Iris Flower Classification ML Project

Classification using Decision Tree

Author - SHIVANGI CHAUHAN

LGMVIP- Data Science

Iris is a flowering plant with showy flowers. Our task is train a decision tree classifier model that would classify the dataset and predict the appropriate specicis for the given input. The dataset contains information about three species of this plant. so, our output must be among one of these three species

The iris dataset has four attributes namely:

1. sepal length
2. sepal width
3. petal length
4. petal width
   `Target variable: species (Iris-setosa, Iris-versicolor, Iris-virginica)`

Importing all the required libraries

In [68]:

```
import pandas as pd
import numpy as np
import plotly
import plotly.express as px
import plotly.offline as pyo
import cufflinks as cf
from plotly.offline import init_notebook_mode,plot,iplot

import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.metrics import accuracy_score
import os
```

In [69]:

```
pyo.init_notebook_mode(connected=True)
cf.go_offline()
```

Loading the Iris Dataset

In [70]:

```
df=pd.read_csv('C:/Users/hp/OneDrive/Desktop/Iris.csv')
```

In [71]:

```
df
```

Out[71]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [72]:

```
df.shape
```

Out[72]:

```
(150, 6)
```

In [89]:

```
df.drop('Id',axis=1,inplace=True)
```

In [90]:

```
df
```

Out[90]:

```
     SepalLength  SepalWidth  PetalLength  PetalWidth         Species
  0          5.1         3.5          1.4         0.2     Iris-setosa
  1          4.9         3.0          1.4         0.2     Iris-setosa
  2          4.7         3.2          1.3         0.2     Iris-setosa
  3          4.6         3.1          1.5         0.2     Iris-setosa
  4          5.0         3.6          1.4         0.2     Iris-setosa
...          ...         ...          ...         ...             ...
145          6.7         3.0          5.2         2.3  Iris-virginica
146          6.3         2.5          5.0         1.9  Iris-virginica
147          6.5         3.0          5.2         2.0  Iris-virginica
148          6.2         3.4          5.4         2.3  Iris-virginica
149          5.9         3.0          5.1         1.8  Iris-virginica
```

150 rows × 5 columns

Renaming columns label into suitable label

In [91]:

```
SepalLengthCm':'SepalLength','SepalWidthCm':'SepalWidth','PetalLengthCm':'PetalLength','Peta
```

In [92]:

```
df
```

Out[92]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

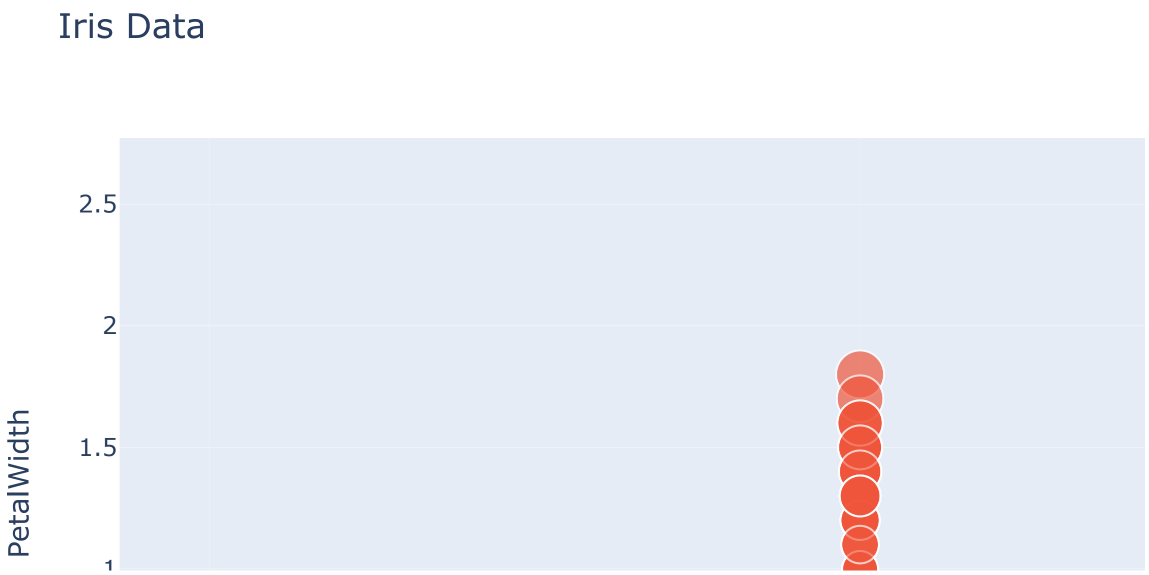
150 rows × 5 columns

Visualizing our Data

Scatter Plot

In [93]:

