گزارش فاز دوم پروژه بازیابی اطلاعات

اعضا گروه به ترتیب الفبا:

علیرضا دیزجی 96105745

آتنا ساقی 96105818

صبرینه مختاری 96110107

مقدمه

قبل از استفاده از الگوریتم‌ها, ابتدا تمامی لغات منحصربفرد از بررسی توکن‌های موجود در سه فایل prepared_english.csv از فاز قبل و prepared_train.csv و prepared_test.csv از فاز دوم (پس از انجام پیش‌پردازش‌های لازم), بدست آورده‌شده‌اند که در نهایت ۱۱۹۵۱ ترم متفاوت بدست آمد و تصمیم بر آن شد که پس از ترکیب ستون‌های title و description به یک ستون واحد در داده‌های آموزش و تست, هر سمپل به شکل یک‌بردار ۱۱۹۵۱تایی و به کمک روش tf-idf درآورده‌شوند.

در ادامه در بخش اول توضیحات مختصری در ارتباط با پیاده‌سازی الگوریتم‌ها, در بخش دوم الگوریتم انتخاب‌شده و روند پیش بینی برچسب داده فاز قبل و در بخش سوم خروجی الگوریتم‌ها نشان‌داده‌می‌شوند.

1. بخش اول

a. الگوریتم KNN

در بررسی این الگوریتم از روش cross validation استفاده شده‌است تا تمامی داده‌ها هم به‌عنوان "آموزش" و هم "اعتبار" مورد استفاده قرار گیرند و نتایج داده‌های اعتبار نشان‌می‌دهند که بهترین دقت به‌ازای k برابر با ۹ بدست می‌آید.

b. الگوریتم Naive Bayes

برای این الگوریتم ابتدا احتمال prior هر دسته محاسبه شد. پس از آن احتمال $p(tk|c)$ برای لغات مختلف در هر دسته محاسبه شد. پس از این که احتمال های فوق به دست آمد، مرحله پیش بینی

برچسب را شروع کردیم. در این بخش لغات داده ی تست (مربوط به 2 ستون title و description) استخراج شد و با استفاده از $arg\ max[log\ p(c) + \sum_{1\leq k\leq n} log\ p(tk|c)]$ دسته ی نهایی هر داکیومنت پیش بینی شد و نتایج به دست آمده من جمله accuracy، precision و ... بررسی شد و در جداول بخش 3 نیز قابل مشاهده می باشد.

c. الگوریتم SVM

در این بررسی با استفاده از کتابخانه Sklearn مدل svm را می سازیم . kernel مناسب و استفاده شده در این مدل rbf با مقدار gamma=0.001 می باشد. از بین C های خواسته شده بیشترین دقت مربوط به مقدار 1 می باشد.

d. الگوریتم Random Forest

در بررسی این الگوریتم نیز به دلیل مشابه در KNN از cross_validation استفاده‌شده‌است. اما جنگل تصادفی چیست؟ درخت تصمیم‌گیری یکی‌از الگوریتم‌های مورد استفاده در یادگیری ماشین است که هرچند بدون هیچ پارامتری می‌توان نتایج خوبی را در مسائل رگرسیون و دسته‌بندی گرفت, اما بسیار وابسته به داده ورودی است به‌طوری‌که اندکی تغییر در یکی از فیچرها می‌تواند ساختار درخت را به‌کلی عوض کند, در نتیجه معضل اورفیت کاملا مشهود است که برای رفع آن دو راهکار اساسی پیشنهاد شده‌است:

1. از چندین درخت تصمیم‌گیری برای همان مسئله استفاده‌شود و بسته به نوع مسئله خروجی نهایی با استفاده از میانگین یا اکثریت خروجی درخت‌ها مشخص شود.
2. در هر گره, لازم به بررسی هریک از فیچرهای باقی‌مانده برای اینکه کدام‌یک داده‌ها را بهتر جدا می‌کند نیست, بلکه به‌طور جایگزین این فیچرها به‌صورت تصادفی انتخاب می‌شوند.

این دو مورد باعث پیدایش جنگل‌های تصادفی شده‌اند که در کنار استفاده از مزایای درخت تصمیم‌گیری, معایب آن را تا حد خوبی می‌پوشانند.

2. بخش دوم

با توجه به نتایج گرفته شده از دسته بندی داده تست که در جداول بخش 3 مشخص است، بهترین الگوریتم با توجه به معیار accuracy، الگوریتم SVM بود. بنابراین از این الگوریتم برای پیش بینی برچسب داده فاز قبل با مقدار C=1 استفاده شد. پس از آن که برچسب با الگوریتم SVM مشخص شد داده به دو قسمت تقسیم شد و نتیجه در دو فایل جداگانه ذخیره شد. پس از این مرحله کاربر دسته ی مدنظر (که بر حسب برچسب داده شده به داده می باشد) را انتخاب می کند و با توجه به دسته ی انتخاب شده فایل داده مشخص می شود(همانطور که گفته شد داده فاز قبل بر اساس برچسب دریافتی در 2 فایل گوناگون ذخیره شده اند). سپس کوئری که درخواست می کند بررسی می شود و مشابه فاز قبل جست و جو انجام می شود.

3. بخش سوم

- **نتایج قسمت آموزش**

	acc	f1_pos	pr_pos	re_pos	f1_neg	pr_neg	re_neg
KNN	0.502	0.296	0.514	0.207	0.616	0.500	0.801
NB	0.958	0.959	0.941	0.978	0.957	0.977	0.938
SVM	0.684	0.690	0.734	0.652	0.678	0.639	0.723
RF	0.628	0.625	0.634	0.615	0.631	0.622	0.641

- **نتایج قسمت تست**

	acc	f1_pos	pr_pos	re_pos	f1_neg	pr_neg	re_neg

KNN	0.525	0.016	0.500	0.008	0.687	0.525	0.992
NB	0.664	0.674	0.625	0.732	0.653	0.713	0.603
SVM	**0.691**	0.629	0.731	0.552	0.735	0.668	0.816
RF	0.663	0.635	0.663	0.609	0.695	0.671	0.720