بسمه تعالی

تمرین اول

پیاده سازی درخت تصمیم برای دسته بندی و تقریب تابع

استاد درس: دکتر آرش عبدی هجراندوست

تدریس‌یار آموزشی: ایمان براتی

نکات تمرین :

- مهلت تحویل ۱۴۰۱/۱۲/۱۹ ساعت ۲۳:۵۵
- مهلت ارسال به هیچ وجه قابل تغییر نیست .
- مواردی که بعد از تاریخ فوق ارسال شوند قابل قبول نبوده و نمره ای نخواهند داشت .
- انجام تمرین تک نفره است. لطفا به تنهایی انجام شود، در غیر اینصورت نمره منفی در نظر گرفته خواهد شد .
- کل محتوای ارسالی زیپ شود و نام فایل زیپ ارسالی HW1_StudentNumber_YourName باشد .
- محتوای ارسالی دارای راهنما (read me) جهت تسهیل اجرا باشد .
- زبان برنامه نویسی دلخواه است. (پیشنهاد : پایتون)
- در صورت استفاده از زبان پایتون فایل کد ترجیحا به فرمت ipynb. بوده و **فایل کد حتما به صورت اجرا شده آپلود گردد و از وجود خروجی سلول ها اطمینان حاصل نمایید .**
- موارد ارسال شده در تاریخی که بعدا مشخص خواهد شد و متعاقبا اعلام می‌گردد به صورت آنلاین نیز تحویل گرفته خواهند شد (صرفا آنچه در LMS طبق تاریخ فوق تحویل داده شده است بعدا به صورت حضوری تست شده و توضیح داده می‌شود).
- تنها تکالیفی که به LMS و قبل از مهلت ارسال، فرستاده می شوند بررسی خواهند شد .
- در صورت داشتن هر گونه سوال می‌توانید سوال خود را در گروه تلگرامی درس مطرح کنید .
- حداقل **یک ساعت قبل از مهلت ارسال** را احتیاط هرف قرار دهید، تا مشکلات غیرقابل پیش بینی مانند موارد زیر باعث عدم آپلود پاسخ ها در LMS و ارسال آنها از طریق ایمیل نشوند :

(قطعی اینترنت، تنظیم نبودن دقیق ساعت سایت با ساعت گرینویچ، کرش سیستم عامل و نیاز به فرمت، بارش زیبای شهاب سنگ از آسمان و ...)

بخش اول (پیاده‌سازی اولیه)

در ابتدا، می‌خواهیم رده‌بند[1] درخت تصمیم را از بیخ و بن (بدون استفاده از کتابخانه‌ی آماده)، برای داده‌های گسسته، مطابق شبه کد ارائه شده در اسلایدهای کلاس، خودمان پیاده سازی نماییم.

function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent_examples*) **returns** a tree

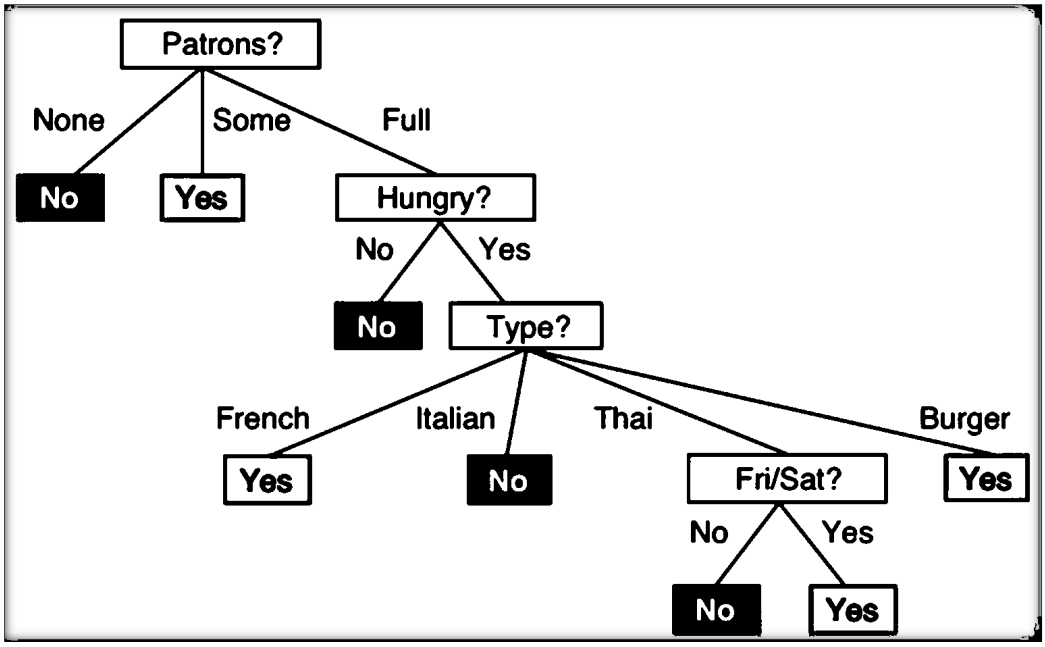
if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*)
else if all *examples* have the same classification **then return** the classification
else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*)
else
  $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
  tree $\leftarrow$ a new decision tree with root test A
  for each value v_k of A **do**
    $exs \leftarrow \{e : e \in examples$ **and** $e.A = v_k\}$
    subtree $\leftarrow$ DECISION-TREE-LEARNING(*exs*, *attributes* $- A$, *examples*)
    add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
  return *tree*

