محمدرضا کیانی‌فر
۸۱۰۱۹۴۳۸۸

گزارش کار

توضیح کلی داده‌ها و تعیین فیچر ها

در این پروژه ما داده‌ی نزدیک به ۵۰۰۰ ایمیل را داشتیم که حدود ۸۶٪ آن‌ها ایمیل‌های ham (سالم) و ۱۴٪ شان ایمیل‌های spam بودند. هر سطر از داده دارای دو ستون بود، متن آن و نوع آن ایمیل. این یک مسئله‌ی supervised learning بود یعنی مسئله‌ای که داده‌های ما لیبل دارند و باید از طریق تعیین فیچرهایی و یادگیری ماشین از روی آن فیچرها، برای دیتاهای جدید بتواند پیشبینی مناسبی (spam یا ham) بکند.

دو فیچر را من برای انجام پروژه در نظر گرفتم. یکی طول متن هر ایمیل و مقایسه‌ی آن نسبت به میانگین طول ایمیل‌های هر نوع. و دومی کلمات موجود در هر ایمیل و نسبت تکرار آن‌ها در ایمیل‌های از یک نوع نسبت به کل کلمات آن نوع ایمیل مثلا برای کلمه‌ی free، تکرار آن در ایمیل‌های اسپم نسبت به تعداد کل کلمات ایمیل‌های اسپم. این فیچر به اندازه‌ی کلماتی از یک ایمیل هست که معتبر باشند. (قبلا آن‌ها را بررسی کرده باشیم و حروف اضافه نباشند.)

توضیح کلیت الگوریتم پیاده‌سازی شده

ابتدا من با نسبت ۹۰ به ۱۰، ۵۰۰۰ داده‌ی ایمیل را تقسیم به دیتای train و test کردم. سپس با استفاده از دیتای train فیچرهایی که بالا معرفی شد را train کردم که به دو شکل در آمد:

۱. برای فیچر طول متن ایمیل‌ها، دو پارامتر به دست آمد که یکی میانگین طول ایمیل‌های ham بود و دیگری میانگین طول ایمیل‌های spam که این فیچر برای ایمیل‌های ham عدد ۷۰ کاراکتر شد و برای ایمیل‌های spam، حدود ۱۴۰ کاراکتر.

۲. برای تک تک کلمات مجاز موجود در متن ایمیل‌های train، دو عدد را به دست آوردم: یکی تعداد تکرار در ایمیل‌های ham و دیگری تعداد در ایمیل‌های spam. همچنین مجموع تعداد کلمات مجاز در ایمیل‌های ham و spam داده‌ی train را نیز محاسبه کردم. (برای ایمیل‌های ham حدود ۴۳۰۰۰ کلمه و برای ایمیل‌های spam حدود ۱۲۵۰۰ کلمه)

سپس برای ایمیل‌های test (و البته باز هم train) با استفاده از فیچرهای learn شده، پیشبینی خود را به روش زیر حساب کردم که برگرفته از روش Naive Bayesian بود. پایه‌ی آن بررسی رابطه‌ی زیر به ازای دو مقدار زیر بود:

$$P(Spam\,|\,e_1, e_2, \ldots, e_n) \propto P(e_1\,|\,Spam)P(e_2\,|\,Spam)\ldots P(e_n\,|\,Spam)P(Spam)$$
$$P(Ham\,|\,e_1, e_2, \ldots, e_n) \propto P(e_1\,|\,Ham)P(e_2\,|\,Ham)\ldots P(e_n\,|\,Ham)P(Ham)$$

که در آن e_i، فیچر شماره‌ی i ام می‌باشد که یکی از دو نسبت طول متن نسبت به میانگین طول متن ایمیل‌های ham یا spam است و دیگری، نسبت تکرار یک کمله در ایمیل‌های یک نوع نسبت به تعداد کل کلمات در ایمیل‌های همان نوع است. با محاسبه‌ی دو مقدار فوق، می‌توان احتمال spam بودن و احتمال ham بودن یک ایمیل به شرط شواهد آن را بررسی کرد و هر کدام که بیشتر بود، پیشبینی من آن مورد خواهد بود.

