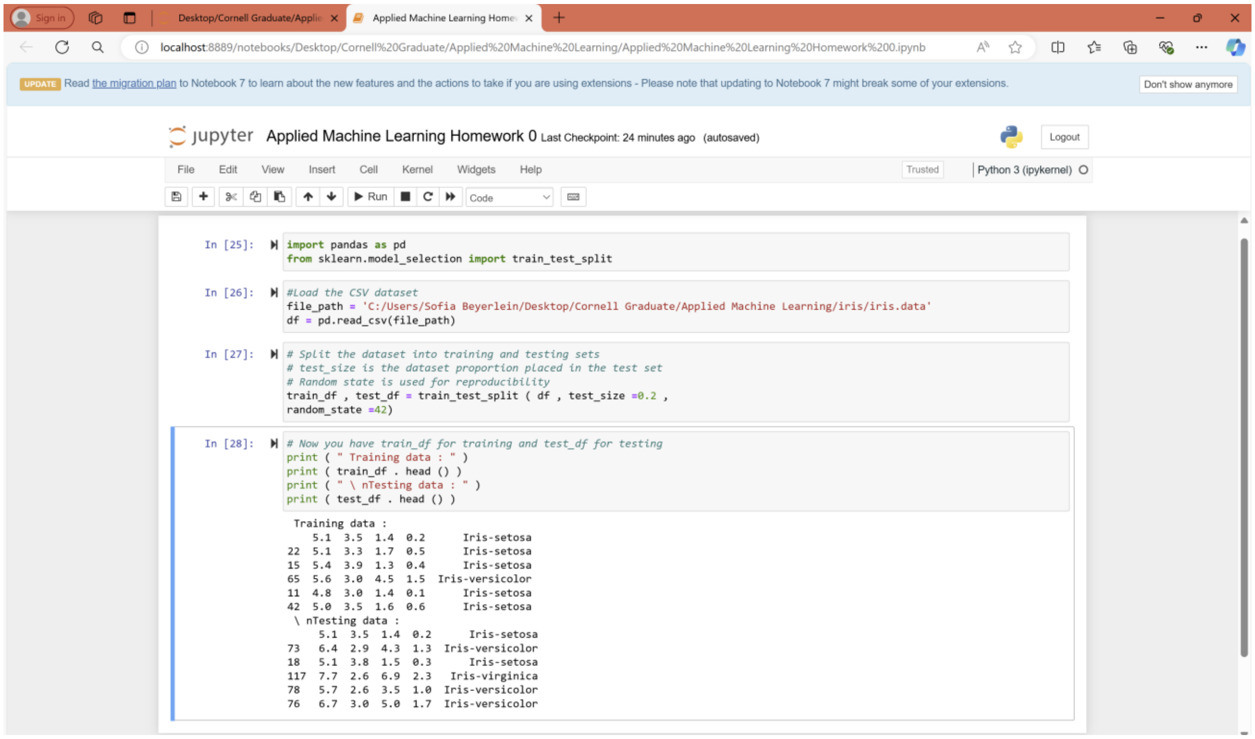
Applied Machine Learning Homework 0

1. There are 4 attributes features/attributes per sample. There are 3 different species (Iris Setosa, Iris Versicolour, and Iris Virginica) and there's 50 samples per each specie so a total of 150 instances.

2. Figure out how to parse the dataset you downloaded. Load the samples into an N × p array, where N is the number of samples and p is the number of attributes per sample. Additionally, create a N -dimensional vector containing each sample's label (species).


Sign in
Desktop/Cornell Graduate/Applie
Applied Machine Learning Home
localhost:8889/notebooks/Desktop/Cornell%20Graduate/Applied%20Machine%20Learning/Applied%20Machine%20Learning%20Homework%200.ipynb
Don't show anymore
jupyter
Applied Machine Learning Homework 0 Last Checkpoint: 24 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

