**بسم الله الرحمن الرحیم**

**پروژه پردازش متن**

**برای محصولات دیجیکالا**

تیم ارائه دهنده: داده کاوانِ فونیکس

ناظر و سرپرست تیم: دکتر علیرضا وفایی صدر

IPMدرخواست دهنده: ستاد علوم شناختی

زمستان 1398

فهرست مطالب

|  |  |  |
| --- | --- | --- |
| 2 | ------------------------------------------------------------ | **مقدمه** |
| 2 | ------------------------------------------------------------ | **تیم داده کاوانِ فونیکس** |
| 2 | ------------------------------------------------------------ | **تعریف پروژه** |
| 2 | ------------------------------------------------------------ | **آشنایی با دیتاست و داده ها** |
| 3 | ------------------------------------------------------------ | **پیش پردازش و تمیز کردن داده** |
| 4 | ------------------------------------------------------------ | **آنالیز و بصری سازی داده** |
| 10 | ------------------------------------------------------------ | **کلاسه بندی کردن ستون هدف** |
| 13 | ------------------------------------------------------------ | **ساخت بردار اعداد از متون** |
| 14 | ------------------------------------------------------------ | **مدلسازی داده و یادگیری ماشین** |
| 18 | ------------------------------------------------------------ | **بصری سازی مدل** |
| 20 | ------------------------------------------------------------ | **کارها و تحقیقات آتی** |

مقدمه

پردازش متن یا NLP (Natural Language Processing) که به آن متن کاوی (Text Mining) هم گفته می شود به مجموعه عملیاتی (Processes) بر روی متون اطلاق می شود که متن را برای کامپیوتر قابل فهم کند. زیرا کامپیوتر و ماشین عدد را می فهمد و نه متن و زبان طبیعی را. بعد از پردازش متن می‌توان از قدرت کامیپوتر در محاسبات و [یادگیری ماشین](https://chistio.ir/%db%8c%d8%a7%d8%af%da%af%db%8c%d8%b1%db%8c-%d9%85%d8%a7%d8%b4%db%8c%d9%86-machine-learning-%da%86%db%8c%d8%b3%d8%aa%d8%9f/) استفاده کنیم تا اطلاعات ارزشمندی را از میان این داده‌ها استخراج کنیم. پردازش متن را می توان در زیرشاخه ی هوش مصنوعی و یا علم داده قرار داد.

تیم داده کاوانِ فونیکس

تیم ما، یعنی داده کاوانِ فونیکس در کارگاه آموزشی علم داده در پژوهشکده فیزیک پژوهشگاه دانش های بنیادین IPM برای انجام پروژه های یادگیری ماشین و علم داده با تدریس و هدایت دکتر علیرضا وفایی صدر تشکیل شد. ما یک تیم شش نفره هستیم که بر روی پروژه هایی از جمله تشخیص و پیش بینی عیوب توربین های گازی شرکت نفت و گاز توربوتک و پروژه متن کاوی داده های توییتر کار کرده ایم و همچنان در حال کار بر روی پروژه های مختلف هستیم.

تعریف پروژه

هدف از انجام این پروژه، انجام متن کاوی بر روی نظرات مختلف در مورد محصولات فروشگاه اینترنتی دیجیکا است. تحلیل نظرات کاربران بر اساس تعداد لایک ها و دیسلایک های موجود، مزایا و معایبی که در در هر نظر برای محصول بیان شده است.

آشنایی با دیتاست و داده ها

فایل دیتاستی که به تیم برای پردازش و تحلیل داده شد، شامل 100000 داده (نظرات در مورد محصولات دیجیکالا) بود دارای 12 ستون زیر:

شناسه محصول

عنوان محصول به فارسی

دسته بندی (عنوان) محصول به انگلیسی

شناسه کاربر

تعداد لایک های هر کامِنت

تعداد دیسلایک های هر کامنت

وضعیت تایید شدن کامنت

کاربر محصول را توصیه می کند یا نه (شامل recommended, not\_recommended, no\_idea)

عنوان کامنت

کامنت محصول

مزایای محصول

معایب محصول

قرار شد بر اساس تعداد لایک ها و دیسلایک های هر نظر، کار تحلیل نظرات انجام شود و همچنین یک دسته بندی برای تعداد لایک ها و دیسلایک ها انجام شود به طوری که در آینده پیش بینی شود یک نظر مشخص چه تعداد لایک و دیسلایک خواهد داشت.

پیش پردازش و تمیز کردن داده

برای پردازش متون (چه فارسی و چه انگلیسی) اولین قدم پیش پردازش متن و تمیز کردن متون است که به آن Text Preprocessing گفته می شود. زیرا نظرات و متونی که غیر ادبی و غیر کتابی باشند و در شبکه های اجتماعی توسط مردم عام نوشته شده باشد بسیار در هم و دارای نویز بوده زیرا هر شخص بر اساس سلیقه خود اقدام به نگارش متن می کند. پس باید تمام متون به حالتی یکپارچه و استاندارد دربیاید و کلمات اضافی و بی استفاده از آنها حذف شود.

برای همین با استفاده از کتابخانه هضم Hazm کار تمیز کردن و نرمال سازی متن را انجام دادیم به صورت زیر:

نرمالسازی متون و حذف فاصله ها و علایم و اعراب گذاری ها.

حذف کلمات پر تکرار و بی استفاده StopWords ها.

حذف شکلک ها، حذف لینک ها، حذف Tab ها. حذف ایمیل، حذف کلمات انگلیسی، حذف کاراکترها و سمبل های خاص مثل علامت سوال، علامت تعجب، کروشه و ...

حذف فاصله های اضافی بین کلمات و حذف فاصله های اضافی بین سطرها.

انجام عملیات lemmatization یا برگرداندن کلمات به اصل ریشه. مثلا تبدیل "رفتم" به "رو" و ...

این عملیات پاکسازی متون را برای ستون های متنی در دیتاست مثل ستون نظرات، ستون مزایا، ستون معایب و ستون عنوان محصول انجام دادیم.

همچنین در 4 ستون فوق اگر دارای مقدار خالی بودند، با متن پیش فرض "فیلد خالی" پر کردیم.

برای ستون توصیه ها (recommend) هم عملیات mapping را بر اساس ارزش هر مقدار انجام دادیم. بدین صورت که داده های خالی را در این ستون همان no\_idea یا نظری ندارم، در نظر گرفتیم و سپس بر اساس وزن هر مقدار، به مقدارِ "توصیه نمی کنم" عدد صفر، به مقدارِ "نظری ندارم" عدد 1 و به مقدارِ "توصیه می کنم" عدد 2 را اختصاص دادیم.

در نهایت تصمیم گرفتیم که سطرهایی که حاوی لایک و دیسلایک همزمان صفر هستند را حذف کنیم چون اطلاعاتی را برای پیش بینی به ما نخواهند داد و ارزش چندانی برای ما ندارند.

کارهایی که می توان برای پیش پردازش متن انجام داد بسیارند و زمان زیادی خواهد برد و به علت کمبود وقت ما به همین روش ها بسنده کردیم.

عملیات پیش پردازش متن در فایل Text Preprocessing.ipynb موجود است.

دیتاست پاکسازی شده را در فایلی به اسم digi\_clean2.xlsx قرار دادیم. عملیات پیش پردازش داده و پاکسازی متون توسط *آرمیتا رضوی* انجام شد.

آنالیز داده و بصری سازی داده

برای اینکه درک کاملی از داده ها داشته باشیم اقدام به آنالیز و تصویر سازی داده ها کردیم تا دید بهتری از داده ها داشته باشیم. همچنین کاربر با نگاه کردن به نمودار و شکل ها راحت تر متوجه داده می شود تا اینکه بخواهد کدهای آماری را بخواند.

آنالیز داده روی داده های پاکسازی شده انجام شده و همچنین نظرات و سطرهایی که دارای لایک و دیسلایک همزمان صفر بوده اند و برایمان ارزشی نداشتند را در این آنالیز و تصویر سازی، حذف کرده ایم. که از 100 هزار سطر به مقدار 73469 سطر رسیدیم. 26531 کامنت، همزمان بدون لایک و دیسلایک بودند.

در این بخش گزارش به مهمترین آنالیزها می پردازیم:

⁕ بیشترین نظرات مربوط به محصولات AC و HW و کمترین مربوط به GC و FF است. (شکل 1)

شکل 1

⁕ از بین 73469 کامنت، حدود 12000 نظر حاوی کلید واژه ی "توصیه نمی کنم" است. حدود 34000 نظر حاوی کلید واژه ی "نظری ندارم" است و حدود 25000 نظر حاوی کلید واژه ی "توصیه می کنم" است. (شکل2)

0 : توصیه نمی کنم – 1 : نظری ندارم – 2 : توصیه می کنم

شکل 2

مقادیر فیلد توصیه بر اساس هر دسته بندی محصولات را در شکل3 مشاهده می کنید:

شکل3

برای مثال در خرید محصولات AC حدود 20% توصیه نشده است، 40% نظری نداشته اند و 40% هم توصیه کرده اند. جالب است که در دسته محصولات GC کلا هیچ توصیه ای نشده است و همگی گفته اند که نظری ندارند!

