

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv("/content/iris.csv")
```

```
data
```

	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
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Id               150 non-null   int64
1   SepalLengthCm   150 non-null   float64
2   SepalWidthCm    150 non-null   float64
3   PetalLengthCm   150 non-null   float64
4   PetalWidthCm    150 non-null   float64
5   Species         150 non-null   object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
data.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

```
data.isnull()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
	0	False	False	False	False	False
	1	False	False	False	False	False
	2	False	False	False	False	False
	3	False	False	False	False	False
	4	False	False	False	False	False

	145	False	False	False	False	False
	146	False	False	False	False	False

```
data.isnull().sum()
```

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

```
data.head(10)
```

	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
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

```
data.columns
```

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

```
data.rename(columns={'id': 'id', 'SepalLengthCm': 'sepal_length', 'SepalWidthCm': 'sepal_width', 'PetalLengthCm': 'petal_length', 'PetalWidthCm':
```

```
plt.scatter(data['Species'], data['petal_length'])
plt.title("iris")
plt.xlabel('Species')
plt.ylabel('petal_length')
plt.show()
```

```
iris

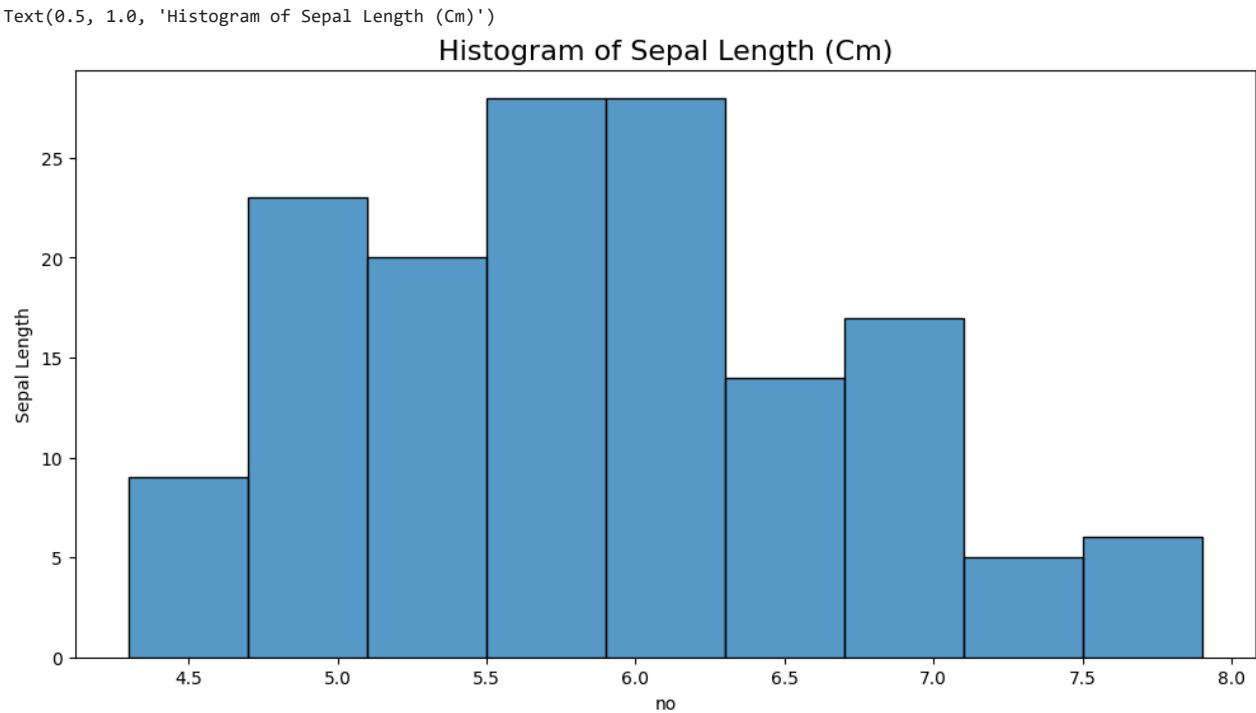
print(data["Species"].unique())

['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']

data['Species'].value_counts()

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

plt.figure(figsize=(12, 6))
sns.histplot(data.sepal_length)
plt.xlabel('no')
plt.ylabel('Sepal Length')
plt.title('Histogram of Sepal Length (Cm)', size=16)
```