برای اطمینان از صحت پیاده سازی صورت گرفته، میتوانید داده های ۱۲ تایی مثال رستوران (مطرح شده در کلاس درس) را مورد بررسی قرار دهید.

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

[1] classifier

برای این کار تمام ۱۲ داده را به عنوان مجموعه آموزشی در نظر بگیرید(بدون مجموعه‌ی آزمایشی مجزا) و الگوریتم را برای این داده‌ها اجرا نمایید و سعی کنید ساختار درخت آموزش دیده شده را نمایش دهید، خروجی صحیح مطابق تصویر زیر می‌باشد(البته ممکن است دو ویژگی آنتروپی یکسانی داشته باشند و ساختار درخت شما در یک لایه متفاوت باشد).



بخش دوم (دسته‌بندی[۲])

در این مرحله می‌خواهیم با استفاده از رده‌بند درخت تصمیمی که در مرحله قبل پیاده سازی کرده‌اید، یک مسئله دسته‌بندی با داده‌های واقعی را حل کنیم. برای این کار از مجموعه دادگان این لینک می‌کنیم که در کنار این فایل با نام part2.csv نیز آمده است که یک مجموعه داده برای تشخیص تومور سرطانی با استفاده از تعدادی ویژگی می‌باشد.

در ابتدای کار می‌بایست داده‌ها را به دو بخش داده‌های آموزشی[۳] و داده‌های آزمایشی[۴] تقسیم کنید. نحوه بخش بندی داده‌ها به دو بخش آموزش و آزمایش به صورت کاملا اختیاری و به دلخواه خودتان است (مثلا ۹۰٪ ، ۱۰٪ – ۷۰٪ ، ۳۰٪ و ...)

همانطور که می‌دانید ورودی‌های درخت تصمیم باید به صورت گسسته باشد. برای گسسته سازی ورودی‌های از نوع پیوسته (مانند مجموعه داده فعلی) روش‌های مختلفی وجود دارد. ساده‌ترین ایده آن است که برای چنین ویژگی هایی، بازه مینیمم تا ماکزیمم اعداد در مجموعه آموزشی را به تعدادی بازه مساوی تقسیم کنید (چه تعداد؟ تعدادهای مختلف را آزمایش کنید) ایده های بهتر برای گسسته سازی مانند مرتب سازی و انتخاب نقاط برش در هر گره از درخت بر اساس نمونه هایی که در آن گره حاضرند را نیز امتحان کنید. همچنین می‌توانید

[2] Classification
[3] Training Set
[4] Test Set

ایده های مطرح شده در کلاس یا ایده‌های جدید و خلاقانه خود نیز استفاده کنید و نتایج آن را با حالت های قبل (بازه های مساوی یا انتخاب نقاط برش بر حسب مرتب سازی) مقایسه کنید.

