Analysis of Iris Flower Dataset

```
# Importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Collection

```
# Reading dataset
df = pd.read_csv('C:\\Users\\shiwa\\Downloads\\Iris Flower - 
Iris.csv')  # With pandas
# data = sns.load_dataset('Iris Flower - Iris', data_home = 'C:\\
Users\\shiwa\\Downloads') # With seaborn

df.head()
```

```
   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
```

```
df.describe()
```

```
                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
```

```
df.shape
```

```
(150, 6)
```

```
df.columns
```

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

```
df['Species'].value_counts()
```

```
Species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

Data Cleaning

Null Values

```
# Checking for null values
df.isnull().sum()
```

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

Duplicated Values

```
# Checking for duplicated values
df.duplicated().sum()
```

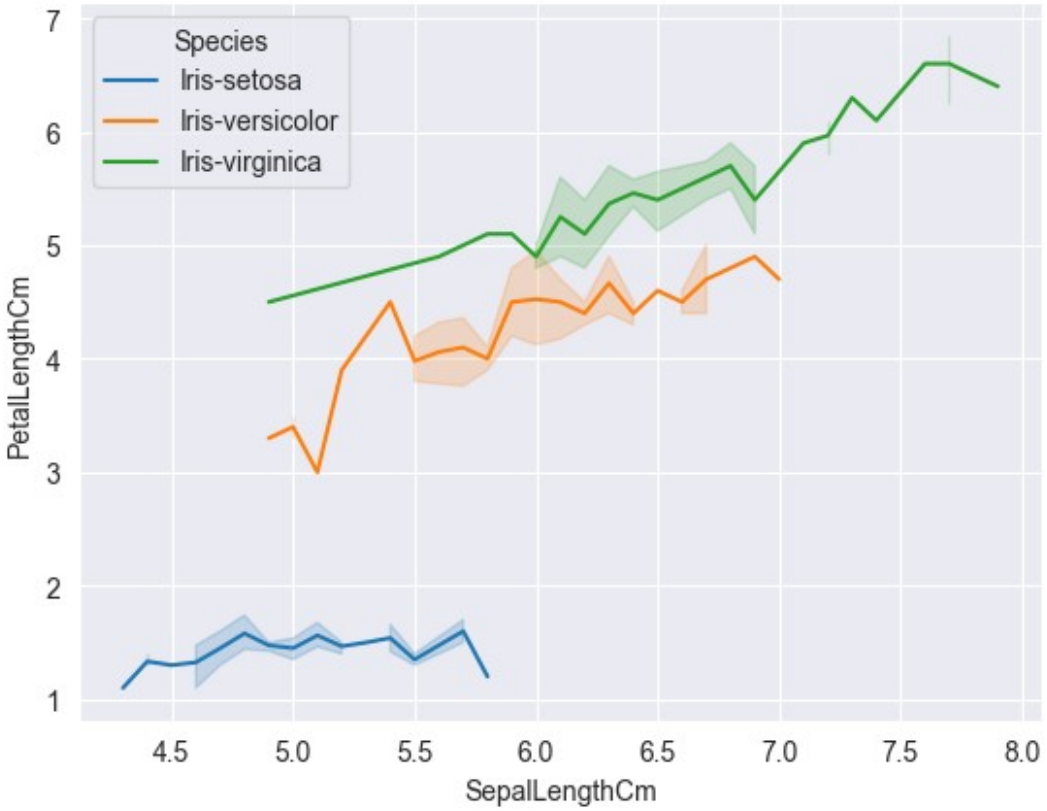
```
0
```

Data Visualization

Line Plot

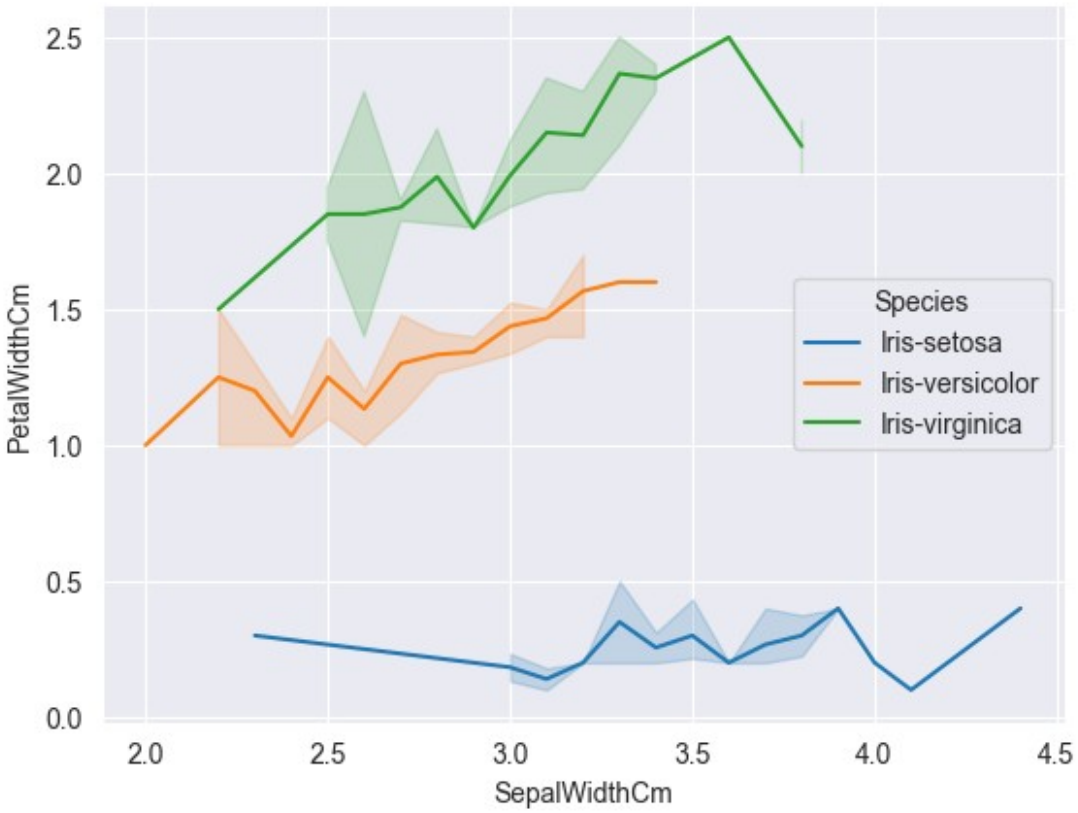
```
# Plotting line chart between sepalLength and petalLength
sns.set_style('dark')
```

```
sns.lineplot(df, x="SepalLengthCm", y="PetalLengthCm", hue="Species")
plt.grid(True)
plt.show()
```



