوییم.

۴-۲ آزمایش درون دامنه‌ای

نتایج به شکل زیر می‌باشند.

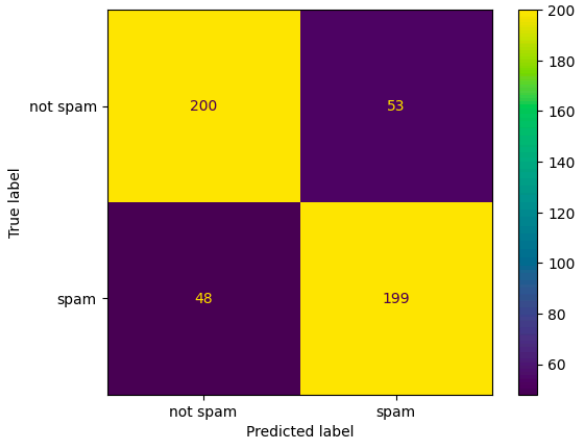
Dataset	FeatureExtraction	Method	Precision	Recall	F1
Parenting	Bert	NN	0.80	0.80	0.80
	BagOfWords	RF	0.76	0.75	0.75
	TFIDF	kNN	0.78	0.78	0.78
Gun	Bert	NN	0.69	0.79	0.73
	BagOfWords	RF	0.49	0.50	0.49
	TFIDF	kNN	0.69	0.65	0.67
MeToo	Bert	NN	0.84	0.90	0.87
	BagOfWords	RF	0.48	0.50	0.49
	TFIDF	kNN	0.84	0.72	0.76

همانطور که در جدول بالا می‌بینیم، بهترین عملگرد بین تمامی مدل‌ها در هر دیتاست متعلق به شبکه عصبی و مدل برت می‌باشد و پس از آن kNN از $RandomForest$ بهتر عمل کرده است.

۴-۲-۱ دیتاست *Parenting*

شبکه عصبی

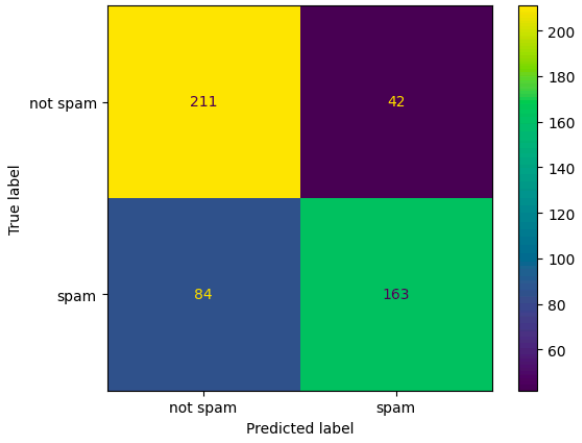
شبکه عصبی در اینجا خوب عمل کرده که شاید توازن داده دلیل اصلی آن باشد.



شکل ۴-۱: $Parenting - WithinDomain - NN$

جنگل تصادفی

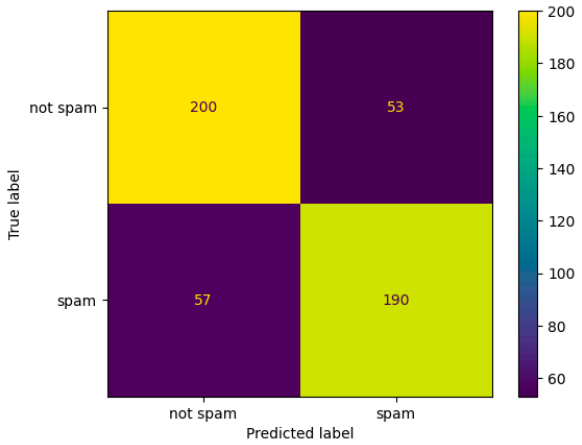
مشکل این روش تعداد زیاد داده‌های اسپمی است که به اشتباه عادی تشخیص داده شده‌اند.



شکل ۴-۲: $Parenting - WithinDomain - RF$

kNN

در این قسمت میبینیم که نتایج تفاوت کمی با روش شبکه عصبی دارد.

