دوره جامع پایتون:
بخش یادگیری ماشین
جلسه بیستم

دکتر ذبیح اله ذبیحی

درخت تصمیم Decision Tree

- درخت تصمیم نقشه‌ای از نتایج احتمالی یکسری از انتخاب‌ها یا گزینه‌های مرتبط بهم است به طوری که به یک فرد یا سازمان اجازه می‌دهد تا اقدامات محتمل را از لحاظ هزینه‌ها، احتمالات و مزایا بسنجد.

- درخت‌های تصمیم می‌گویند زیرا می‌توانند یک تصمیمِ خاص (مثلاً اینکه به یک شخص وام بدهیم یا نه) را بر اساسِ اطلاعاتِ گذشته اتخاذ کنند.
- درخت تصمیم گیری ساختار شرطی (if ....then ...else) را دارد. این ساختار Decision Treeرا برای ساختار برنامه ها به ساگی مناسب می کند. همچنین برای مسئله های دسته بندی براساس ویژگی های داده ها مناسب است. برای مثال به طور مؤثر برای تعیین گونه های یک حیوان مورد استفاده قرار بگیرد.

درخت تصمیم چگونه کار می‌کند؟

- کسانی که بازی بیست سوالی انجام داده اند به سادگی می‌توانند درخت تصمیم گیری را درک کنند. در این بازی یک نفر موضوع خاصی را در ذهن خود در نظر می‌گیرد و شخص دیگری سعی می‌کند با پرسش تعدادی سوال که جواب آنها بلی و خیر است موضوع مورد نظر شخص اول را شناسایی کند. در درخت تصمیم نیز تعدادی پرسش وجود دارد و با مشخص شدن پاسخ هر سوال یک سوال دیگر پرسیده می‌شود. اگر سوالها درست و سنجیده پرسیده شوند، تعداد کمی از پرسش‌ها برای پیش بینی رکورد جدید کافی می باشد.

انواع متغیرها در درخت تصمیم

- متغیرهای عددی یا پیوسته: مانند سن، قد، وزن و... که مقدار خود را از مجموعه اعداد حقیقی می‌گیرند.
- متغیرهای رده‌ای یا گسسته : مانند نوع، جنس، کیفیت و... که به‌صورت دو یا چند مقدار گسسته هستند. در مواردی مانند آیا این شخص دانش‌آموز است؟ که دو جواب بله و خیر داریم، این متغیر از نوع طبقه‌ای خواهد بود.

- متغیرهای مستقل، متغیرهایی هستند که مقدار آن‌ها، مبنای تصمیم گیری ما خواهند بود و متغیر وابسته، متغیری است که بر اساس مقدار متغیرهای مستقل، باید مقدار آنرا پیش‌بینی کنیم. متغیرهای مستقل با گره‌های میانی نشان داده می‌شوند و متغیرهای وابسته، با برگ نشان داده می‌شوند. حال هر یک از این دو نوع متغیر مستقل و وابسته، می‌تواند گسسته یا پیوسته باشد.
- چنانچه متغیری وابسته عددی باشد دسته بندی ما یک مساله رگرسیون و چنانچه طبقه‌ای باشد، دسته بندی از نوع، رده‌بندی Classificationاست.

اصطلاحات مهم مربوط به درخت تصمیم

- گره ریشه: این گره حاوی تمام نمونه‌های موجود هست و سطح بعدی اولین تقسیم مجموعه اصلی به دو مجموعه همگن‌تر است. گره تصمیم: زمانی که یک گره به زیرگره‌های بعدی تقسیم می‌شود، آن را یک گره تصمیم می‌نامیم.
- برگ / گره پایانه: گره‌هایی که تقسیم نمی‌شوند یا به عبارتی تقسیم پیاپی از طریق آن‌ها پایان می‌یابد، برگ یا گره پایانه نام دارند.
- هرس کردن: هنگامی‌که ما از یک گره تصمیم، زیر گره‌ها را حذف کنیم، این عمل هرس کردن نامیده می‌شود. درواقع این عمل متضاد عمل تقسیم کردن است.
- انشعاب / زیردرخت: بخشی از کل درخت را انشعاب یا زیر درخت می‌گویند.
- گره‌های پدر و فرزند: گره‌ای که به چندین زیر گره تقسیم می‌شود را گره والد یا گره پدر برای زیر گره‌های آن می‌گویند. درحالی‌که زیر گره‌هایی که والد دارند، به‌عنوان گره‌های فرزند شناخته می‌شوند.

