بهره اطلاعاتی و آنتروپی

مسئله‌ی ۱.

هنگامی که یک درخت تصمیم می‌سازیم، در هر مرحله ویژگی با بیشترین بهره اطلاعاتی را انتخاب می‌کنیم. حال می‌خواهیم رابطه بهره اطلاعاتی را با مفهوم KL-divergence بررسی کنیم. معیار KL-divergence یک مفهوم مهم در شاخه تئوری اطلاعات است که می‌توان آن را یک معیار فاصله بین دو توزیع $p(x)$ و $q(x)$ دانست.

$$KL(p||q) = -\sum_x p(x) \log_2 \frac{q(x)}{p(x)}$$

- ثابت کنید $I(X,Y) \equiv KL(p(x,y)||p(x)p(y))$
- نشان دهید بهره اطلاعاتی متقارن است. $(I(X,Y) = I(Y,X))$
- در چه شرایطی داریم: $I(X,Y) = ۰$

مسئله‌ی ۲.

می‌دانیم که آنتروپی برای متغیر تصادفی X که گسسته باشد به صورت زیر تعریف می‌شود:

$$H(X) = -\sum_x p(x) \ln p(x)$$

اگر X پیوسته باشد، آنتروپی آن به چه صورت محاسبه می‌شود؟ آنتروپی $H(X)$ را برای متغیر تصادفی X که از توزیع نرمال با میانگین μ با واریناس $\sigma^۲$ پیروی می‌کند را به‌دست آورید.

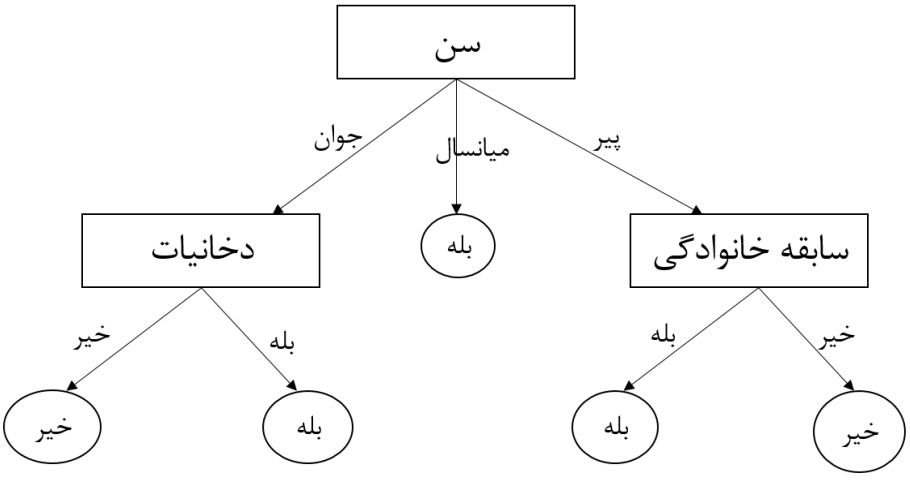
درخت تصمیم

مسئله‌ی ۳.

داده زیر را در نظر بگیرید.

سن	فعالیت بدنی	دخانیات	سابقه خانوادگی سرطان	سرطان
جوان	زیاد	خیر	بله	خیر
جوان	زیاد	خیر	خیر	خیر
میانسال	زیاد	خیر	بله	بله
پیر	متوسط	خیر	بله	بله
پیر	کم	بله	بله	بله
پیر	کم	بله	خیر	خیر
میانسال	کم	بله	خیر	بله
جوان	متوسط	خیر	بله	خیر
جوان	کم	بله	خیر	بله
پیر	متوسط	بله	بله	بله
جوان	متوسط	بله	خیر	بله
میانسال	متوسط	خیر	خیر	بله
میانسال	زیاد	بله	بله	بله
پیر	متوسط	خیر	خیر	خیر

درخت زیر برای آن پیشنهاد داده شده است.



اگر با الگوریتم ID3 بخواهیم درخت تصمیم را بسازیم باز هم درخت ما به همین شکل خواهد بود؟ درخت تصمیم را بسازید و توضیح دهید.

مسئله‌ی ۴.

به موارد زیر در مورد درخت تصمیم پاسخ دهید.