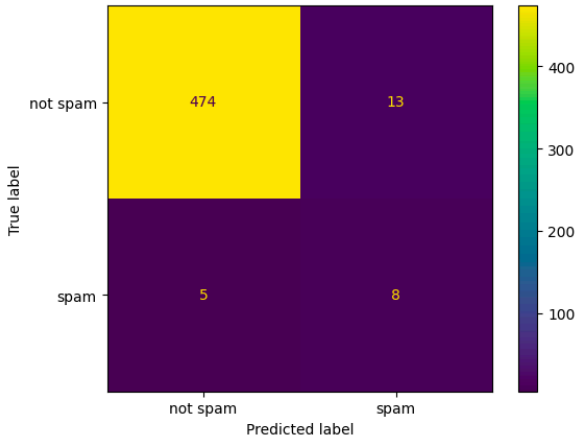


شکل ۴-۳: $Parenting - WithinDomain - kNN$

۴-۲-۲ دیتاست Gun

شبکه عصبی

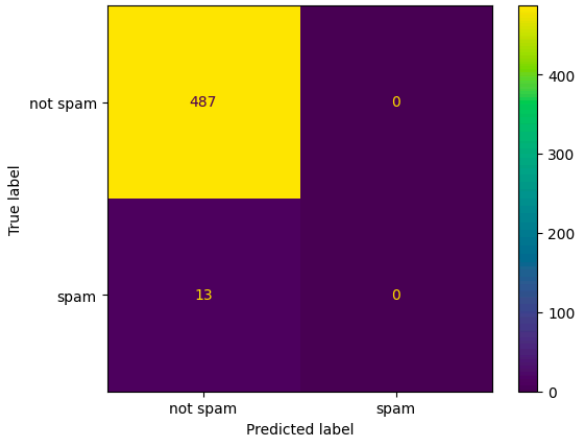
در اینجا داده‌های غیر اسپم به خوبی تشخیص داده شده‌اند ولی در اسپم‌ها با اینکه اکثریت درست تشخیص داده شده‌اند با توجه به تعداد کم نتیجه قابل قبول نیست.



شکل ۴-۴: $Gun - WithinDomain - NN$

جنگل تصادفی

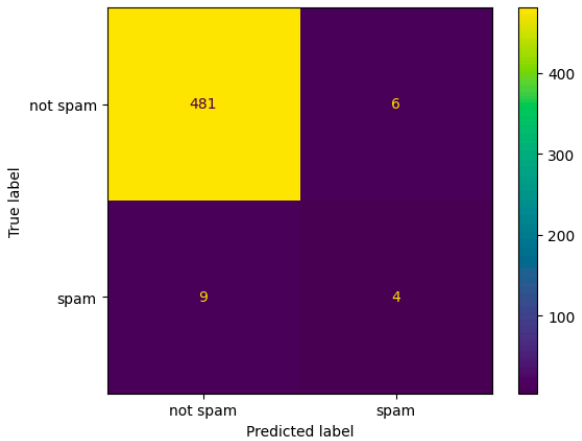
داده‌های غیر اسپم کاملا صحیح و داده‌های اسپم همه غلط پیش‌بینی شده‌اند که این نتیجه به دلیل تعداد داده کم اسپم است.



شکل ۴-۵: $Gun - WithinDomain - RF$

kNN

میبینیم که در مقایسه با جنگل تصادفی، kNN توانسته با وجود داده‌های بسیار کم تعدادی از اسپم‌ها را شناسایی کند.