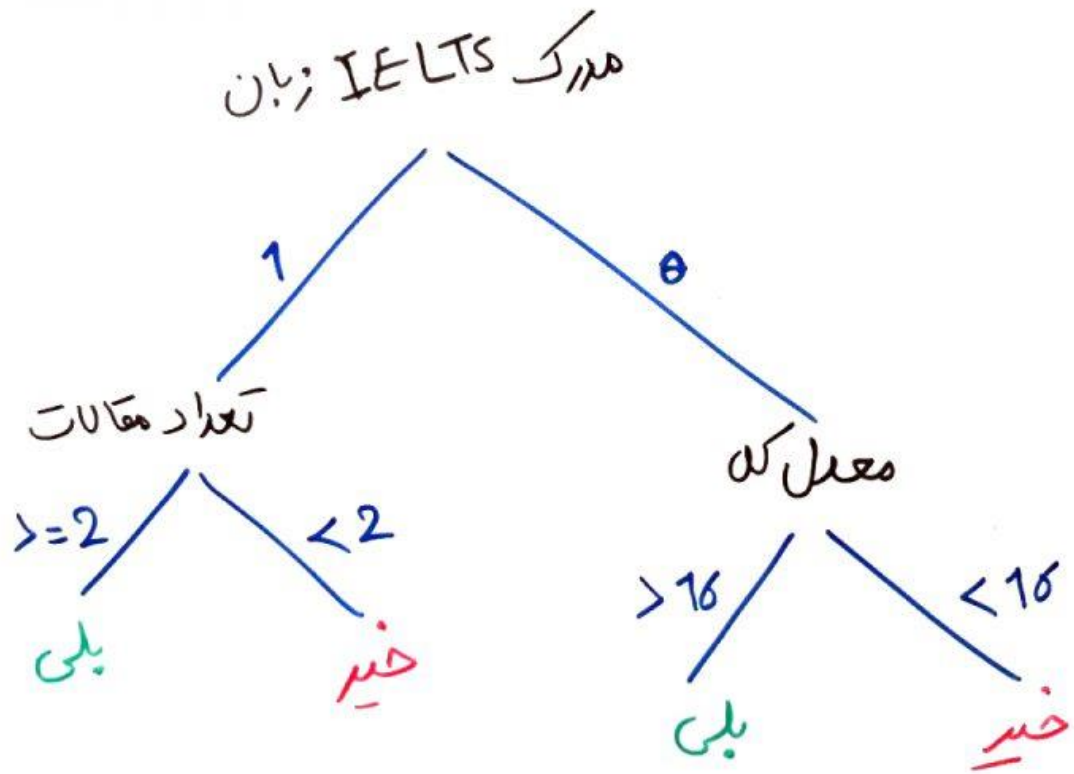
- درخت تصمیم‌گیری سه نوع Node (گِره) مختلف وجود دارد که عبارتند از:

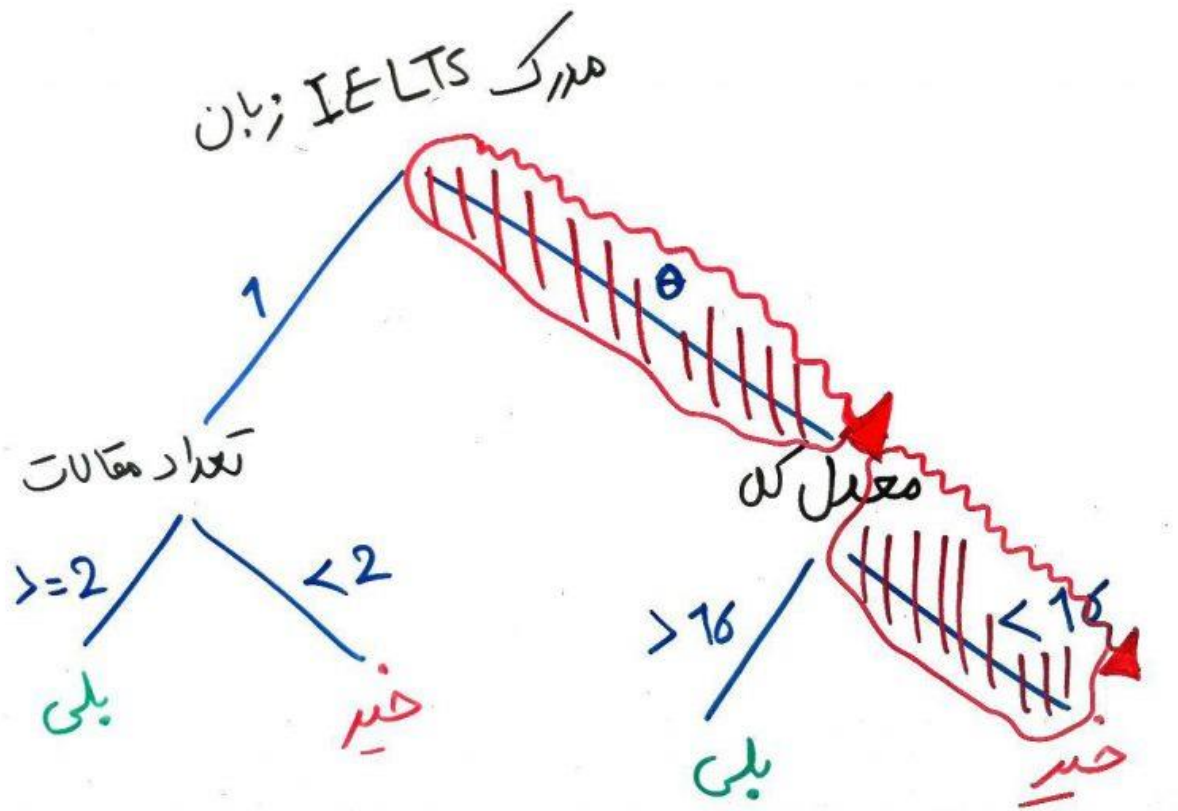
- - نُودهای تصادفی
- - نُودهای تصمیم‌گیری
- - نُودهای پایانی

مثال

- با استفاده ها از داده های موجود از جدول (داده‌های مختلفِ دانشجوهای قبلی) و مدل درخت تصمیم گیری تعیین کنید که آیا دانشجوی جدیدی که معَدل ۱۵.۵ دارد، تعداد ۵ مقاله ارائه کرده است ولی مدرک IELTSزبان ندارد. همچنین ۳ سال سنوات تحصیلی دارد. در آزمون دکتری قبول می‌شود یا خیر؟
- هر ردیف (اطلاعات دانشجوی سال قبل)۴ ویژگی دارند. ۱. معدل کل (عدد) ۲. تعداد مقالات (عدد) ۳. مدرک IELTSزبان دارند (۰ یا ۱)؟ ۴. سنوات تحصیلی (عدد). و همچنین یک ستونِ برچسب که اگر دانشجو در در مقطع دکتری قبول شده بود، بلی و اگر قبول نشده بود، خیر است

دکتری قبول شده؟	سنوات تحصیلی	مدرک زبان	تعداد مقاله	معدل کل	
۱	۳	۱	۳	۱۹/۵	۱
۰	۴	۱	۰	۱۶/۵	۲
۰	۳	۰	۰	۱۵	۳
۱	۲/۵	۱	۲	۱۷	۴
۱	۲/۵	۰	۲	۱۸/۵	۵
۰	۲/۵	۱	۱	۱۵/۵	۶
۱	۳	۱	۳	۱۹	۷


مدرک IELTS زبان
1
0
تعداد مقالات
معدل کل
>=2
<2
>16
<16
بلی
خیر
بلی
خیر


مدرک IELTS زبان
θ
1
تعداد مقالات
>=2
<2
بلی
خیر
معدل کل
>16
<16
بلی
خیر

- درخت تصمیمِ ساخته شده به ما می‌گوید که این دانشجو احتمالاً نمی‌تواند در آزمون دکتری قبول شود. اینَ طبقه‌بندی با استفاده از درختی که توسط داده‌های گذشته ساخته بود، انجام شد.

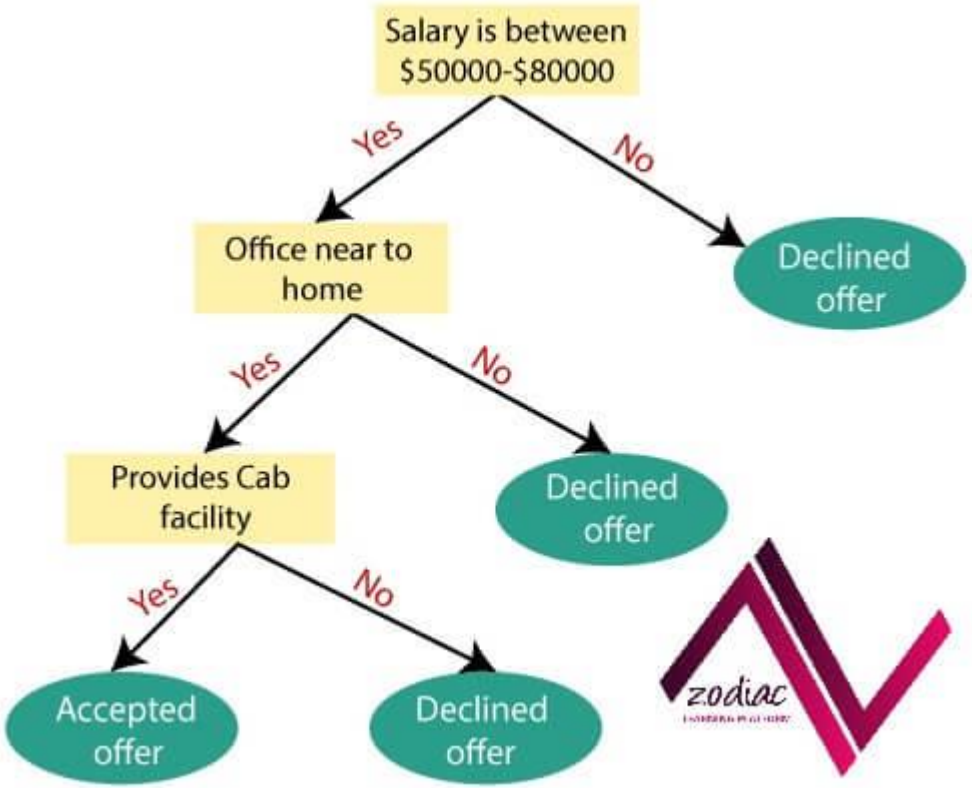
```
import matplotlib.pyplotas plt
from sklearn.treeimport DecisionTreeClassifier
from sklearn.treeimport plot_tree
x
=[[19.5,3,1,3],[16.5,0,1,4],[15,0,0,3],[17,2,1,2.5],[18.5,2,0,2.5],[15.5,1,1,2.5],[19,3,1
,3]]
y = [1,0,0,1,1,0,1]
model= DecisionTreeClassifier()
model.fit(x, y)
print(model.predict([[15.5,5,0,3]]))
plot_tree(model)
plt.show()
```

مثال

- مثلا فرض کنید میخواهید در خصوص احتمال بارش باران و پیش بینی و تصمیم اینکه چتر همراه خودتان بیرون ببرید یا نه تصمیم بگیرید. این سوال از خودتان مطرح میکنید که الان چه فصلی است ، بهار ، تابستان ، پاییز یا زمستان. حال تصمیم شما انشعاب پیا می کند ، ممکن است بگویید الان تابستان است و احتمال باران آمدن صفر است پس بدون شک تصمیم بعدی این است که چتر همراه خود نبرید. اما اگر زمستان باشد ، به سراغ سوال بعدی می روید ، که آیا آسمان ابری است یا صاف ؟؟؟ اگر آسمان صاف باشد مثلا احتمال بارش ۲۰٪ است و شاید چتر نبرید اما اگر ابری باشد احتمال بارش ۹۰ ٪ است پس بدون شک چتر ببرید.

- مثال بالا یک مثال ساده بود که تنها ۲ معیار فصل و ابری بودن در ان بود هر چند می شد این معیار ها بسیار زیاد باشند و یک درخت بزرگ شکل بگیرد.

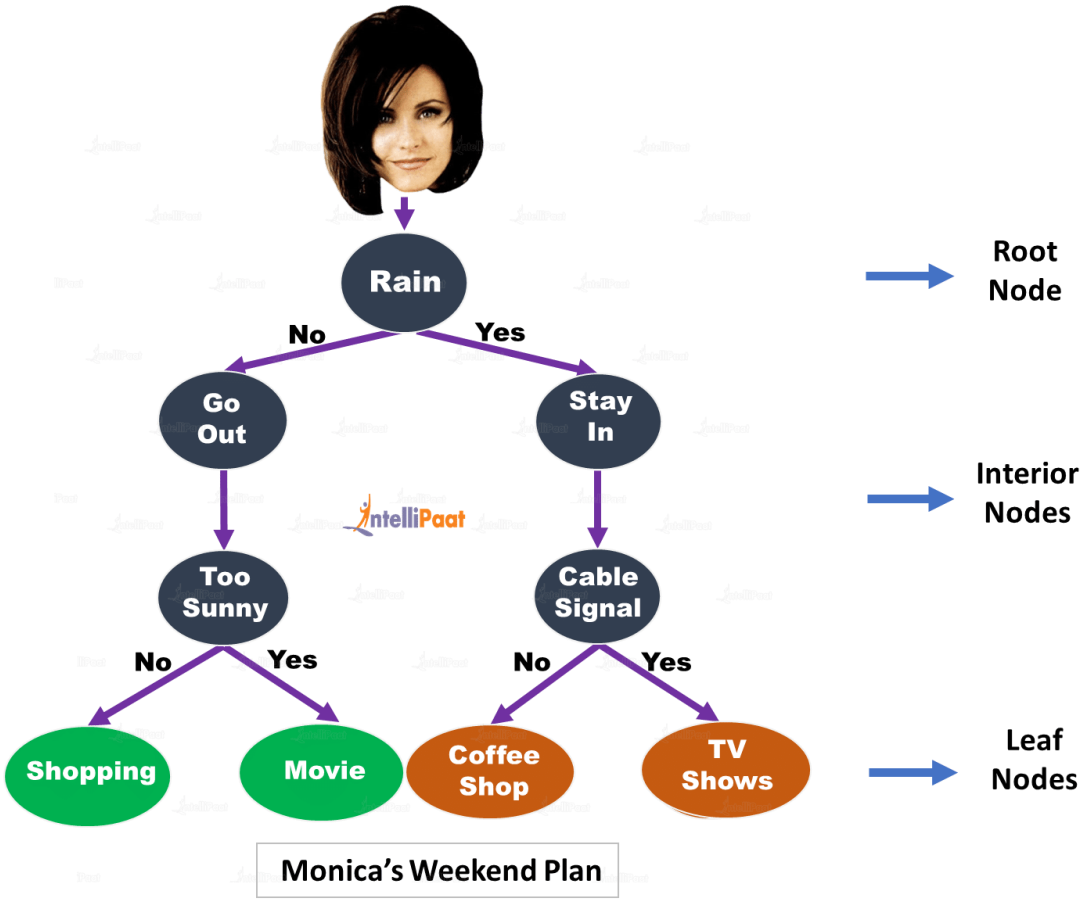
تمرین


Salary is between $50000-$80000
Yes
No
Office near to home
Declined offer
Yes
No
Provides Cab facility
Declined offer
Yes
No
Accepted offer
Declined offer
zodiac

- حال قوانین و دانش درخت در اصل بصورت زیر است :

- اگر حقوق بین ۵۰ -۸۰ هزار دلار نباشد پیشنهاد رد است
- اگرحقوق بین ۵۰-۸۰ هزار دلار باشد و دفتر کنار خانه نباشد ، پیشنهاد رد می شود
- اگر حقوق بین ۵۰ – ۸۰ هزار دلار باشد ، دفتر کنار خانه باشد ، امکانات فراهم نشود ، پشنهاد رد است
- اگر حقوق بین ۵۰ – ۸۰ هزار دلار باشد ، دفتر کنار خانه باشد ، امکانات فراهم شود ، پیشنهاد قبول می شود

