

Iris dataset description:

Iris Dataset Overview:

The Iris dataset is a classic dataset used in machine learning and analysis .It includes measurements of four

features from three iris flower species: Iris setosa,Iris versicolor and Iris virginica.

Features: 1.Sepal Length(cm) 2.Sepal Width(cm) 3.Petal Length(cm) 4.Ptal Width(cm)

Species: 1.Iris setosa 2.Iris versicolor 3.Iris virginica

Dataset Summary: .Samples=150 .Features=4 .Feature Types:Floating_point numbers(cm)
.Species=3

Load the dataset

```
In [1]: import pandas as pd
df=pd.read_csv("Iris.csv")
df.head()
```

```
Out[1]:
```

	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

Data Information

```
In [2]: df.shape
```

```
Out[2]: (150, 6)
```

```
In [3]: df.describe()
```

Out[3]:

	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

In [4]: `df["Species"].value_counts()`

Out[4]:

```

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

```

In [5]:

```

# deleting the column
df=df.drop(columns=['Id'])
df.head()

```

Out[5]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [6]: `df.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   SepalLengthCm   150 non-null   float64
1   SepalWidthCm    150 non-null   float64
2   PetalLengthCm   150 non-null   float64
3   PetalWidthCm    150 non-null   float64
4   Species         150 non-null   object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB

```

In [7]: `df.columns`

Out[7]:

```

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

```

```
In [8]: df.rename(columns={'SepalLengthCm': 'SepalLength', 'SepalWidthCm': 'SepalWidth', 'PetalLengthCm': 'PetalLength', 'PetalWidthCm': 'PetalWidth'})
```

```
In [9]: df.head()
```

```
Out[9]:
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Preprocessing of Dataset

```
In [10]: df.isnull().sum()
```

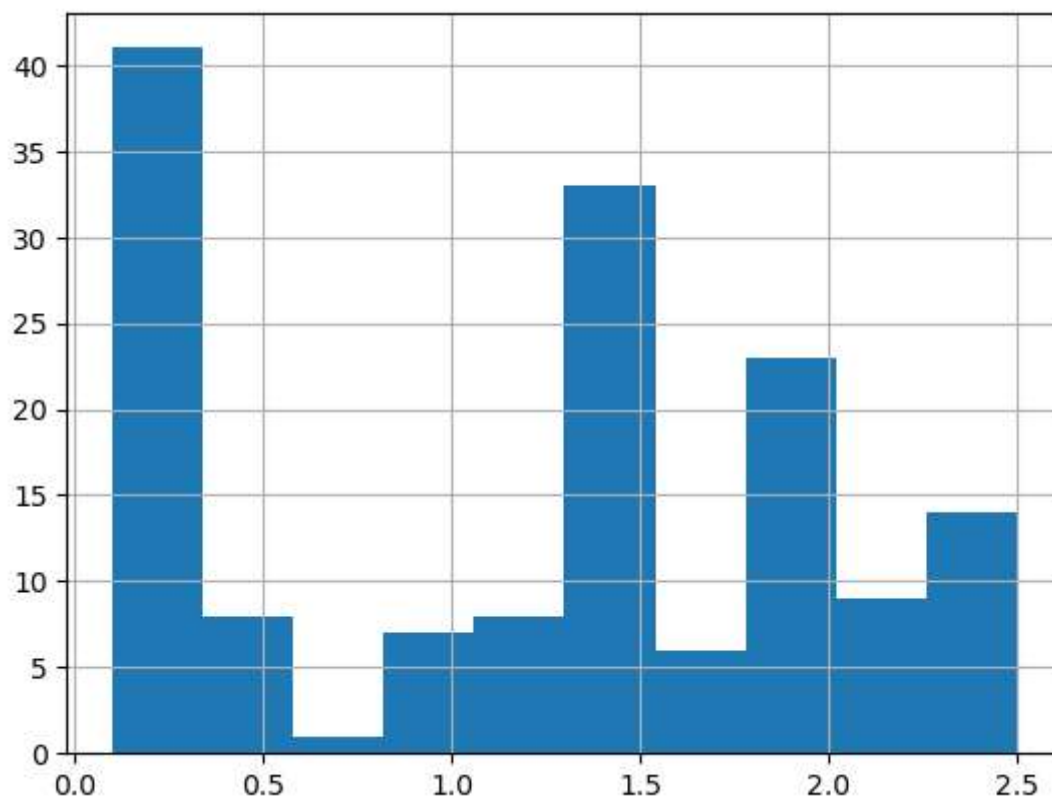
```
Out[10]: SepalLength    0  
SepalWidth    0  
PetalLength    0  
PetalWidth    0  
Species    0  
dtype: int64
```

