In [ ]:

```
Task 1: PERFORM DATA CLEANING
clean a dataset by removing missing values and outliners
By NIVEDHA M
```

In [ ]:

```
#IMPLEMENTING THE DEPENDENCIES
```

In [1]:

```
import pandas as pd
import numpy as np
iTask 1: PERFORM DATA CLEANING
clean a dataset by removing missing values and outliners
By NIVEDHA Mmport seaborn as sns
```

In [12]:

```
#DATA READING
```

In [2]:

```
gender_data = pd.read_csv("gender_submission.csv")
print(gender_data)
```

```
     PassengerId  Survived
0            892         0
1            893         1
2            894         0
3            895         0
4            896         1
..           ...       ...
413         1305         0
414         1306         1
415         1307         0
416         1308         0
417         1309         0

[418 rows x 2 columns]
```

In [ ]:

```
#DATA CLEANING
#Fill the missing values for passenger id and survival columns.In order to fill the miss
#will fill the missing values of both the columns by taking the mean of all columns
```

In [8]:

```
#fill passengerID column
gender_data["PassengerId"].fillna(gender_data["PassengerId"].mean(),inplace = True)
gender_data["PassengerId"].isna().sum()
```

Out[8]:

```
0
```

In [9]:

```
#fill survived column
gender_data["Survived"].fillna(gender_data["Survived"].mean(), inplace=True)
gender_data["Survived"].isna().sum()
```

Out[9]:

```
0
```

In [ ]:

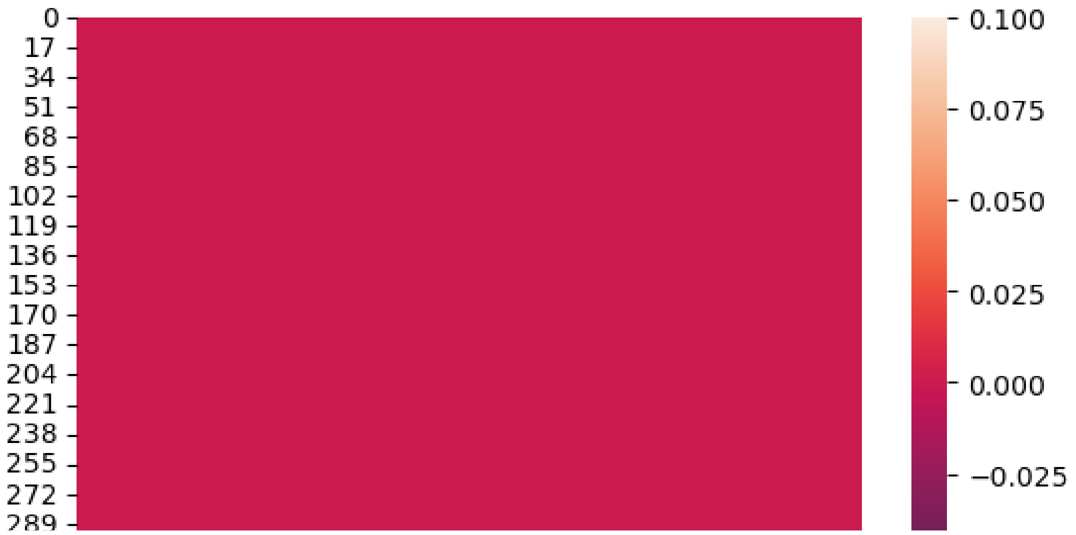
```
#Alternatively we will visualize the null value using heatmap
#we will use heatmap method by passing only records which are null
```

In [10]:

```
sns.heatmap(gender_data.isna())
```

Out[10]:

```
<AxesSubplot:>
```



In [ ]:

```
#we can conclude from the above heatmap that there is no null value left in our dataset
```

In [ ]:

```
Task 2 : Calculate Summary  Statistics
Calculate summary statistics(mean, median, mode, standard deviation) for a dataset
By NIVEDHA M
```

In [ ]:

```
#Implementing the Dependencies
```

In [1]:

```
import pandas as pd
import numpy as np
```

In [ ]:

```
#Data Reading
```

In [2]:

```
gender_data = pd.read_csv("gender_submission.csv")
print(gender_data)
```

```
     PassengerId  Survived
0            892         0
1            893         1
2            894         0
3            895         0
4            896         1
..           ...       ...
413         1305         0
414         1306         1
415         1307         0
416         1308         0
417         1309         0

[418 rows x 2 columns]
```

In [ ]:

```
#Using the describe() to find the statistics(mean, median, mode, standard deviation)
```

In [3]:

```
#Calculating the statistics (mean, median, mode, standard deviation)
gender_data.describe()
```

Out[3]:

	PassengerId	Survived
count	418.000000	418.000000
mean	1100.500000	0.363636
std	120.810458	0.481622
min	892.000000	0.000000
25%	996.250000	0.000000
50%	1100.500000	0.000000
75%	1204.750000	1.000000
max	1309.000000	1.000000

In [5]:

```
gender_data.median()
```

Out[5]:

```
PassengerId    1100.5
Survived          0.0
dtype: float64
```

In [6]:

```
gender_data.mode()
```

Out[6]:

	PassengerId	Survived
0	892	0.0
1	893	NaN
2	894	NaN
3	895	NaN
4	896	NaN
...	...	...
413	1305	NaN
414	1306	NaN
415	1307	NaN
416	1308	NaN
417	1309	NaN

418 rows × 2 columns

In [ ]:

```

```

```
#TASK 3 : Visualization using Histogram
#Create a histogram or bar chart to visualize the distribution of data in a dataclasses_to_dicts
#By NIVEDHA M
```

```
#Implementing the Dependencies
```

```
import pandas as pd
import seaborn as sns
```

```
#Reading the datasets
```

```
iris_data = pd.read_csv("Iris.csv")
print(iris_data)
```

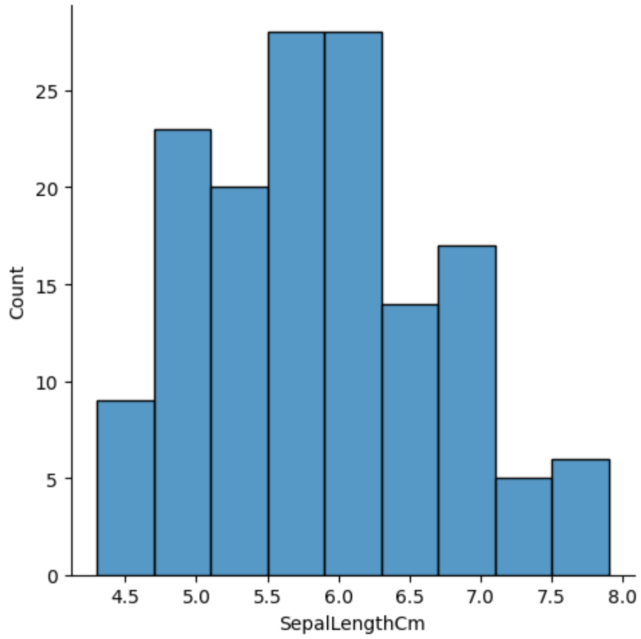
```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           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]
```

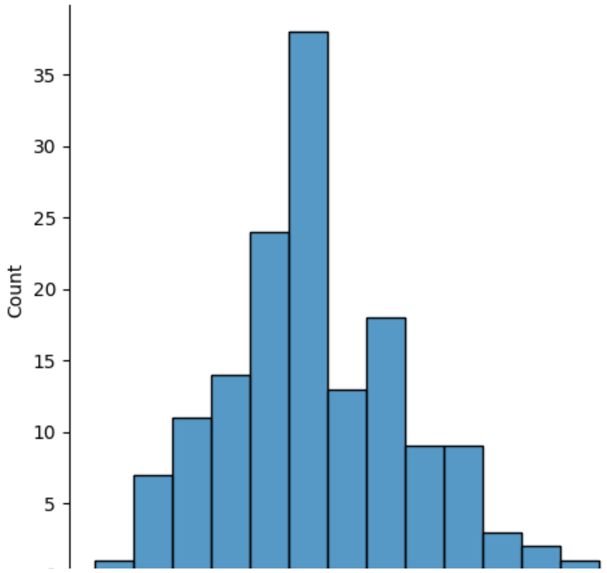
```
#plotting the histogram for SepalLength
sns.displot(x = "SepalLengthCm", data = iris_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x78389ae7aef0>
```



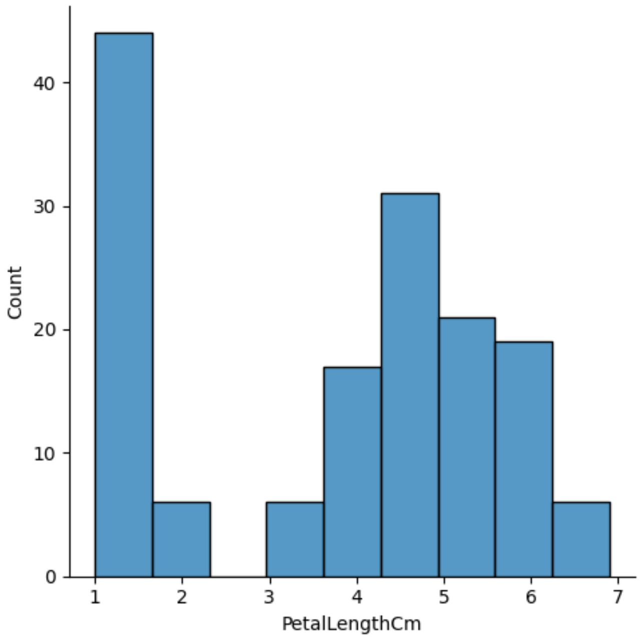
```
#plotting the histogram for SepalWidth
sns.displot(x = "SepalWidthCm", data = iris_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7838d2440d30>
```



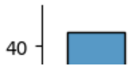
```
#plotting the histogram for PetalLength
sns.displot(x = "PetalLengthCm", data = iris_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x78389a8c7ac0>
```



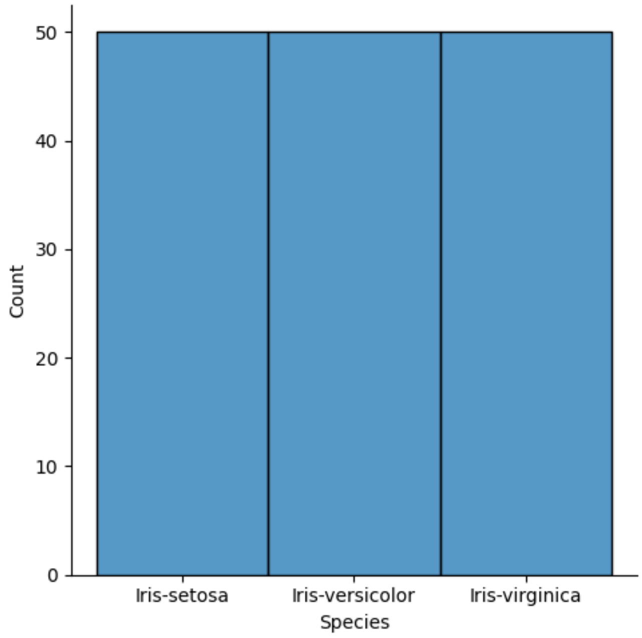
```
#plotting the histogram for PetalWidth
sns.displot(x = "PetalWidthCm", data = iris_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x78389855b0a0>
```



```
#plotting the histogram for Species
sns.displot(x = "Species", data = iris_data)
```

```
<seaborn.axisgrid.FacetGrid at 0x7838986246d0>
```


Count
Iris-setosa
Iris-versicolor
Iris-virginica
Species