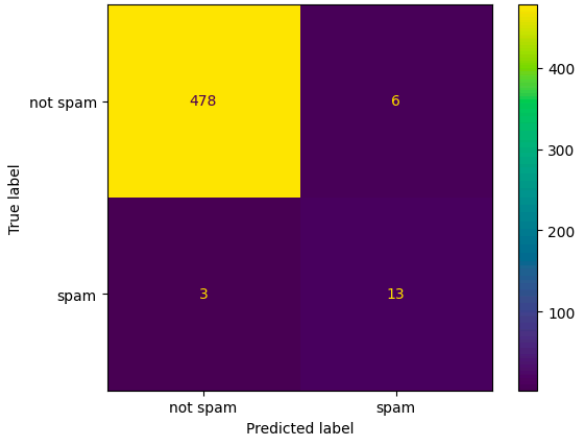


شکل ۴-۶: $Gun - WithinDomain - kNN$

۴-۲-۳ دیتاست $MeToo$

شبکه عصبی

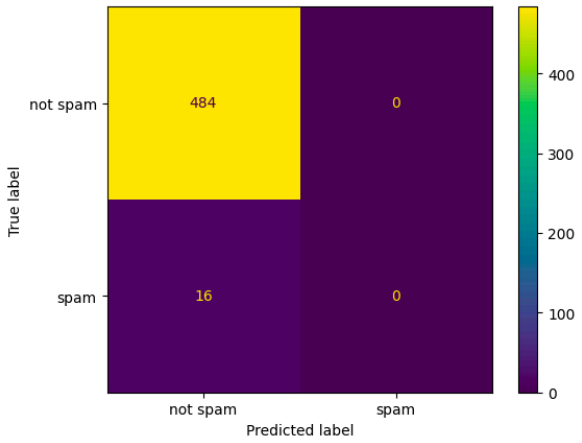
شبکه عصبی در داده‌های $MeToo$ با وجود نامتوازن بودن داده‌ها که مانند داده‌های Gun، بهتر از داده‌های Gun عمل کرده.



شکل ۴-۷: $MeToo - WithinDomain - NN$

جنگل تصادفی

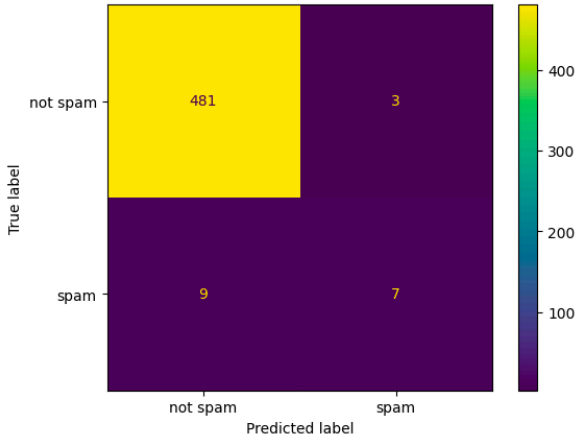
عینا مانند داده‌های Gun، در اینجا هم جنگل تصادفی همه نرمال‌ها را صحیح و همه اسپم‌ها را غلط تشخیص داده.



شکل ۴-۸: $MeToo - WithinDomain - RF$

kNN

و باز هم مثل داده‌های Gun، kNN در اینجا هم بهتر از جنگل تصادفی عمل کرده.



شکل ۴-۹: $MeToo - WithinDomain - kNN$

۴-۳ آزمایش میان دامنه‌ای

اهمیت این آزمایش از این نظر است که می‌تواند مشخص کند که آیا مدل زبانی ما توانایی تشخیص اسپم بودن را تنها در زمینه‌ای که لرن شده دارد یا توانایی تشخیص دیگر موضوعات را نیز داراست.

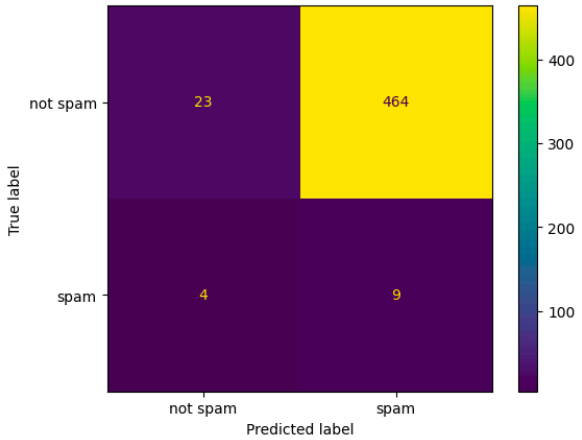
LearnDataset	TestDataset	Precision	Recall	F1
Parenting	Gun	0.44	0.37	0.06
	Metoo	0.52	0.50	0.04
Gun	Parenting	0.70	0.53	0.40
	MeToo	0.48	0.50	0.49
MeToo	Parenting	0.69	0.53	0.41
	Gun	0.99	0.62	0.68

همانطور که می‌بینیم مدلی که با داده‌های $Parenting$ لرن شده نمیتواند به خوبی در مورد داده‌های دیگر تصمیم گیری کند که یکی از دلایل بزرگ این موضوع متوازن نبودن داده‌های دیگر است.