Data Analysis

```
In [11]: import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [12]: df['PetalWidth'].hist()
```

```
Out[12]: <Axes: >
```



```
In [13]: df['PetalWidth'].value_counts()
```

```
Out[13]: 0.2    28
         1.3    13
         1.8    12
         1.5    12
         1.4     8
         2.3     8
         1.0     7
         0.4     7
         0.3     7
         0.1     6
         2.1     6
         2.0     6
         1.2     5
         1.9     5
         1.6     4
         2.5     3
         2.2     3
         2.4     3
         1.1     3
         1.7     2
         0.6     1
         0.5     1
         Name: PetalWidth, dtype: int64
```

```
In [14]: df["PetalLength"].value_counts()
```

```
Out[14]: 1.5    14
          1.4    12
          5.1     8
          4.5     8
          1.6     7
          1.3     7
          5.6     6
          4.7     5
          4.9     5
          4.0     5
          4.2     4
          5.0     4
          4.4     4
          4.8     4
          1.7     4
          3.9     3
          4.6     3
          5.7     3
          4.1     3
          5.5     3
          6.1     3
          5.8     3
          3.3     2
          5.4     2
          6.7     2
          5.3     2
          5.9     2
          6.0     2
          1.2     2
          4.3     2
          1.9     2
          3.5     2
          5.2     2
          3.0     1
          1.1     1
          3.7     1
          3.8     1
          6.6     1
          6.3     1
          1.0     1
          6.9     1
          3.6     1
          6.4     1
          Name: PetalLength, dtype: int64
```

```
In [15]: df["SepalWidth"].value_counts()
```

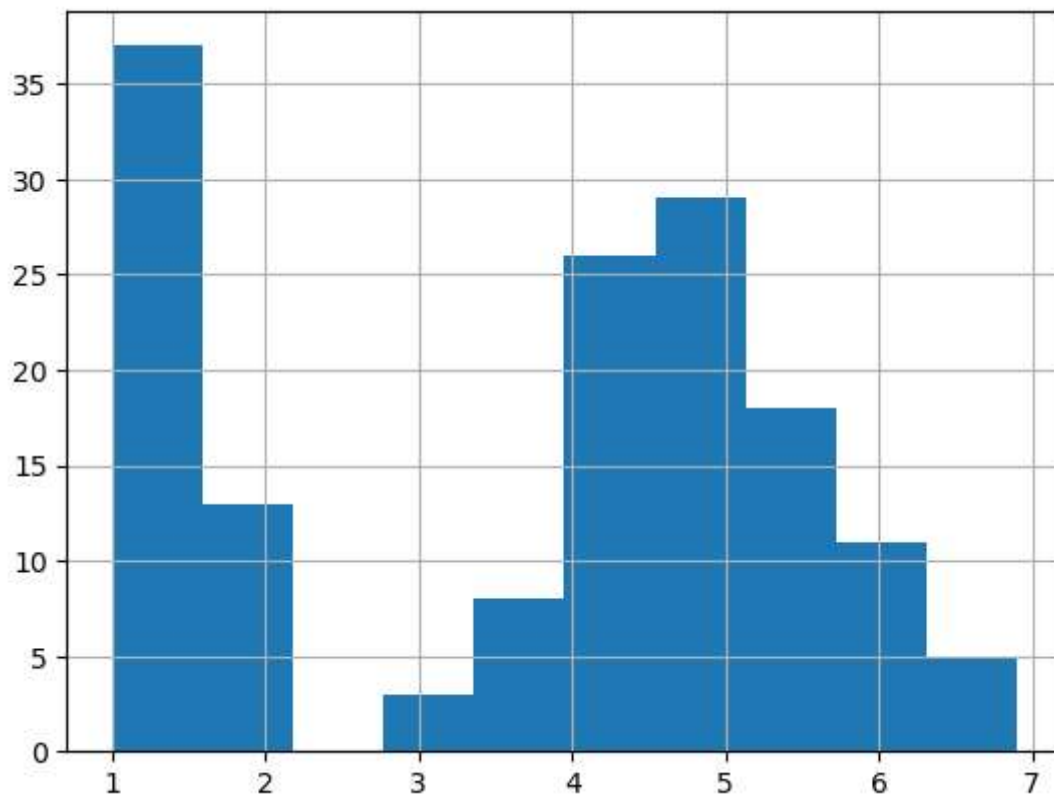
```
Out[15]: 3.0    26
          2.8    14
          3.2    13
          3.1    12
          3.4    12
          2.9    10
          2.7     9
          2.5     8
          3.5     6
          3.3     6
          3.8     6
          2.6     5
          2.3     4
          3.7     3
          2.4     3
          2.2     3
          3.6     3
          3.9     2
          4.4     1
          4.0     1
          4.1     1
          4.2     1
          2.0     1
          Name: SepalWidth, dtype: int64
```

```
In [16]: df["SepalLength"].value_counts()
```

```
Out[16]: 5.0    10
          5.1     9
          6.3     9
          5.7     8
          6.7     8
          5.8     7
          5.5     7
          6.4     7
          4.9     6
          5.4     6
          6.1     6
          6.0     6
          5.6     6
          4.8     5
          6.5     5
          6.2     4
          7.7     4
          6.9     4
          4.6     4
          5.2     4
          5.9     3
          4.4     3
          7.2     3
          6.8     3
          6.6     2
          4.7     2
          7.6     1
          7.4     1
          7.3     1
          7.0     1
          7.1     1
          5.3     1
          4.3     1
          4.5     1
          7.9     1
          Name: SepalLength, dtype: int64
```

