محمدرضا شریعتی، امیرحسین ایمانی، علی وحیدی، امیررضا دواچی

هدف این پروژه طراحی الگوریتمی برای قرار دهی آگهی تبلیغات در کنار متون مرتبط میباشد.

گام های طی شده برای میسر کردن این هدف عبارتند از:

۱. ساختن مدل زبانی با استفاده از داده های بدست آمده از سایت های فارسی
۲. استخراج مجموعه دادهای مربوط به محصولات
۳. تمیز کردن و آشنا شدن با داده های استخراج شده مربوط به محصولات
۴. انتخاب مدل مناسب برای پیشنهاد دادن تبلیغ مربوط به متن
۵. اعتبار سنجی مدل

ساختن مدل زبانی

روش های متفاوتی برای ساختن مدل های زبانی وجود دارند. در این پروژه از روش TF-IDF استفاده شده است. این مدل در مقایسه با دیگر مدل های فرکانسی مانند Bag of Words یا One Hot عملکرد بهتری داشته و میتواند معنای موجود در جمله را بهتر استخراج کند، زیرا علاوه بر اینکه تعداد تکرار هر کلمه در جمله را در نظر میگرد، با بدست آوردن معیاری برای تکرار شدن آن کلمه در کل داده موجود سعی میکند اهمیت هر کلمه را با توجه به متن مورد نظر محاسبه کند. از طرفی این روش در مقایسه با روش های Distributed Representation ساده تر است و نیاز به قدرت پردازشی و داده کمتری دارد.

برای پردازش متن در این مرحله از کتابخانه هضم استفاده شده است. این کتابخانه امکانات خوبی را برای پردازش و تمیز کردن متن فارسی پیاده سازی کرده است.

بعد از پردازش متن های موجود، تعداد ۲۶۵۴۹۷ کلمه به دست آمد. این عدد همچنین تعداد بعد های feature space داده ها ر انمایش میدهد که بسیار بزرگ است.

استخراج داده های محصولات

برای استخراج داده های مربوط به محصولات و دسته بندی آنها از کتابخانه scrapy استفاده شده است. داده های محصولات از سایت دیجیکالا استخراج شده اند. به این صورت که برنامه به صفحه های مربوط به محصولات مختلف سر زده و دو نوع داده برای هر محصول استخراج میکند. اول دسته بندی که آن محصول در آن قرار میگیرد، دوم توضیحات یا خلاصه معرفی محصول. از دسته بندی هر محصول در لیبل زدن آن محصول استفاده میشود و از متن خلاصه هر محصول برای یادگیری ماشین به منظور حدس زدن دسته بندی محصول استفاده میشود. به این ترتیب وقتی متن جدیدی در اختیار الگوریتم قرار میگیرد سعی میکند دسته بندی که به توضیحات موجود در متن نزدیک است را تشخیص دهد. سپس میتوانیم از محصولات موجود در آن دسته بندی برای تبلیغ در کنار متن ورودی استفاده کنیم.

داده خروجی این محصول یک فایل با فرمت csv میباشد که شامل اسم محصول، دسته بندی و توضیحات محصول در هر سطر است. بعد از گذشتن حدود یک ساعت و نیم این برنامه داده مربوط به ۳۵۶۴ محصول را استخراج کرده بود.

آشنا شدن با داده های استخراج شده

داده های استخراج شده نشان میدهند که:

- در مجموع داده ها مربوط به ۲۶۰ دسته بندی متفاوت میباشند.
- نمودار تعداد رکورد ها برای هر دسته بندی نشان میدهد از تعداد کمی از دسته بندی ها تعداد زیادی محصول استخراج شده در حالی که از تعداد زیادی از دسته بندی ها تعداد کمی محصول استخراج شده است.
- بیشترین طول متن موجود در داده ها ۱۰۷۸ کلمه و کمترین طول ۱ کلمه و میانگین طول ۱۸۰ کلمه میباشد.
- ۵۰ درصد متن های استخراج شده دارای طولی بین ۱۰۷ تا ۲۳۸ کلمه میباشند. این بازه احتمالا بازه مناسبی برای مشخص کرد نطول ورودی الگوریتم باشد.
- تعداد ۲۰۹ خلاصه و ۲۷ نام محصول در داده ها به صورت تکراری بودند که از دیتاست حذف شدند.
- تنها ۳۹ دسته بندی در داده حضور داشتند که برای هرکدام بیشتر از ۲۰ محصول استخراج شده بود.
- ۲۴۲۳ محصول استخراج شده مربوط به این ۳۹ دسته بندی بودند.

انتخاب مدل مناسب

مسئله ای که در تلاش برای حل آنیم از نوع مسائل `multi-class classification` میباشد. مدل های محدودی هستند که به صورت طبیعی قادر به حل اینگونه مسائلند. از این مدل ها میتوان به , `Naive Bayes, Stochastic Gradient Decent, KNN, Decision Tree` و شبکه های عصبی اشاره کرد. راه حل هایی وجود دارد که مدل هایی مثل `LogisticRegression` که اساسا کلاسبندی دو دویی انجام میدهند را به طوری تغییر دهیم که قادر به حل اینگونه مسائل شوند.

دیتا ورودی مدل ما ماتریسی با ابعاد ۲۴۲۳ در ۲۶۵۴۹۷ میباشد که تعداد سطرها نشان دهنده تعداد نمونه ها و تعداد ستون ها نشان دهنده تعداد ویژگی ها میباشد. تعداد بسیار بالای ویژگی ها میتواند باعث دچار شدن به `Curse Of Dimensionality` شود. به این منظور سعی در استفاده از الگوریتمی شده است که میتواند در این شرایط موثر واقع شود. در این شرایط الگورتیمی مانند `KNN` عملا کاربرد خود را از دست میدهد. الگورتیم های دیگر هم میتوانند با توجه به تعداد کم داده به دیتاست `overfit` شوند.

اولین الگوریتم مورد استفاده `Logistic Regression` میباشد که یک الگوریتم ساده و خطی است که معمولا به عنوان یک بیسلاین و اولین الگوریتم مورد استفاده قرار میگیرد تا با در نظر گرفتن نتایج آن با ابعاد مسئله بیشتر آشنا شویم.

معیار هایی که برای سنجش مدل ها مورد استفاده قرار گرفته اند عبارتند از `accuracy` و `f1-score`

لازم به ذکر است به دلیل بیشتر بودن تعداد کلاس ها از دوتا مقادیر این معیار ها به دوشیوه اندازه گیری شده اند:

روش اول شیوه میکرو میباشد که بالا بودن مقدار آن نشان میدهد مدل به طور کلی خوب عمل میکند. این معیار نشان دهنده این نیست که مدل برای هر کلاس پیش بینی های قابل قبولی دارد. به همین منظور این معیار میتواند تاثیر پذیر از کلاس هایی با تعداد زیاد رکورد باشد.

روش دوم شیوه ماکرو وزن دار میباشد که بالا بودن مقدار آن نشان میدهد مدل به صورت فردی برای هر کلاس عملکرد خوبی دارد .این معیار به صورت وزن دار محاسبه میشود به این صورت کلاسی که دارای تعداد بیشتری رکورد است تاثیر بیشتری نیز در مقدار این معیار میگذارد. به کمک این دو معیار میتوان مدل را هم به صورت کلی و هم به صورت فردی برای هر کلاس ارزیابی نمود.

با استفاده از روش cross-validation برای تقسیم کردن داده ها به بخش های train و validation و استفاده از ۳ فولد نتایج بدست آمده برای الگوریتم Logistic Regression به صورت زیر میباشد:

```
accuracy: 94% f1-micro: 94%, f1-weighted: 94%
```

این مقادیر نشان میدهند الگوریتم ساده Logistic Regression توانسته است به خوبی از پس این مساله برآید.

گام بعدی برای انتخاب مدل و بهبود مدل موجود استفاده از PCA برای کم کردن تعداد بعد های فضای ویژگی ها میباشد. این کار میتواند سرعت مدل را برای پیشبینی افزایش دهد، سرعت یادگیری مدل را افزایش دهد، مقدار overfit شدن را کاهش دهد و فضای اشغال شده توسط مدل را کمتر کند.

در این مرحله به دلیل کاهش یافتن تعداد بعدها میتوان از مدل های دیگری همچون KNN, Random Forest, Naive Bayes و
SGD استفاده کرد.

برای به دست آوردن تعداد مناسب بعد برای هریک از مدل های فوق از روش Grid Search استفاده شده است. در این روش بـ ااستفاده از Cross-Validation و امتحان کردن تعداد مقادیر مختلف ابعاد و سپس ارزیابی مدل تعداد مناسب ابعاد برای استفاده در هر مدل به دست می آید. نتایج این مرحله به شرح زیر میباشند:

```
Model: GaussianNB, Best number of dimensions:97
acc: 0.9489164086687306, f1_micro: 0.9489164086687306, f1_weighted:
0.9484676155713491

Model: KNeighborsClassifier, Best number of dimensions: 1938
acc: 0.9592363261093911, f1_micro: 0.9592363261093911, f1_weighted:
0.9580465421594021

Model: RandomForestClassifier, Best number of dimensions: 253
acc:0.957688338493292, f1_micro:0.957688338493292, f1_weighted:0.95694662466677

Model: SGDClassifier, Best number of dimensions: 488
acc:0.977296181630547, f1_micro:0.977296181630547, f1_weighted:0.9767658746940895

Model: LogisticRegression, Best number of dimensions: 606
acc:0.9530443756449948, f1_micro:0.9530443756449948, f1_weighted:0.9495052936262306
```