۴-۳-۱ مدل $Parenting$

داده Gun

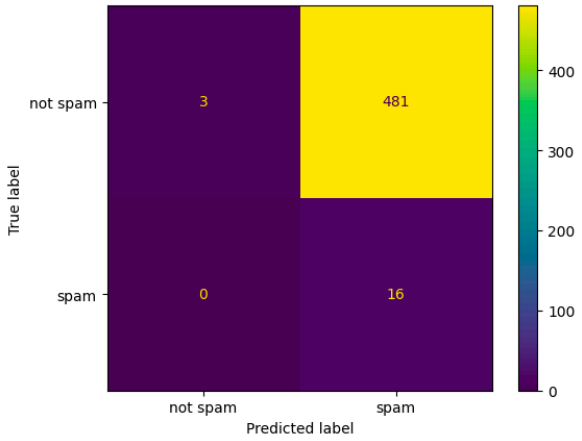
مشکل اصلی مدل در اینجا تشخیص اشتباه داده‌های نرمال به صورت اسپم است.



شکل ۴-۱۰: $Parenting - CrossDomain - GunData$

داده $MeToo$

مدل در اینجا تقریبا تمام داده‌ها را فارغ از برچسب حقیقی اسپم تشخیص داده است.

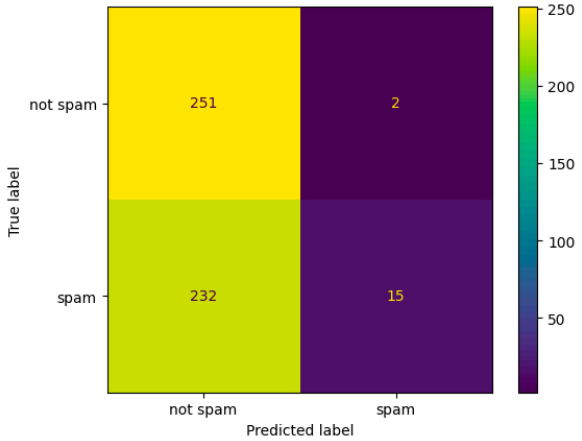


شکل ۴-۱۱: $Parenting - CrossDomain - MeTooData$

۴-۳-۲ مدل Gun

داده $Parenting$

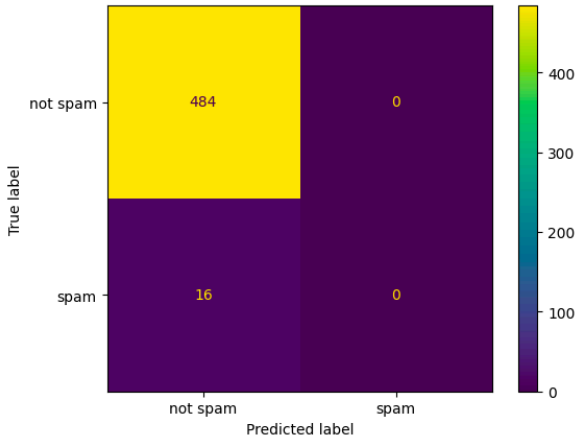
مدل به خوبی داده‌های غیراسپم را تشخیص داده ولی به دلیل زیاد ندیدن داده اسپم، آنها را نیز نرمال تشخیص داده و میبینیم که مدل کاملا به داده غیراسپم سوگیری دارد.



شکل ۴-۱۲: $Gun - CrossDomain - ParentingData$

داده $MeToo$

در اینجا سوگیری به سمت داده غیراسپم کاملا قابل مشاهده است و میبینیم که حتی یک داده از هرکدام از برچسب‌ها هم اسپم تشخیص داده نشده.