سپس با استفاده از الگوریتم نوشته شده درخت تصمیم در بخش قبلی آموزش مدل را بر روی داده‌های آموزشی انجام دهید(در نظر داشته باشید برای پیاده سازی درخت تصمیم **نباید** از توابع آماده استفاده کنید، لذا فرمول آنتروپی و ... را باید خودتان پیاده کنید. استفاده از توابع آماده برای قسمت‌های بعدی بلامانع است (و حتی توصیه میشود) مثلا برای خواندن اکسل، نمایش گرافیکی خروجی درخت برای درک شهودی بهتر از فرآیند نحوه تصمیم گیری درخت تصمیم (که الزامی نیست)، نمایش دقت خروجی و ...

برای انتخاب بهترین ویژگی با توجه موارد تدریس شده از آنتروپی و information gain استفاده می‌شود، معیارهای دیگری مانند gini index نیز برای این کار وجود دارد. در ابتدا معیار gini index را تعریف کرده و چگونگی انتخاب بهترین ویژگی در الگوریتم درخت تصمیم را شرح دهید؛ سپس در کد نوشته شده این معیار را پیاده سازی کرده و آن را جایگزین information gain مبتنی برآنتروپی کنید. درختهای حاصل بر اساس این دو معیار را با هم مقایسه کنید: دقت آنها، اندازه درخت، ترتیب ملاقات ویژگی های از ریشه تا برگها (مثلا آیا ریشه درخت تغییر میکند؟ چند درصد گره های درخت متفاوت می‌شوند، و از این قبیل مقایسه ها) .

سپس مجموعه داده گفته شده را با PCA[5] به ۱۰ بعد کاهش داده و با معیاری (information gain یا gini index) که در آموزش درخت تصمیم بر روی مجموعه داده با تمام ویژگی ها بهتر عمل کرده است، آموزش این مجموعه داده‌ی کاهش بعد داده شده را انجام دهید.

برای جلوگیری از بیش‌برازش[6] از روش‌های توقف زودهنگام[7]برای جلوگیری از برگ‌های با تعداد نمونه‌های بسیار کم (یا gain خیلی کوچک) قبل از ساخت کامل درخت و روش‌های هرس کردن[8] پس از آموزش کامل و ساخت درخت، مورد استفاده قرار می‌گیرد. یک روش برای توقف زودهنگام و یک روش برای هرس کردن را پیاده سازی کنید و با مقادیر مختلف آزمایش کنید. ضمن توضیح مختصر نحوه پیاده سازی هر یک از روشها، نتایج این دو روش را با یکدیگر مقایسه کنید و مزایا و معایب هر یک را بگویید؛

بعد از فرآیند آموزش درخت تصمیم، مقادیر زیر را برای داده‌های آموزشی و داده‌های آزمایشی همه‌ی حالت‌های گفته شده محاسبه کنید: (مورد ٦ و ٧ فقط برای داده‌های آزمایشی رسم و محاسبه شود)

[5] Principal Component Analysis
[6] Overfitting
[7] early stopping or pre-pruning
[8] Pruning or post-pruning

۱) صحت[9] ۲) دقت[10] ۳) فراخوانی[11] ۴) معیار f1[12] ۵) ماتریس درهم‌ریختگی[13] ۶)نمودار ROC[14] ۷)مقدار AUC[15]

مقادیر ۱ تا ۵ با چند ضرب و تقسیم ساده به دست می آیند. مورد ۷ هم بر اساس مورد ۶ محاسبه میشود. برای نمودار ROC، لازم است یک پارامتر آزاد در نظر بگیرید و با تغییر آن، مقادیر مختلف برای FP و TP به دست آورید (برای هر مقدار مشخص از پارامتر مربوطه، هر جفت این مقادیر محاسبه شده و یک نقطه در نمودار را تشکیل می‌دهد.). پارامتر آزاد میتواند حدآستانه (threshold) برای دسته بندی باشد. فرض کنید دربرگها، به جای انتساب برچسب صفر یا یک (برای مساله دو کلاسه)، یک مقدار score برای کلاس مثبت در نظر گرفته شود، بدین ترتیب که نسبت تعداد داده های با خروجی (برچسب) مثبت به کل داده ها در آن برگ را برابر با score مثبت بودن آن برگ در نظر بگیریم (عددی بین صفر (وقتی هیچ داده مثبتی نیست) و یک (وقتی همه داده های آن برگ مثبتند). حال با این درخت دارای Score در برگها، میتوان به هر داده تست، یک مقدار score نسبت داد (معادل احتمال مثبت بودن آن داده) و برای تصمیم گیری نهایی، با یک آستانه گذاری ساده، تصمیم نهایی مثبت یا منفی بودن داده گرفته شود. اینکه این آستانه چقدر باشد، میتواند پارامتری باشد که با تغییر آن، جفت ها FP, TP محاسبه شده و نمودار ROC ترسیم گردد. برای توضیح تکمیلی و نحوه پیاده سازی میتوانید لینک زیر را مطالعه بفرمایید که بیان دیگری ازهمین توضیحات است:

https://stats.stackexchange.com/questions/105501/understanding-roc-curve/105577#105577