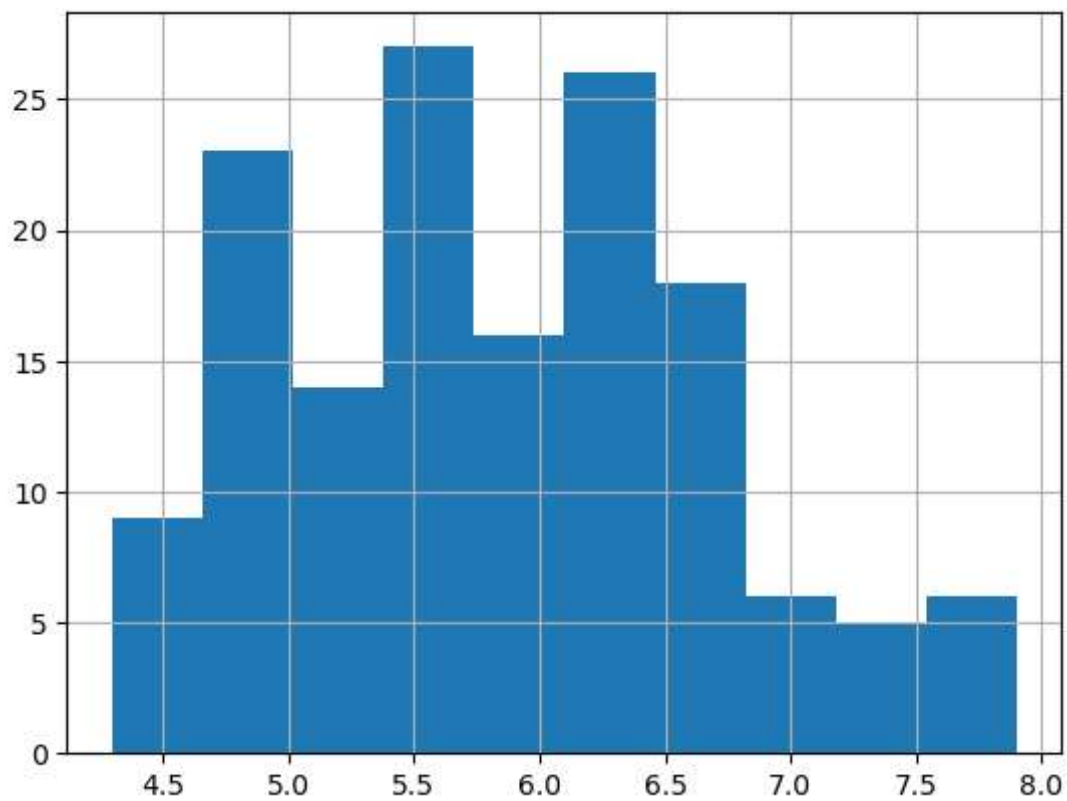
```
In [17]: df['PetalLength'].hist()
```

```
Out[17]: <Axes: >
```



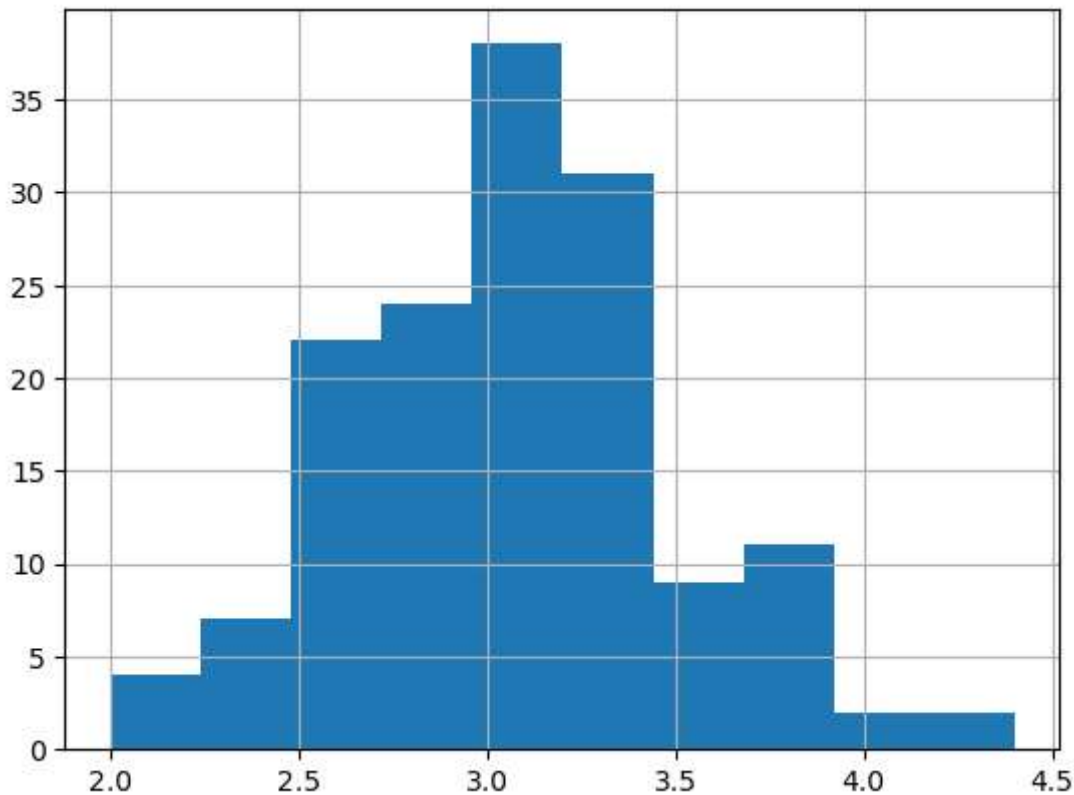
```
In [18]: df['SepalLength'].hist()
```

```
Out[18]: <Axes: >
```



```
In [19]: df['SepalWidth'].hist()
```

```
Out[19]: <Axes: >
```