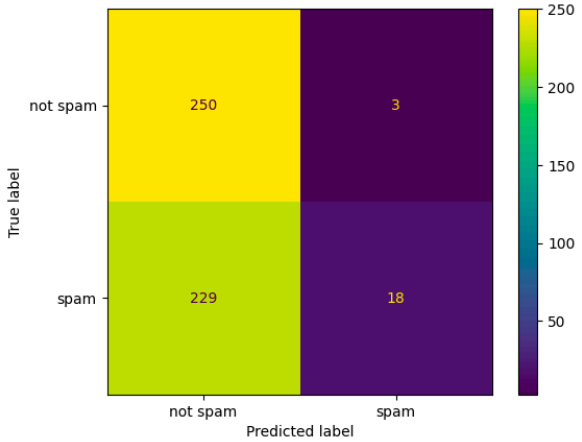


شکل ۴-۱۳: $Gun - CrossDomain - MeTooData$

۴-۳-۳ مدل $MeToo$

داده $Parenting$

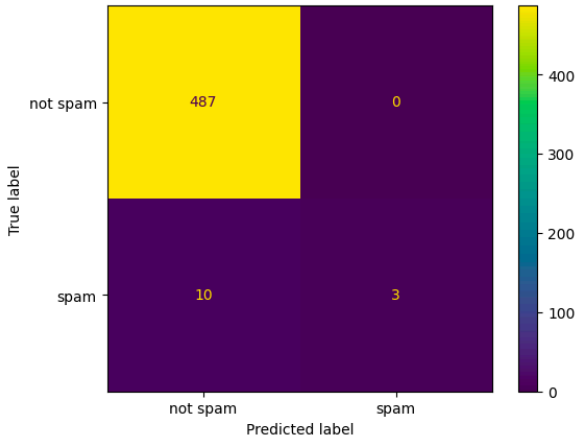
میبینیم که سوگیری این مدل از مدل Gun کمتر است اما بازهم باعث نتایج بدی شده است.



شکل ۴-۱۴: $MeToo - CrossDomain - ParentingData$

داده Gun

بهترین نتیجه بین تمام داده‌ها و مدل‌ها در آزمایش میان دامنه‌ای متعلق به این بخش می‌باشد که داده‌های اسپم Gun را تقریبا مثل خود مدل Gun تشخیص داده است.



شکل ۴-۱۵: $MeToo - CrossDomain - GunData$

۴-۴ نتیجه‌گیری

در مدل‌هایی که داده و مدل یکی بودند نتیجه خوب شد ولی در آزمایش‌های میان‌دامنه‌ای نتیجه به شدت افت کرد و این موضوع نشانگر این است که مدل باید در موضوعات مختلف لرن شود تا بتواند در مورد زمینه‌های مختلف تصمیم‌گیری کند.

فصل پنجم

جمع‌بندی

۵-۱ اقدامات

اقدامات انجام شده در این پروژه به شرح زیر می‌باشد:

۱. بررسی دیتاست و پیش‌پردازش و پاکسازی آن

۲. پیاده‌سازی چند بخش از الگوریتم‌های کلاسیک استفاده شده در مقاله

۳. پیاده‌سازی شبکه عصبی ذکر شده در مقاله با رعایت تمامی جزئیات

۴. مقایسه و دسته‌بندی نتایج بدست آمده

۵. ارائه کامل کد برای تمامی مراحل ذکر شده (لینک در پیوست)

۵-۲ مقایسه نتایج بدست آمده و نتایج مقاله

به مقایسه نتایج بدست آمده در آزمایش‌های درون‌دامنه‌ای با نتایج مقاله در مدل شبکه عصبی می‌پردازیم:

Dataset	Source	Precision	Recall	F1
Parenting	Paper	0.72	0.56	0.63
	Ours	**0.80**	**0.80**	**0.80**
Gun	Paper	**0.76**	0.66	0.70
	Ours	0.69	**0.79**	**0.73**
MeToo	Paper	**0.89**	0.85	0.87
	Ours	0.84	**0.90**	**0.87**

* **نتایج بهتر** $Bold$ **شده‌اند.**

نتیجه‌ای که به وضوح با مقاله اختلاف زیادی دارد مربوط به داده‌های $Parenting$ است که شاید به دلیل $Pre-Process$ باشد.

منابع و مراجع

[1] Kawintiranon, Kornraphop, Singh, Lisa, and Budak, Ceren. Traditional and context-specific spam detection in low resource settings. Machine Learning, 2022.

پیوست

کدهای پروژه در $GoogleColab$ بارگزاری شده و توسط لینک زیر قابل دسترسی است.

Project Source