حوری دهش 9821413

.1

آ) بی نظارت

ب) با نظارت

ج) با نظارت

د) با نظارت

ه) با نظارت

و) بی نظارت

ز) با نظارت

ح) بی نظارت

ط) با نظارت

ی) با نظارت

.2

مثلا یک فروشگاه آنلاین داریم که میوه های تازه را عرضه می کند و می خواهیم مشتریان خود را بر اساس دو ویژگی رنگ (قرمز یا نارنجی) و طعم (شیرین یا ترش) به گروه های مختلف تقسیم کنیم تا بهترین استراتژی بازاریابی را برای هر گروه انتخاب کنیم.

:Gini Index

با استفاده از Gini Index، میتوانیم اثر تقسیم بندی بر روی خالصی گروه ها را ارزیابی کنیم پس:

ابتدا با استفاده از Gini Index، میتوانیم مشتریان را بر اساس ویژگی رنگ تقسیم کنیم، برای مثال اگر میوه‌های قرمز و نارنجی را در نظر بگیریم: اگر تمام سیب ها قرمز باشند و تمام پرتقال ها نارنجی باشند شاخص Gini برابر با صفر خواهد بود زیرا هر دو گروه خالص هستند اما اگر همه میوه ها ترکیبی از سیب و پرتقال باشند شاخص Gini به حداکثر (حدودا 1) میرسد، زیرا هر دو گروه ناخالص هستند. بنابراین، با توجه به این اطلاعات، می‌توانیم تقسیم بندی بهتری انتخاب کنیم که باعث کاهش مقدار Gini index و افزایش خالصی گروه ها شود.

در نهایت از بین Gini Index ها Gini Index را انتخاب میکنیم که از همه کمتر است.

Gain Ratio:

حالا با استفاده از معیار Gain Ratio، میتوانیم به میزان اطلاعاتی که هر تقسیم فراهم می کند و همچنین به تعداد و اندازه گروه های ایجاد شده، توجه کنیم. به عنوان مثال، اگر تقسیمی به ما گروه هایی با سطوح متفاوتی از رنگ و طعم، ولی خالص تر بدهد این تقسیم با استفاده از Gain Ratio بهتر از تقسیمی است که فقط بر اساس یک ویژگی انجام شده است. این به این معنا است که این تقسیم اطلاعات بیشتری فراهم می‌کند و همچنین گروه های خالص تری ایجاد می کند.

در نهایت از بین Gain Ratio ها Gain Ratio را انتخاب میکنیم که از همه بیشتر است.

جاهایی که Gain Ratio و Gini Index استفاده میشوند عبارتند از:

برای Gini Index:

- درخت‌های تصمیم (Decision Trees): در الگوریتم های مانند CART و ID3، Gini Index به عنوان یکی از معیارهای انتخاب ویژگی ها برای جدا کردن داده ها در هر گام از ساخت درخت تصمیم استفاده میشود.
- الگوریتم‌های ماشین لرنینگ: در مسائل طبقه‌بندی، Gini Index معمولا به عنوان یکی از معیارهای سنجش خلوص یا همگنی داده های هر دسته مورد استفاده قرار می گیرد.
- تجزیه و تحلیل اجتماعی و اقتصادی: در برخی از مسائل تحلیل اجتماعی و اقتصادی، مانند تحلیل درآمد و توزیع ثروت، Gini Index به عنوان یک معیار برای اندازه‌گیری تفاوت ها و ناهمسانی ها در توزیع داده ها استفاده میشود.

برای Gain Ratio:

- درخت‌های تصمیم (Decision Trees): مانند Gini Index، Gain Ratio نیز به عنوان یکی از معیارهای انتخاب ویژگی ها برای ساخت درخت های تصمیم در الگوریتم های مانند C4.5 استفاده میشود.
- انتخاب ویژگی ها در مسائل مهندسی ویژگی (Feature Engineering): Gain Ratio می تواند در انتخاب ویژگی های مهم و مفید برای مدل های یادگیری ماشین مورد استفاده قرار بگیرد.
- تحلیل متن: در مسائل مرتبط با پردازش زبان طبیعی و تحلیل متن مانند دسته بندی متن و تشخیص موضوع، ممکن است Gain Ratio به عنوان یکی از معیارهای انتخاب ویژگی ها برای استخراج ویژگی های مهم از متن استفاده شود.

.3

فصل	امکان مرخصی	خودروی شخصی	شهر
تابستان	بله	بله	شیراز
تابستان	نه	بله	تهران
بهار	بله	بله	شیراز
پاییز	بله	نه	شیراز
پاییز	نه	بله	اصفهان
پاییز	بله	نه	شیراز
بهار	نه	نه	شیراز
بهار	نه	بله	رشت
بهار	بله	بله	شیراز
تابستان	نه	بله	تهران

فصل:

	تابستان	بهار	پاییز
شیراز	1	3	2
تهران	2	0	0
اصفهان	0	0	1
رشت	0	1	0

$$gini\ index\left(\text{تابستان}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 - 0 - 0 = \frac{4}{9}$$

$$gini\ index\left(\text{بهار}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - 0 - 0 = \frac{6}{16}$$

$$gini\ index\left(\text{پاییز}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - 0 - 0 = \frac{4}{9}$$

$$gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} gini(i) = \frac{3}{10}\left(\frac{4}{9}\right) + \frac{4}{10}\left(\frac{6}{16}\right) + \frac{3}{10}\left(\frac{4}{9}\right) \approx 0.41$$

امکان مرخصی:

	بله	نه
شیراز	5	1
تهران	0	2
اصفهان	0	1
رشت	0	1

$$gini\ index\left(\text{بله}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{5}{5}\right)^2 - 0 - 0 - 0 = 0$$

$$gini\ index\left(\text{نه}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2 - \left(\frac{1}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = \frac{18}{25}$$

$$gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} gini(i) = \frac{5}{10}(0) + \frac{5}{10}\left(\frac{18}{25}\right) \approx 0.36$$

خودروی شخصی:

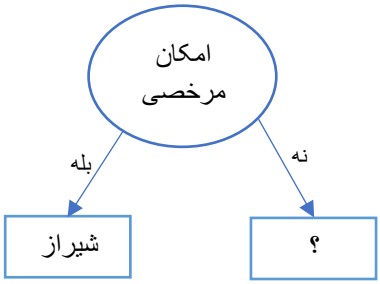
	بله	نه
شیراز	3	3
تهران	2	0
اصفهان	1	0
رشت	1	0

$$gini\ index\left(\text{نه}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{3}{3}\right)^2 - 0 - 0 - 0 = 0$$

$$gini\ index\left(\text{بله}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{2}{7}\right)^2 - \left(\frac{1}{7}\right)^2 - \left(\frac{1}{7}\right)^2 = \frac{34}{49}$$

$$gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} gini(i) = \frac{3}{10}(0) + \frac{7}{10}\left(\frac{34}{49}\right) \approx 0.49$$

حالا ویژگی انتخاب میشود که gini کمتری دارد پس ویژگی امکان مرخصی به عنوان ریشه انتخاب میشود.



فصل:

$$gini\ index\left(\text{تابستان}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

$$gini\ index\left(\text{بهار}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

$$gini\ index\left(\text{پاییز}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

$$gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} gini(i) = \frac{2}{5}(0) + \frac{2}{5}\left(\frac{1}{2}\right) + \frac{1}{5}(0) = 0.2$$

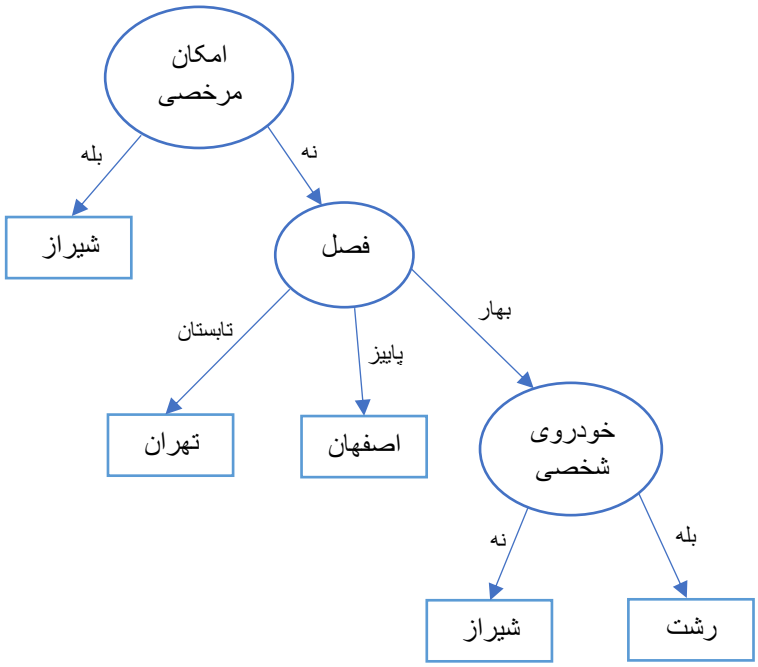
خودروی شخصی:

$$gini\ index\left(\text{نه}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{1}{1}\right)^2 - 0 - 0 - 0 = 0$$

$$gini\ index\left(\text{بله}\right) = 1 - \sum_{i=0}^{c-1} P_i(t)^2 = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = \frac{10}{16}$$

$$gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} gini(i) = \frac{1}{5}(0) + \frac{4}{5}\left(\frac{10}{16}\right) = 0.5$$

حالا ویژگی انتخاب میشود که gini کمتری دارد پس ویژگی فصل انتخاب میشود.



4.

هرس کردن یک مرحله مهم در فرآیند ساخت و بهینه سازی در درخت تصمیم است. این فرآیند به حذف برخی از شاخه های درخت که ممکن است بی اهمیت یا بی نفع باشند و همینطور به کاهش پیچیدگی مدل کمک می کند و این کار باعث میشود که از overfitting جلوگیری شود.

همین طور با استفاده از معیارهایی مانند Gini index یا Entropy می توانیم شاخه هایی که کمترین اطلاعات را فراهم می کنند را شناسایی کنیم و حذف کنیم.

اما استفاده از یک مجموعه مجزا از تاپل ها برای ارزیابی هرس دارای ایراداتی است از قبیل:

- کمبود داده: اگر داده های ما محدود باشند، اختصاص یک مجموعه مجزا برای ارزیابی هرس ممکن است به کاهش مقدار داده های موجود برای آموزش مدل منجر شود که این امر باعث میشود دقت مدل کاهش یابد.
- تعصب در ارزیابی: اگر مجموعه ارزیابی از داده هایی تشکیل شده باشد که خاصیت خاصی دارند (مثلا تمامی نمونه ها از یک کلاس خاص هستند) این می تواند باعث تعصب در ارزیابی هرس شود و نتایج غیر واقعی ارائه دهد.
- از دست دادن اطلاعات: وقتی از مجموعه مجزا از تاپل ها برای ارزیابی هرس استفاده می شود ممکن است برخی از اطلاعات مفید که می تواند به بهبود عملکرد مدل کمک کند، از دست برود به علت اینکه ما داده هایی را که می‌توانستیم برای آموزش مدل استفاده کنیم، برای ارزیابی مدل استفاده کردیم.
- اعتبارسنجی نامعتبر: اگر مجموعه ارزیابی ما نماینده‌ی مناسبی از داده هایی که مدل در زمان واقعی با آنها مواجه میشود نباشد، ممکن است اعتبارسنجی نامعتبری ایجاد کند به عبارت دیگر، اگر مجموعه ارزیابی ما توزیع یا خصوصیات متفاوتی نسبت به داده های آموزش داشته باشد نتایج ارزیابی ممکن است نادرست باشد.
- تفاوت توزیع بین داده های آموزش و داده های ارزیابی: اگر توزیع داده های آموزش با داده های ارزیابی متفاوت باشد، ممکن است مدل بر روی داده های آموزش overfitting کند یعنی مدل ممکن است الگوهایی را یاد بگیرد که فقط برای داده های آموزش صادق است و به داده های جدید، خوب تعمیم داده نشود.

