پروژه نهایی ساختمان داده

نیمسال تحصیلی اول - سال 1403 - 1402

Decision Tree

مقدمه

یکی از پرکاربردترین الگوریتم‌های داده‌کاوی، الگوریتم درخت تصمیم است. در داده‌کاوی، درخت تصمیم یک مدل پیشبینی کننده است به طوری که می‌تواند برای هر دو مدل رگرسیون و طبقه‌ای مورد استفاده قرار گیرد. زمانی که درخت برای کارهای طبقه‌بندی استفاده می‌شود، به عنوان درخت طبقه‌بندی (Classification Tree) شناخته می‌شود و هنگامی که برای فعالیت‌های رگرسیونی به کار می‌رود درخت رگرسیون (Regression Decision Tree)نامیده میشود.

در این پروژه قصد داریم درخت تصمیم را برای طبقه بندی داده ها پیاده سازی کنیم.

توضیحات درخت طبقه بندی

در این درخت شما باید با استفاده داده‌های Train درخت خود را تشکیل بدهید و بعد از آن با داده‌های Test ، خروجی داده‌ها را پیشبینی کنید. در اصل درخت تصمیم به عنوان یک‌تابع عمل می‌کند.
داده های Train شامل دو بخش ویژگی (attributes) و خروجی (labels) هستند.
داده های Test فقط شامل ویژگی ها هستند.

مفاهیم درخت تصمیم به صورت زیر است:

- هر گره داخلی نشان دهنده ویژگی های موجود هست و شاخه های خارج شده از گره نشان دهنده مقادیری هستند که آن ویژگی میتواند بگیرد.
- هر برگ نشان دهنده خروجی است.

آنتروپی(Entropy): آنتروپی معیاری برای محاسبه‌ی ناخالصی است. هر چه آنتروپی یک مجموعه بالاتر باشد به این معنی است که آن مجموعه مقادیر مختلف تری نسبت به مجموعه ای که آنتروپی پایین‌تری دارد، در خودش دارد.

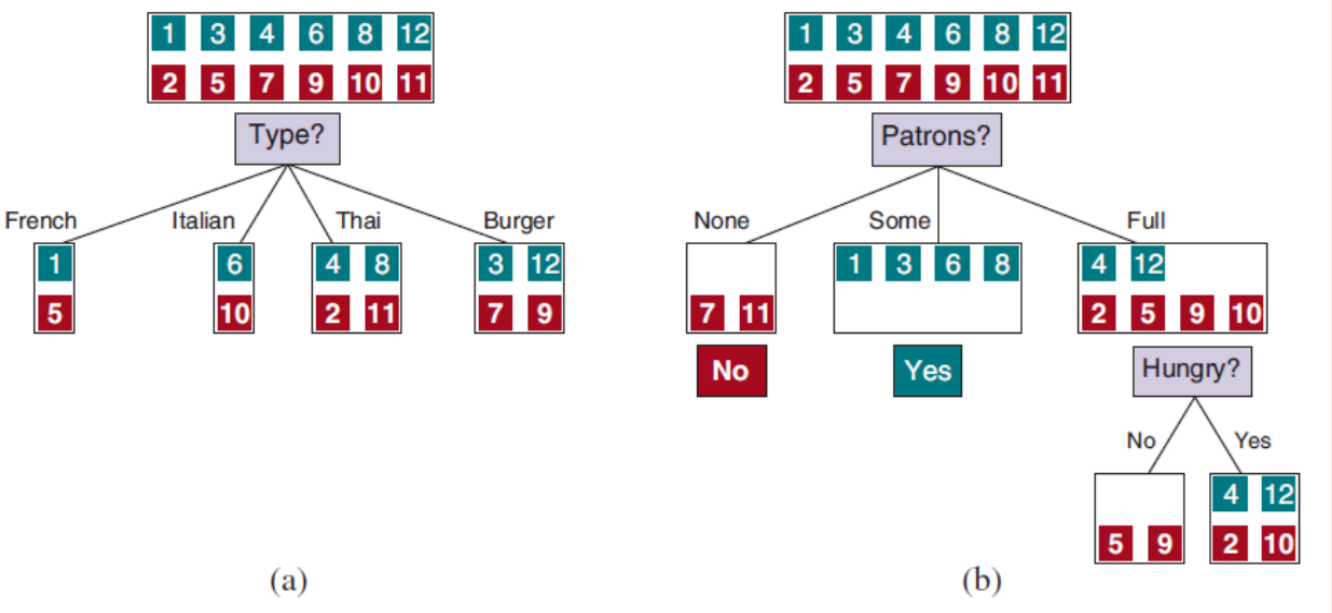
$$E(X) = -\sum_{x \in X} p(x) log\ p(x)$$

که P(x) نشان دهنده احتمال رخ دادن هر عنصر داخل مجموعه است.

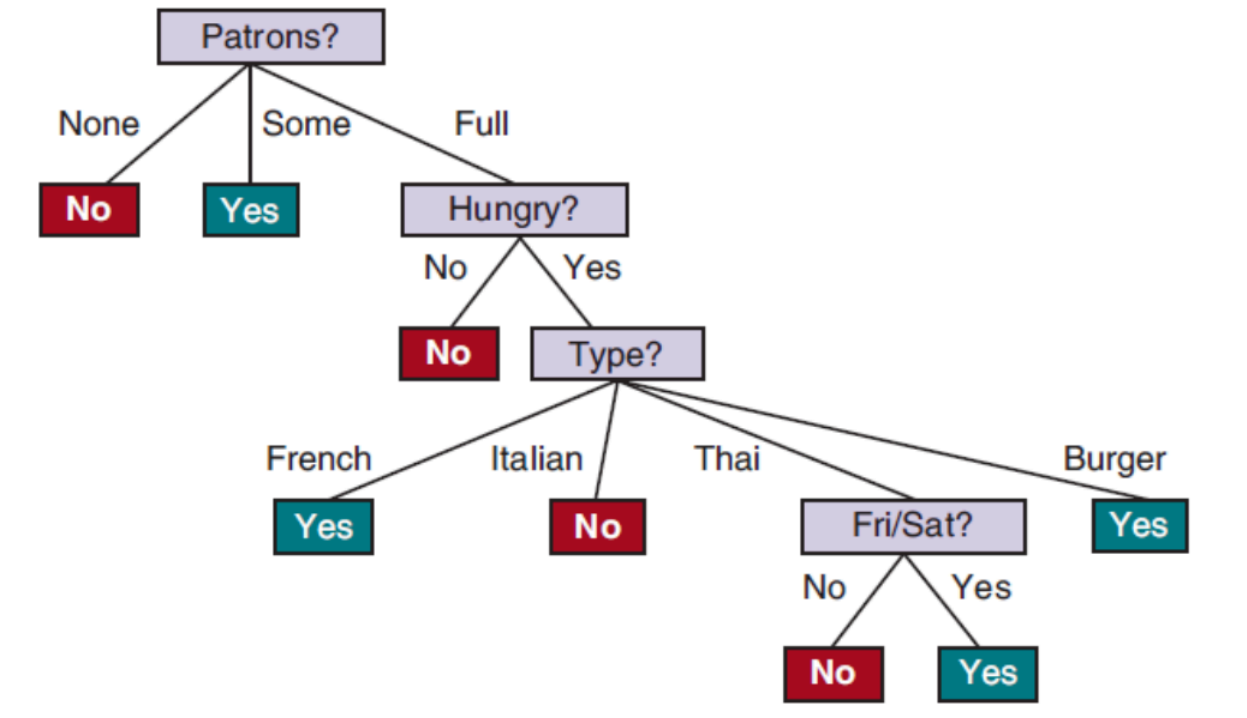
Information gain: برای اینکه در هر قسمت برای تقسیم داده ویژگی درست را انتخاب کنیم باید شاخصی داشته باشیم تا بر اساس آن بهینه‌ترین ویژگی را انتخاب کنیم.

$$IG = E(parent) - \sum w_i\ E(child_i)$$

از بین Gain های بدست آمده ویژگی با بالاترین Gain را برای تقسیم داده انتخاب می‌کنیم.اگر بخواهیم خلاصه تعریف کنیم ، این مقیاس برای ما مشخص میکند که به واسطه‌ی انتخاب این ویژگی، چه‌قدر ویژگی‌هایمان به حد خوبی در خالص سازی داده‌هایمان به ما کمک خواهند کرد و چه قدر اطلاعات از انتخاب این ویژگی به دست می‌آوریم.



Example	Input Attributes										Output
	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- کلاس Tree

```
class Tree{
int depth;
int getDepth();
createTree(float[][] data, float[] labels);
float Entropy(float[] labels);
float iGain(int pEntropy, float[] weight, float[] entropies);
}
```

- متد createTree: آرایه دو بعدی از ویژگی ها دریافت میکند که هر آرایه داخل آن نشان دهنده مقدار های هر ویژگی آن متغیر است. همچنین آرایه تک بعدی دریافت می‌کند که خروجی متناظر با هر متغیر را نشان می‌دهد.
  در نهایت درخت تصمیم ساخته شده را برمی‌گرداند.
- متد getDepth: مقدار depth را بر می‌گرداند که نشان دهنده حداکثر عمق درخت است.
- متد Entropy: آرایه‌ای از خروجی ها دریافت و آنتروپی آن‌ها را محاسبه می‌کند.
- متد iGain: آنتروپی پدر، وزن فرزندان و آنتروپی متناظر با آن‌ها را دریافت می‌کند و Gain را محاسبه می‌کند

کلاس DecisionTreeClassifier

این کلاس درون خود از کلاس Tree استفاده می‌کند.

```
class DTreeClassifire {
    DTreeClassifire(float[][] data, float[] labels);
    float Predict(float[], int depth);
    float[] PredictAll(float[][], int depth);
    float accuracy(int[] labels, int[] labels_predicted);
}
```

- در هنگام ایجاد یک شئ از این کلاس، آرایه از ویژگی ها به همراه خروجی آن ها به متد سازنده کلاس پاس داده می‌شود.
- متد Predict: ویژگی‌ها را دریافت می‌کند و خروجی متناسب با آن را با استفاده از درخت مشخص می‌کند.depth مشخص می‌کند تا چه عمق درخت باید پیشبینی انجام شود.
- متد PredictAll: مثل متد Predict عمل می‌کند با این تفاوت که ویژگی‌های چند متغیر را دریافت و لیستی از خروجی‌های پیشبینی شده برمی‌گرداند.
- متد accuracy: لیستی از خروجی‌های اصلی و خروجی‌های پیشبینی شده دریافت و میزان دقت الگوریتم را برمی‌گرداند.

الگوریتم Random Forest

الگوریتم جنگل تصادفی یا Random Forest یک متد برای طبقه بندی داده ها است که بر اساس مجموعه ای از درختان تصمیم که به صورت تصادفی ساخته شده اند، عمل می کند.

هر درخت تصمیم در این الگوریتم بر روی یک زیر مجموعه تصادفی از داده ها آموزش داده شده و سپس برای پیش بینی یک نمونه جدید، پیش بینی همه درختان گرفته می شود و برچسبی که بیشترین رای را داشته باشد به عنوان پیش بینی نهایی ارائه می شود.

Random Forest به دلیل استفاده از تعداد زیادی درخت تصمیم، معمولاً عملکرد بهتری نسبت به یک درخت تصمیم تکی دارد و همچنین قابلیت کنترل بیش برازش (overfitting) را نیز دارد.

برای توضیحات و مثال بیشتر به لینک زیر مراجعه کنید:

Understand Random Forest Algorithms With Examples

کلاس RandomForestClassifier

```
class RForestClassifier{
    RForestClassifier(float[][] data, float[] labels, int number_of_esimators, int max_depth);
    float Predict(float[] data, int depth);
    float PredictAll(float[][] data, int depth);
    float Accuracy(float[] labels , float[] predicted_labels);
}
```

- تنها تفاوت این کلاس با کلاس DecisionTreeClassifier در number_of_esimator است که همان تعداد درختان جنگل برای تصمیم گیری درباره پیشبینی داده وارد شده خواهد بود.
- دقت کنید که در کلاس RForestClassifier هر درخت تصمیم باید از یک زیر مجموعه تصادفی از داده‌ها تشکیل شود و سپس پیشبینی‌ای که بیشترین رای را در بین همه درختان تصمیم داشت به عنوان پیشبینی نهایی انتخاب شود.

-دیتاهای قرار داده‌شده ویژگی‌های بیمارانی است که باید نوع دیابت آن‌ها را مشخص کنید. با پیاده سازی درست دقت حدود 75% تا 85% را می‌توانید به‌دست بیارید.

HighBP	HighChol	CholCheck	Smoker	Stroke
0 = no high BP 1 = high BP	0 = no high cholesterol 1 = high cholesterol	0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 years	Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no 1 = yes	(Ever told) you had a stroke. 0 = no 1 = yes
				

HeartDiseaseorAt...	PhysActivity	Fruits	Veggies	HvyAlcoholConsu...	AnyHealthcare
coronary heart disease (CHD) or myocardial infarction (MI) 0 = no 1 = yes	physical activity in past 30 days - not including job 0 = no 1 = yes	Consume Fruit 1 or more times per day 0 = no 1 = yes	Consume Vegetables 1 or more times per day 0 = no 1 = yes	Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week) 0 = no	Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc. 0 = no 1 = yes
					

NoDocbcCost	GenHlth	DiffWalk	Sex	Education	Income
Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? 0 = no 1 = yes	Would you say that in general your health is: scale 1-5 1 = excellent 2 = very good 3 = good 4 = fair 5 = poor	Do you have serious difficulty walking or climbing stairs? 0 = no 1 = yes	0 = female 1 = male	Education level (EDUCA see codebook) scale 1-6 1 = Never attended school or only kindergarten 2 = Grades 1 through 8	Income scale (INCOME2 see codebook) scale 1-8 1 = less than $10,000 5 = less than $35,000 8 = $75,000 or more
					

موارد نمره اضافه:

پیاده سازی رندوم فارست.
درخت حاصل را به صورت گرافیکی نمایش دهید.

نکات تکمیلی:

- استفاده از هر زبان برنامه نویسی برای انجام پروژه مجاز است.

- در صورت نیاز می‌توانید متدهای کمکی پیاده‌سازی کنید.

- استفاده از هرگونه کتابخانه خارج از کتابخانه‌های اصلی زبان، مجاز نیست.

- در صورت نیاز به استفاده از دیگر ساختمانهای داده، پیاده‌سازی آنها الزامی است.

- پروژه به صورت *انفرادی* یا در *گروه‌های 2نفره* قابل انجام است (آپلود توسط هر دو عضو الزامی‌ست).
گروه‌ها **می‌بایست** از گیت استفاده کنند.

- هنگام تحویل، *هر دو عضو* گروه باید **تسلط کامل** داشته باشند.

- در صورت مشاهده شباهت غیر متعارف میان پروژه افراد، نمره -100برای هر دو نفر در نظر گرفته میشود.

- تسلط به بخشهای مختلف پروژه در هنگام تحویل الزامی است.

- فایلهای نهایی پروژه خود را در قالب زیر در سامانه vu بارگزاری کنید:

FirstNameLastNames_StudentNumbers_PR2.zip

برای توضیحات بهتر می‌تونید لینک 1 و 2 را مشاهده کنید.

موفق باشید