دقت کنید که در آزمایشها یک مقدار حداقلی از مقادیر دقت مد نظر است (می‌توانید درخت تصمیم نوشته شده با کتابخانه آماده را به عنوان این مقدار مناسب در نظر بگیرید) چنانچه درخت شما مقادیر دقت قابل قبولی نداشت (در مقایسه با درخت آماده موجود در کتابخانه خیلی تفاوت زیادبود) حتما سعی کنید دقت را افزایش و نتایج را بهبود دهید. (با تغییر مجموعه هایپر پارامترها چه در توزیع داده ها و چه در خود مدل و ... .) در غیر این صورت منجر به کسر نمره خواهد شد.

بخش سوم (تقریب تابع)

در این بخش می‌خواهیم با استفاده از درخت تصمیم تقریب تابع (مسئله‌ی رگرسیون) انجام دهیم. برای این کار از مجموعه داده‌ی part3.csv موجود در کنار این فایل، استفاده نماییم که یک مجموعه داده برای تخمین قیمت یک ماشین با استفاده از تعدادی از ویژگی‌های آن ماشین می‌باشد. برای اطلاعات بیشتر درباره این مجموعه داده به این لینک مراجعه نمایید.

[9] Accuracy Score
[10] Precision
[11] Recall
[12] F1-Measure
[13] Confusion Matrix
[14] Receiver Operating Characteristic
[15] Area Under Curve

در ابتدای کار نیاز است پس از تقسیم این مجموعه داده به مجموعه آموزشی و آزمایشی، یک پیش پردازش مناسب بر روی آن انجام داده و با ویژگی‌های آن بیشتر آشنا شوید و برای ویژگی‌های پیوسته، گسسته سازی به مانند بخش قبلی انجام دهید.

در گام بعدی نیاز است تا درخت نوشته شده در بخش قبلی را با کمی تغییرات (که در کلاس درس به آن اشاره شده است)، به یک درخت تصمیم مسئله رگرسیون تغییر دهید.

در برگ‌ها نیز از روش میانگین‌گیری، میانه‌گیری و fit کردن تابع استفاده نمایید. (نتایج این سه روش را می خواهیم در ادامه با یکدیگر مقایسه نماییم)

می‌توان مجددا از هرس کردن برای جلوگیری از بیش برازش (در صورت لزوم) استفاده کنید.

سعی کنید درخت تصمیم ساخته شده را رسم کنید (پیشنهادی) در غیر اینصورت سعی کنید ساختار آن را متوجه شوید و به صورت کامل گزارش نمایید.

برای ارزیابی این مدل، معیارهای ارزیابی برای مسئله رگرسیون را مورد مطالعه قرار دهید و هم برای داده‌های آموزشی و هم برای داده‌های آزمایشی این معیارها را محاسبه و تحلیل نمایید.

آنچه تحویل داده می‌شود:

۱) کداجرایی برنامه با توضیحات لازم برای اجرا

۲) درختی که برای مرحله دوم و سوم پیدا کرده اید (میتوانید گرافیکی نمایش دهید (به هر نحوی که میتوانید) یا به صورت Text با پروتکلی که توضیح می‌دهید و قابل فهم باشد (بشود فهمید در هر گره کدام ویژگی با چه مقادیری خروجی تست شده اند و زیر شاخه هایش کدامند و ...))

۳) گزارش کاملی از مسیر انجام کار، چالش‌هایی که مواجه شده‌اید، اجراهایی که گرفتید و نتایجی که حاصل شده است. گزارش کار از اهمیت بالایی برخوردار است، حجم آن و فرمت استاندارد آن اهمیت ندارد، اما باید نشان دهنده مسیر انجام پروژه، چالشها، راه حلها و نتایج کار شما باشد.