Observation: We can observe that the species **Iris-Setosa** can be easily separted.

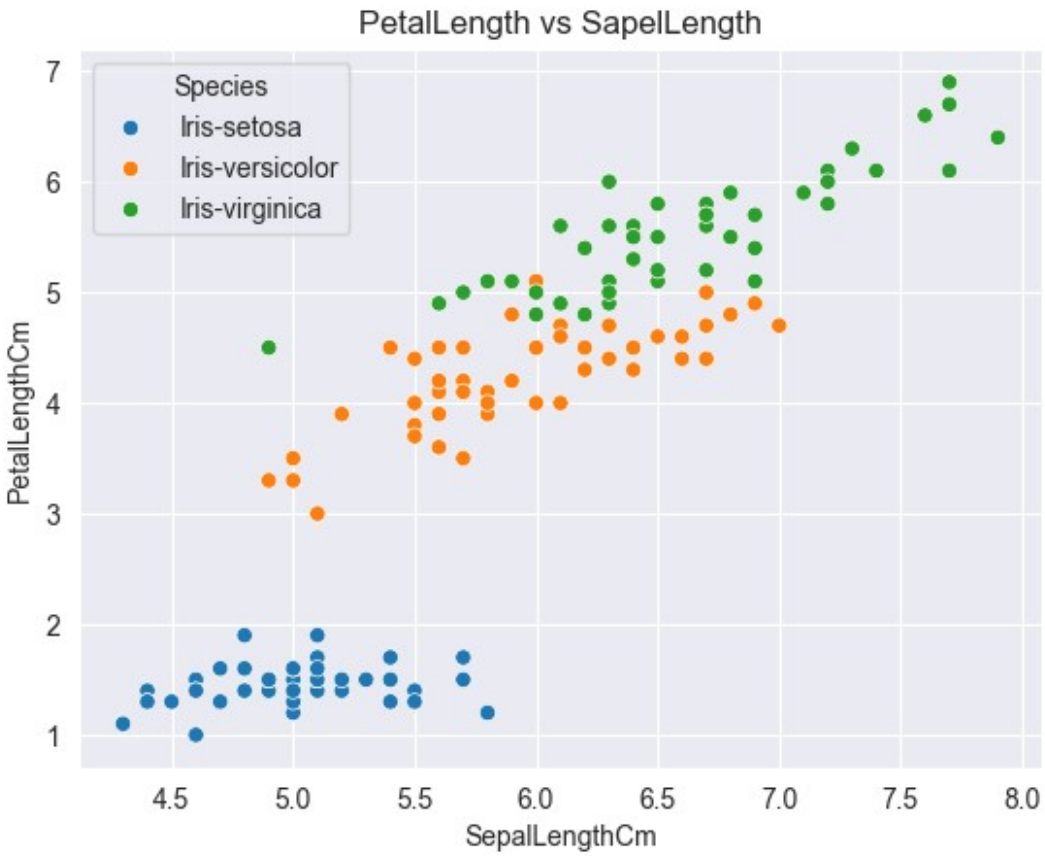
```
# Plotting line chart between sepalwidth and petalwidth
sns.set_style('dark')
sns.lineplot(df, x="SepalWidthCm", y="PetalWidthCm", hue="Species")
plt.grid(True)
plt.show()
```



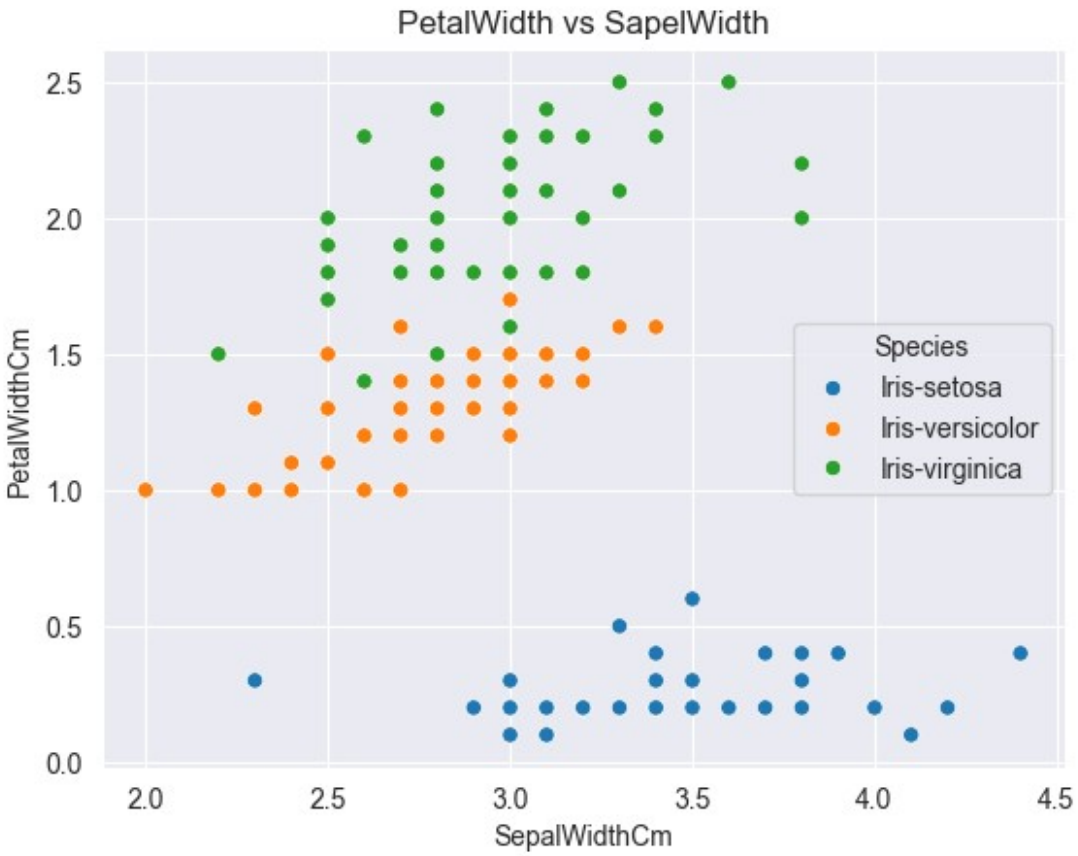
Observation: We can observe that the species **Iris-Setosa** can be easily separted. **Problem:** We are still unable to clearly differenciate the other two species.

Scatter Plot

```
sns.scatterplot(df, x="SepalLengthCm", y="PetalLengthCm",
hue="Species")
sns.set_style('dark')
plt.title('PetalLength vs SapelLength')
plt.grid(True)
plt.show()
```



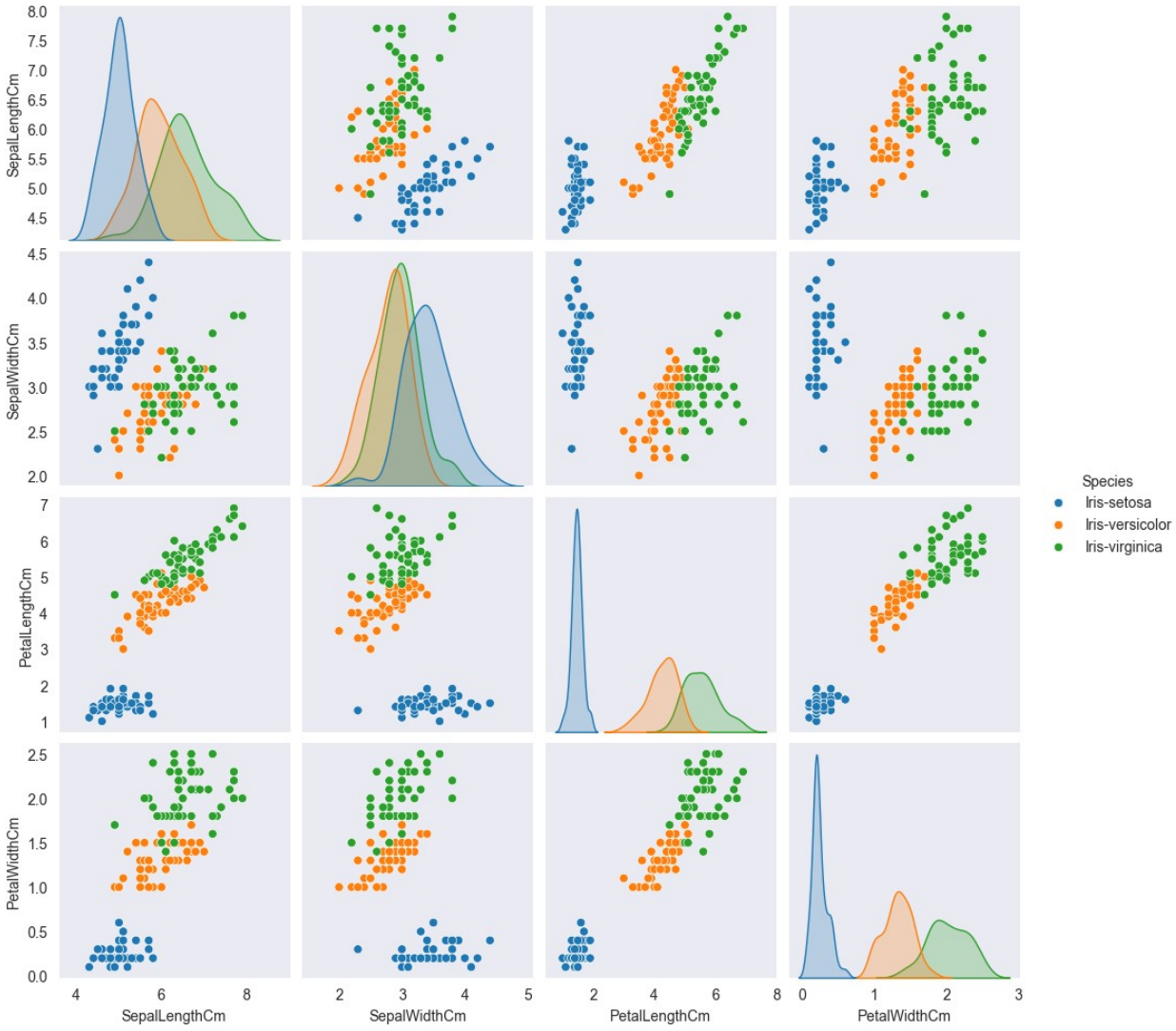

```
sns.scatterplot(df, x="SepalWidthCm", y="PetalWidthCm", hue="Species")
sns.set_style('dark')
plt.title('PetalWidth vs SapelWidth')
plt.grid(True)
plt.show()
```



Observation: Scatter Plot increase the visualization but we are still unable to find the right way to separte all the three species from each other.

Pair Plot

```
# Data Visualization with seaborn
df2 = df.drop(['Id'], axis= 1, inplace = False)
sns.pairplot(df2, hue= "Species")
plt.show()
```



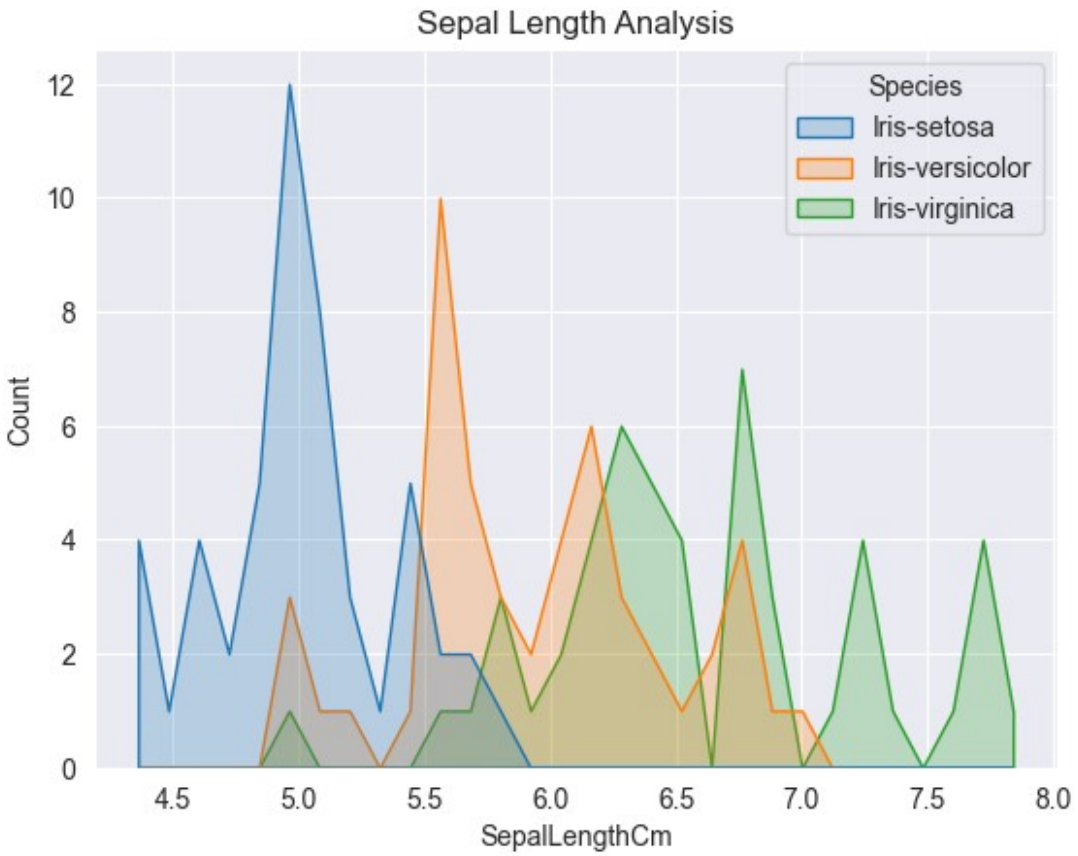
Observation: We can now analyse the right way to perform the classification.

Univariate Analysis

Lets check the importance of each variable ['SepalLengthCm', 'SepalWidthCm', 'PetalWidthCm', 'PetalWidthCm'] for the classification.

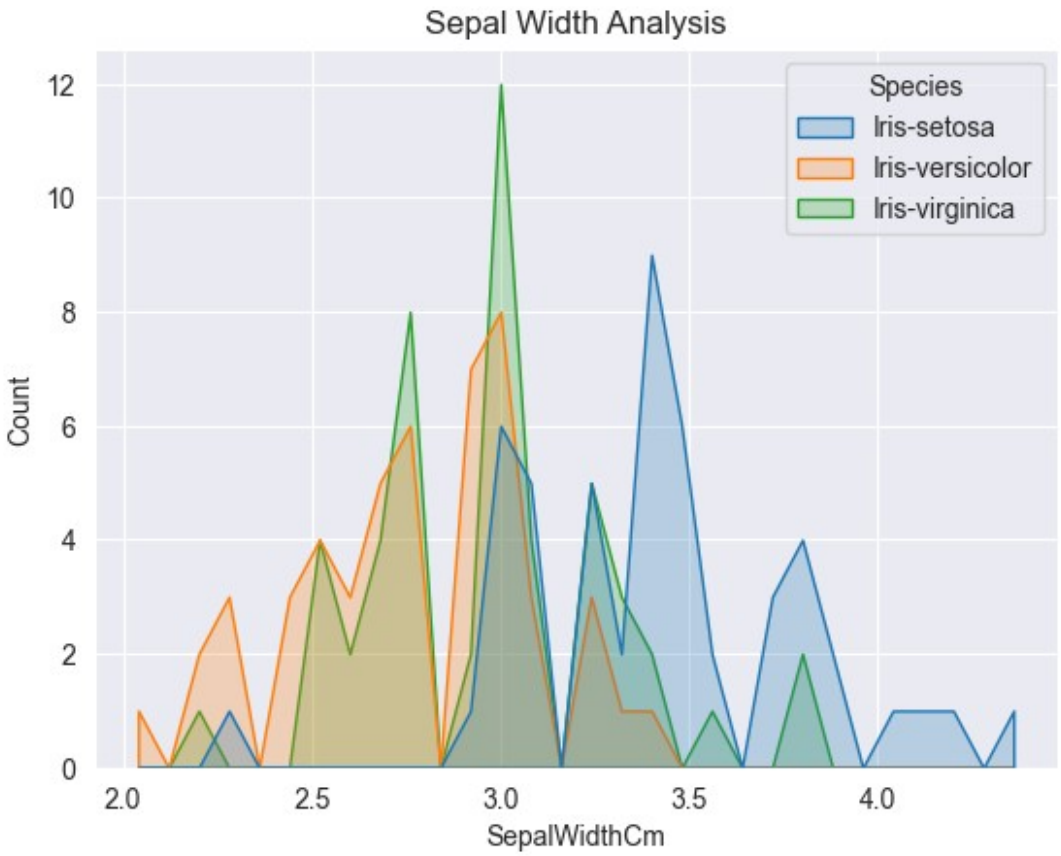
Histogram Plot

```
sns.histplot(df, x="SepalLengthCm", hue="Species", element='poly',
bins=30)
sns.set_style('dark')
plt.title('Sepal Length Analysis')
plt.grid(True)
plt.show()
```



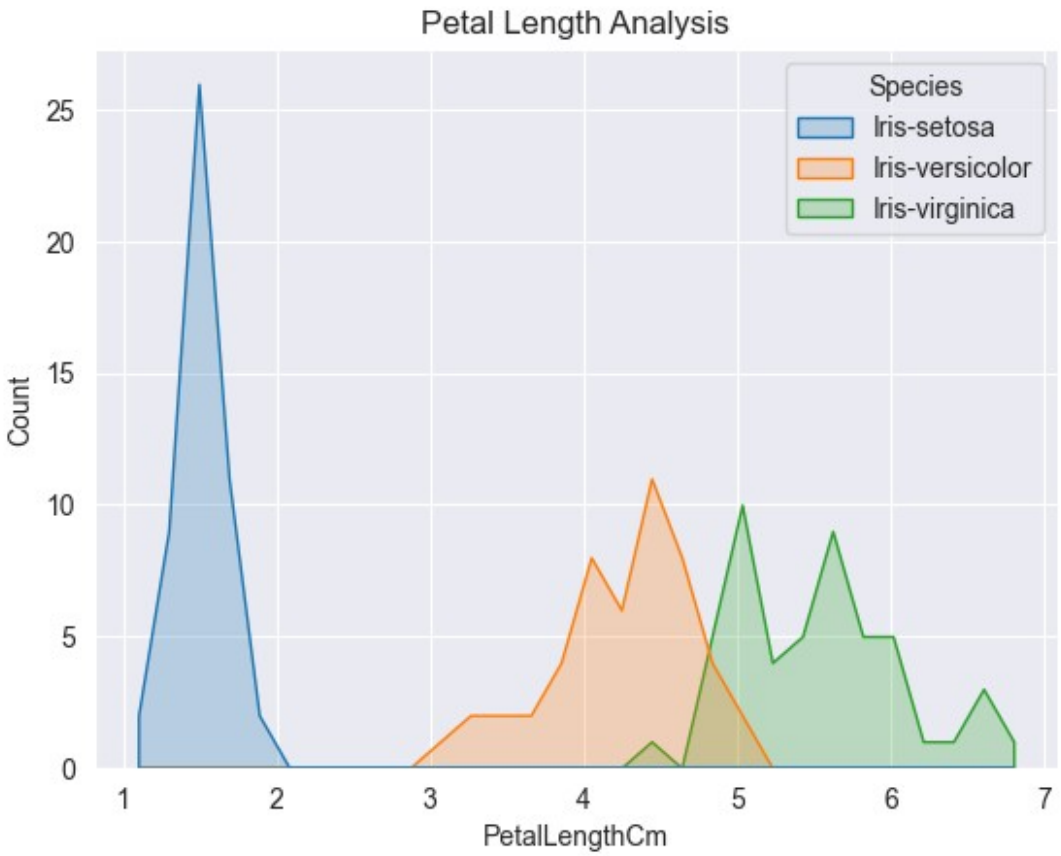
Observation: There is **overlapping** which makes the classification hard for us.

```
sns.histplot(df, x="SepalWidthCm", hue="Species", element='poly',
bins=30)
sns.set_style('dark')
plt.title('Sepal Width Analysis')
plt.grid(True)
plt.show()
```



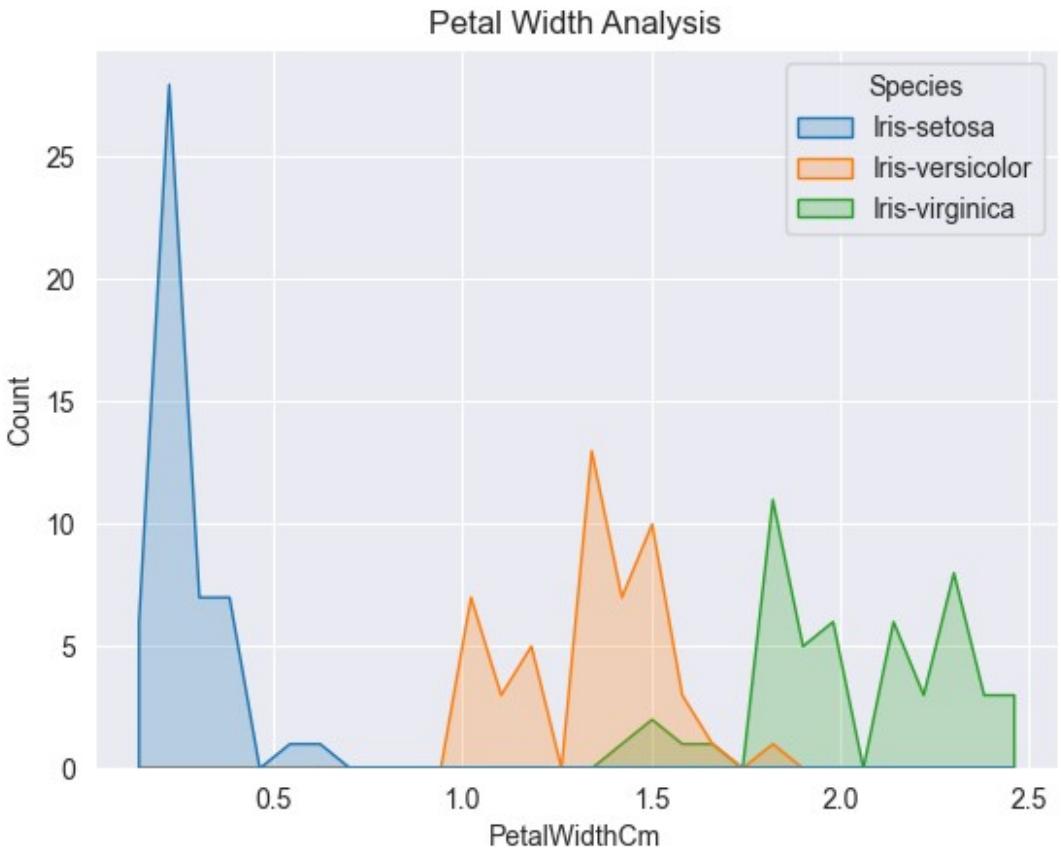
Observation: There is a lot of **overlapping** which makes the classification way more harder than the previous variable.

```
sns.histplot(df, x="PetalLengthCm", hue="Species", element='poly',
bins=30)
sns.set_style('dark')
plt.title('Petal Length Analysis')
plt.grid(True)
plt.show()
```



Observation: This is **best fit** comparable to other variables.

```
sns.histplot(df, x="PetalWidthCm", hue="Species", element='poly',
bins=30)
sns.set_style('dark')
plt.title('Petal Width Analysis')
plt.grid(True)
plt.show()
```

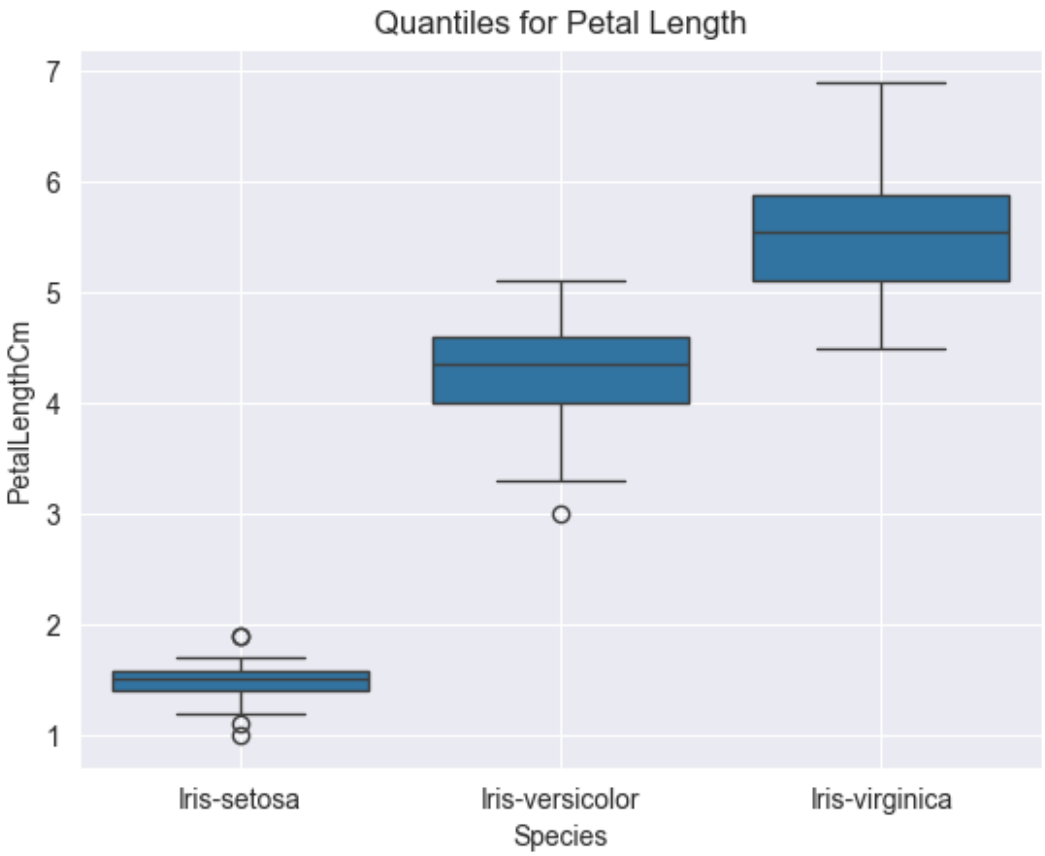


Observation: This is **better** than the first two variables but slightly less preferable than the **best fit**. So we can see that the importance of variables which we can consider is: **PetalLengthCm > PetalWidthCm >> SepalLengthCm >> SepalWidthCm**

Box Plot

Lets check the data's deviation.

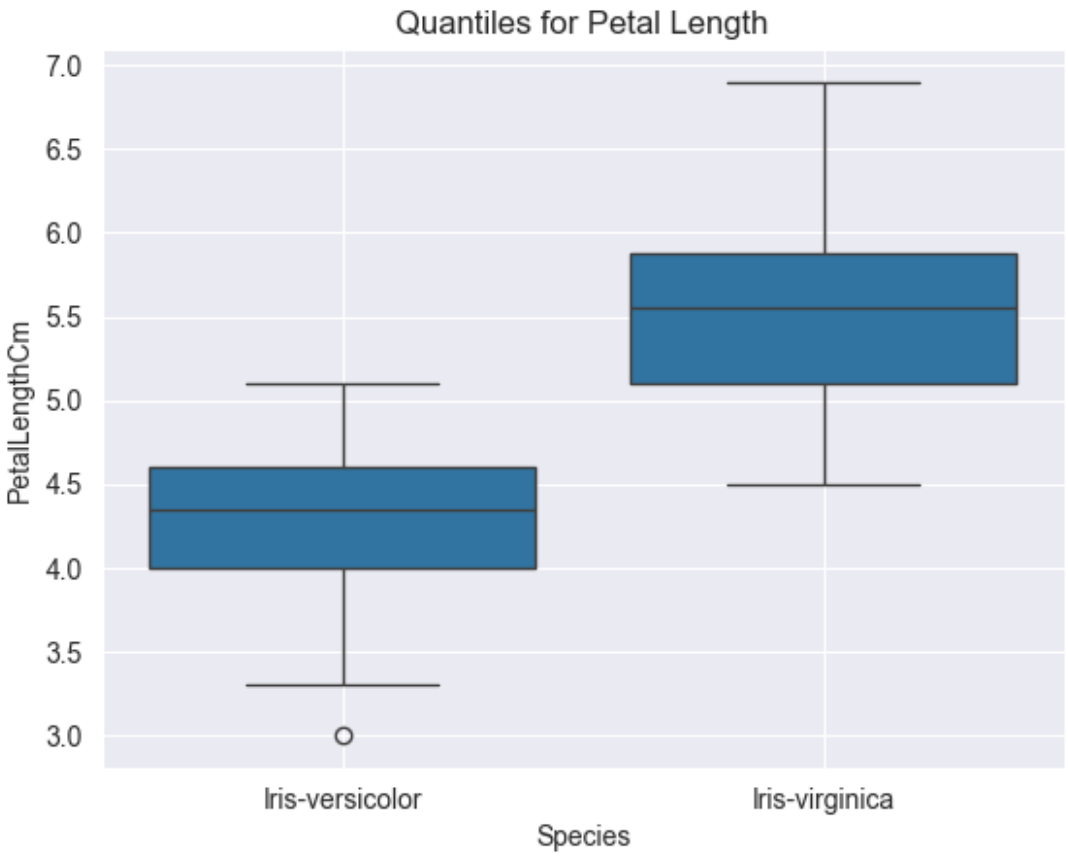
```
sns.boxplot(df,x = 'Species', y = "PetalLengthCm")
sns.set_style('dark')
plt.title('Quantiles for Petal Length')
plt.grid(True)
plt.show()
```



Observation: I can see there are 4 outliers and also I can observe the region where my middle 50% data lies. I want to look versicolor and virginica more closely.

```
vv = pd.read_csv('C:\\Users\\shiwa\\Downloads\\Iris Flower - 
Iris.csv', index_col="Species")
vv.drop(['Iris-setosa'], axis = 0, inplace = True)

sns.boxplot(vv,x = 'Species', y = "PetalLengthCm")
sns.set_style('dark')
plt.title('Quantiles for Petal Length')
plt.grid(True)
plt.show()
```