بر اساس داده های به دست آمده مدل SGD بهترین عملکرد را داشته است. اما همه مدل ها تا حد قابل قبولی کارایی مناسبی نشان داده اند. سرعت یادگیری و حدس برای هر یک از مدل های بالا به صورت زیر میباشد:

```
GaussianNB Mean Score Time: 0.015824635823567707
GaussianNB Mean Train Time: 0.037443955739339195
KNeighborsClassifier Mean Score Time: 2.045650323232015
```

KNeighborsClassifier Mean Train Time: 0.12410640716552734
RandomForestClassifier Mean Score Time: 0.028568267822265625
RandomForestClassifier Mean Train Time: 2.1498297850290933
SGDClassifier Mean Score Time: 0.003428379694620768
SGDClassifier Mean Train Time: 0.31776857376098633
LogisticRegression Mean Score Time: 0.004302422205607097 LogisticRegression
Mean Train Time: 0.2878740628560384

با توجه به نتایج فوق به نظر میرسد مدل Logistic Regression گزینه مناسبی به عنوان مدل نهایی باشد. این مدل علاوه بر اینکه دقت و عملکرد مناسبی دارد، از سرعت یادگیری و حدس بالایی برخوردار است. علاوه بر این خروجی این مدل احتمال میباشد که بر اساس آن میتوانیم با در نظر گرفتن یک مرز، بیشتر از یک دسته بندی برای تبلیغات معرفی کنیم.

اعتبار سنجی مدل انتخاب شده

برای اعتبار سنجی بیشتر مدل بدست آمده میتوان از confusion matrix استفاده کرد. ماتریس بدست آمده نشان دهنده این است که در بیشتر مواقع مدل کلاس را به درستی تشخیص داده است. همچنین مطالعه اشتباهاتی که مدل مرتکب شده است خالی از لطف نمیباشد.

پایه نگهدارنده گوشی و تبلت – گوشی موبایل : این مورد تا حدی قابل انتظار میباشد. احتمالا توضیحات مطرح شده برای پایه نگهدارنده گوشی و تبلت شباهت زیادی به توضیحات گوشی موبایل دارد و طبیعی است که مدل در مواردی پایه نگهدارنده را گوشی موبایل تشخیص دهد

کرم، بالم و لوسیون کودک و نوزاد – کرم ضد آفتاب: این مورد تا حدی قابل انتظار میباشد. دو دسته شباهت های زیادی به یکدیگر دارند و توصیفات دو دسته میتواند شبیه به یکدیگر باشد. طبیعی است که مدل در مواردی کرم و لوسیون کودک را کرم ضد آفتاب تشخیص دهد.

موارد زیادی مشاهده میشود که یک دسته بندی با دفتر چاپی اشتباه گرفته میشود. این مورد میتواند ناشی از کلی بودن توضیحات دسته بندی دفتر چاپی باشد. بهتر است برای دسته بندی هایی که دفتر چاپی تشخیص داده میشوند رکورد های بیشتری استخراج کرد تا مدل متوجه تفاوت هایشان بشود.

رسم نمودار تعداد تشخیص های درست و غلط بر اساس طول متن نشان دهنده آن است که مدل در طول های متفاوت متن به خوبی عمل میکند، اما در متن های کوتاه با طول هایی بین ۰ تا ۱۳۰ کلمه اشتباهات مدل بیشتر میباشد. این مشاهده میتواند نشانگر طول متن مناسب برای ورودی دادن به مدل باشد. بر اساس این نمودار متن هایی با طولی نزدیک به ۱۸۰ کلمه شانس بیشتری برای قرار گیری در دسته بندی درست دارند.

در آخر نیز نشان داده شده است که سایز مدل انتخاب شده بر روی دیسک ۱۹.۰ مگابایت میباشد و مدت زمان لازم برای یادگیری مدل روی دیتا استفاده شده ۳۶.۰ ثانیه میباشد. این موارد نیز میتوانند در انتخاب مدل موثر باشند.

گام آخر اعتبار سنجی مدل برروی داده های استخراج شده از ویکیپدیا میباشد که تاکنون آن هارا مشاهده نکرده است. در این مرحله پنجره ای به طول ۱۰۰ کلمه در نظر گرفته شده است که روی متن استخراج شده مربوط به هر دسته حرکت میکند و آن تکه متن ر ابه مدل ورودی میدهد و در آخر حدس های بدست آمده را با دسته بندی مورد نظر به عنوان لیبل مقایسه میکند و دقت مدل را برای آن دسته بندی خروجی میدهد.