TASK 3- IRIS FLOWER CLASSIFICATION.

name - Nutan Santosh Bhilare
email id- nutan10232@gmail.com domain-data science

importing libraries:

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter("ignore")
```

importing dataset

```
df = pd.read_csv('/content/IRIS.csv')
df.head()
df.points=df.iloc[:, 1:4]
labels=df.iloc[:, 4]
```

```
df['species'],categories =pd.factorize(df['species'])
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	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

Next steps: Generate code with df View recommended plots

```
df.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
df.isna().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

visualization of data:

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


x_train, x_test, y_train, x_test = train_test_split(x,y, test_size=0.2,random_state=42)


classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(x_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

```
print("target labels", df["species"].unique())
```

```
target labels ['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

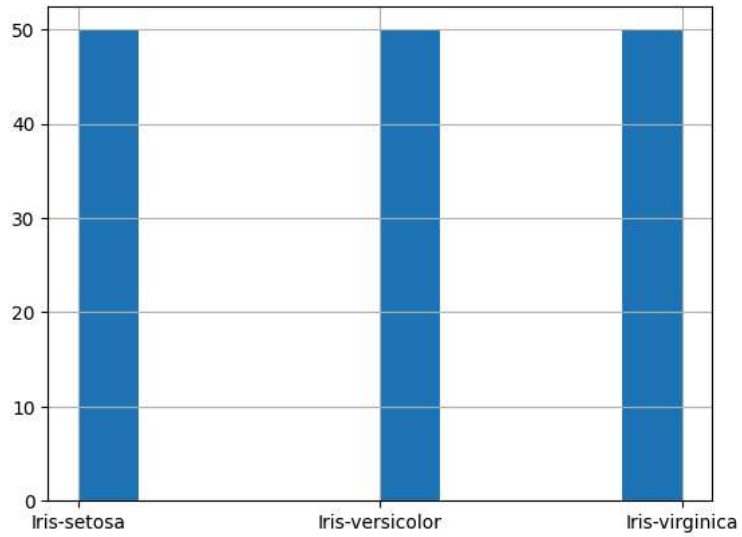
```
df['species'].value_counts()
```

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

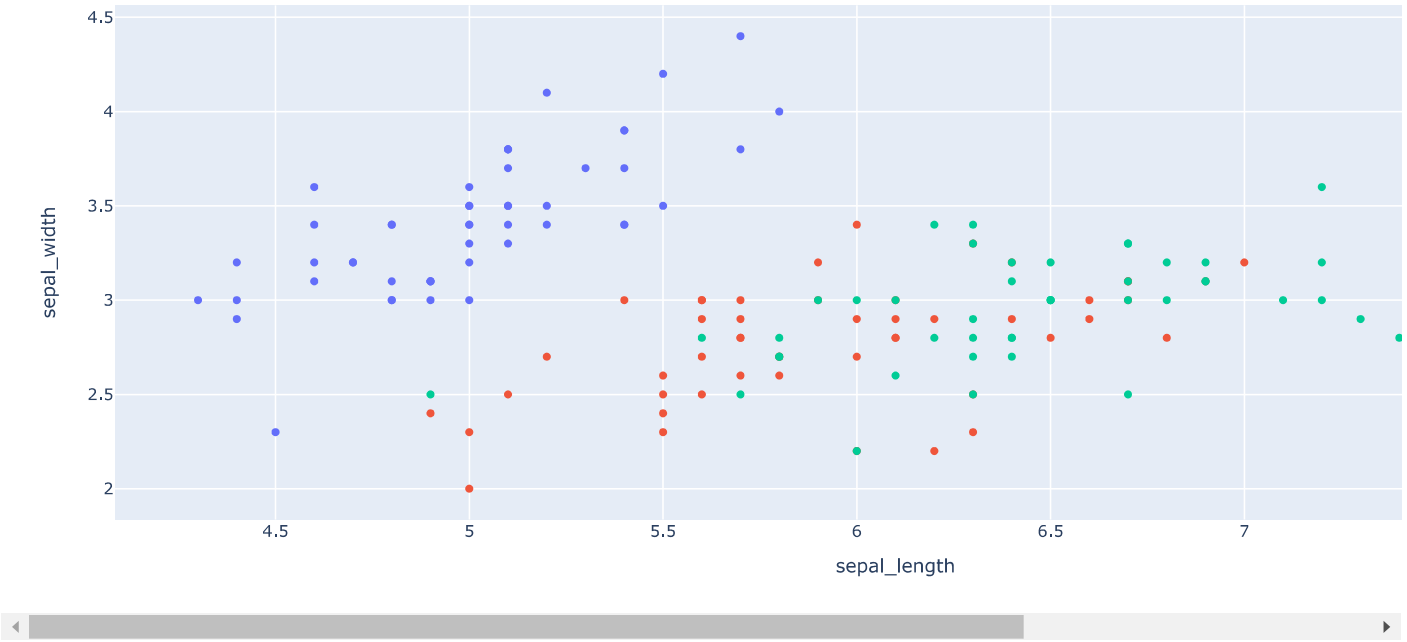
view the count plot of species feature using seaborn:

```
df['species'].hist()
```

```
<Axes: >
```




```
import plotly.express as px
fig = px.scatter(df, x="sepal_length",y = "sepal_width",color="species")
fig.show()
```



Split data into testing and training data

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(df.points, labels, test_size=0.2)
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
Standard_obj = StandardScaler()
Standard_obj.fit(x_train)
x_train_std = Standard_obj.transform(x_train)
x_test_std= Standard_obj.transform(x_test)
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 7,p=2, metric = 'minkowski')
knn.fit(x_train_std, y_train)
print('Training data accuracy {:.2f}'.format(knn.score(x_train_std, y_train)*100))
print('Testing data accuracy {:.2f}'.format(knn.score(x_test_std, y_test)*100))
```

```
Training data accuracy 97.50
Testing data accuracy 93.33
```