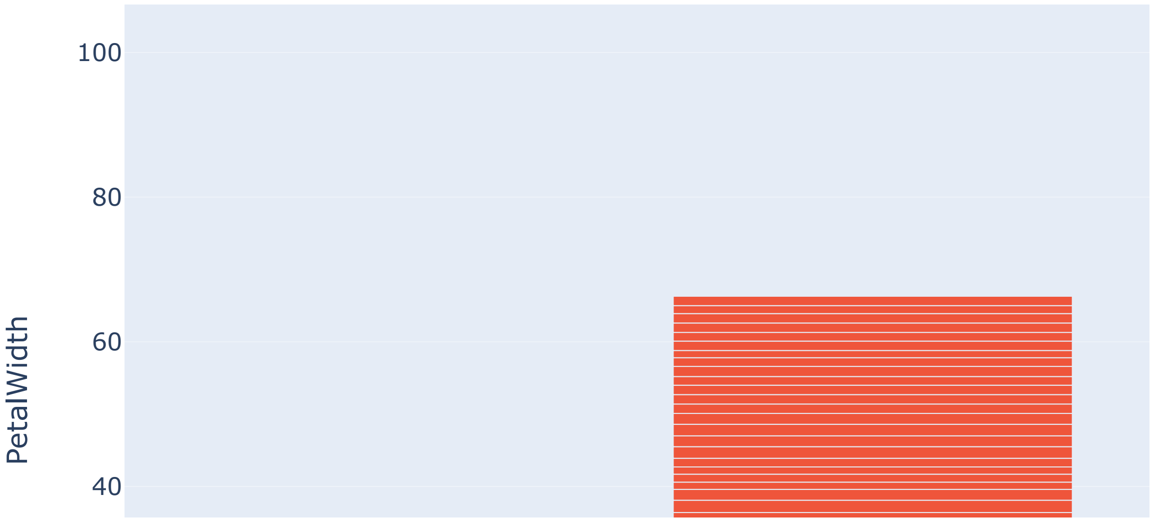
```
px.scatter(df,x='Species', y='PetalWidth',size='PetalWidth',title='Iris Data',color='Specie
```



Bar Plot

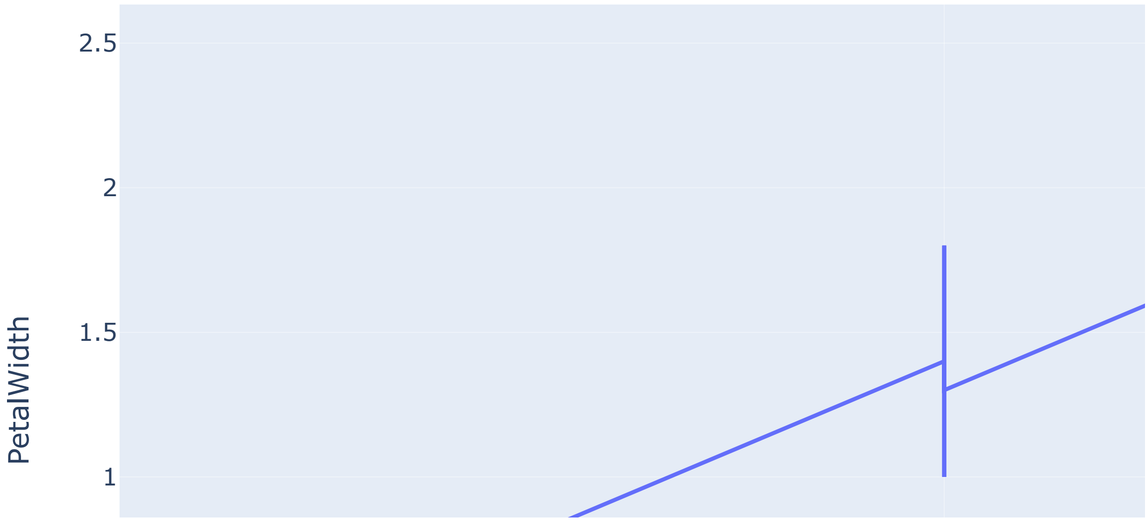
In [94]:

```
px.bar(df,x='Species',y='PetalWidth',color='Species')
```



