



**دانشگاه آزاد اسلامي**

**واحدتهران جنوب**

**گروه فنی و مهندسی**

**«گزارش نهايي پروژه درس یادگیری ماشین»**

**استاد:**

**جناب آقای دکتر صیدی**

**دانشجو:**

**نیلوفر سلگی**

رشته مهندسی کامپیوتر- گرایش هوش مصنوعی

دی ماه 1397

**فصل اول:**

**بیان مساله**

در این مساله هدف، یافتن بهترین مدل(با استفاده از روش های فراگرفته در درس یادگیری ماشین) برای پیش بینی ستون rank در جدول StudentTest می باشد.

با توجه به صورت مساله، دو جدول که شامل StudentTest و train می باشد را در اختیار داریم.

در StudentTest ، ستون rank وجود ندارد و پس از حل مساله بایستی این ستون برای جدول پیش بینی شود.

در ابتدا به صورت مختصر، feature ها(یعنی همان ستون های موجود در جداول)توضیح داده می شود:

Favs:

این ستون، نمایش دهنده تعداد مرتبه های افزودن توییت به لیست علاقه مندی افراد می باشد.

RTs:

این ستون، نمایش دهنده تعداد مرتبه های مجدد توییت شدن یک توییت، می باشد.

Listed:

این ستون نمایش دهنده تعداد افرادی است که با این توییت، فرد را به لیست اضافه نموده اند.

Following:

این ستون بیانگر تعداد اکانت هایی می باشد که یک فرد دنبال می کند.

Followers:

این ستون نشان دهنده تعداد دنبال کنندگان یک اکانت در توییتر می باشد.

Likes:

این ستون بیانگر تعداد like های یک توییت می باشد.

Tweets:

این ستون نمایش دهنده تعداد توییت های منتشر شده از یک اکانت می باشد.

Reply:

این ستون تعداد جواب های برگردانده شده به توییت را شامل می شود.

Rank:

این ستون در واقع target ما در مساله می باشد. چرا که در نهایت بایستی این ستون را برای جدول StudentTest پیش بینی کنیم. این ستون بیانگر رتبه بندی ما برای اکانت های مختلف و توییت های مختلف می باشد.

**فصل دوم:**

**روش ها**

با توجه به بخش پیش پردازش مساله، ابتدا داده ها را در data frame مورد استفاده فراخوانی می کنیم.

تمامی موارد آورده شده در زیر را بر روی جدول train اعمال می کنیم.

داده هایی را که بعضی از feature ها به ازای آن ها مقدار nan دارند برای جلوگیری از بروز خطا، با استفاده از متد replace()‬ با صفر جایگزین می کنیم.

در جدول train یک ستون اضافه نموده و با استفاده از تابع remove punctuation کارکترهای غیر مجاز مانند (@و#و$و ...)را از ستون Tweet content حذف نموده و داده ها را در ستون جدید با نام clear\_‬TweetC وارد می کنیم.

از آنجایی که این ستون، یک ستون با داده های string می باشد و در حل مساله با روش های ذکر شده قابل استفاده نیست، بایستی از طریق word count کلمات آن را به عدد تبدیل کنیم. برای انجام این کار ابتدا با استفاده از stop word اقدام به حذف کلماتی که مهم نیستند مانند am,‬ is,‬ are,‬ this,‬ that و .. می نماییم.

سپس با استفاده از متد count اقدام به محاسبه تعداد کلمات موجود در هر سطر این ستون می نماییم و داده های محاسبه شده را در ستون جدیدی با نام totalwords وارد می نماییم.

ستون هایی که در حل مساله قابل اهمیت نیستند، مانند Tweet content,‬ User Name,‬ URLs,‬ clear\_‬TweetC ,‬ Tweet Id را از جدول train حذف می کنیم.

با استفاده از متد MinMaxScaler داده های موجود را هم رنج می کنیم تا محاسبات به درستی انجام شود.

این جدول حاصل شده را با استفاده از متد split()‬ به سه بخش train,‬ test,‬ validate تقسیم می کنیم.

با استفاده از متد corr()‬ اقدام به بررسی ارتباط feature ها با یکدیگر می نماییم.

تا از بروز هم پوشانی feature های بسیار مرتبط و خطا در محاسبات جلوگیری شود. چرا که ممکن است بدون در نظر گرفتن ارتباط این feature های بسیار به هم مرتبط، موجب حذف هر دو feature شویم در صورتی که این feature ها از اهمیت بسیار زیادی در حل مساله برخوردار باشند.

برای انجام مراحل زیر از داده های split شده train استفاده می کنیم.

جهت پاسخ به قسمت اول مساله با استفاده از روش simple linear regression تابع simple\_linear\_regression را با تعریف ورودی و خروجی متناسب جهت محاسبه شیب و عرض از مبدا ایجاد می کنیم.‬

سپس تابع get\_regression\_predictions را جهت محاسبه predicted\_output یا همان خروجی تخمین زده شده، تعریف می نماییم.

سپس جهت محاسبه خطا، تابع get\_residual\_sum\_of\_squares را ایجاد می کنیم.

در این روش چون از یک معادله خطی استفاده شده است، از توابع برای ارزیابی و انتخاب feature استفاده می کنیم. به این صورت که مدلی با feature یی مناسب تر می باشد، که دارای خطای کمتری باشد.

در قسمت بعدی مساله خواسته شده است با استفاده از multiple regression مدل ارجح را بیابیم.

با توجه به انجام و بررسی فیچرهای high correlated و همچنین هم رنج کردن فیچرها مراحل زیر را انجام می دهیم:

با استفاده از feature های قبلی، feature جدید ایجاد می کنیم. این feature ها ترکیباتی از ضرب یک feature در خودش، لگاریتم گرفتن از feature و مجموع دو feature می باشد.

مدل ها را پلات نموده و با استفاده از تابع get\_resedual\_sum\_of\_squares مدلی که خطای کمتری دارد را انتخاب می کنیم.

برای استفاده از lasso regression و ridge regression ابتدا بایستی feature های مهم را استخراج کنیم. برای انجام این کار چندین راه وجود دارد:

All subset

greedy

Regularize

روش اول بهترین راه می باشد اما در صورتی که تعداد feature ها بسیار زیاد باشد، order بسیار بالایی برابر 2^D+1 دارد و در نتیجه در بیشتر موارد به صرفه نمی باشد.

در روش greedy استفاده از دو الگوریتم forward stepwise و backward stepwise پیشنهاد شده است. در حل این مساله خواسته شده از این دو روش استفاده نماییم.

با تعریف تابع forward\_selected و پس از استخراج feature ها با روش های فوق اقدام به مدل نمودن داده ها و یافتن بهترین مدل می کنیم.

از طریق مدل کردن داده ها با استفاده از تعریف تابع polynomial\_dataframe() و متد sort ، اقدام به مدل نمودن داده و مرتب سازی آنها می کنیم. با استفاده از متد ridge و linear regression اقدام به fit نمودن داده ها با استفاده از داده های StudentTest می نماییم. با استفاده از مدل انتخاب شده و متد predict اقدام به پیش بینی ستون rank برای این جدول می نماییم.