همچنین در این آنالیز، برای بصری سازی داده ها تعداد لایک ها و دیسلایک ها را به سه کلاس تقسیم کردیم:

لایک ها و دیسلایک های زیر 75 تا. لایک ها و دیسلایک های بین 75 تا 150 تا و لایک ها و دیسلایک های بالای 150 تا که آماری بدین شکل به دست آمد:

تعداد کامنت های زیر 75 تا لایک: 73343 عدد - تعداد کامنت های بین 75 و 150 لایک: 102 عدد – تعداد کامنت های بالای 150 تا لایک: 24 عدد

تعداد کامنت های زیر 75 تا دیسلایک: 73379 عدد - تعداد کامنت های بین 75 و 150 دیسلایک: 59 عدد – تعداد کامنت های بالای 150 تا دیسلایک: 31 عدد

که تقریبا تفاوت فاحشی را بین لایک و دیسلایک نشان نمی دهد.

همچنین تعداد لایک ها و دیسلایک ها را برای هر دسته بندی محصول روی نمودار بردیم: (شکل4 و شکل 5)

شکل4

همانطور که مشاهده می شود اکثر محصولات لایک های زیر 75 تا دارند و محصولات AC بیشترین کامنت های حاوی لایک و محصولات دسته بندی GC, FF, BC کمترین کامنت های حاوی لایک ها را دارند و همگی زیر 75 تا لایک هستند.

شکل5

در نمودار دیسلایک ها هم تفاوت چندانی با نمودار لایک ها دیده نمی شود که با توجه به نزدیک بودن تعداد لایک ها و دیسلایک ها امری طبیعی است. این آنالیزها توسط *آرمیتا رضوی* انجام شد.

**کلمات ابری یا Word Cloud:**

سعی کردیم درکی از تعداد کلمات رایج و پر تکرار در کامنت ها و کلیدواژه های مزایا و معایب هر محصولات پیدا کنیم. برای همین اقدام به تولید word cloud برای کلمات پر تکرار در نظرات محصولات کردیم.

در ابتدا 60 کلمه پر تکرار در کل نظرات و کامنت ها راتصویر سازی کردیم: (شکل 6)

شکل 6

در کل کامنت ها و از بین 60 کلمه پر تکرار، کلمات "البته"، "جنس"، "گرفتم"، "هیچ" و "رنگ" بیشتر تکرار شده است.

سپس سراغ 30 کلمات پر تکرار که در معایب کالا ذکر شده است و بیشتر از 150 دیسلایک خورده اند رفتیم و کلمات ابری را برای این نظرات رسم کردیم: (شکل 7)

شکل7

در کل معایب کالاها و از بین 30 کلمه پر تکرار، کلمات "ندارد"، "کمی"، "عدم"، "ضعیف" و "پایین" بیشتر تکرار شده است که برای کلمات منفی در ستون معایب کالا، طبیعی است.

سپس سراغ 30 کلمات پر تکرار که در مزایای کالا ذکر شده است و بیشتر از 150 لایک خورده اند رفتیم و کلمات ابری را برای این نظرات رسم کردیم: (شکل 8)

شکل 8

در کلِ مزایای کالاها و از بین 30 کلمه پر تکرار، کلمات "بوی"، "مناسب"، "شیک"، "سریع" و "قوت" بیشتر تکرار شده است که برای کلمات مثبت در ستون مزایای کالا، دور از انتظار نیست.

همچنین کلمات ابری و پرتکرار را برای کامنت ها بر اساس دسته بندی هر محصول و تعداد لایک ها و دیسلایک هایشان در آوردیم. و همچنین کلمات پر تکرار را برای نظراتی که در رابطه با چندین محصول خاص وجود داشته است هم در آنالیز آورده شده است که که توسط خانم ها *آرمیتا رضوی* و *زهرا نفریه* انجام شده که همه این نمودار ها و آنالیزها را می توانید در فایلهایی با اسمهای Text Exploration، مشاهده کنید. به علاوه، تمامی عکس های word cloud ها در پوشه word clouds تحویل داده خواهد شد.

