Name : Aniruddha Joshi

Division: AI & DS

Batch: B3

Roll No: 51

PRN: 12011234

Decision Tree classifier - Iris dataset

The Iris flower data set or Fisher's Iris data set is a multivariate data set. The data set consists of 150 samples from each of three species of Iris (Iris setosa-0, Iris virginica-1 and Iris versicolor-2). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other

objective: Classify a new flower as belonging to one of the 3 classes present in the data set. Importing the required libraries

Problem Statement: Implement Decision tree classifier for fisher iris data set and evaluate the performance.

In [3]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn.datasets as datasets
%matplotlib inline
```

In [4]:

```
iris_set = datasets.load_iris()
iris_data = pd.DataFrame(iris_set.data, columns=iris_set.feature_names)
iris_data["species"] = iris_set.target
iris_data.head()
```

Out[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [5]:

```
iris_data.shape
```

Out[5]:

```
(150, 5)
```

In [6]:

```
iris_data.columns
```

Out[6]:

```
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'species'],
      dtype='object')
```

In [7]:

```
#how many data points for each class are present?
iris_data["species"].value_counts()
```

Out[7]:

```
0    50
1    50
2    50
Name: species, dtype: int64
```

In [8]:

```
iris_data.isnull().sum()
#ANY MISSING VALUES?
```

Out[8]:

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
species              0
dtype: int64
```

In [9]:

```
#The variables have no missing values
```

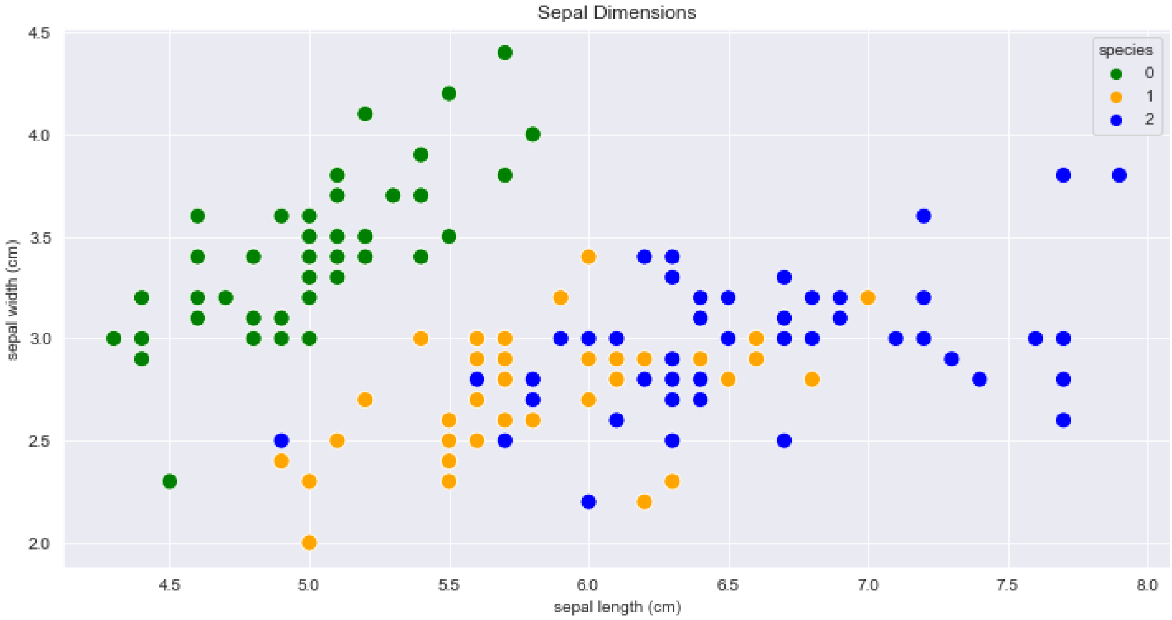
Visualising the sepal and petal dimensions

