# پروژه درس داده کاوی

## ( مقایسه الگوریتم ID3 و CART )

عنوان مقاله:

مقایسه الگوریتم ID3 و CART

مقدمه:

در پژوهش زیر قصد داریم الگوریتم ID3 و CART را بر روی مجموعه داده IRIS بررسی کنیم و تفاوت استفاده از این دو روش را بیابیم.

در ابتدا به معرفی درخت تصمیم و توضیح گره و برگ میپردازیم سپس مزایا و معایب هر یک از دو الگوریتم ذکرشده را بررسی میکنیم. در ادامه نتایج بدست آمده با استفاده از برنامه پایتون را تفسیر و تحلیل میکنیم.

بدنه تحقیق:

درخت تصمیم و نحوه عملکرد آن:

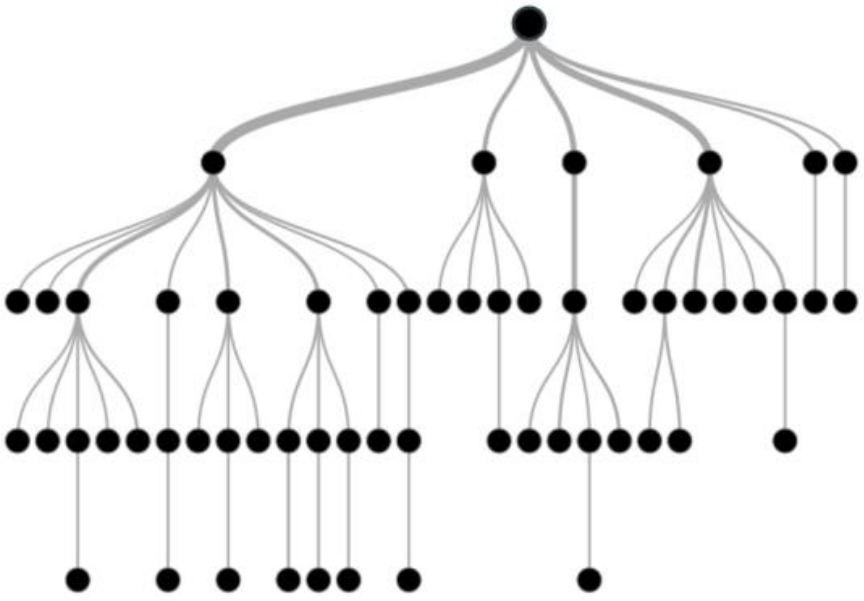
درخت تصمیم (Decision Tree) یک روش رایج برای نشان دادن فرآیند تصمیم‌گیری به وسیله ساختاری درخت مانند و شاخه‌دار است. این روش یکی از رویکردهای دسته‌بندی (Classification) و رگرسیون (Regression)در یادگیری ماشین به حساب می‌آید.

درخت تصمیم، روشی در یادگیری ماشین برای ساختاربندی به الگوریتم است. یک الگوریتم درخت تصمیم برای تقسیم ویژگی‌های مجموعه داده از طریق تابع هزینه (Cost Function) مورد استفاده قرار می‌گیرد. این الگوریتم قبل از انجام بهینه‌سازی و حذف شاخه‌های اضافه، به گونه‌ای رشد می‌کند که دارای ویژگی‌های نامرتبط با مسئله است؛ به همین دلیل، عملیات هرس کردن (Pruning) برای حذف این شاخه‌های اضافه در

آن انجام می‌شود. در الگوریتم درخت تصمیم، پارامترهایی ازجمله عمق درخت تصمیم را نیز می‌توان تنظیم کرد تا از بیش‌برازش یا پیچیدگی بیش از حد درخت تا جای امکان جلوگیری شود.

درخت تصمیم در قالب مدل سازی پیش بینی کننده، به نگاشت تصمیم ها یا راه حل های مختلف برای بدست آوردن خروجی کمک میکند. درخت تصمیم از گره های محتلفی ایجاد شده است. گره ریشه محل شروع درخت تصمیم به حساب می آید که معمولا تمام مجموعه داده مسیله را شامل میشود. گره های برگ نقطه پایانی هر شاخه درخت یا خروجی نهایی مجموعه ای از تصمیم ها هستند. هر شاخه درخت تصمیم در یادگیری ماشین فقط دارای یک گره برگ است.

در درخت تصمیم، ویژگی داده ها در گره های داخلی شاخه ها و نتیجه آن ها در برگ هر شاخه نشان داده میشود. به دلیل اینکه درخت تصمیم ساختار ساده ای در نشان دادن یک مدل دارد، بسیار محبوب است. شکل زیر ساختار کلی یک درخت تصمیم با گره ها و برگ های آن را نشان میدهد.



شکل1- ساختار درخت تصمیم

**درخت تصمیم چگونه کار می کند؟**

در درخت تصمیم برای پیش‌بینی کلاس‌های مورد نظر مجموعه داده مسئله، رویکرد الگوریتم از گره ریشه درخت آغاز می‌شود. این الگوریتم، مقادیر ویژگی‌های ریشه را با ویژگی‌های داده‌ها مقایسه و بر اساس این مقایسه، شاخه‌ها را دنبال می‌کند و به گره بعدی می‌رود. برای گره بعدی، الگوریتم دوباره مقدار ویژگی داده‌ها را با زیر گره‌های دیگر مقایسه می‌کند و روند ایجاد درخت را پیش می‌برد. این رویکرد تا رسیدن به گره برگ یا گره انتهایی درخت ادامه پیدا می‌کند. فرآیند کامل روش کار کردن درخت تصمیم را می‌توان با ارائه آن به صورت الگوریتم زیر بهتر درک کرد:

- مرحله اول :شروع روند کار الگوریتم درخت تصمیم از گره ریشه آغاز می‌شود که شامل مجموعه داده کامل مسئله است.
- مرحله دوم :با استفاده از روش سنجیدن انتخاب ویژگی (Attribute Selection Measure ) بهترین ویژگی در مجموعه داده انتخاب می‌شود.
- مرحله سوم :تقسیم کردن گره ریشه به زیرمجموعه‌هایی که شامل مقادیر مناسب و ممکن برای بهترین ویژگی‌ها باشند.
- مرحله چهارم :تولید گره درخت تصمیمی که شامل بهترین ویژگی‌ها باشد.
- مرحله پنجم :با استفاده از زیرمجموعه‌های ایجاد شده از مجموعه داده در مرحله سوم این رویکرد، درخت‌های تصمیم جدید به صورت بازگشتی ایجاد می‌شوند. این روند تا جایی ادامه دارد که دیگر نمی‌توان گره‌ها را بیشتر طبقه‌بندی کرد و گره نهایی به عنوان گره برگ یا انتهایی به دست می‌آید.

روش سنجش انتخاب ویژگی درخت تصمیم در یادگیری ماشین چیست؟

در زمان پیاده‌سازی درخت تصمیم ، یکی از مهم‌ترین و اساسی‌ترین مسائلی که پیش می‌آید این است که بهترین ویژگی برای گره ریشه و گره‌های فرعی دیگر چگونه انتخاب شود؟ بنابراین برای حل چنین مسائلی، روشی وجود دارد که به آن معیار یا سنجش انتخاب ویژگی (Attribute Selection Measure) یا «ASM»گفته می‌شود. با این روش می‌توان به راحتی بهترین ویژگی را برای گره ریشه و دیگر گره‌های درخت انتخاب کرد. روش سنجش انتخاب ویژگی دارای دو رویکرد رایج به نام‌های زیر است:

بهره اطلاعاتی(Information Gain)

شاخص جینی(Gini Index)

چه زمانی تقسیم شدن انشعاب های درخت تصمیم متوقف می شوند؟

از آنجایی که معمولاً یک مسئله دارای مجموعه داده بزرگی است و این حجم بالای داده‌ها باعث ایجاد تعداد تقسیم‌بندی و انشعاب‌های بالایی می‌شود، درخت بزرگ و پیچیده‌ای به وجود می‌آید. چنین درختانی باعث ایجاد بیش‌برازش خواهند شد، بنابراین زمان توقف تقسیم شاخه‌های درخت باید بررسی و مشخص شود. روش‌های جلوگیری از بیش‌برازش در این حالت و زمان توقف تقسیم‌بندی شاخه‌های درخت تصمیم در ادامه ارائه شده‌اند:

- یکی از روش‌های انجام این کار، تنظیم حداقل تعداد ورودی‌های آموزشی برای استفاده در هر برگ است. برای مثال فقط می‌توان ۱۰ مسافر برای تصمیم‌گیری در مسئله زنده یا مرده بودن استفاده کرد و هر برگی با کمتر از ۱۰ مسافر را نادیده گرفت.
- روش دیگر، تنظیم حداکثر عمق مدل است. حداکثر عمق به طول طولانی‌ترین مسیر از یک رشته تا یک برگ اشاره دارد.

هرس کردن درخت تصمیم چیست؟

کارایی درخت تصمیم در یادگیری ماشین می‌تواند با استفاده از روش‌های هرس کردن افزایش پیدا کند. هرس کردن به معنی حذف شاخه‌هایی است که دارای ویژگی‌هایی با اهمیت کمتر در هدف مسئله هستند. با استفاده از این روش می‌توان پیچیدگی درخت را کاهش داد و سپس قدرت و دقت پیش‌بینی الگوریتم با کاهش بیش‌برازش افزایش پیدا خواهد کرد. هرس کردن می‌تواند از ریشه یا برگ‌ها آغاز شود.

الگوریتم درخت تصمیم ID3 چیست؟

الگوریتم ID3 در سال ۱۳۶۵ شمسی (۱۹۸۶ میلادی) توسط «Ross Quinlan» توسعه یافته است. این الگوریتم یک درخت چند مسیره (Multiway) ایجاد می‌کند و برای هر گره مانند الگوریتم‌های حریصانه ویژگی گسسته و گروهی پیدا خواهد کرد. این نوع از الگوریتم‌های درخت تصمیم در یادگیری ماشین بیشترین اطلاعات را در مسائلی با اهداف گسسته به دست می‌آورند. معمولاً درخت‌ها به بزرگترین اندازه ممکن خود در مسائل رشد پیدا می‌کنند، سپس یک مرحله هرس برای بهبود عملکرد و توانایی درخت روی داده‌های درخت انجام می‌شود.

الگوریتم ID3 یک روش برای ساخت درخت تصمیم‌گیری در یادگیری ماشین است که به ما کمک می‌کند داده‌ها را به دسته‌های مختلف تقسیم کنیم. این الگوریتم با شروع از یک مجموعه داده، در هر مرحله بهترین ویژگی را انتخاب می‌کند که بیشترین اطلاعات را برای تقسیم داده‌ها فراهم کند. به این ترتیب، داده‌ها به زیرمجموعه‌های کوچکتری تقسیم می‌شوند تا زمانی که همه داده‌های هر زیرمجموعه به یک دسته خاص تعلق داشته باشند.

مزایای ID3 :

سادگی و وضوح :الگوریتم ID3 به دلیل ساختار ساده و قابل فهم خود، به راحتی قابل پیاده‌سازی و توضیح است.

استفاده از آنتروپی :با استفاده از آنتروپی برای محاسبه همگنی، ID3می‌تواند ویژگی‌هایی را که بیشترین اطلاعات را ارائه می‌دهند، شناسایی کند.

ساختار درختی :درخت تصمیم تولید شده توسط ID3 به راحتی قابل تفسیر است و می‌تواند به عنوان یک مدل بصری برای تصمیم‌گیری استفاده شود.

کارایی در داده‌های کوچک ID3 :معمولاً در مجموعه‌های داده کوچک و با ویژگی‌های گسسته عملکرد خوبی دارد.

عدم نیاز به پیش‌پردازش پیچیده :این الگوریتم به طور معمول نیاز به پیش‌پردازش پیچیده داده‌ها ندارد و می‌تواند به سادگی بر روی داده‌های خام اعمال شود.

با این حال، باید توجه داشت که ID3 دارای محدودیت‌هایی نیز هست، به ویژه در مدیریت مقادیر گمشده و ویژگی‌های پیوسته، که در الگوریتم‌های پیشرفته‌تر مانند C4.5 بهبود یافته است.

معایب ID3:

عدم توانایی در مدیریت مقادیر گمشده :الگوریتم ID3 نمی‌تواند به خوبی با داده‌های دارای مقادیر گمشده کار کند، که می‌تواند منجر به کاهش دقت مدل شود.

عدم توانایی در پردازش ویژگی‌های پیوسته ID3 :به طور عمده برای ویژگی‌های گسسته طراحی شده است و در پردازش ویژگی‌های پیوسته محدودیت دارد.

حساسیت به داده‌های نویزی ID3 :ممکن است به داده‌های نویزی حساس باشد و در نتیجه درخت تصمیم تولید شده ممکن است به شدت تحت تأثیر داده‌های غیرمعمول قرار گیرد.

تمایل به درختان عمیق ID3 :ممکن است درختان تصمیم عمیق و پیچیده‌ای تولید کند که منجر به بیش‌برازش می‌شود، به ویژه در مجموعه‌های داده بزرگ.

عدم توانایی در تعمیم : به دلیل ساختار درختی و پیچیدگی ممکن، ID3ممکن است در تعمیم به داده‌های جدید ضعیف عمل کند.

این معایب باعث شده‌اند که الگوریتم‌های پیشرفته‌تری مانند C4.5 و CART توسعه یابند که این محدودیت‌ها را برطرف کنند.

الگوریتم درخت تصمیم CART چیست؟

الگوریتم CART توسط دو محقق به نام‌های Leo Breiman و همکاران در سال 1986 معرفی شد. هدف از توسعه این الگوریتم ساخت درخت‌هایی بود که بتوانند داده‌ها را به صورت موثر و کارآمد طبقه‌بندی یا پیش‌بینی کنند CART .مبنای بسیاری از الگوریتم‌های دیگر یادگیری ماشین است. الگوریتم CART یکی از تکنیک‌های محبوب در یادگیری ماشین است که برای مسائل طبقه‌بندی و رگرسیون مورد استفاده قرار می‌گیرد. این الگوریتم به دلیل سادگی، قابلیت تفسیر و عملکرد خوب در بسیاری از کاربردها، بسیار مورد توجه قرار گرفته است.

الگوریتم CART از متغیرهای هدف پیوسته و عددی یا همان مسائل رگرسیون نیز پشتیبانی می‌کند و مجموعه قوانین را محاسبه نمی‌کند. این الگوریتم با استفاده از ویژگی و آستانه‌ای که بیشترین اطلاعات را در هر گره ایجاد می‌کند، درخت‌های دودویی را می‌سازد.

ساختارCART:

درخت تصمیم‌گیری CART یک ساختار درختی است که شامل گره‌ها و لبه‌ها می‌باشد:

گره‌ها :نمایانگر ویژگی‌ها یا صفات داده‌ها هستند.

لپه‌ها :نمایانگر تصمیمات یا پیش‌بینی‌ها هستند.

درخت CART از دو نوع گره تشکیل شده است:

گره‌های داخلی :که نشان‌دهنده ویژگی‌هایی هستند که برای تقسیم داده‌ها استفاده می‌شوند.

گره‌های برگ :که نشان‌دهنده تولید پیش‌بینی نهایی هستند.

روند کار الگوریتم:

روند کار الگوریتم CART به صورت زیر است:

انتخاب بهترین ویژگی و آستانه :در هر گره، بهترین ویژگی و آستانه‌ای که داده‌ها را به دو گروه تقسیم کند، پیدا می‌شود. این انتخاب با استفاده از معیار جینی انجام می‌شود.

تقسیم داده‌ها :داده‌ها بر اساس ویژگی و آستانه انتخاب‌شده به دو زیرمجموعه تقسیم می‌شوند.

مدیریت عمق درخت :این روند تا رسیدن به شرایط توقف مانند حداکثر عمق، تعداد کم نمونه‌ها در گره‌ها یا عدم کاهش بیشتر انحراف معیار ادامه می‌یابد.

پیش‌بینی :برای پیش‌بینی، ورودی جدید به درخت هدایت شده و گام به گام از گره‌های داخلی به سمت گره برگ که پیش‌بینی نهایی است، دنبال می‌شود.

مزایایCART:

قابلیت تفسیر :درخت‌های تصمیم به راحتی قابل تفسیر و بصری‌سازی هستند، که باعث می‌شود تصمیمات قابل توضیح باشند.

عدم نیاز به پیش‌پردازش زیاد داده‌ها CART : می‌تواند با داده‌های گسسته و پیوسته کار کند بدون اینکه نیاز به نرمال‌سازی یا استانداردسازی داده‌ها باشد.

عملکرد خوب در اندازه‌های مختلف داده :این الگوریتم در حجم‌های مختلف داده‌ها به خوبی عمل می‌کند.

معایبCART :

حساسیت به داده‌های نویزی :درخت‌های تصمیم به داده‌های نویزی حساس‌اند و ممکن است به راحتی دچار بیش‌برازش شوند.

عدم قابلیت تعمیم خوب :اگر عمق درخت خیلی زیاد باشد، مدل ممکن است به داده‌های آموزشی بیش‌برازش شده و عملکرد ضعیفی روی داده‌های تست داشته باشد.

در این پژوهش از مجموعه داده معروف IRIS استفاده خواهیم کرد تا دو الگوریتم ID3 و CART را با هم مقایسه کنیم.

مجموعه داده‌های آیریس یک مجموعه کلاسیک برای طبقه‌بندی، یادگیری ماشین و تجزیه و تحلیل داده‌هاست.

این مجموعه داده شامل: ۳ کلاس (گونه‌های مختلف آیریس) با ۵۰ نمونه هر کدام و همچنین چهار ویژگی عددی در مورد این کلاس‌ها است: طول کاسبرگ، عرض کاسبرگ، طول گلبرگ و عرض گلبرگ.

یک گونه، آیریس ستوسا، از دو گونه دیگر "خطی جداشدنی" است. این به این معنی است که می‌توانیم یک خط (یا یک ابرصفحه در فضاهای با ابعاد بالاتر) بین نمونه‌های آیریس ستوسا و نمونه‌های مربوط به دو گونه دیگر بکشیم.

به منظور اجرای هدف مورد نظر با استفاده از نرم افزار پایتون کتابخانه های مورد نیاز را فراخوانی میکنیم سپس داده ها را بازخوانی کرده و آنها را به دو بخش آموزش و تست تقسیم میکنیم. مدل را بر روی داده های آموزش برازش میکنیم و کارایی آن را بررسی میکنیم سپس بر روی داده های تست اعمال کرده و نتایج را بررسی میکنیم.

```
import numpy as np

import pandas as pd

from sklearn.datasets import load_iris
```

```
from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn import tree

import matplotlib.pyplot as plt
```

#بارگذاری داده‌های Iris

```
iris = load_iris()

X = iris.data

y = iris.target
```

#تبدیل به DataFrame

```
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

iris_df['species'] = iris.target
```

#نمایش 5 ردیف اول

```
   sepal length (cm)  sepal width (cm)  petal length
(cm)  petal width (cm)   \
0                5.1               3.5
1.4               0.2
1                4.9               3.0
1.4               0.2
2                4.7               3.2
1.3               0.2
```

```
3                  4.6                3.1
1.5                 0.2
4                  5.0                3.6
1.4                 0.2

   species
0        0
1        0
2        0
3        0
4        0
```

sepal length (cm), sepal width (cm), petal length (cm), petal width (cm)مشخصات فیزیکی گل‌ها هستند.

- Species نوع گل را نشان می‌دهد که به صورت عددی کدگذاری شده است0 ، 1 و 2 برای سه نوع مختلف گل

```
print(iris_df.head())
```

#تقسیم داده‌ها به داده‌های آموزشی و تست

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

#ایجاد مدل درخت تصمیم با الگوریتم CART

```
model = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=42)
```

#آموزش مدل

```
model.fit(X_train, y_train)
```

DecisionTreeClassifier
```
DecisionTreeClassifier(max_depth=3, random_state=42)
```

#پیش‌بینی با استفاده از داده‌های تست

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

#محاسبه دقت مدل

```
accuracy = np.mean(y_pred == y_test)

print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 1.00
```

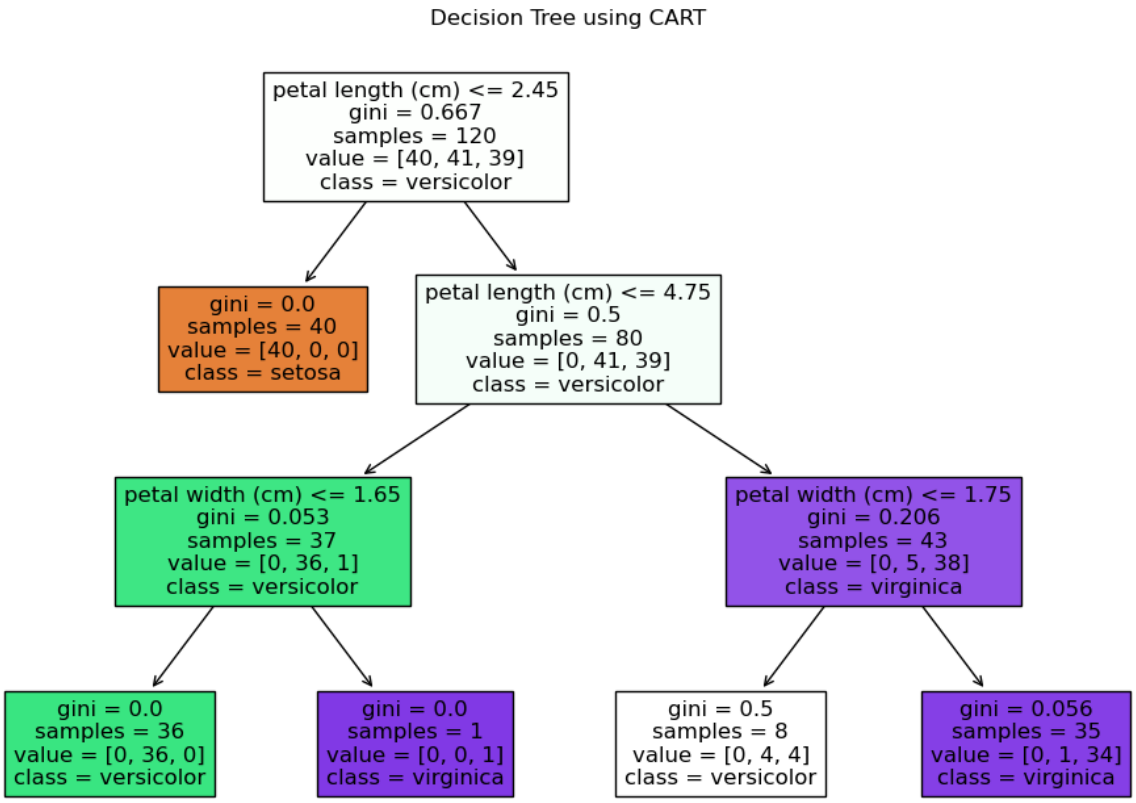
#نمایش درخت تصمیم

```
plt.figure(figsize=(12,8))

tree.plot_tree(model, filled=True, feature_names=iris.feature_names,
class_names=iris.target_names)

plt.title("Decision Tree using CART")
```

```
plt.show()
```

- توضیحات کد
- ابتدا ما کتابخانه‌های مورد نیاز را وارد می‌کنیم.
- داده‌های Irisرا بارگذاری می‌کنیم و به دو دسته داده‌های آموزشی و تست تقسیم می‌کنیم.
- یک مدل درخت تصمیم با استفاده از DecisionTreeClassifierاز scikit-learnایجاد می‌کنیم و آن را آموزش می‌دهیم.
- سپس با استفاده از داده‌های تست پیش‌بینی می‌کنیم و دقت مدل را محاسبه می‌کنیم.
- در نهایت، درخت تصمیم را به صورت بصری نشان می‌دهیم.

ویژگی پیش‌بینی شده: گونه‌های مختلف گیاه آیریس.




شکل2- ساختار درخت تصمیم با الگوریتم CART

این تصویر یک درخت تصمیم (Decision Tree) را با استفاده از الگوریتم CART نشان می‌دهد. در اینجا توضیحات مربوط به هر بخش درخت آورده شده است:

1. گام اول (پایه درخت):
   - طول گلبرگ:2.45 <= (cm)
     - مقدار Gini: 0.667
     - تعداد نمونه‌ها: 120
     - کلاس: 40 نمونه از نوع setosa، 0 نمونه از نوع versicolorو 39 نمونه از نوع virginica.

2. گام دوم:

- اگر طول گلبرگ 2.45 =>، به دو شاخه تقسیم می‌شود:
  - طول گلبرگ:4.75 => (cm)
    - مقدار0.5 :Gini
    - تعداد نمونه‌ها: 80
    - کلاس: 40 نمونه از نوع setosaو 40 نمونه از نوع .versicolor
  - عرض گلبرگ:1.65 => (cm)
    - مقدار0.053 :Gini
    - تعداد نمونه‌ها: 37
    - کلاس: 36 نمونه از نوع versicolorو 1 نمونه از نوع .virginica

3. گام سوم:

- اگر عرض گلبرگ :1.75 =>
  - مقدار0.5 :Gini
  - تعداد نمونه‌ها: 43
  - کلاس: 43 نمونه از نوع .versicolor
- اگر عرض گلبرگ :1.75 <
  - مقدار0.5 :Gini
  - تعداد نمونه‌ها: 35
  - کلاس: 35 نمونه از نوع .virginica

نکات کلیدی:

- مقدار :Gini معیاری برای سنجش خلوص گروه‌ها در درخت تصمیم است. مقدار نزدیک به 0 نشان‌دهنده خلوص بالای کلاس‌ها است.
- تعداد نمونه‌ها :نشان‌دهنده تعداد داده‌هایی است که در هر گام مورد بررسی قرار می‌گیرند.
- کلاس‌ها :نشان‌دهنده نوع گلها setosa، versicolor، virginicaهستند که در این درخت تصمیم طبقه‌بندی شده‌اند.

این درخت تصمیم به ما کمک می‌کند تا با استفاده از ویژگی‌های گلبرگ‌ها، نوع گل را پیش‌بینی کنیم.

ایجاد مدل درخت تصمیم با الگوریتم ID3

در اینجا، ما از DecisionTreeClassifier استفاده خواهیم کرد و از entropy به عنوان معیار تقسیم‌شده استفاده می‌کنیم تا عملکردی مشابه ID3 داشته باشیم.

#ایجاد مدل درخت تصمیم با الگوریتم ID3 با استفاده از entropy

```
model_id3 = DecisionTreeClassifier(criterion='entropy', max_depth=3,
random_state=42)
```

#آموزش مدل

```
model_id3.fit(X_train, y_train)
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=42)
```

#پیش‌بینی با استفاده از داده‌های تست

```
y_pred_id3 = model_id3.predict(X_test)
```

#محاسبه دقت مدل

```
accuracy_id3 = np.mean(y_pred_id3 == y_test)
print(f"Accuracy (ID3): {accuracy_id3:.2f}")
```

```
Accuracy (ID3): 1.00
```

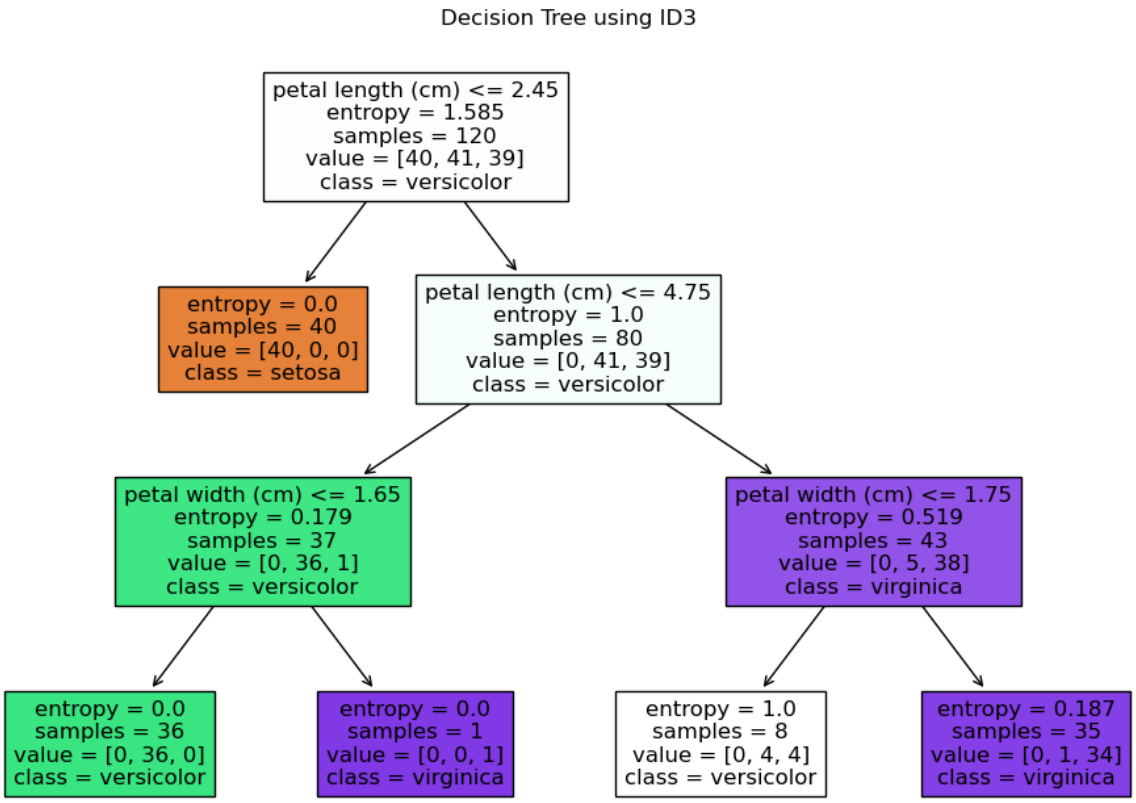
#نمایش درخت تصمیم

```
plt.figure(figsize=(12,8))
tree.plot_tree(model_id3, filled=True, feature_names=iris.feature_names,
class_names=iris.target_names)
plt.title("Decision Tree using ID3")
plt.show()
```

توضیحات کد:

- در این کد، به همین صورت که پیش از این توضیح داده شد، داده‌های Iris بارگذاری می‌شود و به مجموعه‌های آموزش و تست تقسیم می‌شود.
- سپس، یک درخت تصمیم با استفاده از معیار entropy برای پیاده‌سازی الگوریتم ID3 ساخته می‌شود.
- مدل آموزش داده می‌شود، و پیش‌بینی‌ها و دقت مدل محاسبه می‌شود.
- در نهایت، درخت تصمیم به صورت بصری نمایش داده می‌شود.

با این تنظیمات، عملکرد مشابه ID3 با استفاده از scikit-learn بدست می‌آید.




شکل3- ساختار درخت تصمیم با الگوریتم ID3

این تصویر یک درخت تصمیم (Decision Tree) را با استفاده از الگوریتم ID3 نشان می‌دهد. در اینجا توضیحات مربوط به هر بخش درخت آورده شده است:

1. **گام اول (پایه درخت):**
   - **طول گلبرگ:2.45 <= (cm)**
     - **انتروپی)** 0.0 :این نشان‌دهنده خلوص کامل است(
     - **تعداد نمونه‌ها**40 :
     - **کلاس** 40 :نمونه از نوع **setosa**.
2. **گام دوم:**

- اگر طول گلبرگ 2.45 =>، به دو شاخه تقسیم می‌شود:
  - **طول گلبرگ:4.75 => (cm)**
    - **انتروپی**0.0 :
    - **تعداد نمونه‌ها**40 :
    - **کلاس** 40 :نمونه از نوع .**versicolor**
  - **عرض گلبرگ:1.65 => (cm)**
    - **انتروپی**0.179 :
    - **تعداد نمونه‌ها**37 :
    - **کلاس** 36 :نمونه از نوع **versicolor**و 1 نمونه از نوع .**virginica**

3. **گام سوم:**

- اگر عرض گلبرگ :1.75 =>
  - **انتروپی**0.0 :
  - **تعداد نمونه‌ها**43 :
  - **کلاس** 43 :نمونه از نوع .**versicolor**
- اگر عرض گلبرگ :1.75 >
  - **انتروپی**0.187 :
  - **تعداد نمونه‌ها**35 :
  - **کلاس** 35 :نمونه از نوع .**virginica**

**نکات کلیدی:**

- **انتروپی** :معیاری برای سنجش عدم قطعیت یا بی‌نظمی در داده‌ها است. مقدار نزدیک به 0 نشان‌دهنده خلوص بالای کلاس‌ها است.
- **تعداد نمونه‌ها** :نشان‌دهنده تعداد داده‌هایی است که در هر گام مورد بررسی قرار می‌گیرند.

- **کلاس‌ها** :نشان‌دهنده نوع گل‌ها setosa ، versicolor، virginica هستند که در این درخت تصمیم طبقه‌بندی شده‌اند.

این درخت تصمیم به ما کمک می‌کند تا با استفاده از ویژگی‌های گلبرگ‌ها، نوع گل را پیش‌بینی کنیم.

بحث و نتیجه گیری:

با توجه به شاخص‌های عملکردی، CART به طور کلی به عنوان الگوریتم بهتری شناخته می‌شود، به ویژه در شرایطی که داده‌ها پیچیده و بزرگ هستند. این الگوریتم به دلیل دقت بالاتر، کنترل بهتر بر روی overfitting و قابلیت کار با داده‌های عددی و کیفی، معمولاً انتخاب بهتری است.لذا با توجه به شاخص جینی و انتروپی، شاخص جینی کارکرد بهتری دارد پس در این پژوهش الگوریتم CART مناسب تر است.

در مقابل، ID3 می‌تواند در شرایط ساده و برای آموزش‌های اولیه مناسب باشد، اما در مقایسه با CART در شرایط پیچیده‌تر عملکرد کمتری دارد.