تولید کلمات ابری بر اساس دسته بندی کالا توسط خانم زهرا نفریه:

کار بررسی کلمات پر تکرار در کامنت ها، مزایا و معایب کالا بر اساس هر دسته بندی کالا انجام شد که نتایج به صورت زیر است:

نمونه ای از دسته بندی لایک ها و دیسلایک ها که در ابتدای آنالیز داده ها بیان شد:

|  |  |  |
| --- | --- | --- |
| معیار طبقه بندی | طبقه بندی Like و Dislike |  |
| Like = 150 dislike= -75 | High Like & Low dislike | 1 |
| Like = -75  Dislike= between | Low Like & Medium dislike | 2 |
| Like= between  Dislike = -75 | Medium Like & Low dislike | 3 |

دسته بندی محصولات:

[‘IT’ ‘AC’ ‘HW’ ‘MO’ ‘PC’ ‘PA’ ‘TC’ ‘TS’ ‘MA’ ‘HA’ ‘AV’ ‘FA’ ‘HC’ ‘BC’ ‘DF’ ‘GC’ ‘GF’ ‘FF’]

همچنین مجموع تعداد لایک ها و دیسلایک ها را برای کامنت ها بر طبق هر دسته بندی به دست آوردند:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | دسته بندی محصولات | ستون نظرات یا کامنت ها | |  |
| مجموع لایک | مجموع دیسلایک | طبقه بندی لایک و دیسلایک |
| 1 | IT | 56 | 56 | Low Like & Low dislike |
| 2 | AC | 12 | 12 | Low Like & Low dislike |
| 3 | HW | 0 | 0 | No Like & No dislike |
| 4 | MO | 102 | 102 | Medium Like & Medium dislike |
| 5 | PC | 39 | 39 | Low Like & Low dislike |
| 6 | PA | 10 | 10 | Low Like & Low dislike |
| 7 | TC | 0 | 0 | No Like & No dislike |
| 8 | TS | 0 | 0 | No Like & No dislike |
| 9 | MA | 19 | 19 | Low Like & Low dislike |
| 10 | HA | 0 | 0 | No Like & No dislike |
| 11 | AV | 46 | 46 | Low Like & Low dislike |
| 12 | FA | 29 | 29 | Low Like & Low dislike |
| 13 | HC | 0 | 0 | No Like & No dislike |
| 14 | BC | 0 | 0 | No Like & No dislike |
| 15 | DF | 0 | 0 | No Like & No dislike |
| 16 | GC | 0 | 0 | No Like & No dislike |
| 17 | GF | 0 | 0 | No Like & No dislike |
| 18 | FF | 0 | 0 | No Like & No dislike |

کلاسه بندی کردن ستون هدف

هدف اصلی این پروژه تعیین میزان لایک ها و دیسلایک ها در هر کامنت و پیش بینی بر اساس مدل های یادگیری ماشینی است که در آینده با توجه به هر کامنت بتوان میزان لایک ها و دیسلایک های آن را تشخیص داد.

از آنجا که مقادیر لایک ها و دیسلایک ها مقادیری پیوسته بودند تصمیم گرفتیم آنها را به مقادیر گسسته دسته بندی و کلاسه بندی کنیم. چهار کلاس را در نظر گرفتیم: کلاس لایک ها، کلاس خیلی لایک ها، کلاس دیسلایک ها و کلاس خیلی دیسلایک ها.

در ابتدا یک دید آماری از لایک ها و دیسلایک ها به دست آوردیم که در زیر مشاهده می کنید: (شکل 9)

شکل 9

شکل 9

همینطور نمودار توزیع داده را برای این دو ستون رسم کردیم: (شکل10)

شکل 10

متاسفانه توزیع داده های لایک و دیسلایک اصلا توزیع نرمالی نبود و برای کلاسه بندی کردن داده ها طوری که تعداد داده های یکسان در هر کلاس قرار گیرد و توازن بین کلاس ها برقرار شود مجبور شدیم از روش هایی غیری از مد و میانه و چارک ها استفاده کنیم.

*روش اول کلاسه بندی ستون هدف:*

یک روش که توسط آقای *وحید غفوریان* پیشنهاد شد بدین صورت بود که کلاس بندی لایک ها و دیسلایک ها را بر اساس عدد 4 در نظر بگیریم به صورت زیر که مشاهده می کنید: (شکل 11)

شکل 11

در این نوع کلاس بندی بر اساس تعداد لایک ها و دیسلایک های کمتر و بیشتر از 4، پنج نوع کلاس برای ستون هدف تعیین شد که البته لایک ها و دیسلایک های همزمان صفر هم به عنوان کلاس صفر، در نظر گرفته شده است.

از آنجا که تعداد داده ها در 5 کلاس، متوازن نبود، از عملیات تکرار داده ها یا up-sampling استفاده شد و تعداد داده ها در هر کلاس به 26000 داده رسید تا توازن بین داده های کلاس ستون هدف برقرار شود.

همین نوع کلاس بندی هم دوباره انجام شد ولی بدون کلاس صفر و کلاسی که لایک و دیسلایک همزمان صفر داشت را در نظر نگرفتیم و ستون هدف را 4 کلاسه کردیم.

*روش دوم کلاسه بندی ستون هدف:*

در روش دیگری که خانم *رقیه فرجی* ارائه و پیشنهاد دادند بدین صورت لایک ها و دیسلایک ها را طبقه بندی کردیم که در ابتدا داده های پرت و یا outlier که خارج از بازه سه انحراف معیار STD بودند مشخص گردید و کنار گذاشته شد. با انجام اینکار آستانه جداسازی را میانگین در نظرگرفتیم و در نتیجه توازن و تقارن داده ها حول میانگین بهتر شد و کلاس بندی لایک ها و دیسلایک ها را روی مرزبندی جدید انجام دادیم. کلاس حاوی تعداد لایک هایی کمتر از میانگین، کلاس بزرگتر مساوی میانگینِ لایک ها و به همین ترتیب دو کلاس برای دیسلایک ها. جمعا چهار کلاس برای ستون هدف به دست آمد. از داده های پرت بعدا برای داده های تست و آموزشی در مدل، استفاده کردیم.

برای اینکه تعداد داده های هر کلاس برابر شود از re-sampling استفاده شد و تعداد سطرها در هر کلاس به 30000 رسید. همچنین داده هایی که تعداد لایک ها و دیسلایک هایشان همزمان صفر بود را حذف کردیم و در کلاس بندی نیاوردیم.

*روش سوم کلاسه بندی ستون هدف*

در کلاسه بندی دیگری که توسط خانم فرجی پیشنهاد شد این بود که داده هایی که لایک های آنها صفر ولی دیسلایک آنها بیشتر از صفر هستند را به عنوان just-dislike درنظر بگیریم و برعکس داده هایی که دیسلایک های آنها صفر ولی لایک آنها بیشتر از صفر هستند را به عنوان just-like درنظر بگیریم. با این روش، تعداد کلاس های ستون هدف شش کلاسه شد.

ساخت بردار اعداد از متون

بعد از اینکه کار کلاسه بندی ستون های لایک و دیسلایک انجام شد تصمیم گرفتیم داده های را به الگوریتم های یادگیری ماشین و همچنین به شبکه عصبی بدهیم تا بتوان در آینده پیش بینی کرد که هر کامنت محصول در دیجیکالا در چه کلاسی از لایک و دیسلایک قرار می گیرد.

از آنجا که تمامی الگوریتم ها و مدلهای یادگیری ماشین، ورودی را به صورت اعداد می گیرند و درکی از متن ندارند باید متون را به بردارهایی از اعداد تبدیل کرده و به هر کلمه وزن و ارزشی را اختصاص دهیم که به اینکار وکتورایز vectorize کردن متن گفته می شود.

روش های مختلفی برای اینکار وجود دارد که ما از دو روش TF-IDF برای کار با مدل Naive Bayes و همچنین از روش Tokenizer برای آماده سازی داده ها برای کار با شبکه های عصبی در مدل های deep learning استفاده کردیم.

ابتدا داده ها را به داده های train و validation تقسیم کردیم به نسبت 70 به 30.

ساخت بردار متن برای ستون کامنت ها با TF-IDF : (شکل 12)

شکل 12

ساخت بردار متن برای ستون کامنت ها با Tokenizer : (شکل 13)

شکل 13

مدلسازی داده و یادگیری ماشین

برای کار مدلسازی، از دو مدل استفاده کردیم، مدل MultinomialNB که همان مدلی از Naive Bayes است از کتابخانه sk-learn در پایتون، و دیگری شبکه های عصبی کانولوشنی یا CNN از مدلهای deep learning .

مدلسازی انجام شده توسط خانم رقیه فرجی

در ابتدا دیتاستی که دارای شش کلاس متوازن برای ستون هدف بود را به مدل MultinomialNB دادیم که دقتی معادل 42% داد. سپس دیتاست چهار کلاسه نامتوازن را که در فایل unbalance\_classified.csv قرار دارد به مدل MultinomialNB داده شد که دقتی معادل 40% گرفتیم. مدلی از MultinomialNB در زیر دیده می شود: (شکل 14)

شکل 14

کانفیوژن ماتریکس مدل MultinomialNB برای چهار کلاسه متوازن: (شکل 15)

شکل 15

هچنین کانفیوژن ماتریس چهار کلاسه برای داده های نامتوازن هم در عکس های پروژه موجود است.

سپس خانم فرجی دیتاست چهار کلاسه متوازن خود را که در فایل balance\_classified.csv قرار دارد به مدل شبکه عصبی داد و شبکه را آموزش داد که دقتی معادل 75% برای چهار کلاسه به دست آورد. معماری شبکه عصبی را در شکل زیر مشاهده می کنید: (شکل 16)

شکل 16

خلاصه مدلسازی خانم فرجی در جدول زیر دیده می شود: (شکل 17)

|  |  |  |  |
| --- | --- | --- | --- |
| **دقت پیش بینی مدل** | **مدل** | **متوازن یا نامتوازن** | **تعداد کلاس بندی** |
| 50% | Naive Bayes | متوازن | 4 |
| 43% | Naive Bayes | نامتوازن | 4 |
| 75% | CNN | متوازن | 4 |
| 50% | CNN | نامتوازن | 4 |
| 42% | Naive Bayes | متوازن | 6 |

شکل 17

تمامی مدلهای این بخش در فایل final\_model2.ipynb قرار دارد.

مدلسازی انجام شده توسط آقای وحید غفوریان

همانطور که گفتیم آقای غفوریان دو نوع کلاسه بندی متوازن برای ستون هدف تولید کرد. 4 کلاسه و 5 کلاسه. دیتاست چهار کلاسه در فایل digi\_VahidEditation\_balanced\_4class.csv و دیتاست پنج کلاسه در فایل digi\_VahidEditation\_balanced\_5class.csv قرار دارد.

هر دو دیتاست روی مدل شبکه عصبی دیگری آموزش دیده شد و برای 5 کلاسه دقت 62% و برای 4 کلاسه دقت 74% به دست آورد. (فایل final\_model3.ipynb)

نمونه ای از مدل شبکه عصبی انجام شده در زیر دیده می شود: (شکل 18)

شکل 18

مدلسازی انجام شده توسط آرمیتا رضوی

در ابتدا لایک های بزرگتر مساوی 1 را یک کلاس در نظر گرفتیم و زیر 1 را کلاسی دیگر.

سپس دیسلایک های بزرگتر مساوی 2 را یک کلاس و زیر 2 را کلاسی دیگر در نظر گرفتیم و ستون هدف را چهار کلاسه کردیم. ولی تعداد داده های کلاس را متوازن نکردیم و برای تست به مدل دادیم که برای مدل MultinomialNB دقت 40% و برای شبکه عصبی دقت 30% داد. امری طبیعی بود چون کلاس ها اصلا متوازن نبودند و مرزبندی هم مرزبندی متقارنی نبود.

سپس دیتاست چهار کلاسه متوازن خانم فرجی (classified\_3.csv) را بر روی مدل شبکه عصبی آقای غفوریان اجرا کرده و دقتی معادل 69% گرفته شد.

نتیجه کلی مدلسازی به این صورت است که بهترین حالت، 4 کلاسه کردن ستون هدف و به صورت داده های متوازن می باشد که بالاترین دقت 75% را داده است.

فایل این مدلسازی به اسم final\_model.ipynb است که مدل 69% در آن قرار دارد.

بصری سازی مدل

یک نمونه از معماری مدل شبکه عصبی که روی چهار کلاسه متوازن دقت 75% داد را با tensor-board بصری سازی کردیم: (شکل 19)

شکل 19

متاسفانه وضوح عکس کیفیت خوبی ندارد ولی می توانید عکس مورد نیاز را در فولدر images مشاهده کنید.

همچنین مدل شبکه عصبی ای که دقت 69% داد را انتخاب کرده و وزنها و کلمات موجود در لایه ی Embedding آن را در دو فایل جدا به اسم های vecs.tsv و meta.tsv ذخیره کردیم و در سایت projector.tensorflow.org فایل ها را آپلود و نمایی سه بعدی از ماتریس بردار کلمات به دست آوردیم. در این بصری سازی می توان دید که کدام کلمات به هم نزدیکتر هستند و ارتباط بیشتری با هم دارند و همچنین می توان کلمات همسایه هر کلمه را پیدا کرد. این روش برای کلاستر بندی کلمات بسیار مفید است. در واقع این بصری سازی همان بصری سازی word2vec می باشد. این دو فایل در فایل های پروژه موجود است و می توانید با مراجعه به سایتی که ذکر شد و آپلود فایل ها تصویری کامل و جامع از ماتریس کلمات داشته باشید: (شکل 20)

شکل 20

کارها و تحقیقات آتی

از آنجا که تیم داده کاوان فونیکس فقط شش رو برای پردازش متن این دیتاست زمان داشت، ما تمام تلاش خود را کردیم که بهترین نتیجه را در این زمان کوتاه تحویل دهیم. دیدی کلی از داده به دست آوردیم، داده ها را پیش پردازش کرده و پاکسازی کردیم. تعداد کلمات موجود در هر کامنت را به دست آوردیم. یکسری آنالیز روی داده ها انجام داده و داده ها را تصویر سازی کرده و روی نمودار بردیم. کلمات پرتکرار را برای یکسری از کامنت ها بر اساس تعداد لایک ها و دیسلایک ها به دست آورده و word cloud آنها را کشیدیم. همچنین word cloud های مختلفی از کامنت ها و متون مزایا و معایب بر اساس دسته بندی کالاها و یکسری کالاهای خاص کشیدیم و کلمات مثبت و منفی را درآوردیم که همگی در پوشه عکس ها موجود است.

به این نتیجه رسیدیم که بیشتر کاربران روی کامنت ها برای توصیه کردن خرید محصولات، نظری خاصی ندارند و بیشترین کامنت ها دارای لایک و دیسلایک زیر 75 عدد هستند.

همچنین انواع مختلفی برای کلاسه بندی کردن میزان لایک ها و دیسلایک ها در نظر گرفتیم و برای هر نوع مرزبندی و کلاسه بندی، سعی کردیم داده ها را متوازن کنیم و روی هر کدام مدلسازی انجام دادیم.

که در نهایت به دیتاست 4 کلاسه و متوازن رسیدیم که دقتی معادل 75% روی شبکه عصبی به ما داد.

این فایل مستند به همراه تمامی کدها و عکس های پروژه در آدرس github گروه فونیکس به آدرس زیر قرار دارد که می توانید با مراجعه به این آدرس، به فایل های پروژه دسترسی داشته باشید:

https://github.com/phoenix-dataminers/Digikala

کارهای بسیار زیادی می توان در آینده روی این دیتاست انجام داد. می توان عملیات پیش پردازش و feature engineering بهتر و بیشتری انجام داد. می توان ارتباط و شباهت بین هر کامنت را درآورد و همچنین کلمات کلیدی را از بین کامنت ها برای محصولات استخراج کرد.

می توان کار دسته بندی کامنت ها را یا همان کلاستربندی را انجام داد که بتوان کامنت ها را در سه دسته مثبت، منفی و خنثی برای sentiment analyze قرار داد و کارهای بسیاری که در صورت داشتن زمان و تمایل می شود انجام داد.

همچنین می توان با صرف وقت بیشتری و دادن کلمات مزایا و معایب کالا علاوه بر نظرات، دقت پیش بینی مدل را افزایش داد.

در آخر اعلام می کنیم که این کار نتیجه تلاش گروهی و همفکری جمعی اعضای تیم داده کاوان فونیکس بوده و همه به طور یکسان با همدلی سعی کردند پروژه را به بهترین نحو به سرانجام برسانند.

همچنین لازم می دانیم از سرکار خانم لیلا نوربالا که داده ها را در اختیار ما قرار دادند و همچنین جناب آقای دکتر علیرضا وفایی صدر که به عنوان مربی به تیم ما اعتماد کرده و این فرصت را به ما دادند و همچنین از ستاد علوم شناختی IPM نهایت تشکر را به عمل بیاوریم.

پایان