به نام خدا

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

تمرین سری سوم داده کاوی

توضیحات:

- پاسخ به تمرین‌ها باید به صورت انفرادی صورت گیرد و در صورت مشاهده هرگونه تقلب نمره صفر برای کل تمرین منظور خواهد شد.
- تمیزی و خوانایی گزارش تمرین از اهمیت بالایی برخوردار است.
- لطفا کدها و گزارش تمرین خود را در قالب یک فایل ZIP با نام **«HW3_StudentNumber.zip»** در سایت درس در مهلت معین بارگزاری نمایید.
- در صورت داشتن اشکال می‌توانید از طریق ایمیل **datamining.spring2021@gmail.com** با تدریس‌یار درس در ارتباط باشید.

نیم‌سال دوم ۰۰-۹۹

بخش اول: سوالات تئوری

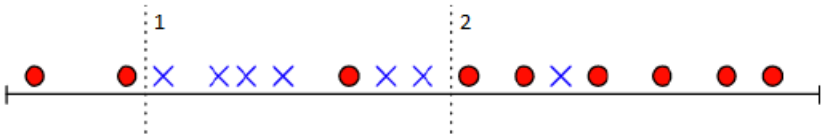
۱) فرض کنید یک نرم افزار فیلتر برای تشخیص اتوماتیک spam وجود دارد. این سیستم بر اساس حضور و یا عدم حضور بعضی کلمات در متن ایمیل، نوع آن را تشخیص می‌دهد. در زیر داده‌های آموزشی برای این نرم‌افزار را مشاهده می‌کنید:

'study'	'free'	'money'	Category
1	0	0	Regular
0	0	1	Regular
1	0	0	Regular
1	1	0	Regular
0	1	0	Spam
0	1	0	Spam
1	1	0	Spam
0	1	0	Spam
0	1	1	Spam
0	1	1	Spam
0	1	1	Spam
0	0	1	Spam

الف) اگر این نرم افزار مقدار $p(spam) = 0.1$ را در نظر بگیرد، توضیح دهید که آیا این کار منطقی است؟ چرا؟

ب) بر اساس قاعده بیز و $p(spam) = 0.1$ مشخص کنید که جمله "money for psychology study" در چه دسته‌ای قرار می‌گیرد.

2) در شکل زیر داده‌های دو کلاس بر روی یکی از متغیرهای پیوسته خود رسم شده‌اند .در صورتی که بخواهیم برای درخت تصمیم این ویژگی پیوسته را به یک ویژگی گسسته باینری تبدیل نماییم، کدام یک از نقاط 1 و یا 2 برای این کار مناسب‌تر هستند؟



3) جدول زیر شرایط مسابقه تنیس زمانی که رافائل نادال و راجر فدرر در آن برنده بوده‌اند را نشان می‌دهد.

الف) درخت تصمیم را برای آن رسم نمایید و در هر مرحله بهره اطلاعات برای هر ویژگی را محاسبه کنید. (F = برد راجر فدرر، N = برد رافائل نادال)

زمان	نوع مسابقه	زمین مسابقه	حداکثر قدرت	نتیجه
صبح	مستر	چمن	۱	F
بعدازظهر	گرند اسلم	شنی	۱	F
بعدازظهر	دوستانه	ترکیبی	۰	N
بعدازظهر	مستر	شنی	۱	N
بعدازظهر	گرند اسلم	چمن	۱	F
بعدظهر	گرند اسلم	سخت	۱	F
بعدازظهر	گرند اسلم	سخت	۱	F
صبح	مستر	چمن	۱	F
بعدازظهر	گرند اسلم	شنی	۱	N
شب	دوستانه	سخت	۰	F
شب	مستر	ترکیبی	۱	N
بعدازظهر	مستر	شنی	۱	N
بعدازظهر	مستر	چمن	۱	F
شب	گرند اسلم	سخت	۱	F
بعدازظهر	گرند اسلم	شنی	۱	F

ب) نمونه‌های زیر را به عنوان نمونه اعتبار سنجی در نظر بگیرید و خطای صحت درخت روی این نمونه‌ها را بدست آورید.

زمان	نوع مسابقه	زمین مسابقه	حداکثر قدرت	نتیجه
صبح	مستر	چمن	۱	F
بعدازظهر	گرند اسلم	شنی	۱	N
بعدازظهر	مستر	ترکیبی	۰	F
صبح	مستر	شنی	۱	N
شب	دوستانه	سخت	۰	F
شب	گرند اسلم	ترکیبی	۱	F

4) boosting چیست و چگونه باعث افزایش دقت می‌شود؟ یکی از روش هایی که از ایده boosting استفاده میکند، gradient boosting می‌باشد که برای حل مسئله رگرسیون مجموعه ای از درخت‌های تصمیم را نتیجه می‌دهد. در مورد این روش تحقیق کرده و در حد یک پاراگراف توضیح دهید.

5) دو روش هرس کردن درخت (post-pruning و pre-pruning) را با هم مقایسه کرده و مزایا و معایب هرکدام را ذکر کنید.

