

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: df=pd.read_csv(r'C:\Users\mohan\Downloads\archive (4)\IRIS.csv' )

In [3]: df.info()

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

In [4]: df.head()

Out[4]:
```

	sepal_length	sepal_width	petal_length	petal_width	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 [5]: df.tail()

Out[5]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

```


In [6]: df.describe()

Out[6]:
```

	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

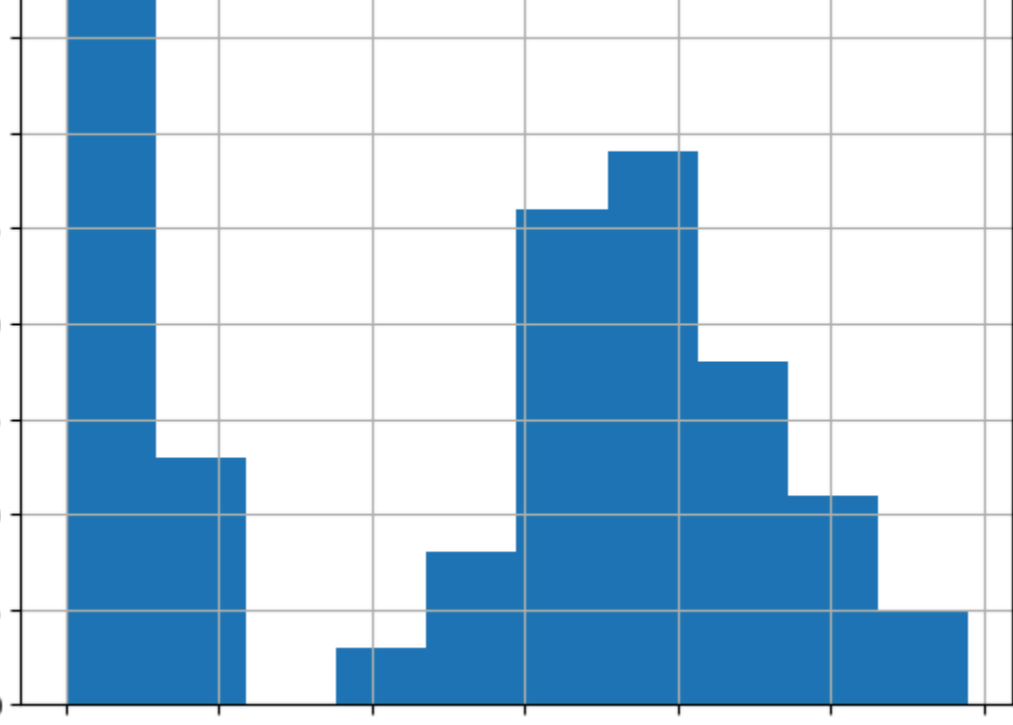
```


In [7]: df.isnull().sum()

Out[7]:
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64

In [8]: df['petal_length'].hist()

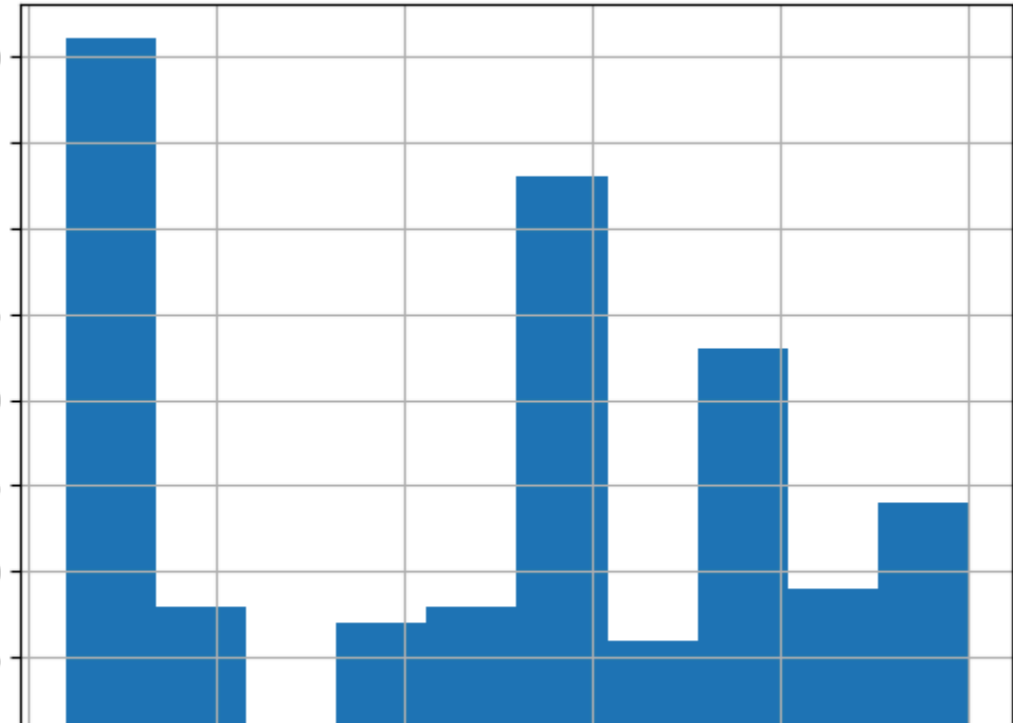
Out[8]:
<Axes: >
```



```


In [9]: df['petal_width'].hist()

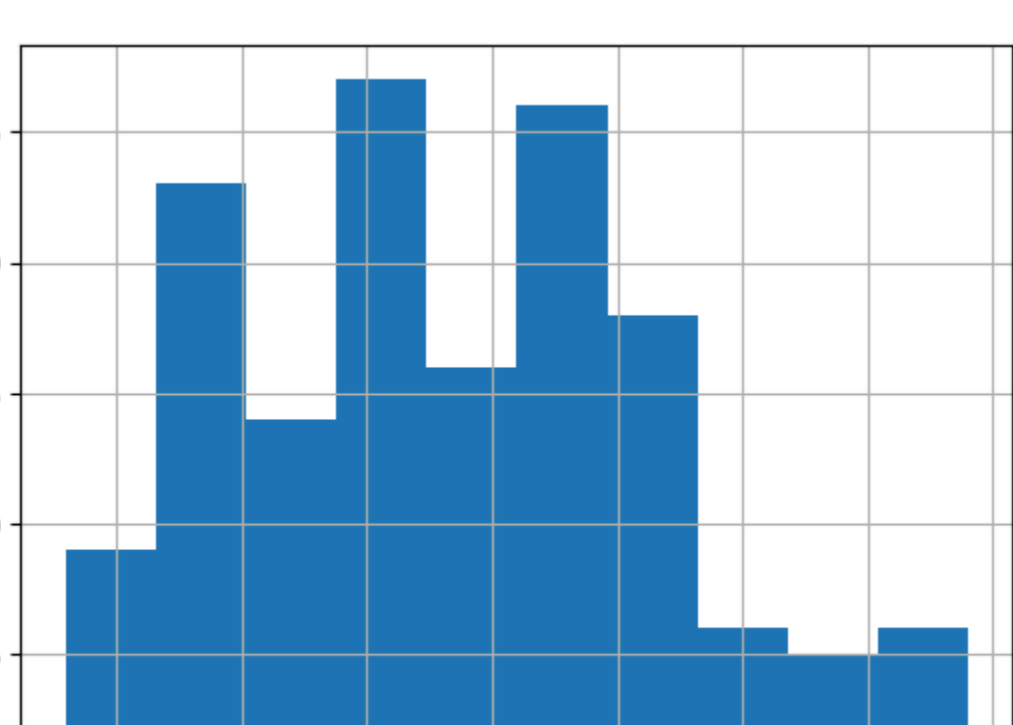
Out[9]:
<Axes: >
```



```


In [10]: df['sepal_length'].hist()

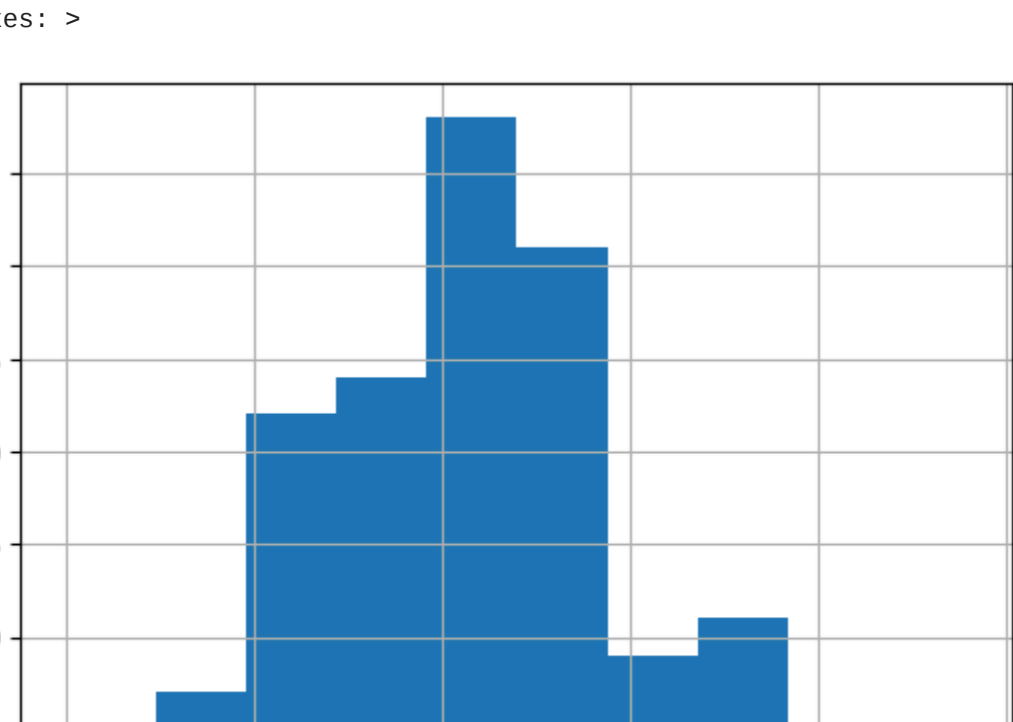
Out[10]:
<Axes: >
```



```


In [11]: df['sepal_width'].hist()

Out[11]:
<Axes: >
```



```


In [12]: colors=['red','green','blue']
Species=['Iris-setosa','Iris-versicolor','Iris-virginica']
for i in range(3):
    x=df[df['species']==Species[i]]
    plt.scatter(x['sepal_length'],x['sepal_width'],c=color[i],label=Species[i])
plt.title('Sepal length Vs Sepal width')
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.legend()

In [13]: colors=['orange','purple','yellow']
Species=['Iris-setosa','Iris-versicolor','Iris-virginica']
for i in range(3):
    x=df[df['species']==Species[i]]
    plt.scatter(x['petal_length'],x['petal_width'],c=color[i],label=Species[i])
plt.title('Petal Length Vs Petal width')
plt.xlabel('Petal length')
plt.ylabel('Petal width')
plt.legend()

In [14]: from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
df['species']=labelencoder.fit_transform(df['species'])
df.head()

Out[14]:
```

	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

```


In [15]: df.tail()