In [11]:

```
sns.set_style("darkgrid")
```

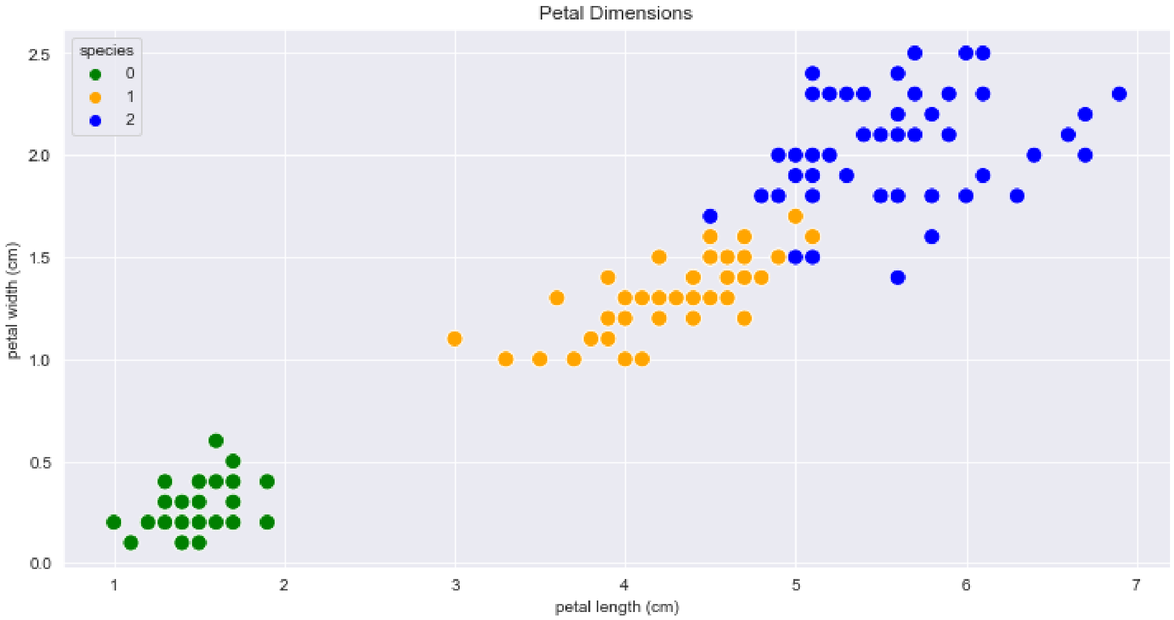
In [12]:

```
plt.figure(figsize=(12, 6))
plt.title('Sepal Dimensions')

sns.scatterplot(x=iris_data["sepal length (cm)"],
                y=iris_data["sepal width (cm)"],
                hue=iris_data["species"],palette = ["green","orange","blue"],
                s=100);

```



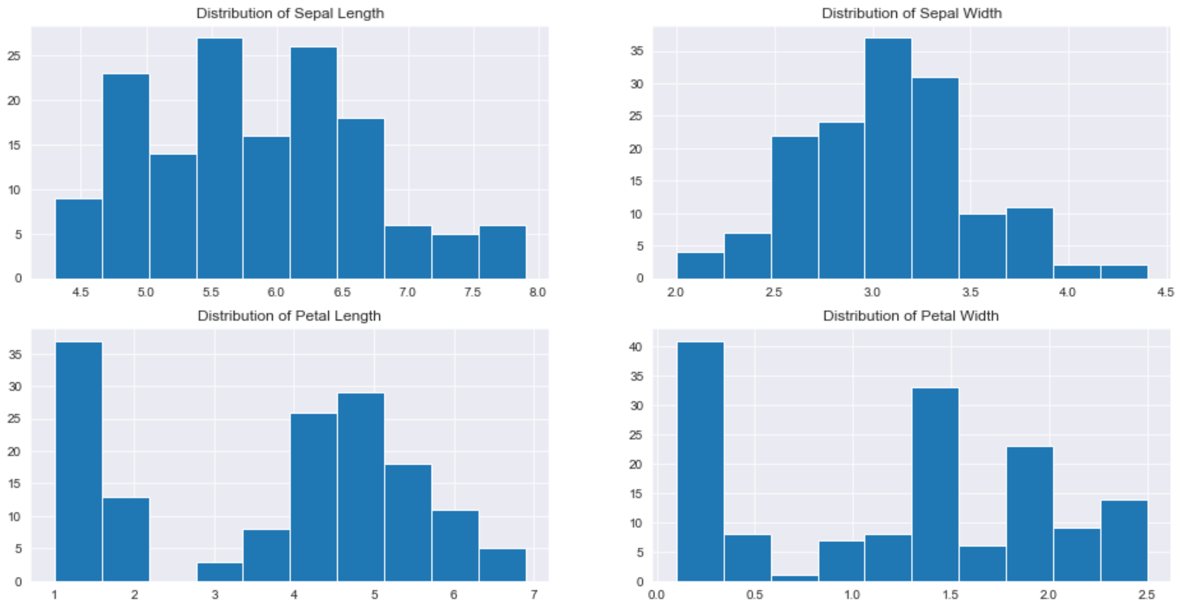
In [13]:

```
plt.figure(figsize=(12, 6))
plt.title('Petal Dimensions')

sns.scatterplot(x=iris_data["petal length (cm)"],
                y=iris_data["petal width (cm)"],
                hue=iris_data["species"],palette = ["green","orange","blue"],
                s=100);
```



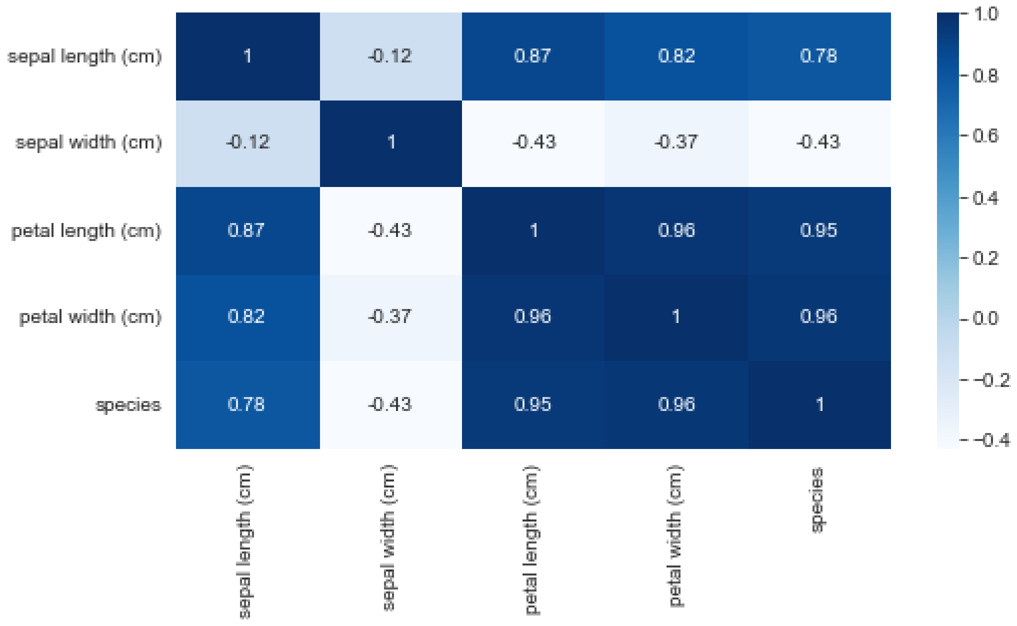
In [14]:

```
fig, axes = plt.subplots(2, 2, figsize=(16, 8))

#plot for sepal length
axes[0,0].set_title('Distribution of Sepal Length')
axes[0,0].hist(iris_data["sepal length (cm)"]);

#plot for sepal width
axes[0,1].set_title('Distribution of Sepal Width')
axes[0,1].hist(iris_data["sepal width (cm)"]);

#plot for petal length
axes[1,0].set_title('Distribution of Petal Length')
axes[1,0].hist(iris_data["petal length (cm)"]);

#plot for petal width
axes[1,1].set_title('Distribution of Petal Width')
axes[1,1].hist(iris_data["petal width (cm)"]);
```



In [15]:

```
plt.figure(figsize=(8,4))
sns.heatmap(iris_data.corr(), annot=True, cmap='Blues');
```



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
sepal length (cm)	1	-0.12	0.87	0.82	0.78
sepal width (cm)	-0.12	1	-0.43	-0.37	-0.43
petal length (cm)	0.87	-0.43	1	0.96	0.95
petal width (cm)	0.82	-0.37	0.96	1	0.96
species	0.78	-0.43	0.95	0.96	1

Splitting The Data into Training And Testing Dataset

In [16]:

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(iris_data, test_size = 0.2)
```

In [17]:

```
train.shape, test.shape
```

Out[17]:

```
((120, 5), (30, 5))
```

In [18]:

```
iris_data.columns
```

Out[18]:

```
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'species'],
      dtype='object')
```

In [19]:

```
train_x = train[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)','petal width
train_y = train.species
test_x = test[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)','petal width (c
test_y = test.species
```

In [34]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
dtree = DecisionTreeClassifier()
dtree.fit(train_x,train_y)
predictions = dtree.predict(test_x)
print("The accuracy of Decision Tree is:", metrics.accuracy_score(predictions, test_y))
```

```
The accuracy of Decision Tree is: 0.9333333333333333
```

In [44]:

```
import sklearn.metrics
```

In [46]:

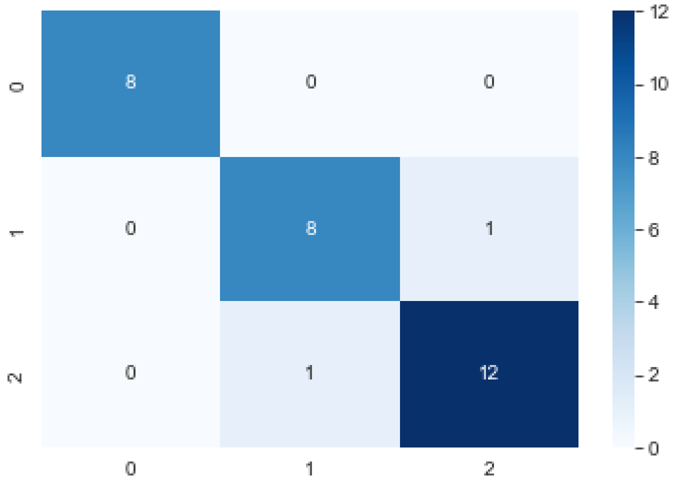
```
print(sklearn.metrics.classification_report(test_y, predictions))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       0.89      0.89      0.89         9
           2       0.92      0.92      0.92        13

    accuracy                           0.93        30
   macro avg       0.94      0.94      0.94        30
weighted avg       0.93      0.93      0.93        30
```

Confusion Matrix -

In [48]:

```
import seaborn as sns
cf_matrix = sklearn.metrics.confusion_matrix(test_y, dtree.predict(test_x))
ax = sns.heatmap(cf_matrix, annot=True, cmap='Blues')
```



In [ ]: