به نام خدا

پروژه‌ی نهایی

درس هوش مصنوعی

شکیبا بلبلیان خواه - ۸۱۰۱۹۶۴۲۶

درباره

مسائل regression به دسته مسائلی گفته می‌شود که مدل باید بر اساس تعدادی از ویژگی‌ها، یک مقدار پیوسته را پیش‌بینی کند. بر این اساس و بر اساس نوع مدل انتخاب شده پیش‌پردازش‌هایی بر روی داده باید صورت بگیرد و روش‌هایی نیز برای ارزیابی مدل انتخاب شود که در ادامه در رابطه با آن صحبت خواهد شد.

پیش پردازش داده‌ها

از مهم‌ترین بخش‌های هر پروژه‌ی آموزش ماشین، پیش پردازش داده و یافتن feature های وابسته به متغیر هدف می‌باشد. (در واقع اگر یک ویژگی هیچ وابستگی به متغیر هدف نداشته باشد، در فرایند آموزش دادن مدل کمک کننده نخواهد بود و تاثیری نخواهد گذاشت.) بر این اساس پیش‌پردازش‌های متعددی روی دیتاست داده‌شده پیاده‌سازی شد، که در ادامه به آن پرداخته می‌شود. قابل ذکر است که در مدل‌های یادگیری ماشین، ما به دنبال ویژگی‌های عددی هستیم و داده‌های categorical یا text سودی برای ما نخواهد داشت. از سویی دیگر در این پروژه، هر سه نوع داده‌ی عددی، categorical و free text وجود دارد که با توجه به حجم داده‌های متنی و توضیحات موجود در آن، پیش پردازش دیتاست از اهمیت بالاتری برخوردار است.

نکته: از آن‌جایی که پیش پردازش‌های اولیه را هر سه نوع داده‌ی train، test و predict نیاز داشتند، فازهای ابتدایی بر روی کل دیتاست پیاده سازی شد و در مراحل بعدی داده‌ها تفکیک شدند.هم‌چنین محاسبه‌ی وابستگی میان داده‌ها پس از آماده‌سازی آن‌ها انجام گرفت

تغییر مقیاس داده‌ها (Rescaling)

به صورت کلی تغییر مقیاس برای داده‌هایی که با فاصله کار می‌کنند مانند KNN لازم می‌باشد اما از آنجایی که در این پروژه از Linear Regression استفاده می‌کنیم، اعمال rescaling ضرورت خاصی ندارد. هم‌چنین تنها ستونی که در ادامه‌ی پیش پردازش با مقادیر عددی مختلف باقی خواهد ماند(یعنی مقادیر سطرهای آن باینری نمی‌باشد) ستون image_count است که در محاسبه میزان وابستگی، به دست می‌آید که وابستگی بسیار کمی بین داده‌های آن و قیمت موبایل‌ها وجود دارد.

- **حذف کردن داده‌های پرت (Deleting Outliers)**

در هر دیتاستی، امکان آن وجود دارد که متغیر مقداری نامعمول نسبت به کلیت مقادیر دیگر داشته باشد. برای حذف outlierها دو راه پیاده سازی شد. در روش اول، از z-score با ۳ threshold استفاده شد. در این روش به صورت خودکار انحراف و میانگین داده‌ها محاسبه شده و مقادیری که اختلاف آن‌ها از حالت معمول بیش از مقدار threshold باشد، حذف خواهند شد.

در روش دوم ابتدا دیتاست بررسی شد. بر اساس داده‌ها به نظر می‌رسید به صورت کلی گوشی‌ها با برند نوکیا از سایر برندها کم‌قیمت تر هستند. پس سطرهایی که برند آن‌ها نوکیا بود با min مقدار کمتر (۳۰۰۰۰ تومان) و سایر برندها با min مقدار کمی بیشتر (۴۰۰۰۰ تومان) و سقف مقدار ۵۰۰۰۰۰۰ تومان انتخاب شدند. روش دوم به صورت کلی بهتر عمل کرد که دلیل‌های متعددی از جمله نحوه‌ی پیش پردازش متن‌ها می‌تواند داشته باشد.

- **پردازش داده‌های موضوعی (Label Encoding & One-Hot Encoding)**

در regression به صورت کلی از آنجایی که امتیازی برای متغیرهای عددی قائل می‌شود، اگر متغیرهای موضوعی به صورت label encoding پردازش شوند، امکان دارد مدل خوب آموزش نبیند. بر این اساس دو ستون شهر و برند نیاز به پردازش به صورت one-hot encoding داشتند که این تغییر بر روی آن‌ها اعمال شد.

- **پردازش داده‌های متنی**

پردازش داده‌های متنی در این دیتاست با مسائل گوناگونی مواجه است. در وهله‌ی اول، داده‌های بسیاری دارای غلط املایی‌های مختلف می‌باشند. هم‌چنین متون فارسی به صورت محاوره نوشته شده‌اند و در بعضی موارد برای تاکید برخی حروف تکرار شده‌اند. مهم‌ترین داده‌هایی که از ستون‌های title و desc می‌توان به دست آورد، مدل گوشی و وضعیت آن است که با وجود مسائل کنونی کار را دشوار کرده است. برای مثال برای موبایل samsung galaxy s5 نوشتارهای متنوعی چون سامسونگ گلکسی اس۵، سامسونگ galaxy اس5، سامسونگ galaxy s5، سامسونگ galaxy اس۵ و... وجود دارد که عملا باعث سرشکن شدن آمار داده‌ها می‌شود. برای بهتر کردن این مشکل، یک استراتژی پیاده شد که در ادامه به آن می‌پردازیم.

  - **پیش پردازش کلمات:**

در ابتدا می‌دانیم روش‌های معمولی برای پردازش کلمات وجود دارد که آن‌ها را پیاده‌سازی می‌کنیم. کتابخانه‌ی هضم یک نوع تابع نرمالایز را پشتیبانی می‌کند که در آن فاصله‌ی میان پیشوند و پسوندها را درست می‌کند (affix_spacing). از این نرمالایزر به منظور بهبود فواصل استفاده

می‌کنیم. هضم توابع دیگری مانند نرمالایزر برای ادبیات غیر رسمی را هم پشتیبانی می‌کند، اما با اعمال آن‌ها نتیجه‌ی مطلوبی حاصل نشد.
در گام بعد متن‌های فارسی را با استفاده از tokenizer کتابخانه‌ی هضم، به کلمات می‌شکنیم. هم‌چنین پس از بهبود متون فارسی، برندها را اصلاح کرده و فقط بخش انگلیسی آن را نگه می‌داریم(برای زیباتر بودن زمان one hot encoding). از آن‌جایی که پس از بهبود متون فارسی، فقط کلماتی انگلیسی در اختیار داریم، متد stem کتابخانه‌ی nltk را به منظور قراردادن ریشه‌ی کلمات مشابه هم برای بهبود داده‌ها بر روی دو ستون title و desc اعمال می‌کنیم.

- **بهبود متون فارسی:**

ایده‌ی این روش بر اساس همان galaxy s5 توضیح داده شده در ابتدای این بخش، پیاده شد. بر این اساس، از مجموع کلمات به دست آمده ابتدا آن‌هایی را انتخاب می‌کنیم که یا ترکیبی از اعداد و حروف انگلیسی و اعداد فارسی‌اند. در صورتیکه عدد فارسی وجود داشته باشد، می‌توانند حروف فارسی هم در خود داشته باشند. حال با استفاده از کتابخانه‌ی unidecode اعداد و حروف فارسی موجود در کلمات انتخاب شده را به انگلیسی تبدیل می‌کنیم. واضح است که این کتابخانه آنقدر دقیق عمل نمی‌کند اما تاثیر خوبی روی مدل‌های برای مثال تبدیل اس۵ به s5 دارد. بدین ترتیب در نهایت مجموعه‌ای از کلمات انگلیسی خواهیم داشت که در مراحل بعدی تعدادی از آن‌ها انتخاب خواهد شد.

- **ساختن Bag of Words:**

توجه: این قسمت پس از تقسیم بندی داده‌ها و جدا کردن سطرها با مقدار قیمت ۱- و هم‌چنین تقسیم باقی سطرها به دو دسته‌ی test و train با نسبت ۰.۲ و ۰.۸ پیاده‌سازی می‌شود و فقط روی دسته‌ی train انجام می‌شود. ویژگی‌های دو دسته‌ی test و دسته‌ای که قیمت ۱- دارند بر اساس همین bag of words ساخته شده در این قسمت محاسبه می‌شود.
از روش‌های معمول در مسائل regression ساختن bag of word می‌باشد. البته که روش‌های متعددی برای لحاظ کردن تاثیر کلمات (مانند تعداد تکرار، حضور یا عدم حضور، شاخص tf-idf) وجود دارد که در قسمت پایانی به آن‌ها خواهیم پرداخت. در این پروژه و به علت نبود زمان کافی، روش حضور/عدم حضور را انتخاب کرده‌ایم. ابتدا یک دیکشنری از کلمات موجود در ستون‌های title و desc (که آن‌ها را با هم ادغام کرده و در ستون text part) نوشته‌ایم می‌سازیم. سپس از دیکشنری کلیدهایی را که کمتر از ۱۵۰ بار آمده‌اند (کلمات نادر) و بیشتر از ۲۵۰۰ بار تکرار (کلمات عادی و غالبا مشترک) و هم‌چنین آن‌هایی که فقط مقدار عددی دارند (چرا که در رابطه با مقدار عددی معلوم نیست که متعلق به قیمت است، به مدل موبایل است، به اطلاعات داخلی

موبایل ربط دارد و...) را حذف می‌کنیم.حال ستون text part تمامی سطرهای داده‌ی train را بررسی می‌کنیم. برای هر سطر در صورتی که هر یک از کلمات bag of word در آن وجود داشت، به ازای آن کلمه‌ی موجود در bag در ستون متناظرش در دیتافریم ۱ قرارداده و اگر وجود نداشت، صفر قرار می‌دهیم. این کار را برای داده‌های train و آن‌هایی که باید بعدا predict شوند نیز با همین bag of word فعلی در آینده انجام خواهیم داد.

- **پردازش تاریخ**

یکی دیگر از ستون‌هایی که داده‌های آن به صورت خام بی‌ارزش هستند، تاریخ ایجاد آگهی می‌باشد. ساعت تاثیر چندانی در قیمت موبایل ندارد. روز هفته هم همینطور است (در واقع تاریخ دقیق روز ایجاد می‌تواند کمک کننده باشد چرا که به نوعی نشان‌دهنده‌ی وضعیت نوسانات بازار در آن روز ها بوده اما روز هفته و ساعت کمک چندانی نیست. با این حال، برای نشان دادن، از روز هفته ستون is weekend را می‌سازیم که نشان می‌دهد آیا زمان ایجاد در آخر هفته بوده یا خیر. در بخش وابستگی‌ها مشاهده خواهد شد تاثیر این ستون بسیار ناچیز است و بر همین اساس آن را حذف می‌کنیم.

- **محاسبه‌ی وابستگی میان ویژگی‌ها**

برای محاسبه‌ی وابستگی بین ستون‌های داده از pearson استفاده می‌کنیم. این ضریب به صورت خلاصه اعلام می‌کند که قدرت رابطه‌ی خطی میان دو متغیر چه قدر است. محاسبه وابستگی میان برند و قیمت و شهر و قیمت بدین صورت کار خیلی درستی نیست اما به صورت حدودی می‌توان گفت برای برند از آنجایی که مقدار وابستگی منفی است، برندی مانند اپل قیمت بالاتری نسبت به نوکیا و ZTE خواهد داشت که البته دلایلی چون تعداد نمونه‌های موجود از هر برند نیز اثرگذار است. در این بین مشاهده می‌شود که آخر هفته بودن یا نبودن و هم‌چنین تعداد تصاویر وابستگی خیلی کمی با قیمت هر کالا دارند و بنابراین از داده‌ها حذف می‌شوند.

- **جداسازی داده‌های آموزش و تست و پیش‌بینی**

همانطور که بیان شد پیش از ساختن bag of word نیاز داریم تا داده‌ها تقسیم شوند. ابتدا سطرهایی با قیمت ۱- را جدا می‌کنیم تا بعدا بر اساس مدل، قیمت آن‌ها را پیش‌بینی کنیم. سپس از داده‌های باقی‌مانده با استفاده از train_test_split دو دهم از داده‌ها را به تست و ۰.۸ را به train اختصاص می‌دهیم.

ساختن مدل، Train و Test آن و پیش‌بینی مقادیر خواسته شده

انتخاب مدل

به صورت کلی مدل‌های غالب در مسائل regression، مدل‌هایی چون Linear Regression، KNN، درخت تصمیم و شبکه‌های عصبی می‌باشد. شبکه‌های عصبی به علت زمان‌بر بودن پیاده‌سازیشان، در این پروژه استفاده نشدند. اما هر سه مدل رگرشن خطی، knn و جنگل تصادفی(به جای درخت تصمیم به علت آنکه مانع از overfitting می‌شود) امتحان شد و در میان آن‌ها linear regression بهترین نتیجه را داد. علت دقیق این اتفاق را نمی‌شود کامل بیان کرد چرا که به نوع پیش‌پردازش داده‌ها بسیار بستگی دارد. اگر معیارهای دیگری برای پیش‌پردازش انتخاب می‌شد، امکان داشت نتیجه خلاف حالت موجود باشد.

اما از مزایای linear regression می‌توان به محاسبه‌ی سریع آن اشاره کرد و مدل به دست آمده از آن نسبت به جنگل تصادفی ساده‌تر بوده و از پیچیدگی کمتری برخوردار است.

آموزش و تست مدل موجود

برای این کار کافیست، ستون price را به عنوان متغیر هدف و باقی دیتافریم را که هم‌اکنون همه‌ی آن‌های به داده‌های عددی تبدیل شده‌اند را با تابع fit به مدل موردنظر می‌دهیم. پس از با تابع predict متغیر هدف را بر اساس ورودی‌ها به دست می‌آوریم و بدین ترتیب وارد فاز ارزیابی مدل می‌شویم.

معیارهای ارزیابی (Evaluation)

روش‌های ارزیابی متعددی برای تشخیص میزان کارایی مدل در حالتی که مقادیر پیوسته را تخمین می‌زنیم وجود دارد، که در این پروژه به سه‌تا از آن‌ها پرداخته شد. در این پروژه سعی شده ضمن بالابردن مقدار R Square، از میزان MSE و MAE کاسته شود.

- **MSE:**

یا همان mean squared error به صورت تقریبی معروف‌ترین متریک ارزیابی مدل می‌باشد. این معیار میانگین مربع خطا میان مقادیر پیش بینی شده و مقادیر واقعی را می‌یابد. با توجه به آنکه قیمت پایه‌ی ما در این دیتاست در مقیاس ۱۰۰۰ تومان می‌باشد، شاید دیدن چنین مقدار بزرگی برای MSE در نتیجه‌ی نهایی دور از ذهن نباشد.

- **MAE:**

یا همان mean absolute error که تفاوتش با mse در این است که میانگین قدر مطلق اختلاف دو مقدار واقعی و پیش‌بینی شده را محاسبه می‌کند.

- **R Squared Value:**

که در خروجی با عنوان score نشان داده شده است در واقع معیاری برای نشان دادن میزان خوب بودن فیت شدن مدل بر روی داده است. البته که متغیرهای متفاوتی برای ارزیابی فیت شد وجود دارد. به صورت کلی هرچه مقدار آن بالاتر باشد بهتر است اما کم بودن آن لزوما به این معنی نیست که مدل به خوبی نمی‌تواند عمل کند.

- **پیش‌بینی موارد خواسته شده**

در نهایت با ساخته شدن و آموزش دیدن مدل نهایی، اطلاعات مربوط به سطرهایی که باید پیش‌بینی شوند را به آن می‌دهیم و مقادیر پیش بینی شده با index متناظرشان در فایل output.csv در پوشه‌ی بارگذاری شده ذخیره شده‌اند.

ایده‌هایی برای بهبود مدل

- **هوشمند سازی متن فارسی و روش‌های مختلف Bag of Words**

در سایت دیوار اجناس دست دوم قرار دارند و این بدین معنی است که طیف وسیعی از قیمت را به خاطر دلایل مختلف خرابی،آکبند بودن کالا، خراب بودن شارژر و ... و هم‌چنین قیمت‌گذاری اختیاری و دلخواه شاهد هستیم. از طرفی متن توضیحات و عنوان به نوعی تبلیغی محسوب می‌شود و صادقانه نیستند که کامل بتوان به آن اکتفا کرد. از این لحاظ در وهله‌ی اول بهتر است روش بهتری برای پالایش کردن متون فارسی داشته باشیم. یکی از این روش‌ها فراهم آوردن دایره‌ی لغات مرتبط به هم و قرار گرفته در یک دسته‌ی خاص است. این بدین معنی است که بیرون کشیدن مدل موبایل از مجموعه داده‌های فارسی و انگلیسی با غلط‌های املایی ابتدا با دقت بیشتری انجام می‌شود (که نیاز به بهبود متن داریم). در گام دوم، برای مثال توضیحی مثل شارژر خراب، گلس شکسته و ... بار منفی برای کالا به همراه دارند و از قیمت آن می‌کاهند. پس باید متون پردازش شوند و این قبیل کلمه‌ها (حتی مجموعه‌ای متوسط) از داده‌های train به دست آوریم و بر اساس آن به متن امتیاز دهیم.

یکی دیگر از مسائل مهم، سیاست پیاده‌سازی شده در bag of word است. از آنجایی که بعضی پیام ها بعضی کلمات را به صورت اغراق شده بیان کرده بودند (مثلا: نو نوو نووی نووو) حذف کلمات تکراری از متن راحت نبود و از طرفی در نظر گرفتن آن‌ها نیز بار منفی برای مدل به همراه داشت و به دلیل نبود زمان کافی فقط به حالت باینری اکتفا شد. در صورتیکه با پالایش پیام‌های متنی (مطابق پاراگراف قبل) و استفاده از شاخصی مانند tf-idf نتیجه‌ی بهتری می‌توانیم بگیریم. این شاخص میزان اهمیت هر کلمه را در هر متن بیان می‌کند. در صورتیکه کلمه مقدار شاخص بالایی داشته باشد یعنی اطلاعات به خصوص‌تری با خود

دارد (البته که کلمات نادر را باید از این دست حذف کرد) و هرچه کمتر باشد یعنی معمول‌تر است. در این راستا تلاش کردم تا از این شاخص استفاده کنم اما به دلیل وجود کلمات متعدد و پالایش نشده، نمی‌توانستم مجموعه‌ای منطقی و مناسب را به عنوان کلمات نهایی دیتاست انتخاب کنم (بهبود متون فارسی بعد از تلاش برای شاخص tf-idf انجام شد.) به همین دلیل سیاست خود را عوض کردم. با این حال قسمت‌های مربوط به آن را (با وجود درهم ریختگی!) در فایل Failed TF-IDF در پوشه‌ی پروژه قرار داده‌ام.

- **مدل استفاده شده**

همانطور که پیش‌تر بیان شد؛ از آنجایی که خروجی جنگل تصادفی پیچیده‌تر و دور تر از سادگی موجود در رگرشن خطیست، با داشتن داده‌های مناسب به نظر می‌رسد انتخاب بهتر و معقول‌تری باشد. هم‌چنین در بعضی مقالات به اهمیت KNN در تخمین مقادیر پیوسته اشاره شده بود.

- **آموزش و تقسیم داده‌های train و test بر اساس برند موبایل به صورت جداگانه**

از دیگر کارهای کمک کننده، آموزش دادن مدل بر حسب برندهای مختلف است. می‌دانیم که برند موبایل عامل تاثیرگذاری بر روی قیمت آن است(برای مثال در همین دیتاست تلفن‌ها با برند apple قیمت بالاتری از سایرین داشتند) بر همین اساس ابتدا باید بر اساس توزیع برندهای مختلف، آن‌ها را به دسته‌های train و test تقسیم کنیم، سپس برای هر برند به صورت جداگانه یک bag of words داشته باشیم که ارزش آن را تشخیص دهد. بدین ترتیب می‌توانیم مدل بهینه‌تری داشته باشیم.