Out[15]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

```


In [16]: df.corr()

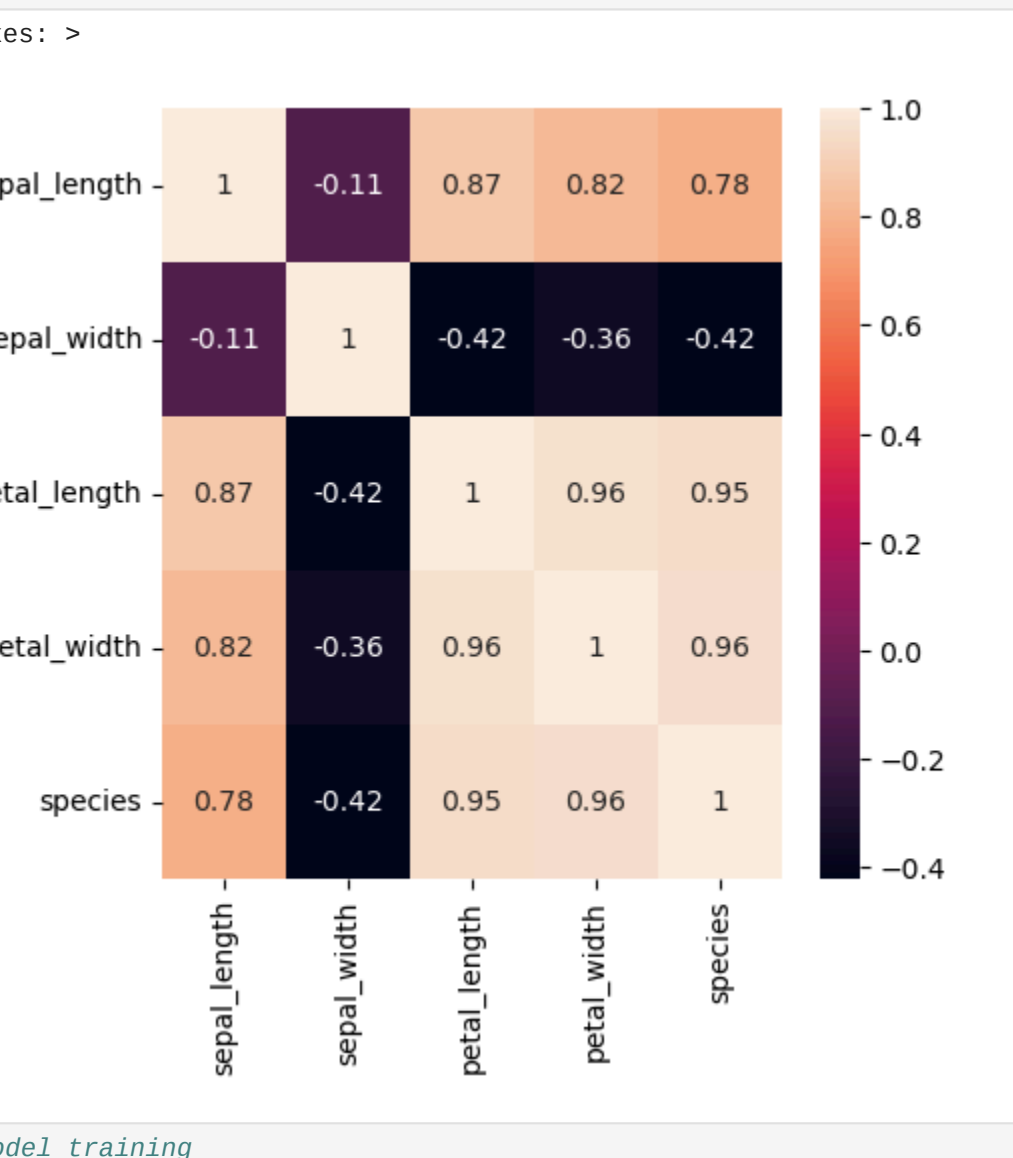
Out[16]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
sepal_length	1.000000	-0.109369	0.871754	0.817954	0.782561
sepal_width	-0.109369	1.000000	-0.420516	-0.356544	-0.419446
petal_length	0.871754	-0.420516	1.000000	0.962757	0.949043
petal_width	0.817954	-0.356544	0.962757	1.000000	0.956464
species	0.782561	-0.419446	0.949043	0.956464	1.000000

```


In [17]: corr=df.corr()
fig,ax=plt.subplots(figsize=(5,5))
sns.heatmap(corr,annot=True)

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



```


In [18]: #model training
x=df[['sepal_length','sepal_width','petal_length','petal_width']]
y=df['species']

In [19]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)

In [20]: from sklearn.linear_model import LogisticRegression
log=LogisticRegression(random_state=0)
log.fit(x_train,y_train)

Out[20]:
LogisticRegression
LogisticRegression(random_state=0)

In [21]: print(log.predict(x_test))

[2 1 0 2 0 2 0 1 1 2 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 2 1 1 2 0 2 0 0]

In [22]: print(y_test)

114 2
62 1
33 0
187 2
7 0
189 2
48 0
86 1
76 1
71 1
134 2
51 1
73 1
54 1
63 1
37 0
78 1
90 1
45 0
16 0
121 2
66 1
24 0
8 0
126 2
22 0
44 0
97 1
93 1
26 0
137 2
84 1
27 0
127 2
132 2
59 1
18 0
83 1
61 1
92 1
112 2
2 0
141 2
43 0
10 0
Name: species, dtype: int32

In [23]: import warnings
warnings.filterwarnings("ignore")
res=log.predict([[5.9,3.0,5.1,1.8]])
if(res==0):
    print("Iris-setosa")
elif(res==1):
    print("Iris-versicolor")
elif(res==2):
    print("Iris-virginica")
else:
    print("undefined")

Iris-virginica

In [ ]:
```