- یکی از مشکلات درخت تصمیم، بالابودن خطای واریانس آن است. توضیح دهید جنگل تصادفی چگونه این مشکل را حل می‌کند.
- آیا ساخت درخت تصمیم به طور حریصانه و یا با کمک گرفتن از معیار هایی همچون بهره اطلاعاتی، همیشه بهترین درخت را به ما می‌دهد؟ توضیح دهید.

مسئله‌ی ۵.

فرض کنید که n ویژگی دودویی داریم. به این صورت که $X = \langle X_1, ...X_n \rangle$ و $X_i \in \{0, 1\}$ و n بزرگتر از ۳ می‌باشد. تابعی که می‌خواهیم آن را یاد بگیریم $Y = X_1 \vee X_2 \vee X_3$ می‌باشد. فرض کنید که داده آموزش ما تمام 2^n حالت را دارا می‌باشد. حال برای $n = 4$ به سوالات زیر پاسخ دهید.

- چه تعداد اشتباه درخت تصمیم یک برگی انجام می‌دهد؟ (درخت تصمیم یک برگی حتی یک بار نیز داده را تقسیم نمی‌کند)
- آیا تقسیمی وجود دارد که تعداد اشتباهات را حداقل به اندازه یک کاهش دهد؟
- آنتروپی Y برای درخت تصمیم یک برگی را بدست آورید.
- آیا تقسیمی وجود دارد که آنتروپی Y را به مقدار ناصفری کاهش دهد؟ توضیح دهید.

kNN

مسئله‌ی ۶.

موارد زیر را توضیح دهید.

- در درخت هاي تصمیم گیري می‌بایست تمامی نمونه هاي آموزش در اختیار باشد تا درخت ساخته شود. بنابراین اگر تعدادي نمونه جدید به داده آموزش اضافه شود، می بایست درخت یاد گرفته شده را به روز رسانی کرد. آیا kNN هم این مساله را دارد؟ چرا؟
- در kNN در صورتی که نویز داده‌ها زیاد باشد، k را چگونه تغییر دهیم؟ چرا؟
- در kNN افزایش و کاهش k چه تاثیری بر بایاس و واریانس مدل دارد؟

بخش عملی

در این بخش به پیاده‌سازی دو الگوریتم درخت تصمیم و k نزدیکترین همسایه می‌پردازیم. داده‌ای که استفاده می‌کنیم در مورد بیماری قلبی می‌باشد که توضیحات بیشتر ویژگی‌های آمده در آن در کنار آن قرار داده شده‌است.
برای انجام این بخش باید ۳ اسکریپت پایتون به نام های utils و DecisionTree و kNN داشته باشید.

- یک تابع برای خواندن داده بنویسید.
  ورودی: آدرس داده و نام ستون هدف
  خروجی: دو داده X و Y

- یک تابع برای بر زدن [۱] داده بنویسید.
  ورودی: دو داده X و Y
  خروجی: دو داده بر خورده X و Y

- یک تابع برای تقسیم داده به دو داده آموزش و تست بنویسید. در این تابع حق استفاده از دو تابع split و array_split از کتابخانه numpy را ندارید.
  ورودی: داده و نسبت تقسیم
  خروجی: دو داده مورد نظر

- یک کلاس برای درخت تصمیم پیاده‌کنید.
  توجه کنید که درخت تصمیم پیاده شده توسط شما حداقل باید موارد زیر را دارا باشد:
  ـ پارامتر max_depth که مشخص کنند حداکثر عمق درخت می‌باشد.
  ـ پارامتر threshhold که به این صورت کار می‌کند: مثلا اگر مقدار آن ۰.۸ باشد، وقتی بیش از ۸۰٪ نمونه ها در یک نود درخت یک برچسب یکسان داشتند، به آن نود همان برچسب تعلق گیرد و دیگر گسترش داده نشود
  ـ تابع fit که دو داده X و Y را گرفته و مدل را می‌سازد.
  ـ تابع predict که برای داده گرفته شده پیش‌بینی می‌کند.

- یک کلاس برای knn پیاده‌کنید.
  توجه کنید که knn شده توسط شما حداقل باید موارد زیر را دارا باشد:
  پارامتر k برای مشخص کردن تعداد همسایه‌ها
  ـ تابع fit که دو داده X و Y را گرفته و مدل را می‌سازد.
  ـ تابع predict که برای داده گرفته شده پیش‌بینی می‌کند.