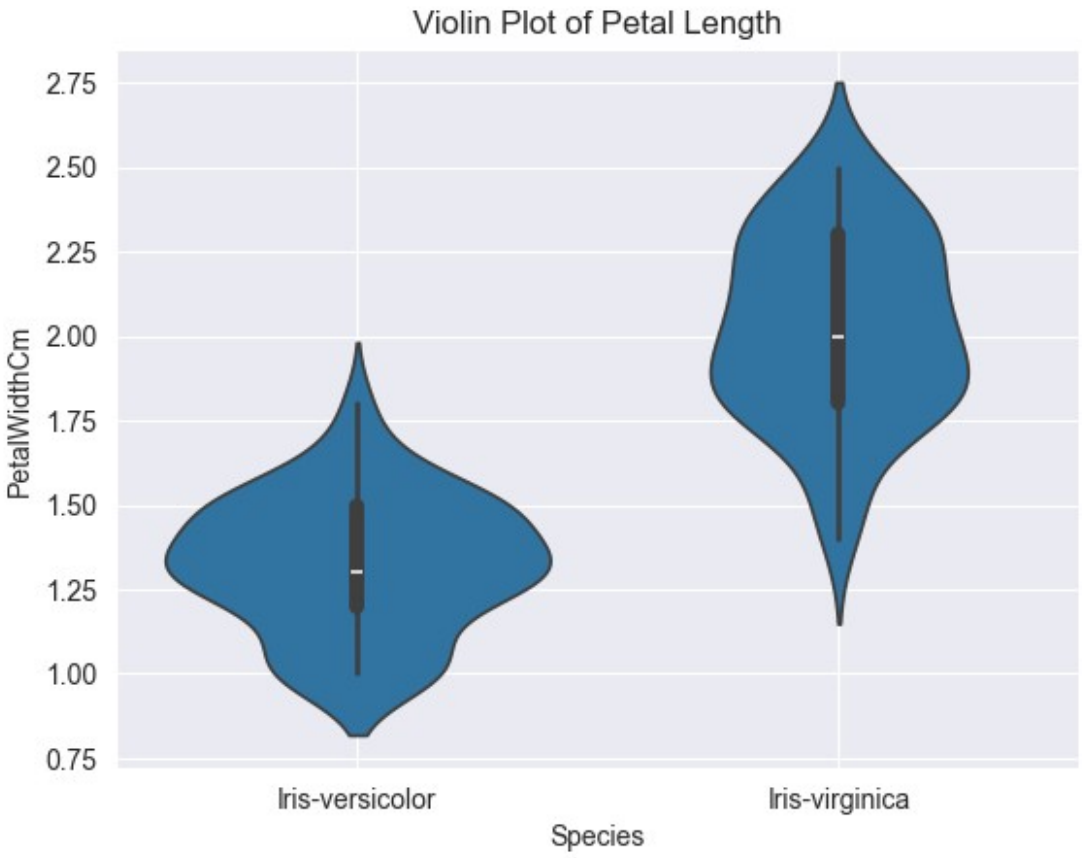
Observation: It is still not clear with this plot lets try violin plot for more clarity.

Violin Plot

```
sns.violinplot(vv,x = 'Species', y = "PetalLengthCm")
sns.set_style('dark')
plt.title('Violin Plot of Petal Length')
plt.grid(True)
plt.show()
```

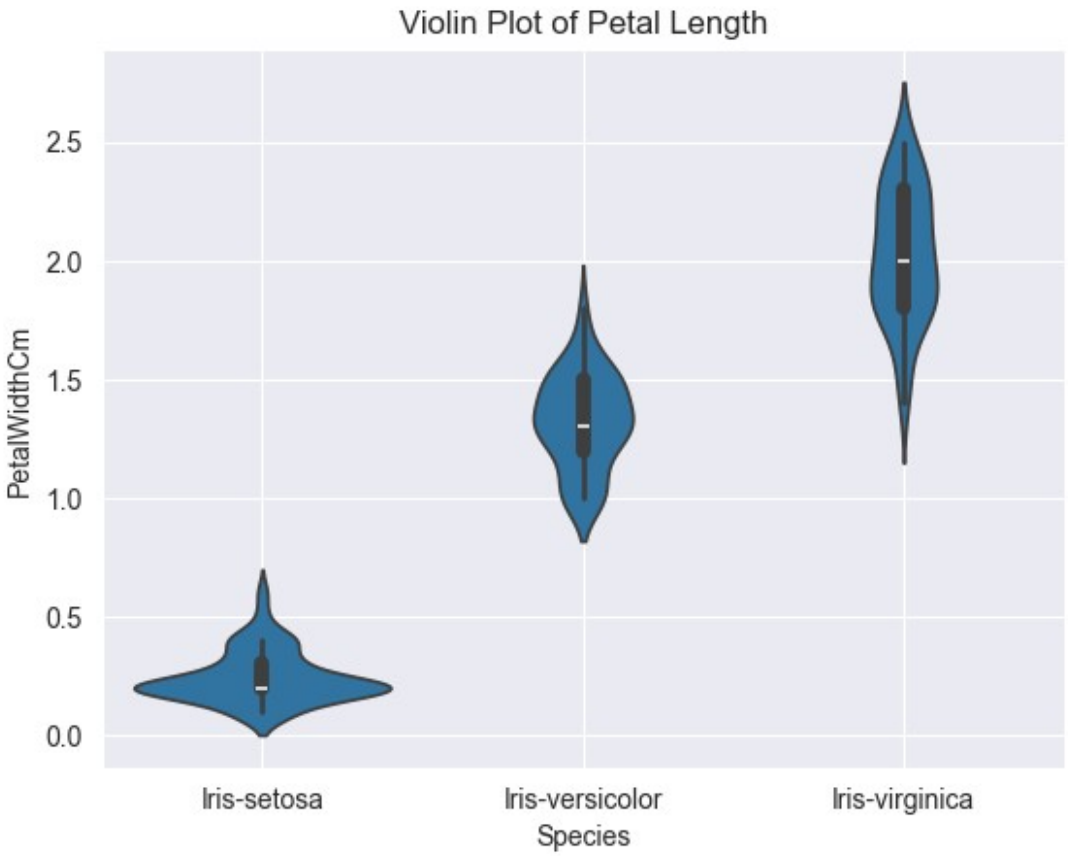
Observation: The difference point is still not clear with petallength so lets try petal width

```
sns.violinplot(vv,x = 'Species', y = "PetalWidthCm")
sns.set_style('dark')
plt.title('Violin Plot of Petal Length')
plt.grid(True)
plt.show()
```



Observation: I can see the gap between the IQR of these two.

```
sns.violinplot(df,x = 'Species', y = "PetalWidthCm")
sns.set_style('dark')
plt.title('Violin Plot of Petal Length')
plt.grid(True)
plt.show()
```



Observation: Now with the help of **Petal Length** I am able to classify **Setosa** from the samples And with the help of **Petal Width** I am able to classify **Versicolor** from the sample So the **rest** of are **Virginica**.