6) به طور کلی نرمال کردن داده‌ها به چه منظور صورت می‌گیرد؟

7) آیا ID3 تضمین می‌کند به جوابی برسد که globally optimum باشد؟ توضیح دهید.

8) نشان دهید accuracy تابعی از precision و recall است.

بخش دوم: سوالات عملی

۱) در این سوال قرار است تا الگوریتم KNN را پیاده سازی کرده و از آن برای کلاس بندی مجموعه segmentation استفاده کنیم. این مجموعه داده خلاصه ای از اطلاعات پیکسل های تصاویر (مثل میانگین میزان قرمز، سبز یا آبی بودن، غلظت رنگ و...) را در بر دارد که بر اساس آن بتوان تصاویر را بین ۷ دسته متفاوت مثل: چمن، پنجره، آسمان و... تقسیم بندی کرد. در این مجموعه داده، ستون اول **برچسب** هر داده است و پس از آن ۱۹ شاخصه برای هر نمونه داده ذکر شده است. ابتدا لازم است همه این داده ها را بخوانید، اما طبیعتا آموزش روی مجموعه داده segmentation.Train و تست روی segmentation.Test انجام می‌شود.

الگوریتم KNN پیچیدگی خاصی ندارد و فاز آموزش آن صرفا به خواندن دادهای آموزش بسنده می‌کند. به همین دلیل برای داده های با حجم بسیار زیاد مناسب نخواهد بود. اما همانطور که در این تمرین خواهیم دید جایی که داده آموزشی ما کوچک باشد، عملکرد مناسبی نشان خواهد داد.

در فاز پیشبینی، ابتدا پارامتر k توسط کاربر تعیین می‌شود. سپس به ازای هر نمونه که می‌خواهیم برچسبش را پیش‌بینی کنیم، ابتدا فاصله‌اش را با همه نقاط مجموعه آموزش حساب می‌کنیم، آنگاه k تا از نزدیک ترین نمونه های آموزش به آن را انتخاب کرده و برچسب هایشان را برسی می کنیم. برچسبی که بیشترین تکرار را داشته باشد، به عنوان خروجی پیش‌بینی اعلام می‌شود. (در صورتی هم که تعداد تکرار دو یا چند برچسب یکسان بود، یکی را تصادفی اعلام می کنیم) هر چند تعیین پارامتر k به کاربر وابسته است، اما عموما آن را عددی فرد در نظر می‌گیرند.

برای این بخش لازم است که شما تابعی بنویسید که با گرفتن ۴ پارامتر: مجموعه آموزش، مجموعه تست، k و نوع معیار فاصله (که یا اقلیدسی و یا کسینوسی است)، لیستی از خروجی های پیش‌بینی شده برای مجموعه تست را به ما برگرداند. پس از نوشتن این تابع، به ازای مجموعه های آموزش و تست داده شده، و به ازای kهای [1,...,8] و دو فاصله اقلیدسی و کسینوسی، خروجی هارا بدست آورده و دقت (درصد پیش‌بینی های صحیح) را محاسبه نمایید. در نهایت در یک نمودار، دقت را به ازای هر k نشان دهید. بهترین نتیجه به ازای کدام معیار فاصله و کدام k به دست می آید؟ دقت کنید که همیشه چنین نیست و در مسائل مختلف، معیار و kهای متفاوتی کاربرد دارند.

برای محسابه شباهت کسینوسی دو بردار هم‌اندازه، لازم است ضرب داخلی آنهارا تقسیم بر حاصل ضرب norm2 شان کنید. همچنین می‌توانید با یکه کردن همه بردارها، شباهت کسینوسی را صرفا با ضرب داخلی آنها محسابه کنید.

$$\text{CosineDist(A, B)} = \frac{A.B}{||A||\,||B||} = \frac{A}{||A||}.\frac{B}{||B||} \xrightarrow{if\ ||A||=||B||=1} = A.B$$

با استفاده از مقدار به دست آمده که میزان **شباهت** کسینوسی دو بردار را نشان می‌دهد می‌توان معیاری برای فاصله کسینوسی در نظر گرفت. مثلا معکوس شباهت یا منفی آن (هر دو صحیح هستند چرا که برعکس شباهت عمل می کنند)

۲) در این سوال می‌خواهیم الگوریتم درخت تصمیم را بر روی دیتاست car.data که در اختیارتان قرار گرفته شده است پیاده کنیم. این مجموعه داده اطلاعاتی در خصوص هر ماشین دارد که در فایل از چپ به راست به ترتیب: هزینه خرید ماشین، هزینه نگه‌داری ماشین، تعداد در های ماشین، ظرفیت افراد قابل حمل توسط ماشین، اندازه صندوق عقب آن و در نهایت میزان امنیت کلی ماشین. در آخرین ستون هم یک **برچسب** برای هر ماشین ارائه شده که به نحوی ارزیابی کلی ماشین است (غیرقابل‌قبول، قابل‌قبول، خوب، عالی). هدف ما دسته بندی بر اساس همین برچسب ها می‌باشد. این مجموعه داده را خوانده و پس از shuffle کردن آنها، با نسبت ۸۰ به ۲۰ به دو بخش آموزش و تست تقسیم نمایید.

دقت فرمایید که نیازی به پیاده سازی خود الگوریتم نیست و می‌توانید از کتابخانه SKlearn استفاده کنید. لینک زیر اطلاعات کافی را در اختیارتان قرار می‌دهد:

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

پارامتر های زیادی برای تغییر در این مسئله وجود دارد، حداقل ۳ عمق متفاوت را امتحان کرده و دقت دسته‌بندی هرکدام را گزارش کنید. به ازای بهترین عمق به دست آمده در این بخش، لازم است که خروجی خود را با معیار Confusion matrix ارزیابی کرده و خطای آموزش و تست را گزارش کنید. همچنین با ابزار های موجود بهترین درخت حاصل را رسم نمایید.

۳) یکی از مهم‌ترین کاربردهای Naïve Bays در مسائل Text Categorization است. در این بخش قصد داریم با استفاده از الگوریتم Naïve Bayes نظرات در مورد فیلم‌های IMDB موجود در فایل IMDB_review_labels.txt را به دو نوع منفی و مثبت کلاس‌بندی کنیم. دقت فرمایید که نیازی به پیاده سازی خود الگوریتم نیست و می‌توانید از کتابخانه SKlearn استفاده کنید. لینک زیر اطلاعات کافی در اختیارتان قرار می‌دهد:

https://scikit-learn.org/stable/modules/naive_bayes.html

همچنین، برای انجام این بخش تمرین، ضمن گزینه های موجود دیگر، می‌توانید با فرض گوسی بودن توزیع featureها از Gaussian Naïve Bayes استفاده نمایید.

ابتدا باید پیش‌پردازشهای لازم را بر روی دیتاست انجام دهید:

- حذف علامت‌های نوشتاری و Tokenizing
- حذف Stop word ها
- حذف تگ‌های HTML
- ریشه‌یابی و Stemming
- رفع حساسایت به حروف بزرگ و کوچک
- اعمال POS-Tagging
- و...

در مرحله بعد باید ماتریس tf-idf را تشکیل دهید. tf به معنای تعداد تکرار یک کلمه[1] در یک سند بوده و idf به معنای معکوس تعداد سندهایی[2] که یک کلمه در آن‌ها آمده است، می‌باشد. هرچند فرمول هایی با تفاوت های اندک برای این منظور وجود دارد، ما از فرمول زیر برای محاسبه tf-idf برای یک کلمه در یک سند خاص استفاده می‌کنیم:

$$w_{t,d} = \log(1+\mathrm{tf}_{t,d}) \times \log_{10}(N/\mathrm{df}_t)$$

[1] Term Frequency
[2] Inversed Document Frequency

اگر این مقدار را به ازای هر **متن** و **کلمه** محاسبه کنیم، یک جدول بدست می‌آید که به آن ماتریس tf-idf می‌گوییم. ابعاد این ماتریس برابر *تعداد کل کلمات غیر تکراری* × *تعداد کل متون* خواهد بود. برای کسب اطلاعات بیشتر در مورد شاخص tf-idf به این آدرس مراجعه کنید.

حال از ماتریس tf-idf به عنوان فیچرها یا ورودی الگوریتم Naïve Bayes (به این صورت که هر متن تبدیل به یک بردار شده) و از labelهای موجود در دیتاست به عنوان خروجی آن استفاده کرده و نهایتا خطای اموزش و تست را گزارش کنید.

۴) در بخش نهایی این تمرین قصد داریم تا با استفاده از SVM عملیات binary classification را انجام دهیم. مشابه قسمت قبل نیازی به پیاده سازی خود الگوریتم نیست و می‌توانید از کتابخانه SKlearn استفاده کنید. لینک زیر اطلاعات کافی در اختیارتان قرار می‌دهد:

https://scikit-learn.org/stable/modules/svm.html

پیشنهاد می‌شود برای این تمرین از SVC استفاده شود.

ابتدا می‌خواهیم مجموعه داده سوال قبل را این بار با استفاده از SVM طبقه بندی کنیم. پس لازم است بردارهای به دست آمده برای آموزش را به عنوان ورودی به مدل SVM بدهید و پس از آموزش آن خروجی را برای داده تست گرفته و دقت آن را محسابه کنید. در مقایسه با سوال ۳ که از بیز ساده استفاده کردیم، دقت کمتر شده یا بیشتر؟ به نظر شما دلیل آن چیست؟

در ادامه مجموعه داده binary_2d.txt را خوانده و آن را هم با نسبت ۹۰ به ۱۰ جدا کنید. دقت کنید که دو عدد اول در هر سطر، مختصات یک نقطه و عدد سوم برچسب آن است.

این بار مدل SVC را با kernel=linear ایجاد کرده و روی داده ها آموزش دهید. سپس خروجی را برای داده های تست محاسبه کرده و دقت را گزارش کنید. مدل SVM با استفاده از ضرایب svm.coef_ طبقه بندی را انجام می‌دهد. **نقاط آموزش و مرز به دست آمده** را با استفاده از این ضرایب رسم کنید.