Correlation matrix

In [20]: `df.corr()`

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_7136\1134722465.py:1: FutureWarning: The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
df.corr()
```

Out[20]:

	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength	1.000000	-0.109369	0.871754	0.817954
SepalWidth	-0.109369	1.000000	-0.420516	-0.356544
PetalLength	0.871754	-0.420516	1.000000	0.962757
PetalWidth	0.817954	-0.356544	0.962757	1.000000

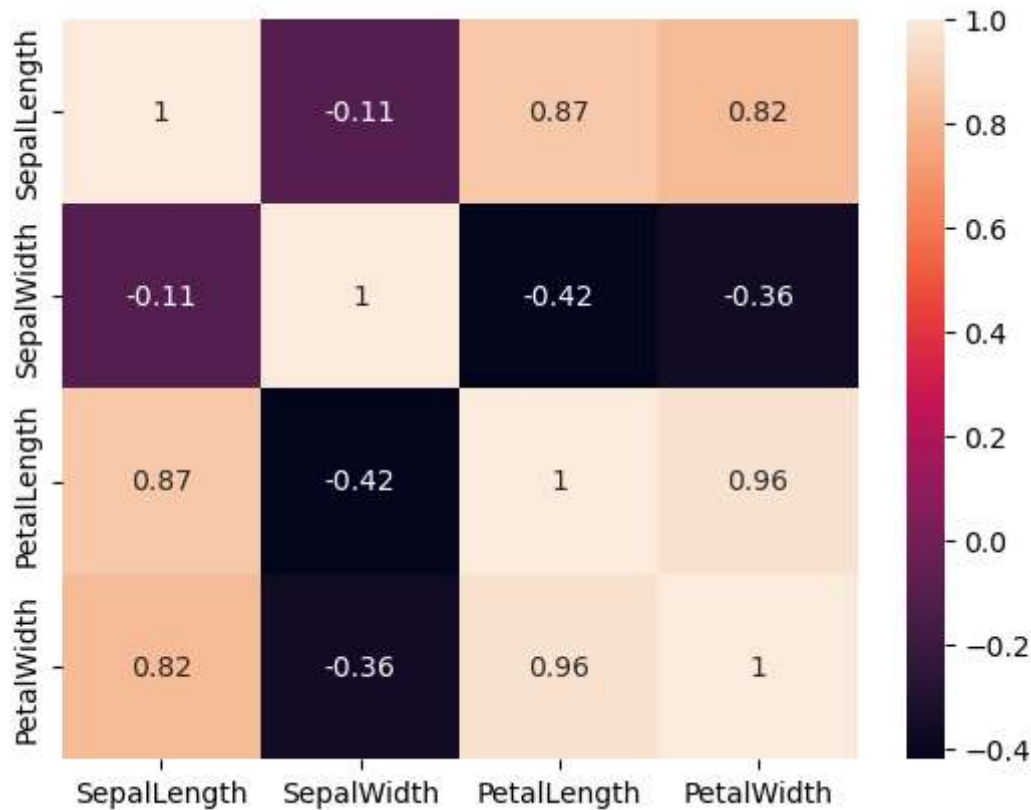
	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength	1.000000	-0.109369	0.871754	0.817954
SepalWidth	-0.109369	1.000000	-0.420516	-0.356544
PetalLength	0.871754	-0.420516	1.000000	0.962757
PetalWidth	0.817954	-0.356544	0.962757	1.000000

In [21]: `sns.heatmap(df.corr(),annot=True)`

C:\Users\Lenovo\AppData\Local\Temp\ipykernel_7136\4277794465.py:1: FutureWarning: The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
sns.heatmap(df.corr(),annot=True)
```

Out[21]: <Axes: >



train and test the model

```
In [22]: from sklearn.model_selection import train_test_split
X=df.drop(columns=['Species'])
y=df['Species']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state=42)
```

```
In [23]: from sklearn.linear_model import LogisticRegression
m=LogisticRegression()
m.fit(X_train,y_train)
```

```
Out[23]: LogisticRegression
LogisticRegression()
```

Accuracy of model

```
In [24]: print("Accuracy of logistic regression:",m.score(X_test,y_test)*100)
```

Accuracy of logistic regression: 100.0

accuracy of the model is 100%

```
In [25]: # deployment of a machine learning model
SepalLength=float(input("Enter your sepal length(between 4cm to 8cm):"))
```

```
SepalWidth=float(input("Enter your Sepal Width(between 2cm to 5cm):"))
PetalLength=float(input("Enter your ptal length( between 1cm to 7cm):"))
PetalWidth=float(input("Enter your Petal width(between 0cm to 3cm):"))

list=[SepalLength,SepalWidth,PetalLength,PetalWidth]

Newdf=pd.DataFrame([list])
y_p=m.predict(Newdf)
y_P
```

```
Enter your sepal length(between 4cm to 8cm):3.0
Enter your Sepal Width(between 2cm to 5cm):4.2
Enter your ptal length( between 1cm to 7cm):5.5
Enter your Petal width(between 0cm to 3cm):1.0
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does
not have valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(
```

```
Out[25]: array(['Iris-virginica'], dtype=object)
```

```
In [ ]:
```