P(spam) و یا P(ham) نیز، احتمال spam بودن یا ham بودن یک ایمیل است که از رابطه‌ی زیر محاسبه شده:

$$P(Spam) = \frac{SpamEmailsCount}{AllEmailCount} \approx 0.14 \quad , \quad P(Ham) = \frac{HamEmailsCount}{AllEmailCount} \approx 0.86$$

توضیح overfit و راهکار تشخیص آن

Overfit نوعی خطای مدل‌سازی است و زمانی رخ می‌دهد که تابع پیشگو بسیار بر اساس داده‌ی train، learn شده باشد و برای آن پیشبینی دقیقی انجام دهد اما برای داده‌های جدید دقت پیشبینی بسیار افت کند. از آن جایی که در دنیای واقعی داده‌ها کمی خطا و نویز دارند، اگر مدل پیشگو کاملا منطبق بر آن باشد، خطاهای داده‌ها را نیز به عنوان داده‌ای درست یادگرفته و احتمال خطا روی داده‌های جدید بیشتر می‌شود.

راهکار قطعی‌ای برای تشخیص overfit وجود ندارد اما یکی از راهکار حدودی تشخیص رخ دادن overfit، از بررسی دقت یا خطای پیشبینی روی داده‌های train و داده‌ی test است. (داده‌ها ابتدا به دو بخش test و tarin شکسته می‌شوند، به کمک داده‌ی train فیچرها learn می‌شوند و سپس مسئله به کمک فیچرهای learn شده، داده‌ی test را پیشبینی می‌کنند.) یعنی اگر خطا در داده‌ی tain بسیار پایین بود و در داده‌ی test اختلاف فاحشی با دقت قبلی داشت احتمال رخ‌داد overfit بالاست. در واقع در این حالت اگر نمودار پیشبینی روی داده‌های train را رسم کنیم، میبینیم که مدل ما کاملا منطبق بر داده‌ی train است یعنی در یک چند جمله‌ای با بالاترین درجه‌ی ممکن آن را مدل کرده‌ایم.

بررسی وجود overfit در راه حل پیاده‌سازی شده

پس از انجام عملیات یادگیری روی داده‌های train، با فیچرهای learn شده یک بار داده‌های train و یک بار داده‌های test را پیشبینی کردم و نتایج نسبتا مشابهی به دست می‌آمد. برای مثال نتایج برای یک بار اجرا به شرح زیر است:

```
Report for Train Data
Recall: 0.9918166939443536
Precision: 0.9409937888198758
Accuracy: 0.9906215921483097
##############################
Report for Test Data
Recall: 0.9384615384615385
Precision: 0.8970588235294118
Accuracy: 0.9774127310061602
```

که در آن پارامترها تقریبا نزدیک هستند و اختلاف محسوسی بین دقت در داده‌ی test و در داده‌ی train مشاهده نمی‌شود. در نتیجه overfit رخ نداده است.

دقت نهایی پروژه بر اساس معیارهای ارزیابی خطا

در ابتدا من ویژگی طول پیام ایمیل را به عنوان یک فیچر در نظر نمی‌گرفتم و precision نسبتا پایین‌تری (٪۸۰) به دست می‌آوردم. با اضافه کردم فیچر طول ایمیل به precision تقریبا ۸۵٪ رسیدم و سپس با نرمال کردن بیشتر کلمات ایمیل به precison بالای ۹۰٪ رسیدم. در نرمال کردن کلمات من ابتدا تمام حروف را lowercase می‌کردم. سپس stop words ها مانند the، is و ... را از کلمات حذف می‌کردم و سپس بعد از تنها نگه‌داشتن کلماتی که حروف الفبا داشتند، نقطه‌گذاری‌ها (punctuations) را نیز حذف می‌کردم. بهینه‌سازی آخری که از آن نام‌بردم، حذف کردن مرحله‌ی نگه داشتن کلمات دارای حروف الفبا بود چرا که برخی اعداد با این کار حذف می‌شدند در صورتی که در تشخیص ham یا spam بودن ایمیل‌ها تاثیرگذار بودند.

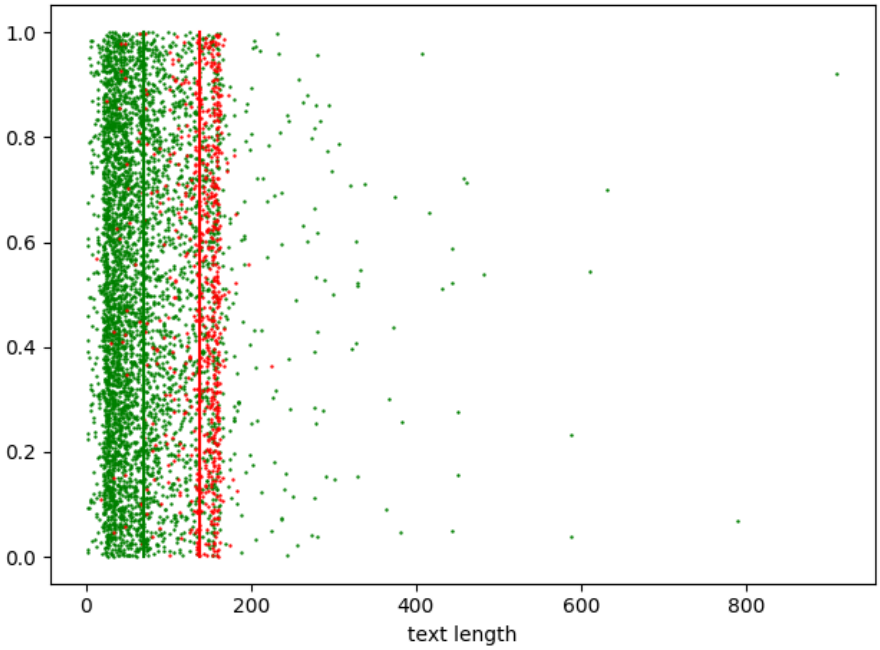
من با نوشتن یک اسکریپت بش، ۱۰ بار برنامه نهایی(شامل تمام بهینه‌سازی‌های گفته شده در بالا) را اجرا کردم و دقت این ده بار اجرا روی داده‌ی تست را میانگین گرفتم که نتیجه‌ی آن به شکل زیر است:

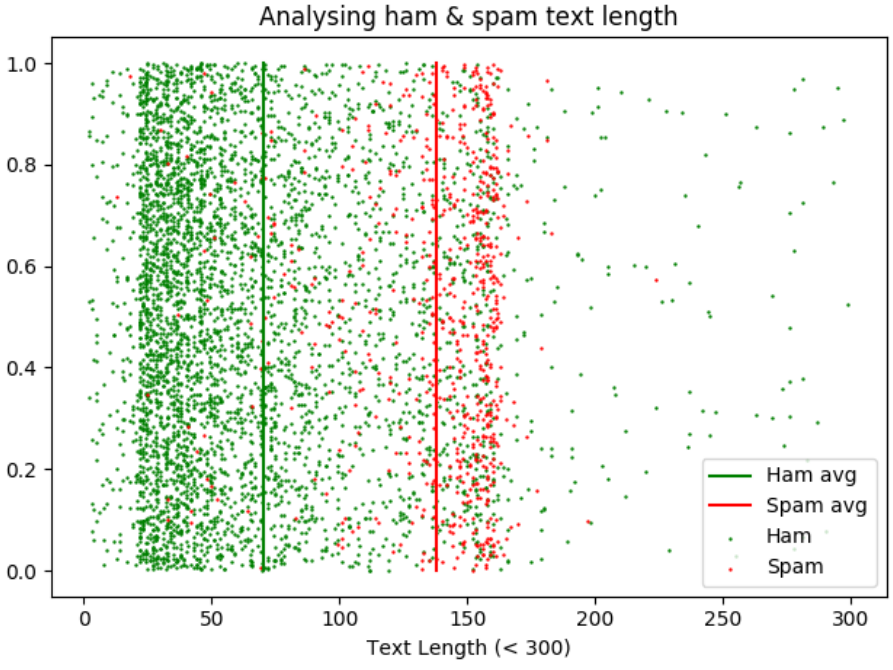
```
Report for Test Data
Recall: 0.9480903935
Precision: 0.9327821764
Accuracy: 0.9836570208
```