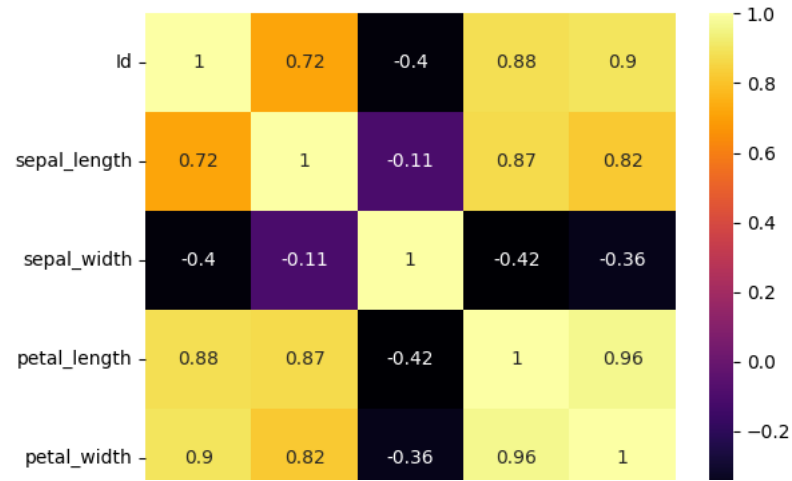
```
correlation = data.corr()
correlation

<ipython-input-25-521f87fcc686>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a f
correlation = data.corr()

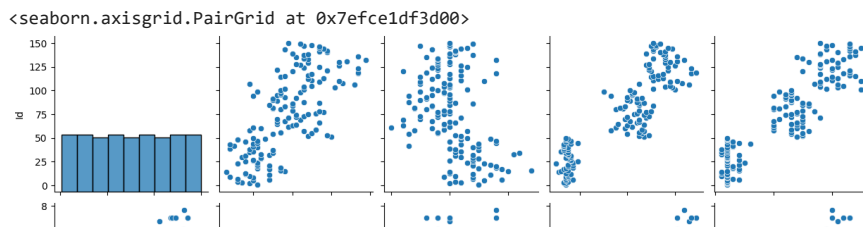
   Id  sepal_length  sepal_width  petal_length  petal_width
Id      1.000000      0.716676   -0.397729      0.882747      0.899759
sepal_length 0.716676      1.000000   -0.109369      0.871754      0.817954
sepal_width -0.397729  -0.109369      1.000000   -0.420516     -0.356544
petal_length 0.882747      0.871754   -0.420516      1.000000      0.962757
petal_width 0.899759      0.817954   -0.356544      0.962757      1.000000

sns.heatmap(data.corr(),annot=True,cmap='inferno')
```

```
<ipython-input-26-c46cf576d981>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a f
sns.heatmap(data.corr(),annot=True,cmap='inferno')
<Axes: >
```



```
sns.pairplot(data)
```



```
from wordcloud import WordCloud

text = " ".join(data.Species)

# Creating word_cloud with text as argument in .generate() method

word_cloud = WordCloud(collocations = False, background_color = 'white').generate(text)

# Display the generated Word Cloud

plt.imshow(word_cloud, interpolation='bilinear')

plt.axis("off")

plt.show()
```

☞

Iris
versicolor
virginica
setosa

```
X = data.iloc[:, :-1]
y = data.iloc[:, -1]
#X = data.drop('species', axis=1)
#y = data['species']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf = DecisionTreeClassifier()

clf.fit(X_train, y_train)

▼ DecisionTreeClassifier
DecisionTreeClassifier()

y_pred = clf.predict(X_test)

fig = plt.figure(figsize=(10, 8))
_ = tree.plot_tree(clf, feature_names=X.columns, class_names=y.unique(), filled=True)
```