In [25]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
```

In [26]:
```
#Load the CSV dataset
file_path = 'C:/Users/Sofia Beyerlein/Desktop/Cornell Graduate/Applied Machine Learning/iris/iris.data'
df = pd.read_csv(file_path)
```

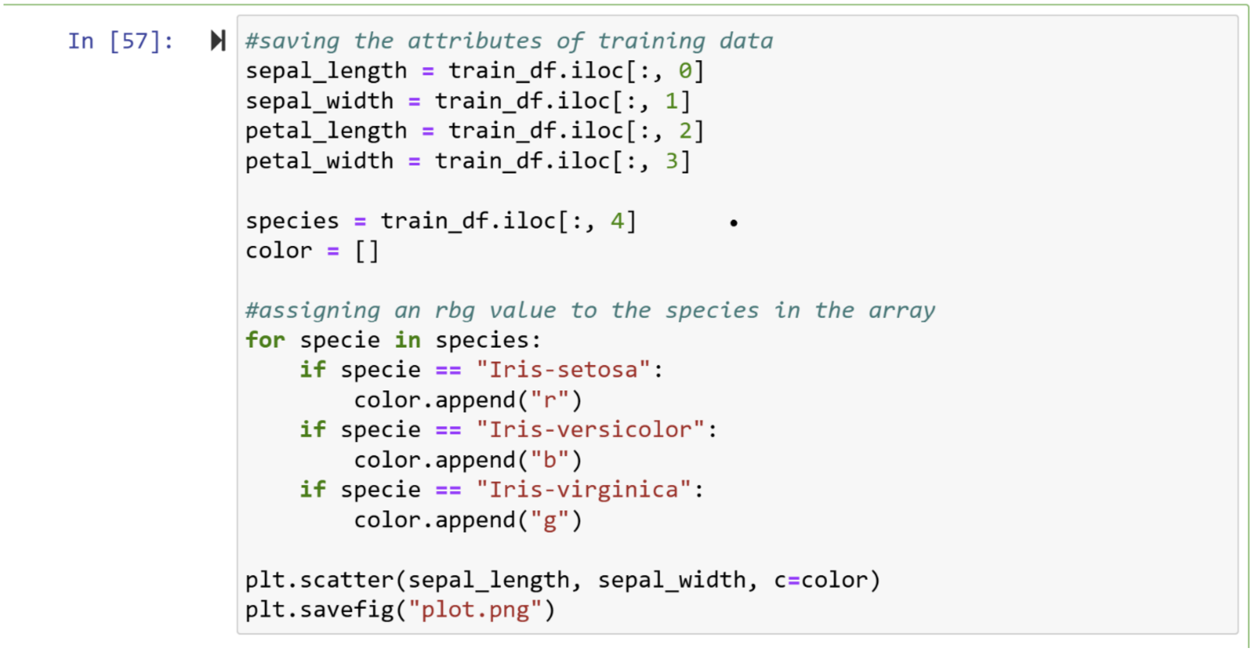
In [27]:
```
# Split the dataset into training and testing sets
# test_size is the dataset proportion placed in the test set
# Random state is used for reproducibility
train_df , test_df = train_test_split ( df , test_size =0.2 ,
random_state =42)
```

In [28]:
```
# Now you have train_df for training and test_df for testing
print ( " Training data : " )
print ( train_df . head () )
print ( " \ nTesting data : " )
print ( test_df . head () )
```

```
 Training data :
    5.1  3.5  1.4  0.2      Iris-setosa
22  5.1  3.3  1.7  0.5      Iris-setosa
15  5.4  3.9  1.3  0.4      Iris-setosa
65  5.6  3.0  4.5  1.5  Iris-versicolor
11  4.8  3.0  1.4  0.1      Iris-setosa
42  5.0  3.5  1.6  0.6      Iris-setosa
 \ nTesting data :
     5.1  3.5  1.4  0.2      Iris-setosa
73   6.4  2.9  4.3  1.3  Iris-versicolor
18   5.1  3.8  1.5  0.3      Iris-setosa
117  7.7  2.6  6.9  2.3   Iris-virginica
78   5.7  2.6  3.5  1.0  Iris-versicolor
76   6.7  3.0  5.0  1.7  Iris-versicolor
```

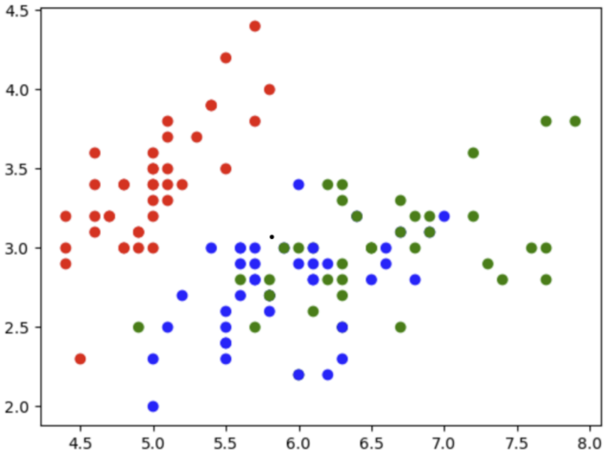
3. I plotted all the graphs using this snippet of code but replacing plt.scatter(sepal_length, sepal_width, color) with the other attribute arrays (e.g. plt.scatter(sepal_length, petal_length, color). I also want to clarify that I am

plotting the training data.

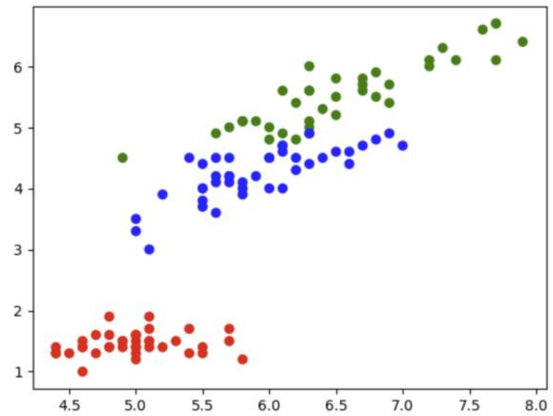

In [57]:

```
#saving the attributes of training data
sepal_length = train_df.iloc[:, 0]
sepal_width = train_df.iloc[:, 1]
petal_length = train_df.iloc[:, 2]
petal_width = train_df.iloc[:, 3]

species = train_df.iloc[:, 4]
color = []

#assigning an rbg value to the species in the array
for specie in species:
    if specie == "Iris-setosa":
        color.append("r")
    if specie == "Iris-versicolor":
        color.append("b")
    if specie == "Iris-virginica":
        color.append("g")

plt.scatter(sepal_length, sepal_width, c=color)
plt.savefig("plot.png")
```

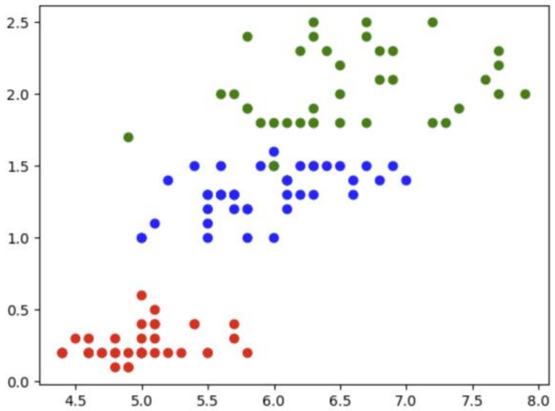
Sepal Length vs. Sepal Width



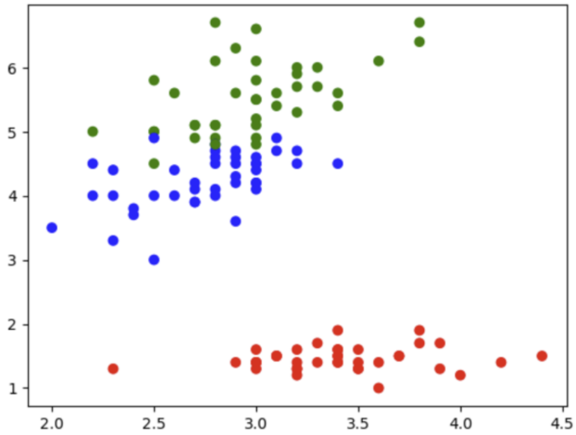
Sepal Length vs. Petal Length



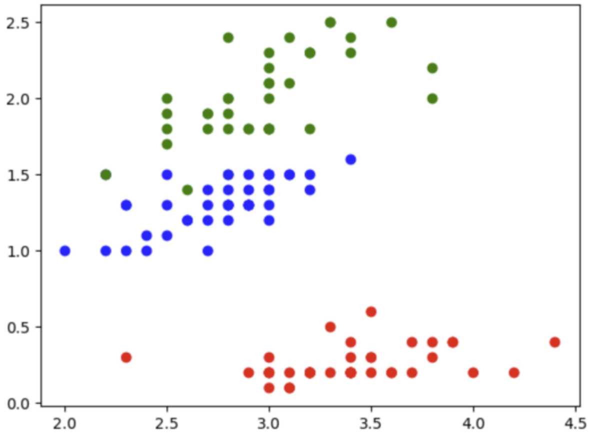
Sepal Length vs. Petal Width



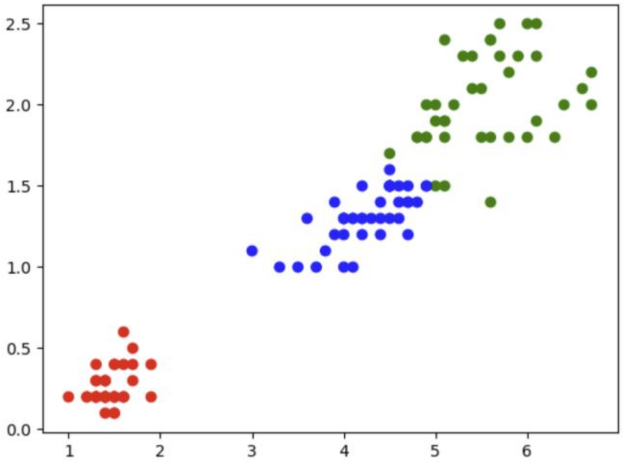
Sepal Width vs. Petal Length



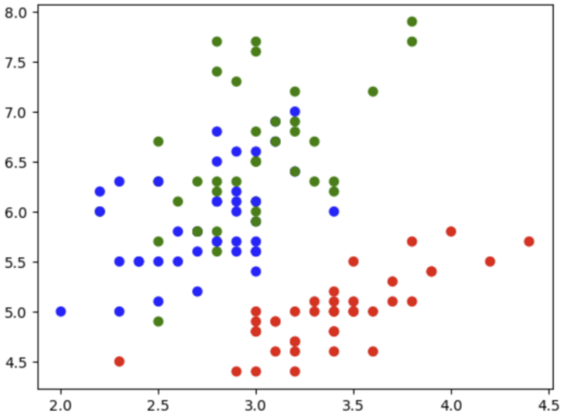
Sepal Width vs. Petal Width



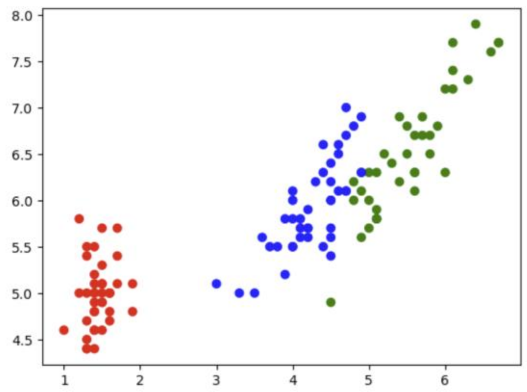
Petal Length vs. Petal Width



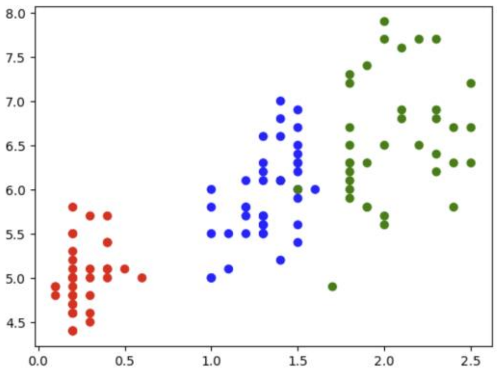
Sepal Width vs. Sepal Length



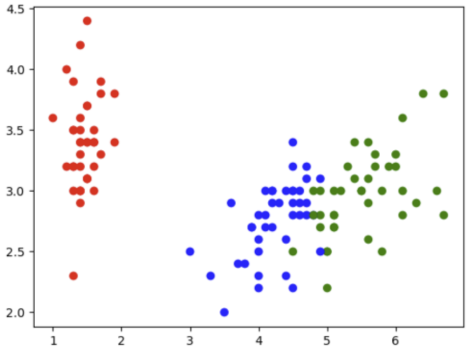
Petal Length vs. Sepal Length



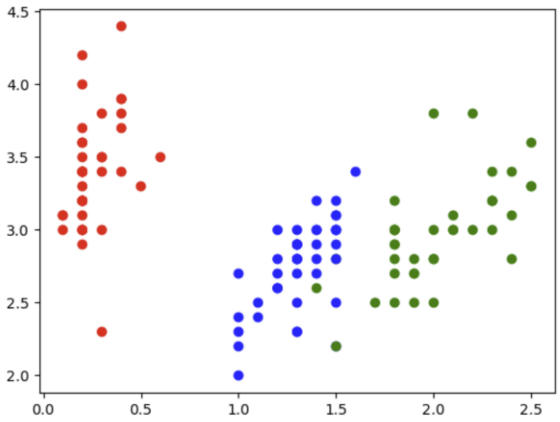
Petal Width vs. Sepal Length



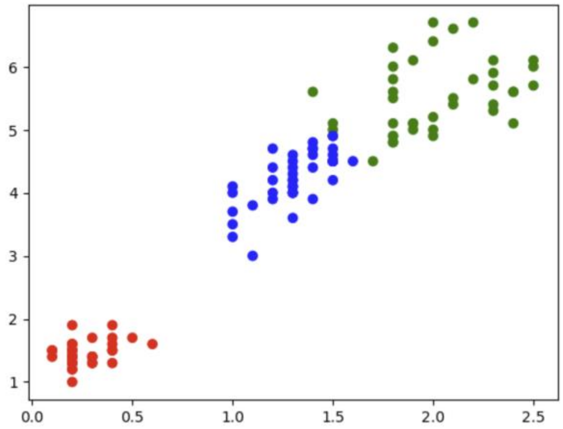
Petal Length vs. Sepal Width



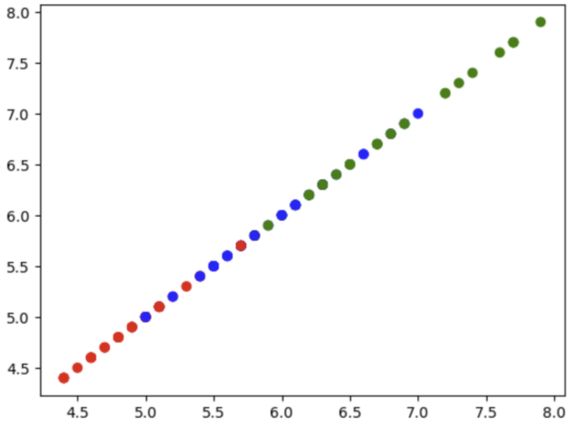
Petal Width vs. Sepal Width



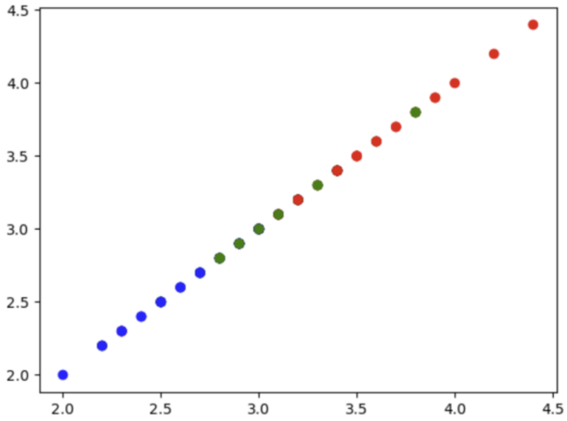
Petal Width vs. Petal Length



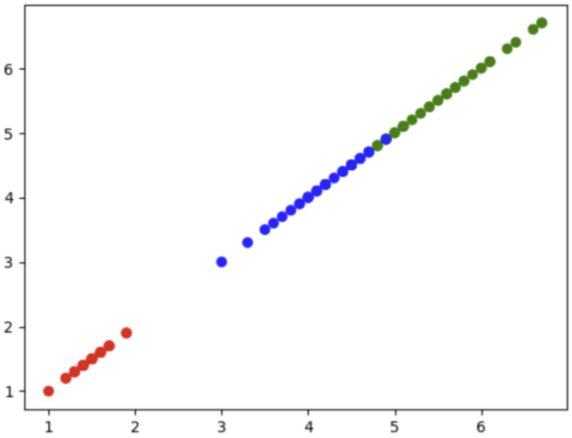
Sepal Length vs. Sepal Length



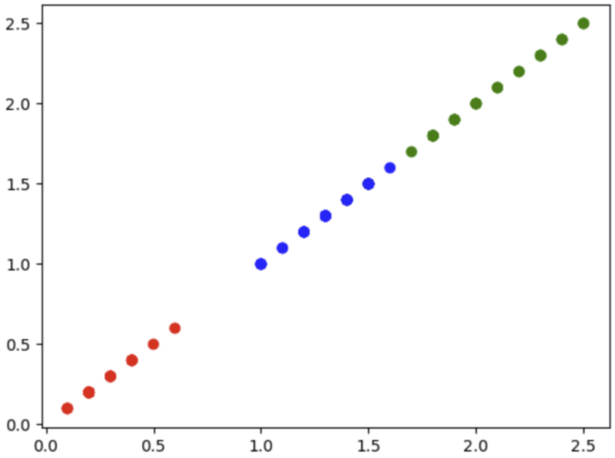
Sepal Width vs. Sepal Width



Petal Length vs. Petal Length



Petal Width vs. Petal Width


2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5