.5

.1

5-1

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[152] import pandas as pd
```

```
dataframe=pd.read_csv('/content/drive/MyDrive/Social_Network_Ads.csv')
dataframe.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

Next steps: View recommended plots

.2

5-2

```
[154] from sklearn.preprocessing import LabelEncoder
```

```
[155] label_encoder = LabelEncoder()
dataframe['Gender'] = label_encoder.fit_transform(dataframe['Gender'])
dataframe.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15668575	0	26	43000	0
3	15603246	0	27	57000	0
4	15804002	1	19	76000	0

Next steps: View recommended plots

5-3

```
[156] x = dataframe.iloc[:, 1:4]
      y = dataframe.iloc[:,4:5]
```

+ Code

```
[157] print("X variable:\n",x)
      print("\nY variable:\n",y)
```

```
X variable:
      Gender  Age  EstimatedSalary
0          1   19            19000
1          1   35            20000
2          0   26            43000
3          0   27            57000
4          1   19            76000
..       ...  ...              ...
395        0   46            41000
396        1   51            23000
397        0   50            20000
398        1   36            33000
399        0   49            36000

[400 rows x 3 columns]
```

```
396        1   51            23000
397        0   50            20000
398        1   36            33000
399        0   49            36000

[400 rows x 3 columns]

Y variable:
      Purchased
0             0
1             0
2             0
3             0
4             0
..          ...
395           1
396           1
397           1
398           0
399           1

[400 rows x 1 columns]
```

.4

5-4

```
[158] from sklearn.model_selection import train_test_split
```

```
[159] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
```

```
[160] print("ابعاد مجموعه آموزشی:")
      print(x_train.shape, y_train.shape)
      print("ابعاد مجموعه تست:")
      print(x_test.shape, y_test.shape)
```

```
ابعاد مجموعه آموزشی:
(280, 3) (280, 1)
ابعاد مجموعه تست:
(120, 3) (120, 1)
```

.5

5-5

```
[161] from sklearn.tree import DecisionTreeClassifier
```

```
[162] classifier = DecisionTreeClassifier(random_state=0)
```

```
[163] classifier.fit(x_train,y_train)
```

```
DecisionTreeClassifier(random_state=0)
```

.6

5-6

```
[164] y_pred_test = classifier.predict(x_test)

      y_pred_train = classifier.predict(x_train)
```

```
print("y_pred_test:\n",y_pred_test)

print("\n**********************************************************************************\n")

print("y_pred_train:\n",y_pred_train)
```

```
y_pred_test:
 [0 0 0 1 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0
 0 1 0 1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 1 0 1 1 0 0 0 1 1
 0 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 0
 0 0 0 1 0 0 0 0 1]

**********************************************************************************

y_pred_train:
 [0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0
```

```
y_pred_test:
 [0 0 0 1 1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 0
 0 1 0 1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 1 0 1 1 0 0 0 1 1
 0 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 0
 0 0 0 1 0 0 0 0 1]

**********************************************************************************

y_pred_train:
 [0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0
 0 0 1 0 1 1 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 1 0
 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 1 0 1 1
 0 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 0
 1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 1 0
 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 1 1 0 0 1
 1 1 0 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 1 1 0 0 1 0 0 1 0 1 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0]
```

.7

5-7

```
[166] from sklearn.metrics import accuracy_score
```

```
[167] accuracy_test = accuracy_score(y_test, y_pred_test)

      accuracy_train = accuracy_score(y_train, y_pred_train)
```

```
print("Train Accuracy:", accuracy_train)

print("\nTest Accuracy:", accuracy_test)
```

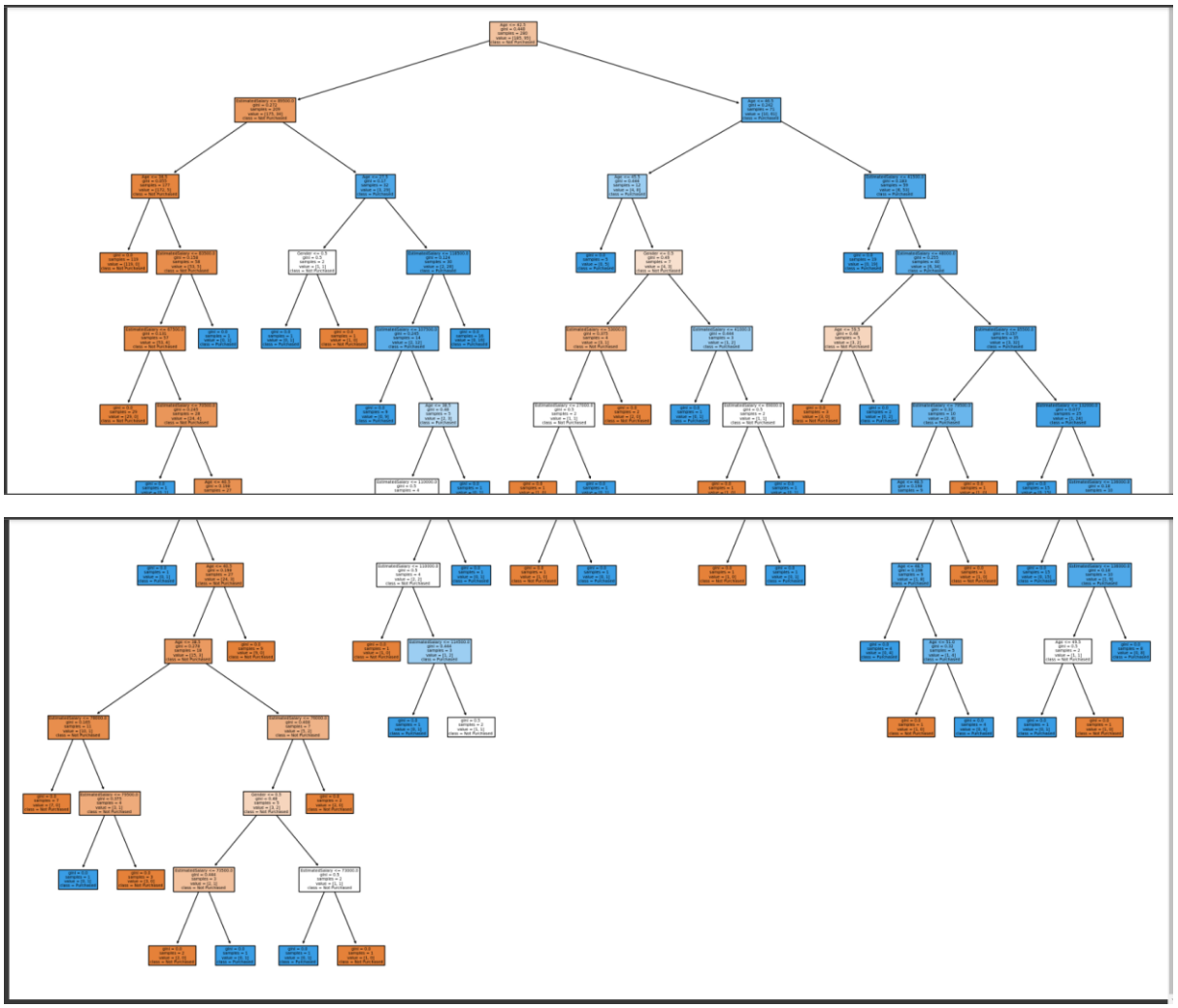
```
Train Accuracy: 0.9964285714285714

Test Accuracy: 0.8333333333333334
```

5-8

```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(25, 20))
plot_tree(classifier, filled=True, feature_names=x.columns, class_names=["Not Purchased", "Purchased"])
plt.show()
```



.9

.10

پارامترهای DecisionTreeClassifier:

- criterion: این پارامتر معیاری را تعیین می کند که برای اندازه‌گیری کیفیت یک شاخه استفاده می شود. مقادیر ممکن عبارتند از gini برای اندازه‌گیری ناخالصی gini و entropy برای اندازه‌گیری اطلاعات انتروپی. همچنین این معیار تعیین می کند که چگونه یک گره بر اساس یک ویژگی خاص تقسیم شود.
- splitter: این پارامتر استراتژی را برای انتخاب شاخه در هر گره را تعیین می کند. مقادیر ممکن عبارتند از best برای انتخاب بهترین شاخه و random برای انتخاب شاخه به صورت تصادفی. همچنین این معیار تعیین می کند که چگونه درخت تصمیم بر اساس ویژگی ها شکافته شود.
- max_depth: این پارامتر عمق حداکثری درخت را تعیین می کند. اگر این مقدار None باشد درخت به صورت کامل رشد می کند تا زمانی که همه گره ها خالص شوند یا تا زمانی که تعداد نمونه ها کمتر از min_samples_split شود. همچنین این معیار تعیین می کند که چقدر درخت تصمیم می تواند عمیق شود. مثلا اگر max_depth بیش از حد بزرگ شود مدل ممکن است دچار overfitting شود و از طرف دیگر اگر max_depth بیش از حد کم باشد مدل ممکن است دچار underfitting شود.
- min_samples_split: این پارامتر حداقل تعداد نمونه های لازم برای شکافتن یک گره را تعیین می کند. اگر این مقدار بیش از حد بزرگ باشد، مدل ممکن است underfitting شود.
- min_samples_leaf: این پارامتر حداقل تعداد نمونه های لازم در یک گره برگ را تعیین می کند. اگر این مقدار بیش از حد بزرگ باشد، مدل ممکن است underfitting شود.
- max_features: این پارامتر تعداد حداکثری ویژگی ها را تعیین می کند که برای بهترین تقسیم در نظر گرفته می شوند. اگر این مقدار کم باشد، مدل ممکن است underfitting شود.

- random_state: این پارامتر بذر را برای ژنراتور تصادفی تعیین می کند، که برای شکافتن گره ها در حالتی که splitter=random استفاده می شود.
- max_leaf_nodes: با تعیین این پارامتر، می توانیم تعداد حداکثری گره های برگ را محدود کنیم. این می‌تواند به کاهش overfitting کمک کند.

.11

5-11

```
[174] from sklearn.model_selection import GridSearchCV
      from sklearn.model_selection import KFold
```

```
CV = KFold(n_splits=10, random_state=100, shuffle=True)

def GrdSrch_Tune(model, X, y, params):
    clf = GridSearchCV(model, params, scoring ='accuracy', cv = CV, n_jobs=-1)
    clf.fit(X, y)

    print("best score is :" , clf.best_score_)
    print("best estimator is :" , clf.best_estimator_)
    print("best Params is :" , clf.best_params_)

    return (clf.best_score_)
```

```
[176] param_DT = {'criterion':['gini', 'entropy', 'log_loss'] ,
                  'splitter':['best', 'random'] ,
                  'max_depth':[None,3,4,5,6] ,
                  'min_samples_split':[2,3,4,5,6] ,
                  'min_samples_leaf':[2,3,4,5,6]}


GrdSrch_Tune(DecisionTreeClassifier(random_state = 0), x_train, y_train, param_DT)
```

```
[176] param_DT = {'criterion':['gini', 'entropy', 'log_loss'] ,
                  'splitter':['best', 'random'] ,
                  'max_depth':[None,3,4,5,6] ,
                  'min_samples_split':[2,3,4,5,6] ,
                  'min_samples_leaf':[2,3,4,5,6]}


GrdSrch_Tune(DecisionTreeClassifier(random_state = 0), x_train, y_train, param_DT)
```

```
best score is : 0.9357142857142856
best estimator is : DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=6,
                       random_state=0)
best Params is : {'criterion': 'entropy', 'max_depth': 3, 'min_samples_leaf': 6, 'min_samples_split': 2, 'splitter': 'best'}
0.9357142857142856
```

```
[177] classifier2 = DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=6,random_state=0)

classifier2.fit(x_train, y_train)

y_pred_train = classifier2.predict(x_train)

y_pred_test = classifier2.predict(x_test)

accuracy_train = accuracy_score(y_train, y_pred_train)

accuracy_test = accuracy_score(y_test, y_pred_test)

print("Train Accuracy:", accuracy_train)
print("\nTest Accuracy:", accuracy_test)
```

```
[177] classifier2 = DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=6,random_state=0)

      classifier2.fit(x_train, y_train)

      y_pred_train = classifier2.predict(x_train)

      y_pred_test = classifier2.predict(x_test)

      accuracy_train = accuracy_score(y_train, y_pred_train)

      accuracy_test = accuracy_score(y_test, y_pred_test)

      print("Train Accuracy:", accuracy_train)
      print("\nTest Accuracy:", accuracy_test)

      Train Accuracy: 0.9357142857142857

      Test Accuracy: 0.8583333333333333
```