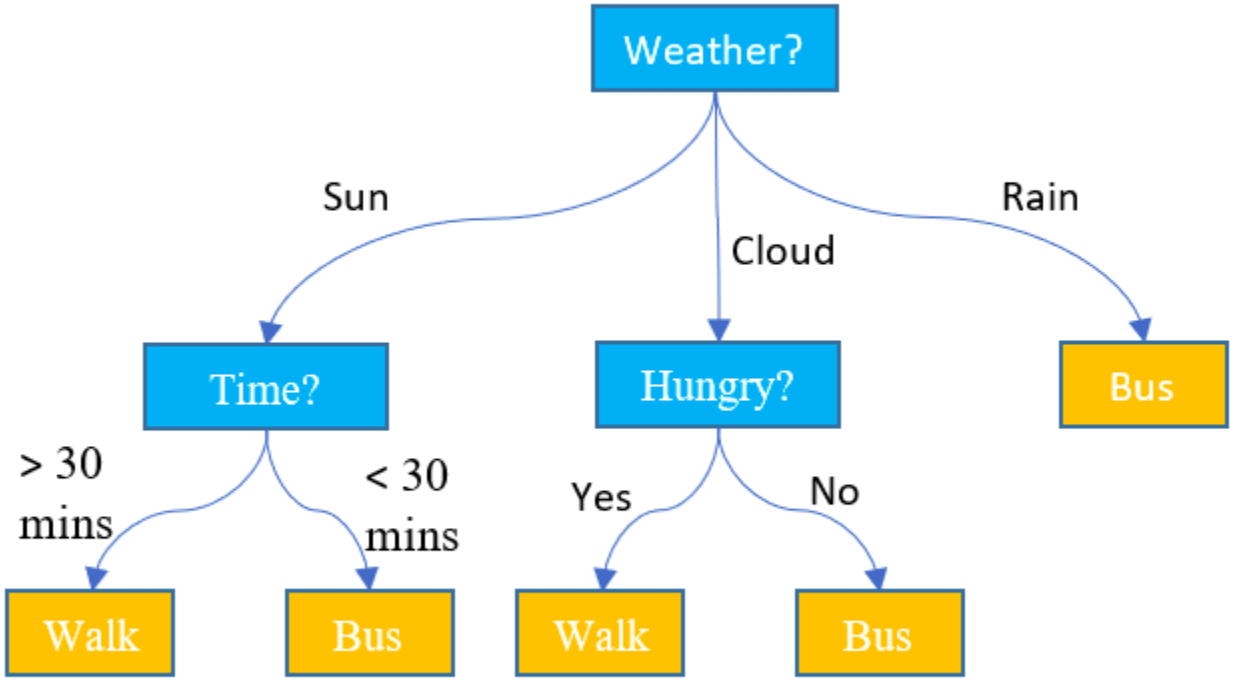
مثال


Rain
No
Yes
Go Out
Stay In
IntelliPaat
Too Sunny
Cable Signal
No
Yes
No
Yes
Shopping
Movie
Coffee Shop
TV Shows
Root Node
Interior Nodes
Leaf Nodes
Monica's Weekend Plan

مثال

- تعیین خریداران احتمالی یک محصول با استفاده از داده های جمعیتی برای هدفمند کردن بودجه تبلیغات محدود
- پیش بینی احتمال تقاضای وام گیرندگان با استفاده از مدل های پیش بینی شده از سابقه وام گیرندگان
- کمک به اولویت بندی روش های درمان بیماران اورژانسی با توجه به سن، سابقه، فشار خون، جنسیت، محل و شدت درد....
- ارزیابی فرصت های توسعه برند برای یک تجارت با استفاده از داده های فروش تاریخی

مثال


Weather?
Sun
Cloud
Rain
Time?
Hungry?
Bus
> 30 mins
< 30 mins
Yes
No
Walk
Bus
Walk
Bus

ادامه درخت تصمیم گیری

- مجموعه داده به دو بخش داده‌های «آموزش» (Training) و «آزمون» (Test) تقسیم می‌شود، مدل روی داده‌های آموزش ساخته و صحت این مدل در مقابل داده‌های «آزمون» آزموده می‌شود.
- برای ایجاد یک درخت تصمیم، همه ی داده‌ها باید عددی باشند. داده های غیر عددی را باید تبدیل به عدد کرد.
- خواهید دید که اگر درخت تصمیم را به تعداد دفعات کافی اجرا کنید، نتایج مختلفی به شما ارائه می‌دهد، حتی اگر داده‌های یکسان به آن دهید. این مسئله به این دلیل است که درخت تصمیم یک پاسخ قطعی ٪۱۰۰ به ما نمی‌دهد. این پاسخ بر پایه احتمال یک خروجی است و پاسخ متفاوت خواهد بود.

مثال

- هدف: ایجاد یک مدل درخت تصمیم در پایتون تا تعیین کند که آیا داوطلبان در یک دانشگاه معتبر پذیرفته می شوند یا خیر.
- دو نتیجه احتمالی وجود دارد: پذیرفته شده (۱) در مقابل رد شده (۰)
- سپس می توان یک رگرسیون لجستیک در پایتون ایجاد کرد ، جایی که:
- متغیر وابسته نشان می دهد که آیا فرد پذیرفته می شود. و ۳ متغیر مستقل نمره GMAT، معدل (GPA)و سالها سابقه کار هستند

gmat	gpa	work_experience	admitted
780	4	3	1
750	3.9	4	1
690	3.3	3	0
710	3.7	5	1
680	3.9	4	0
730	3.7	6	1
690	2.3	1	0
720	3.3	4	1
740	3.3	5	1
690	1.7	1	0
610	2.7	3	0
690	3.7	5	1
710	3.7	6	1
680	3.3	4	0
770	3.3	3	1
610	3	1	0
580	2.7	4	0
650	3.7	6	1
540	2.7	2	0
590	2.3	3	0
620	3.3	2	1
600	2	1	0
550	2.3	4	0
550	2.7	1	0
570	3	2	0
670	3.3	6	1
660	3.7	4	1
580	2.3	2	0
650	3.7	6	1
660	3.3	5	1
640	3	1	0
620	2.7	2	0
660	4	4	1
660	3.3	6	1
680	3.3	5	1
650	2.3	1	0
670	2.7	2	0
580	3.3	1	0
590	1.7	4	0
690	3.7	5	1

بخش اول بارگذاری داده ها و متغیر ها

```
import pandas as pd
candidates = {'gmat':
[780,750,690,710,680,730,690,720,740,690,610,690,710,680,770,610,580,650,540,590,620,600,55
0,550,570,670,660,580,650,660,640,620,660,660,680,650,670,580,590,690],
'gpa':
[4,3.9,3.3,3.7,3.9,3.7,2.3,3.3,3.3,1.7,2.7,3.7,3.7,3.3,3.3,3,2.7,3.7,2.7,2.3,3.3,2,2.3,2.7,3,3.3,3.7,2.3,3.
7,3.3,3,2.7,4,3.3,3.3,2.3,2.7,3.3,1.7,3.7],
'work_experience': [3,4,3,5,4,6,1,4,5,1,3,5,6,4,3,1,4,6,2,3,2,1,4,1,2,6,4,2,6,5,1,2,4,6,5,1,2,1,4,5],
'admitted': [1,1,0,1,0,1,0,1,1,0,0,1,1,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,1,1,0,1,1,0,0,1,1,1,0,0,0,0,1]
}
df= pd.DataFrame(candidates,columns= ['gmat', 'gpa','work_experience','admitted'])
X = df[['gmat', 'gpa','work_experience']]
y = df['admitted']
```

بخش دوم: فراخوانی مدل درخت تصمیم گیری

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier


X_train,X_test,y_train,y_test=
train_test_split(X,y,test_size=0.25,random_state=0)
model= DecisionTreeClassifier()
model.fit(X_train,y_train)
```

بخش سوم: تعیین صحت داده های آموزش و آزمون

```
print("Training set accuracy: {:.3f}".format(model.score(X_train, y_train)))
print("Test set accuracy: {:.3f}".format(model.score(X_test, y_test)))
```

بخش چهارم: پیش بینی

```
y_pred=model.predict(X_test)
print(y_pred)
```

بخش پنجم: خطای مقدار پیش بینی از مقدار واقعی

```
import numpy as np
percentageerror_tree=(y_test-y_pred)
print(np.mean(percentageerror_tree))
```

بخش ششم:رسم درخت تصمیم

```
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
plot_tree(model)
plt.show()
```

الگوریتم Kنزدیک ترین همسایه (KNN)

مثال

- فرض کنید شما یک فروشگاه مواد غذایی و دو دسته مشتری دارید، مشتریانِ دسته اول، کسانی هستند که بیشترِ از ۱۰۰ هزار تومان در هر بار مراجعه َاز شما خریداری می‌کنند و دسته‌ی دوم مشتریانی هستند که در هر مراجعه معمولاً کمتر از ۱۰ هزار تومان خرید می‌کنند. شما که فروشنده‌ی با تجربه‌ای هستید، به سبب چند سالی که در این مغازه فعالیت دارید می‌دانید که مشتریان دسته‌ی اول خانم‌هایی هستند که سن بالای ۴۰ سال دارند و با اتومبیل به فروشگاه شما می‌آیند (مثلاً در مراجعه‌ی آن‌ها به این الگو دقت کرده‌اید). و دسته‌ی دوم، یعنی همان مشتریانی که کمتر از ۱۰ هزار تومان از شما خریداری می کنند. معمولاً آقایانی هستند که سن زیر ۲۵ سال دارند و بدون اتومبیل به فروشگاه می‌آیند. حال فرض کنید، یک مشتری جدید به فروشگاه آمده است. این شخص یک خانم ۴۵ ساله است که با اتومبیل خود به فروشگاه آمده. شما این مشتری جدید را در کدام دسته (با توجه به مشتریان قبلی) قرار میدهید؟ انتظار داریدَ که این شخص چقدر از شما خرید کند؟

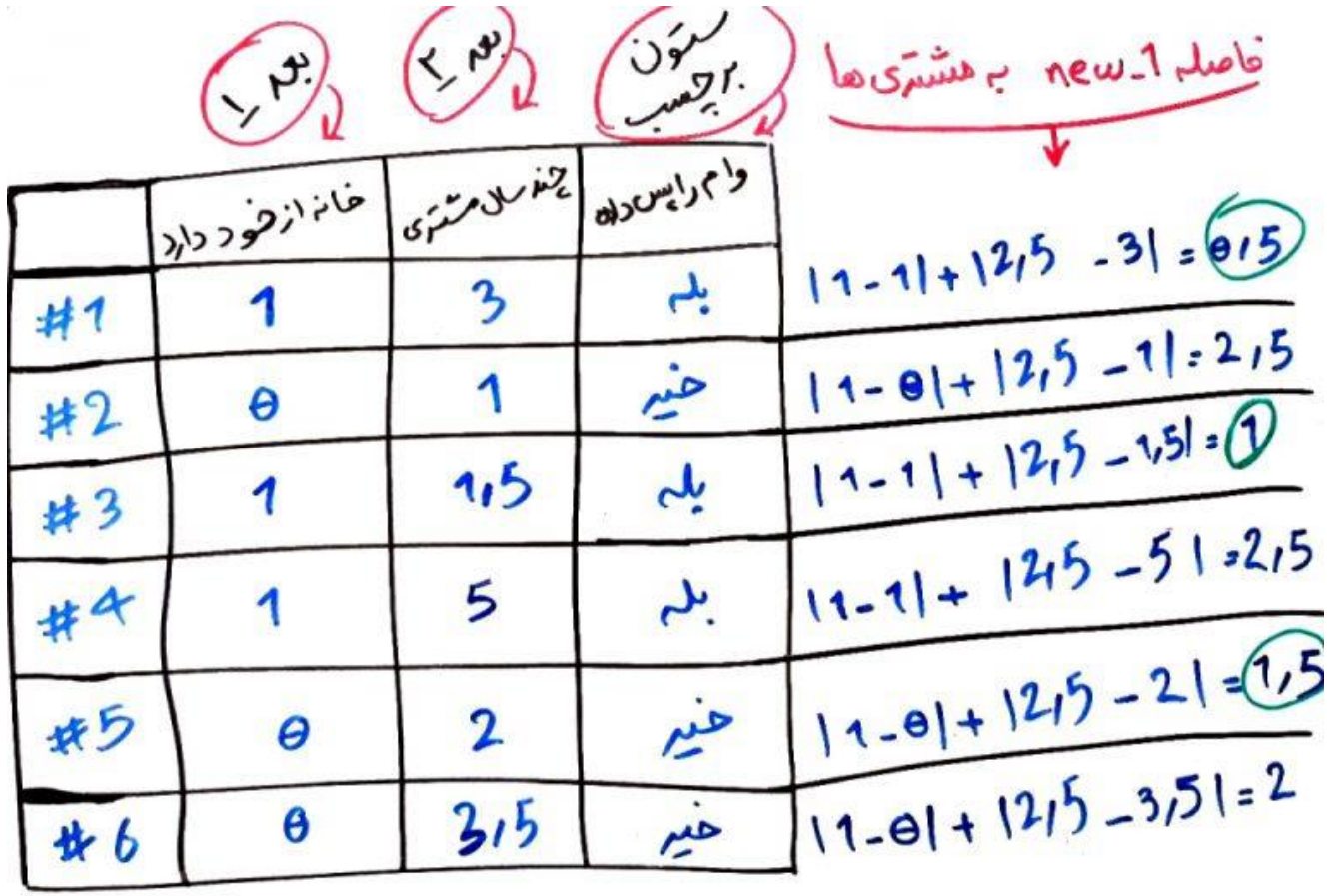
- قطعاً بدون فکر کردن، دسته‌ی اول را برای پاسخ به سوالِ بالا انتخاب می‌کنید. دلیل آن بسیار ساده است. این خانم ۴۵ ساله با اتومبیل، بسیار نزدیک‌تر به مشتریانِ دسته‌ی اول َاست تا مشتریانِ دسته‌ی دوم (دسته‌ی دوم، آقایانی بودند که زیر ۲۵ سال سن داشتند و بدون اتومبیل بَه فروشگاه می‌آمدند). واژه نزدَیک‌تر، در واقع پایه‌ی عملیات طبقه‌بندی K نزدیک‌ترین همسایه می‌باشد. در این الگوریتم، هر کدام از نمونه‌های جدید (مثلاً مشتری‌های جدید) با تَمامی نمونه‌َهای قبلی مقایسه می‌شوند و به هر کدام از نمونه‌های قبلی که نزدیک‌تر باشند، به آن دسته از نمونه‌ها تعلق می گیرند. این دقیقاً همان کاری است که در مثال اول توسط فروشنده‌ی با تجربه‌ی فروشگاه مواد غذایی انجام شد.

مثال

- فرض کنید شما مدیریت یک بانک را بر عهده دارید و می‌خواهید از بینِ کسانی که درخواستِ وام کرده‌اند، آن هایی را انتخاب کنید که به احتمال زیاد می‌توانند وام خود را بازگردانند. فرض کنید برای این کار یک دیتاست از مشتریانِ قبلیِ خود که وام گرفته‌اند را آماده کرده‌اید و به هر کدام از این مشتریانْ یک برچسبِ بله (یعنی توانسته‌اند وام را بازگردانند) و خیر (یعنی نتوانسته‌اند وام را بازگردانند) نسبت داده‌اید. توجه کنید که این‌ها مشتریانِ قبلی شما هستند که وضعیت برگرداندنِ وام توسط آن‌ها مشخص شده است. هر کدام از مشتری‌های شما هم ۲ ویژگی دارد. ویژگی اول: خانه از خود دارد یا خیر (۰ به معنیِ خانه نداشتن و ۱ به معنیِ داشتنِ خانه از خود است)؟ ویژگی دوم: چند سال است که مشتری بانک است (یک عدد بزرگتر از ۰)؟ هر کدام از این ویژگی‌ها برای تمامیِ مشتریان تکرار می شوند. نگاهی به شکل زیر بیندازید (یک دیتاست با ۶ نمونه و ۲ بُعد و دارای برچسب):

- حال فرض کنید دو مشتریِ جدید با ویژگی‌هایی مطابق با جدول زیر متقاضیِ جدیدِ وام هستند و که می‌خواهیم تصمیم بگیریم به آن ها وام بدهیم یا خیر

	خانه از خود دارد	چند سال مشتری بانک است
new_1	۱	۲/۵
new_2	۰	۲


بعد ۱
بعد ۲
ستون پرچسب
فاصله new_1 به مشتری‌ها
خانه از خود دارد
چند سال مشتری
وام را پس داده
#1
#2
#3
#4
#5
#6
بله
خیر
بله
بله
خیر
خیر
|1-1|+|2,5-3| = 0,5
|1-0|+|2,5-1| = 2,5
|1-1|+|2,5-1,5| = 1
|1-1|+|2,5-5| = 2,5
|1-0|+|2,5-2| = 1,5
|1-0|+|2,5-3,5| = 2

فاصله همسایه

- روش Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

- روش Manhattan

$$\sum_{i=1}^{k}|x_i - y_i|$$

- روش Minkowski

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$$

- روش Hamming

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \rightarrow D = 0$$

$$x \neq y \rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

- در روش Kنزدیک‌ترین همسایه، انتخاب بهترین مقدار برای Kبهتر است با اولین برسی داده ها انجام شود. به طور کلی، یک مقدار بزرگ Kدقیق تر است زیرا نویز کلی را کاهش می دهد اما هیچ تضمینی برای اعتبار آن وجود ندارد. اعتبار سنجی متقاطع یکی دیگر از راه های به دست آوردن یک Kخوب با استفاده از یک مجموعه داده مستقل برای اعتبار سنجی Kمی باشد. به لحاظ تجربه، Kمطلوب برای بیشتر مجموعه داده ها بین ۳ تا ۱۰ است. این نتایج بسیار بهتر از 1NNتولید می کند.

مثال

- هدف: ایجاد یک مدل نزدیک ترین همسایه در پایتون تا تعیین کند که آیا داوطلبان در یک دانشگاه معتبر پذیرفته می شوند یا خیر.
- دو نتیجه احتمالی وجود دارد: پذیرفته شده (۱) در مقابل رد شده (۰)
- سپس می توان یک رگرسیون لجستیک در پایتون ایجاد کرد ، جایی که:
- متغیر وابسته نشان می دهد که آیا فرد پذیرفته می شود. و ۳ متغیر مستقل نمره GMAT، معدل (GPA)و سالها سابقه کار هستند

gmat	gpa	work_experience	admitted
780	4	3	1
750	3.9	4	1
690	3.3	3	0
710	3.7	5	1
680	3.9	4	0
730	3.7	6	1
690	2.3	1	0
720	3.3	4	1
740	3.3	5	1
690	1.7	1	0
610	2.7	3	0
690	3.7	5	1
710	3.7	6	1
680	3.3	4	0
770	3.3	3	1
610	3	1	0
580	2.7	4	0
650	3.7	6	1
540	2.7	2	0
590	2.3	3	0
620	3.3	2	1
600	2	1	0
550	2.3	4	0
550	2.7	1	0
570	3	2	0
670	3.3	6	1
660	3.7	4	1
580	2.3	2	0
650	3.7	6	1
660	3.3	5	1
640	3	1	0
620	2.7	2	0
660	4	4	1
660	3.3	6	1
680	3.3	5	1
650	2.3	1	0
670	2.7	2	0
580	3.3	1	0
590	1.7	4	0
690	3.7	5	1

بخش اول:داده ها

```
import pandas as pd
candidates = {'gmat':
[780,750,690,710,680,730,690,720,740,690,610,690,710,680,770,610,580,650,540,590,620,600,550,550,570,670,660,580,650,660,6
40,620,660,660,680,650,670,580,590,690],
'gpa':
[4,3.9,3.3,3.7,3.9,3.7,2.3,3.3,3.3,1.7,2.7,3.7,3.7,3.3,3.3,3,2.7,3.7,2.7,2.3,3.3,2,2.3,2.7,3,3.3,3.7,2.3,3.7,3.3,3,2.7,4,3.3,3.3,2.3,2.7,3.3,1.
7,3.7],
'work_experience': [3,4,3,5,4,6,1,4,5,1,3,5,6,4,3,1,4,6,2,3,2,1,4,1,2,6,4,2,6,5,1,2,4,6,5,1,2,1,4,5],
'admitted': [1,1,0,1,0,1,0,1,1,0,0,1,1,0,1,0,0,1,0,0,1,0,0,0,0,1,1,0,1,1,0,0,1,1,1,0,0,0,0,1]
}
df= pd.DataFrame(candidates,columns= ['gmat', 'gpa','work_experience','admitted'])
X = df[['gmat', 'gpa','work_experience']]
y = df['admitted']


df.head()
```

بخش دوم: مقیاس کردن داده ها

```
df.head()
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df.drop('admitted',axis=1))
scaled_features = scaler.transform(df.drop('admitted',axis=1))
df_feat = pd.DataFrame(scaled_features,columns=df.columns[:-1])
df_feat.head()
```

بخش سوم:فراخوانی مدل نزدیک ترین همسایه

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

X_train,X_test,y_train,y_test=
train_test_split(scaled_features,df['admitted'],test_size=0.30)

model = KNeighborsClassifier(n_neighbors=1)
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
print(y_pred)
```

بخش چهارم: تعیین تعداد همسایه بهینه

```
import numpy as np
error_rate = []
# Might take some time
for i in range(1,28):

    model = KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train,y_train)
    pred_i = model.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))
```

بخش پنجم: رسم نمودار

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.plot(range(1,28),error_rate,color='blue', linestyle='dashed', marker='o',
        markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
plt.show()
```

نرمل سازی داده ها

$$Z = \frac{x - xmin}{xmax - xmin}$$

داده نرمال شده	داده اولیه
۱	۲۰
۰/۷۵	۱۵
۰/۵	۱۰
۰/۲۵	۵
۰	۰

تمرین: با توجه به جدول زیر قیمت یک آپارتمان ۷۰ متری، یک اتاق، طبقه دوم، ۱۷ سال ساخت در منطقه ۲ را تخمین بزنید.

قیمتf	منطقهe	سال ساختd	طبقهc	تعداد اتاقb	متراژa	
۶۲۰	۱	۱۰	۴	۲	۶۰	۱
۹۲۰	۱	۵	۲	۲	۸۰	۲
۳۶۰	۱	۱۵	۱	۱	۴۰	۳
۱۱۲۵	۲	۵	۲	۲	۷۵	۴
۱۱۷۰	۲	۱۰	۴	۳	۹۰	۵
۵۶۰	۲	۲	۱	۱	۳۵	۶
۷۰۰	۲	۱۰	۳	۱	۵۰	۷
۱۶۰۰	۳	۵	۵	۲	۸۰	۸
۱۸۰۰	۳	۱۰	۴	۳	۱۰۰	۹
۱۰۰۰	۳	۲	۱	۱	۵۰	۱۰