- یک تابع برای بررسی میزان دقت بنویسید.
  ورودی: مقادیر پیش‌بینی شده و مقادیر واقعی
  خروجی: دقت پیش‌بینی

- یک تابع برای محاسبه confusion matrix بنویسید.
  ورودی: مقادیر پیش‌بینی شده و مقادیر واقعی
  خروجی: TP-FP-TN-FN

[۱] Shuffle

- یک تابع برای محاسبه classification report بنویسید.
  ورودی: مقادیر پیش‌بینی شده و مقادیر واقعی
  خروجی: Accuracy, Precision, Recall, Specificity, f1Score

- یک تابع(یا کلاس) برای هرس درخت تصمیم با روش chi-square پیاده کنید.(امتیازی)
  تابع(یا کلاس) شما باید مدل درخت آموزش داده شده را دریافت و مدل هرس شده را برگرداند.

- تابع t-test را برای مقایسه دو مدل بنویسید.
  ورودی: مقادیر پیش‌بینی شده توسط دو مدل
  خروجی: نتایج آزمون فرض

توجه کنید تمامی بخش‌های بالا باید توسط خود شما پیاده‌سازی شود و حق استفاده از کتاب‌خانه های آماده (به جز کتاب‌خانه هایی همچون pandas و یا numpy و ... در موارد کلی) را ندارید.
حال در یک فایل Jupyter Notebook توسط توابعی که پیاده‌سازی کردید و با روند زیر می‌خواهیم چند مدل را آموزش داده و نتایج را ببینیم.

- ابتدا داده را بخوانید.
- حال داده را به دو قسمت test و train تقسیم کنید. (۸۰ درصد داده را برای آموزش در نظر بگیرید.)
- برای این داده، درخت تصمیم‌های متفاوت با عمق های متفاوت (از عمق ۱ تا عمق ۱۳) آموزش دهید. دقت مدل را براي داده هاي آموزش و تست به ازاي هر محدوديت عمق محاسبه کنيد. نمودار دقت روي داده هاي آموزش و تست را بر حسب محدوديت عمق درخت رسم کنيد. نمودار را توضیح دهید. درخت با چه عمقی مناسب‌تر می‌باشد؟ آیا می توان دقت دسته بند روي داده هاي آزمون را براي اين عمق به عنوان معيار نهايی عملکرد مدل گزارش کرد؟
- حال kNN های متفاوت را برای این داده آموزش دهید.(تعداد همسایه‌ها را از ۱ تا ۱۵ در نظر بگیرید.) به صورت قسمت قبل برای این مدل‌ها نیز عمل کنید.
- حال با استفاده از روش 5fold-cross validation بهترین مدل برای درخت تصمیم و kNN را با پارامترهای مناسب پیدا کنید.
  ** توجه کنید که حداقل دقت شما برای مدل نهایی درخت تصمیم ۶۵ درصد و برای kNN مقدار ۵۵ درصد باید باشد تا نمره کامل را دریافت کنید.**
- confusion matrix و classification report را برای دو مدل به دست آورید.
- درخت تصمیمی را که با عمق ۱۳ ساخته بودید توسط chi-square هرس کنید.(امتیازی)
- حال با استفاده از t-test ابتدا دو درخت تصمیم هرس شده و نشده را مقایسه کنید و نتایج را تفسیر کنید.(امتیازی)
- در نهایت مدل kNN و درخت تصمیم هرس نشده را مقایسه و نتایج را تفسیر کنید.

توجه کنید که فایل جوپیتر شما باید قابلیت اجرا دوباره را داشته باشد و در هنگام نمره‌دهی، این فایل دوباره اجرا می‌شود و بر اساس آن نمره می‌گیرید.

نکات مهم

- بخش تئوری را در قالب یک فایل pdf با اسم HW1_STD-Num آپلود کنید.
- هر یک از ۳ اسکرپیت گفته شده را به صورت جدا و همینطور هر ۴ فایل را در قالب یک فایل زیپ آپلود کنید.
- ددلاین تمرین ساعت ۲۳:۵۹ روز ۲۶ اسفند می‌باشد.