Task: For the given 'Iris' dataset, create the Decision Tree classifier and visualize it graphically. The purpose is if we feed any new data to this classifier, it would be able to predict the right class accordingly.

Importing the Libraries

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

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

Importing the datasets

In [8]:
```
df=pd.read_csv(r'Downloads\Iris.csv')
print(df)
```

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

            Species
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

[150 rows x 6 columns]
```

Checking for missing values

In [9]:
```
df.isnull().any()
```

Out[9]:
```
Id               False
SepalLengthCm    False
SepalWidthCm     False
PetalLengthCm    False
PetalWidthCm     False
Species          False
dtype: bool
```

Describing the Data

In [10]:
```
df.dtypes
```

Out[10]:
```
Id                int64
SepalLengthCm     float64
SepalWidthCm      float64
PetalLengthCm     float64
PetalWidthCm      float64
Species            object
dtype: object
```

In [11]:
```
df.describe()
```

Out[11]:

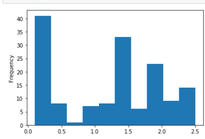
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [12]:
```
df.columns
```

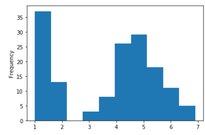
Out[12]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

Histograms of Petal Width and Petal Length

In [13]:
```
df['PetalWidthCm'].plot.hist()
plt.show()
```

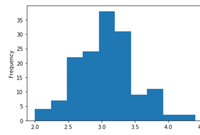


In [14]:
```
df['PetalLengthCm'].plot.hist()
plt.show()
```

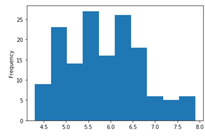


Histograms of Sepal Width and Sepal Length

In [15]:
```
df['SepalWidthCm'].plot.hist()
plt.show()
```



In [16]:
```
df['SepalLengthCm'].plot.hist()
plt.show()
```



Preparing the Data

In [18]:
```
x = df.iloc[:, 1:5].values
y = df.iloc[:, 5].values
```

Splitting Data into Training & Testing Datasets

In [19]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state =0)
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

Decision Tree Classifier

In [20]:
```
from sklearn.tree import DecisionTreeClassifier

dtc = DecisionTreeClassifier( )
dtc.fit(x, y)
print('Decision Tree Classifer Created')
```

```
Decision Tree Classifer Created
```

Visualizing Decision Trees

In [21]:
```
from sklearn.tree import plot_tree

plt.figure(figsize=(20,20))
tree_img= plot_tree(dtc, feature_names=df.columns, class_names=df['Species'].unique().tolist(), precision=4,label="all",filled=True)
plt.show()
```