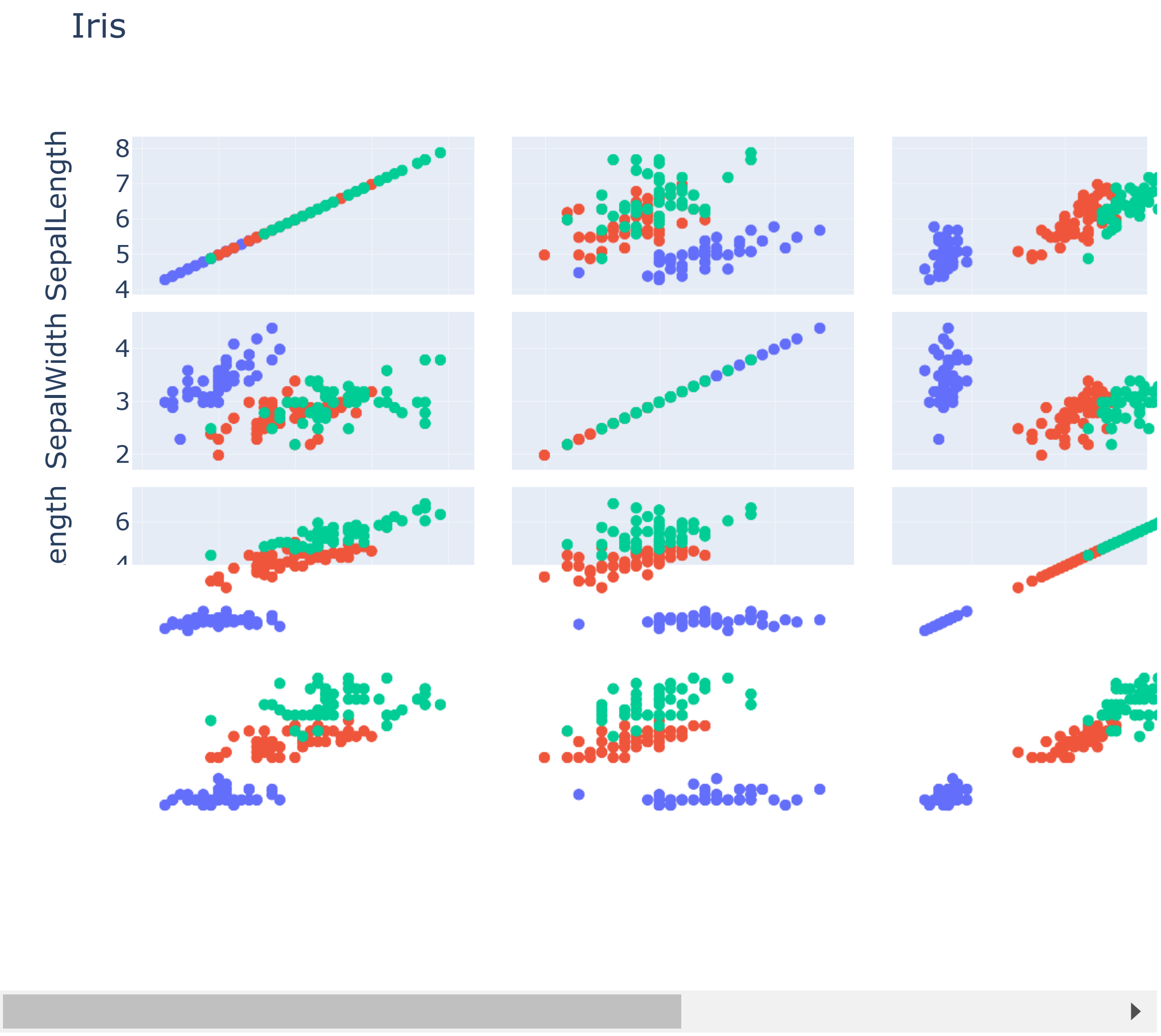
In [95]:

```
px.line(df,x='Species',y='PetalWidth')
```



In [96]:

```
px.scatter_matrix(df,color='Species',title='Iris',dimensions=['SepalLength','SepalWidth','P
```



Data Preprocessing

In [97]:

```
df
```

Out[97]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [98]:

```
x=df.drop(['Species'],axis=1)
```

In [99]:

```
x
```

Out[99]:

	SepalLength	SepalWidth	PetalLength	PetalWidth
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
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [100]:

```
y=df[('Species')]
```

In [101]:

```
y
```

Out[101]:

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

In [102]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(y)
```

In [103]:

```
y
```

Out[103]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [104]:

```
x=np.array(x)
x
```

```
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
```

In [105]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [106]:

```
x_test
```

Out[106]:

```
array([[5.8, 2.8, 5.1, 2.4],
       [6. , 2.2, 4. , 1. ],
       [5.5, 4.2, 1.4, 0.2],
       [7.3, 2.9, 6.3, 1.8],
       [5. , 3.4, 1.5, 0.2],
       [6.3, 3.3, 6. , 2.5],
       [5. , 3.5, 1.3, 0.3],
       [6.7, 3.1, 4.7, 1.5],
       [6.8, 2.8, 4.8, 1.4],
       [6.1, 2.8, 4. , 1.3],
       [6.1, 2.6, 5.6, 1.4],
       [6.4, 3.2, 4.5, 1.5],
       [6.1, 2.8, 4.7, 1.2],
       [6.5, 2.8, 4.6, 1.5],
       [6.1, 2.9, 4.7, 1.4],
       [4.9, 3.1, 1.5, 0.1],
       [6. , 2.9, 4.5, 1.5],
       [5.5, 2.6, 4.4, 1.2],
       [4.8, 3. , 1.4, 0.3],
       [5.4, 3.9, 1.3, 0.4],
       [5.6, 2.8, 4.9, 2. ],
       [5.6, 3. , 4.5, 1.5],
       [4.8, 3.4, 1.9, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [6.2, 2.8, 4.8, 1.8],
       [4.6, 3.6, 1. , 0.2],
       [5.1, 3.8, 1.9, 0.4],
       [6.2, 2.9, 4.3, 1.3],
       [5. , 2.3, 3.3, 1. ],
       [5. , 3.4, 1.6, 0.4],
       [6.4, 3.1, 5.5, 1.8],
       [5.4, 3. , 4.5, 1.5],
       [5.2, 3.5, 1.5, 0.2],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.2],
       [5.2, 2.7, 3.9, 1.4],
       [5.7, 3.8, 1.7, 0.3],
       [6. , 2.7, 5.1, 1.6],
       [5.9, 3. , 4.2, 1.5],
       [5.8, 2.6, 4. , 1.2],
       [6.8, 3. , 5.5, 2.1],
       [4.7, 3.2, 1.3, 0.2],
       [6.9, 3.1, 5.1, 2.3],
       [5. , 3.5, 1.6, 0.6],
       [5.4, 3.7, 1.5, 0.2]])
```

In [107]:

```
x_test.size
```

Out[107]:

```
180
```

In [108]:

```
x_train.size
```

Out[108]:

```
420
```

Decision Tree

Training The Decision Tree

In [109]:

```
from sklearn import tree
DT=tree.DecisionTreeClassifier()
DT=DT.fit(x_train,y_train)
```

In [110]:

```
from sklearn.metrics import accuracy_score
import os
```

In [111]:

```
PDT=DT.predict(x_test)
```

In [112]:

```
ADT=accuracy_score(y_test,PDT)*100
```

In [113]:

```
ADT
```

Out[113]:

```
97.77777777777777
```

In [114]:

```
y_test
```

Out[114]:

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 1, 1, 1, 2, 0, 2, 0,
       0])
```

In [115]:

```
PDT
```

Out[115]:

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 2, 1, 1, 2, 0, 2, 0,
       0])
```

In [116]:

```
pip install graphviz
```

```
Requirement already satisfied: graphviz in c:\users\hp\anaconda3\lib\site-pa
ckages (0.17)
Note: you may need to restart the kernel to use updated packages.
```

In [117]:

```
os.environ["PATH"]+= os.pathsep+(r'C:\Users\hp\Downloads')
```

In [118]:

```
import graphviz
```

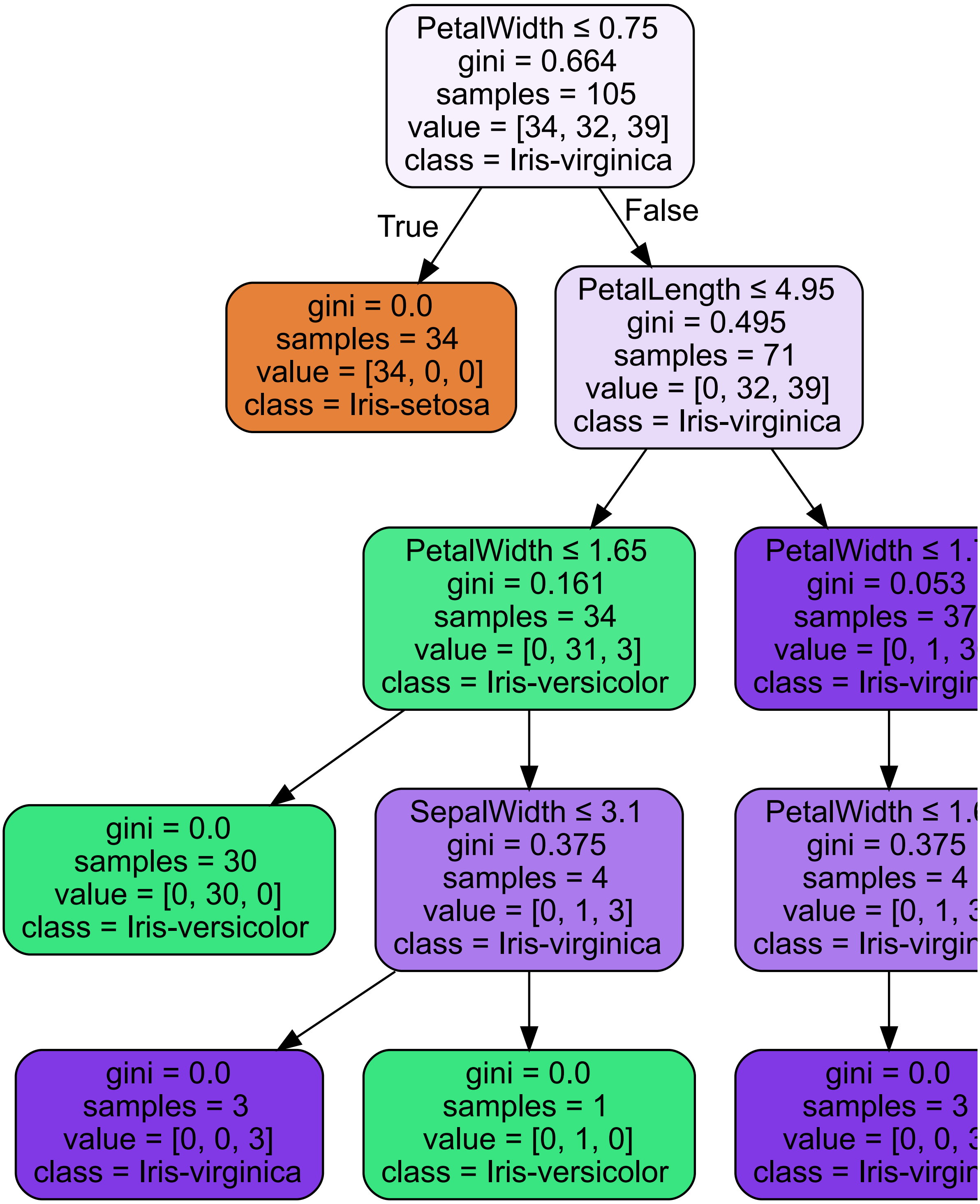
In [119]:

```
ure_names=df.drop(['Species'],axis=1).keys(),class_names=df['Species'].unique(),filled=True,
```

In [120]:

```
graphviz.Source(vis_data)
```

Out[120]:



Let's predict on some custom values

In [126]:

```
Catagory=['Iris-Setosa','Iris-Versicolor','Iris-Virginica']
```

In [127]:

```
X_DT=np.array([[1 ,1, 1, 1]])
X_DT_prediction=DT.predict(X_DT)
```

In [128]:

```
X_DT_prediction[0]
print(Catagory[int(X_DT_prediction[0])])
```

```
Iris-Versicolor
```

In [ ]: