In [ ]:
```
IRIS FLOWER CLASSIFICATION TASK(3)
```

In [4]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [5]:
```
df = pd.read_csv('IRIS.csv')
```

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

Out[6]:

	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 [7]:
```
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 [8]:
```
df.transpose()
print(df)
```

```
     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
..            ...          ...           ...          ...             ...
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

[150 rows x 5 columns]
```

In [9]:
```
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 [11]:
```
df['species'] .value_counts()
```

Out[11]:
```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [9]:
```
df.isna().sum()
```

Out[9]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

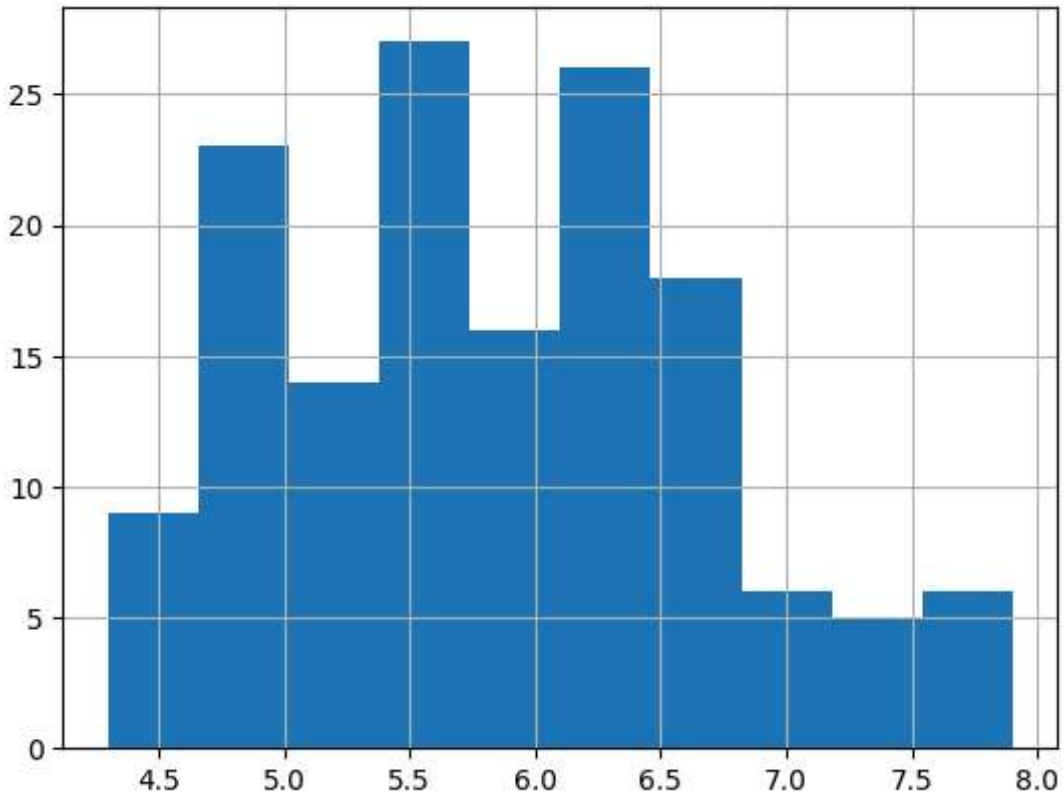
In [10]:
```
df.describe()
```

Out[10]:

	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

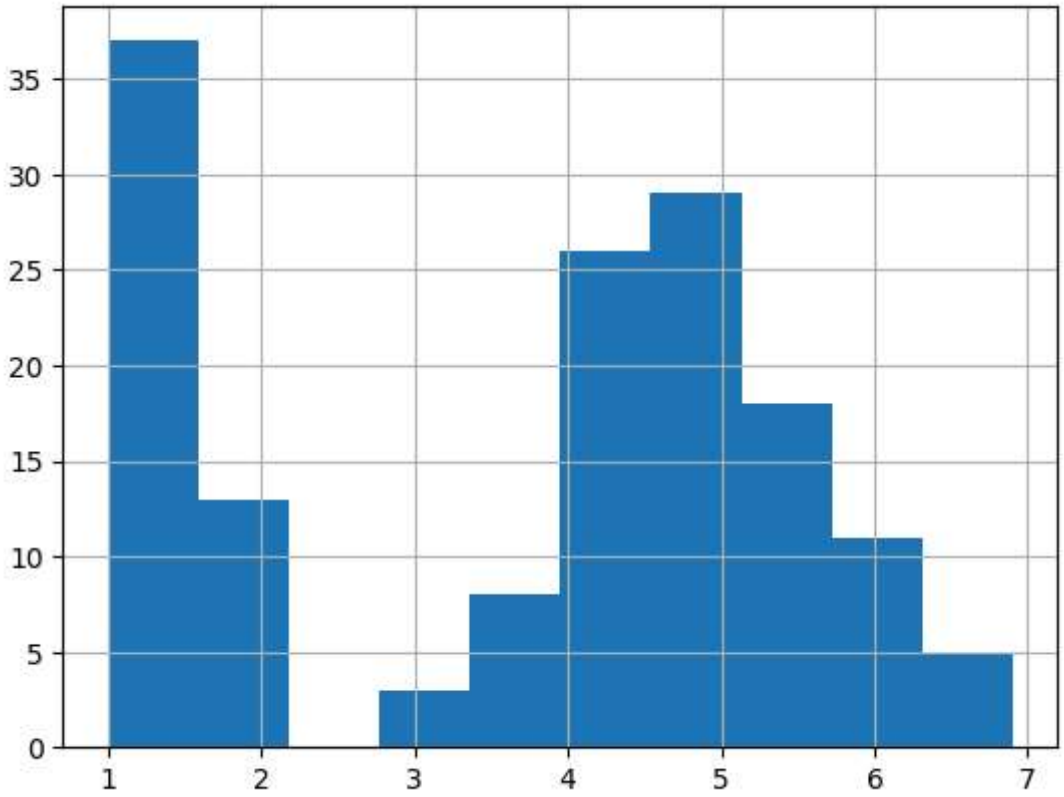
In [15]:
```
df['sepal_length'].hist()
```

Out[15]: <Axes: >



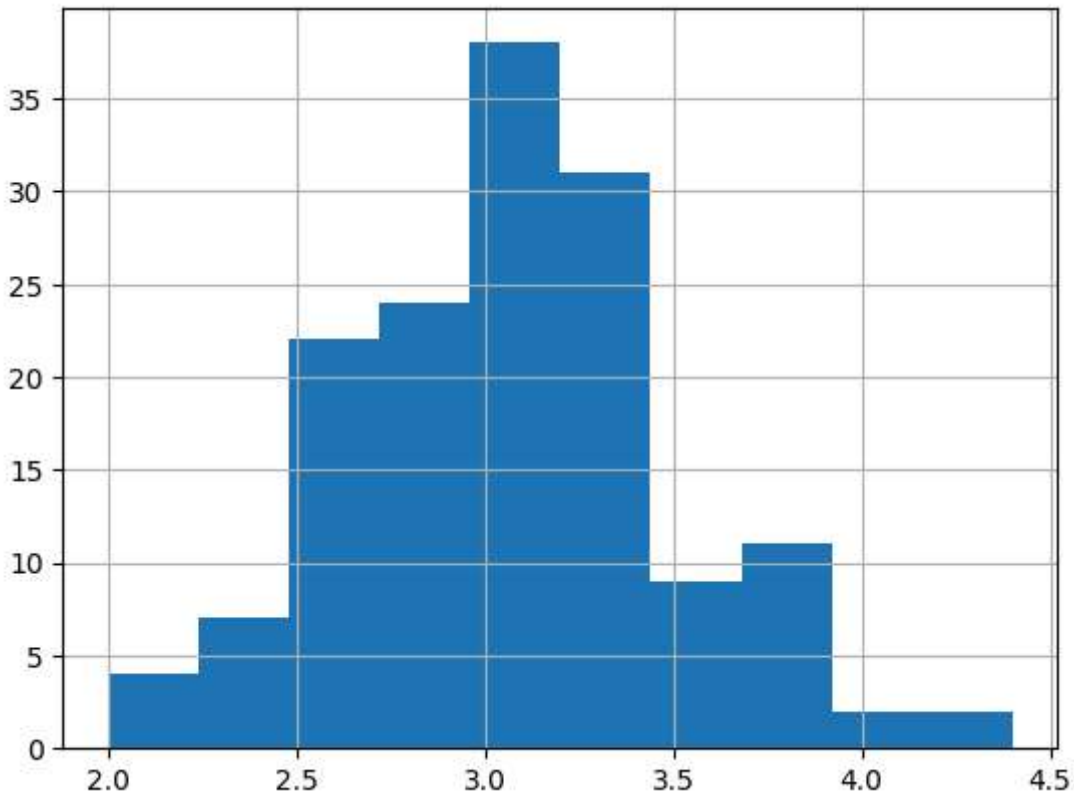
In [18]:
```
df['petal_length'].hist()
```

Out[18]: <Axes: >



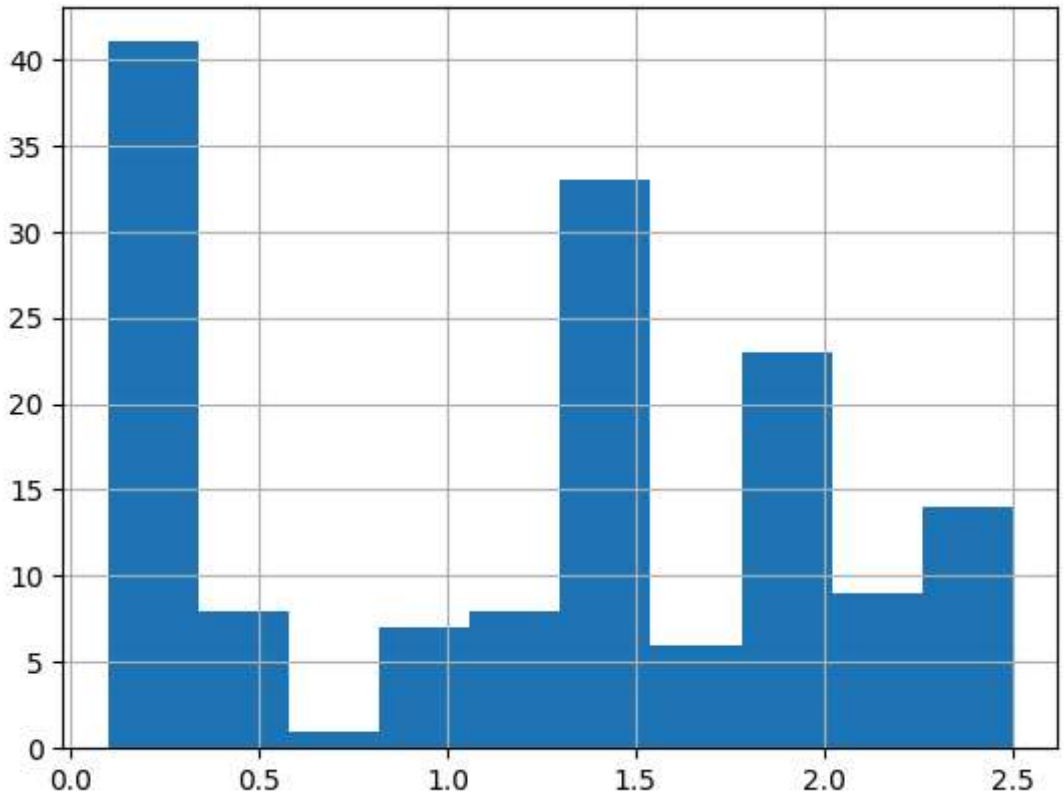
In [19]:
```
df['sepal_width'].hist()
```

Out[19]: <Axes: >



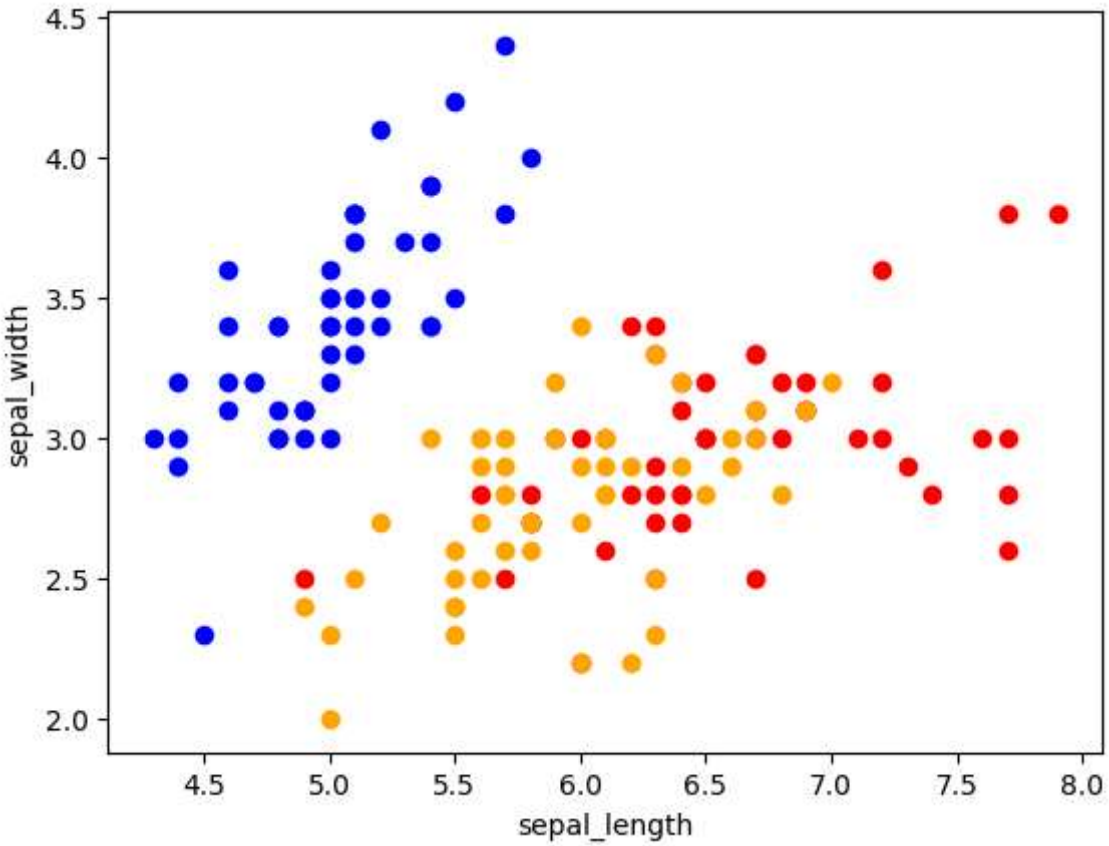
In [20]:
```
df['petal_width'].hist()
```

Out[20]: <Axes: >

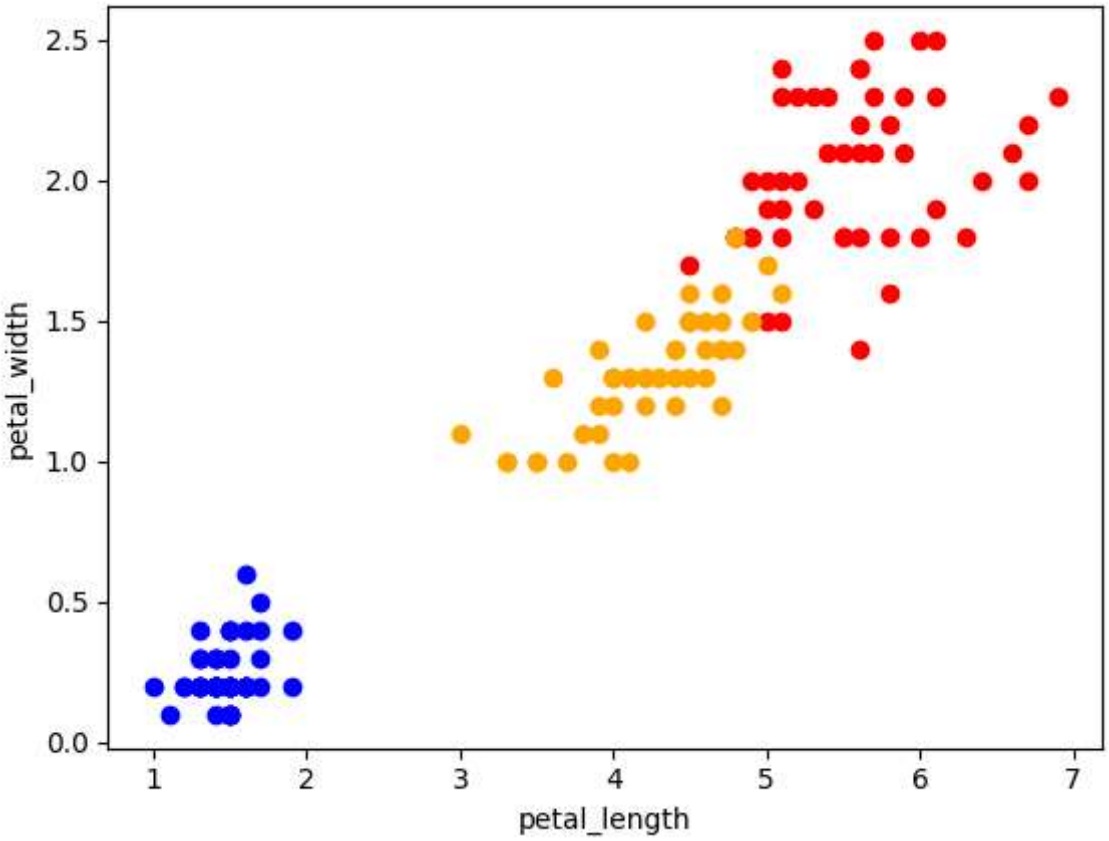


In [11]:
```
colors=['red','orange','blue']
species =['Iris-virginica','Iris-versicolor','Iris-setosa']
```

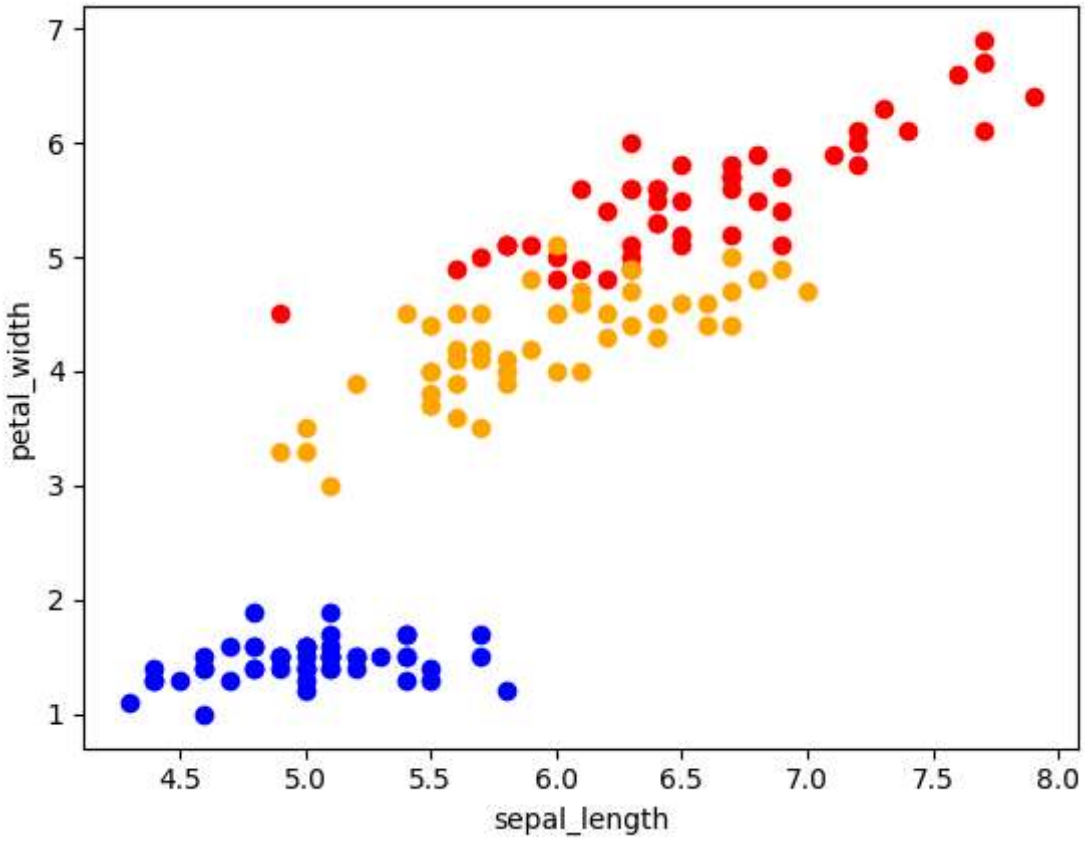
In [12]:
```
for i in range(3):
    x = df[df['species'] == species[i]]
    plt.scatter(x['sepal_length'],x['sepal_width'],c= colors[i], label = species[i]
plt.xlabel("sepal_length")
plt.ylabel("sepal_width")
plt.show()
```



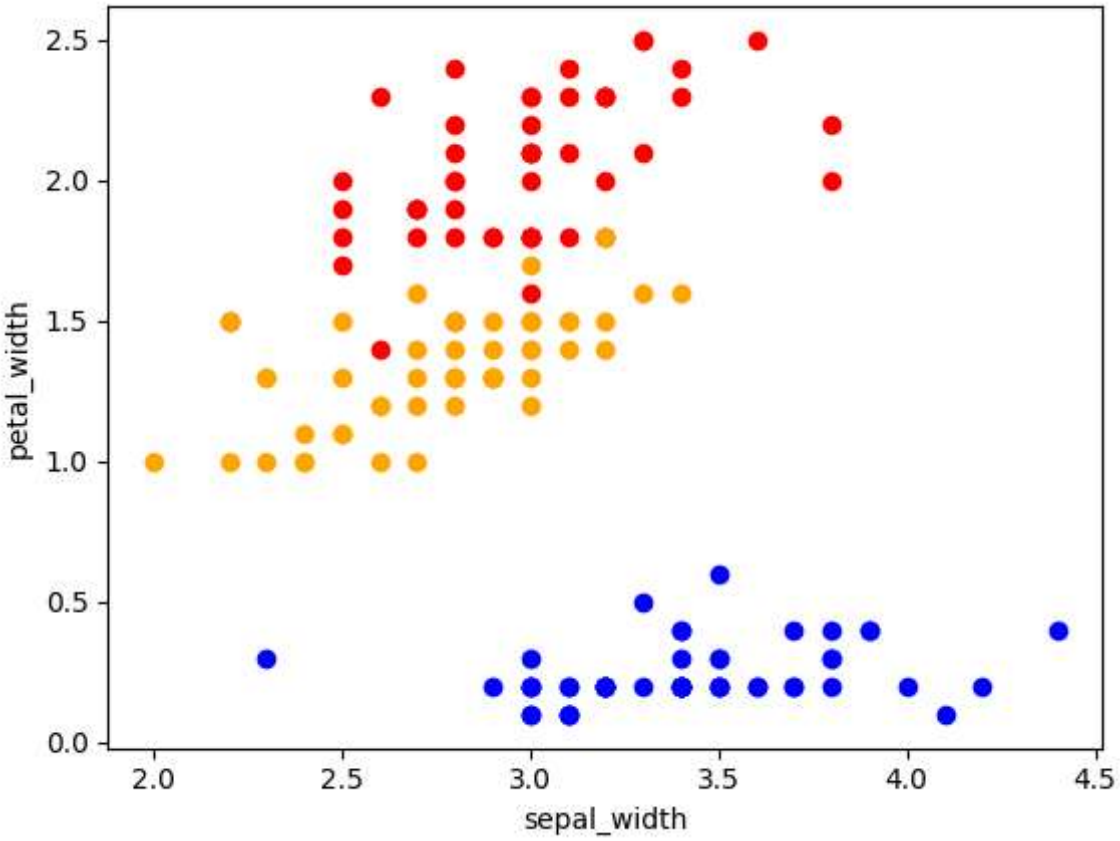
In [13]:
```
for i in range(3):
    x = df[df['species'] == species[i]]
    plt.scatter(x['petal_length'],x['petal_width'],c= colors[i], label = species[i]
plt.xlabel("petal_length")
plt.ylabel("petal_width")
plt.show()
```



In [14]:
```
for i in range(3):
    x = df[df['species'] == species[i]]
    plt.scatter(x['sepal_length'],x['petal_length'],c= colors[i], label = species[i
plt.xlabel("sepal_length")
plt.ylabel("petal_width")
plt.show()
```



In [15]:
```
for i in range(3):
    x = df[df['species'] == species[i]]
    plt.scatter(x['sepal_width'],x['petal_width'],c= colors[i], label = species[i])