دو ویژگی بررسی شده و نمودار های آن ها

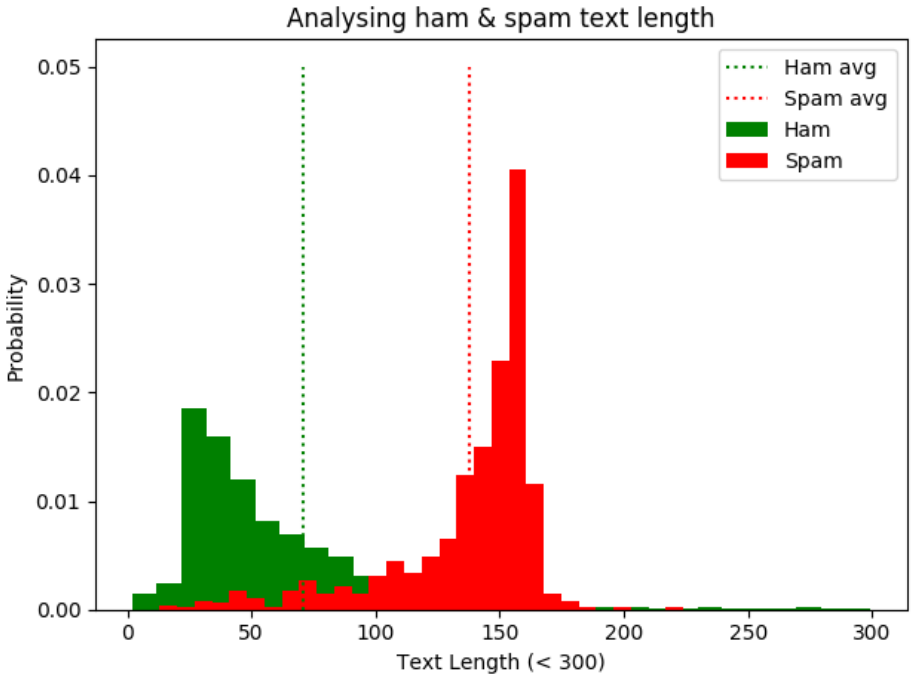
ویژگی طول پیام هر ایمیل

هر ایمیل به یک عدد مدل شد که طول آن را نشان می‌داد. میانگین طول ایمیل‌های ham و spam را نیز به دست آورده بودم که در دومین بخش از گزارش کار به آن اشاره شد. (میانیگن طول ایمیل‌های ham حدود ۷۰ کاراکتر و میانگین طول ایمیل‌های spam، حدود ۱۴۰ کاراکتر)





نمودار با نمایش نقطه‌ای به شکل زیر می‌شد: (برای خواناتر شدن نمودار و روی هم نیفتادن نقاط، یک عدد رندوم بین ۰ تا ۱ به

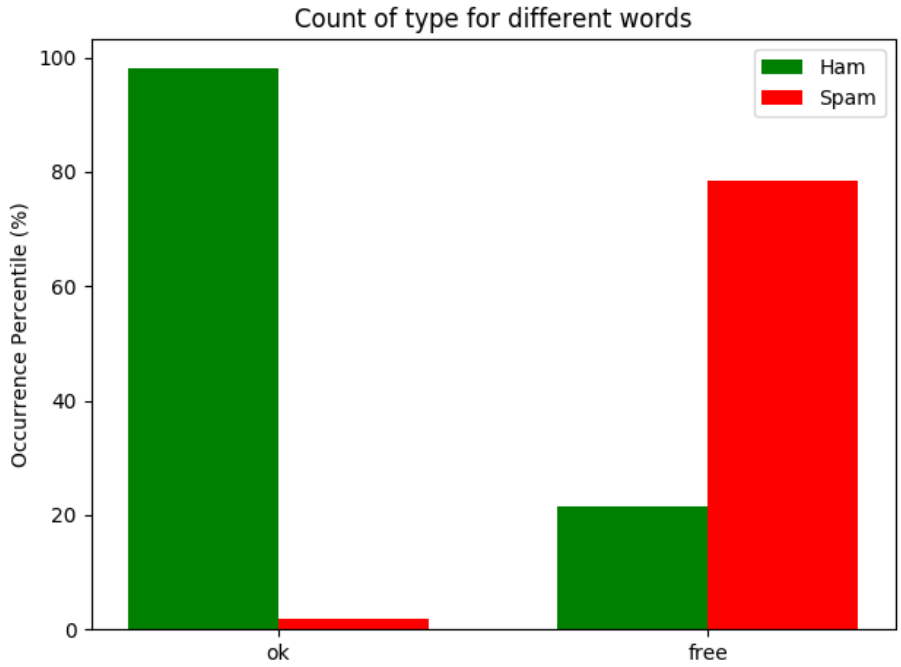


هر طول اختصاص دادم.)

سپس برای نمایش بهتر، ایمیل‌ها با طول بیش از ۳۰۰ را فیلتر کردم که نتیجه آن نمودار زیر شد:

در گام آخر برای درک بهتر خوشه‌بندی داده‌ها بر اساس این فیچر، نمودار هیستوگرام آن را کشیدم:

ویژگی تکرار یک کلمه در ایمیل‌های ham/spam نسبت به کل کلمات ham/spam



برای این موضوع من ابتدا کلمات تاثیرگذار در هر گروه را با یک مرتب‌سازی بر اساس تعداد تکرار پیدا کردم و دو تا از آن‌ها را انتخاب کردم: کلمه‌ی ok که کلمه‌ای تاثیرگذار در ایمیل های ham است و کلمه‌ی free که احتمال spam بودن بالایی دارد و تاثیرگذاری کلمه را در قالب نمودارهای زیر کشیدم.