plt.xlabel("sepal_width")
plt.ylabel("petal_width")
plt.show()
```



In [53]:
```
df.count()
df.transpose()
```

Out[53]:

	0	1	2	3	4	5	6	7	8	9	..
sepal_length	5.1	4.9	4.7	4.6	5.0	5.4	4.6	5.0	4.4	4.9	..
sepal_width	3.5	3.0	3.2	3.1	3.6	3.9	3.4	3.4	2.9	3.1	..
petal_length	1.4	1.4	1.3	1.5	1.4	1.7	1.4	1.5	1.4	1.5	..
petal_width	0.2	0.2	0.2	0.2	0.2	0.4	0.3	0.2	0.2	0.1	..
species	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	Iris-setosa	..

5 rows × 150 columns

In [18]:
```
from sklearn import metrics
```

In [21]:
```
from sklearn.preprocessing import LabelEncoder
import pandas as pd
le = LabelEncoder()
```

In [23]:
```
df['species']= le.fit_transform(df['species'])
df.head()
```

Out[23]:

	species
0	0
1	1
2	2
3	0
4	2

In [22]:
```
data = {'species': ['setosa', 'versicolor', 'virginica', 'setosa', 'virginica']}
df = pd.DataFrame(data)
```

In [56]:
```
label_encoder = LabelEncoder()
```

In [57]:
```
df['species_encoded'] = label_encoder.fit_transform(df['species'])
```

In [58]:
```
print(df)
```

```
     species  species_encoded
0     setosa                0
1  versicolor                1
2   virginica                2
3     setosa                0
4   virginica                2
```

In [59]:
```
df.head()
```

Out[59]:

	species	species_encoded
0	setosa	0
1	versicolor	1
2	virginica	2
3	setosa	0
4	virginica	2

In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import pandas as pd
```

In [15]:
```
column_names = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'clas
df = pd.read_csv('IRIS.csv')
```

In [18]:
```
X = df.drop('sepal_length', axis=1)
y = df['sepal_length']
```

In [19]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [83]:
```
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (120, 4)
X_test shape: (30, 4)
y_train shape: (120,)
y_test shape: (30,)
```