Iris Flowers Classification ML Project

The main objective of the project is to perform a basic supervised machine learning on the iris dataset using logistic regression as the classifier. For visualization, various plots such as the scatter plot, correlation heatmap and pair-plots have been used. Finally, predictions are made using the logistic regression function and accuracy score and a classification report is displayed.

In [1]:

```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\admin\anaconda3\lib
\site-packages (0.11.2)
Requirement already satisfied: scipy>=1.0 in c:\users\admin\anaconda3\l
ib\site-packages (from seaborn) (1.7.3)
Requirement already satisfied: numpy>=1.15 in c:\users\admin\anaconda3
\lib\site-packages (from seaborn) (1.21.5)
Requirement already satisfied: pandas>=0.23 in c:\users\admin\anaconda3
\lib\site-packages (from seaborn) (1.4.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\admin\anacon
da3\lib\site-packages (from seaborn) (3.5.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\anac
onda3\lib\site-packages (from matplotlib>=2.2->seaborn) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\admin\anac
onda3\lib\site-packages (from matplotlib>=2.2->seaborn) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\admin\a
naconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\admin\anaconda3
\lib\site-packages (from matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\admin\anaco
```

In [56]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
log_model = LogisticRegression(solver='lbfgs', max_iter=1000)
```

In [20]:

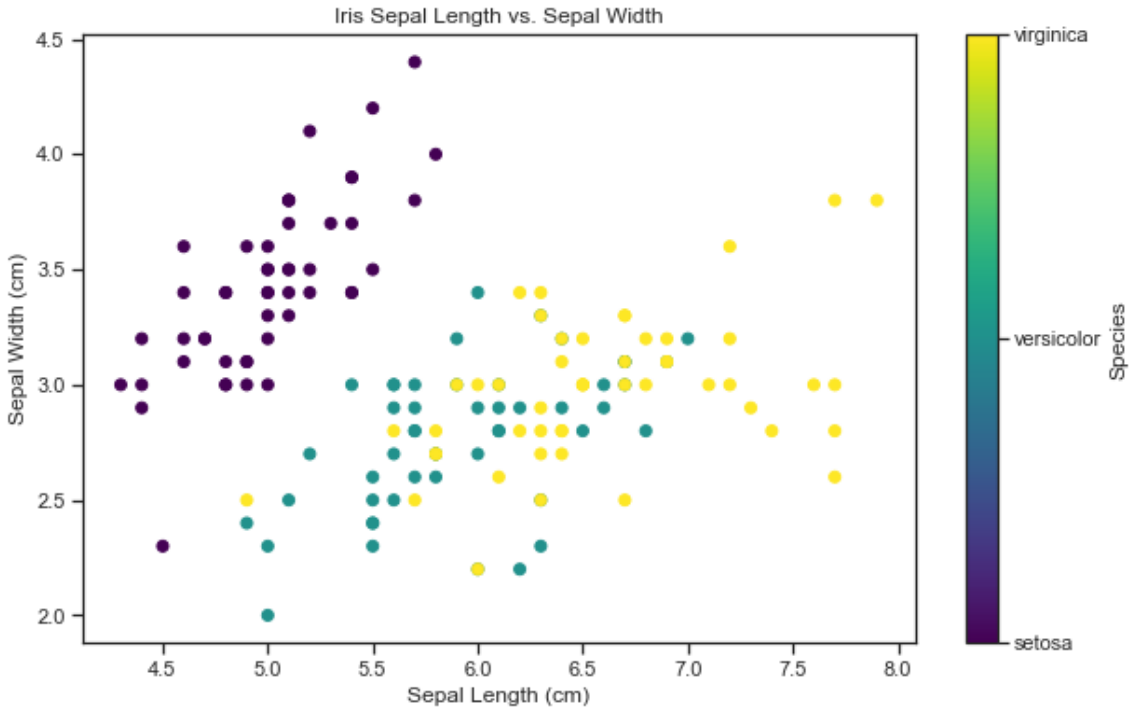
```
# Load the Iris dataset
from sklearn.datasets import load_iris
iris = load_iris()
```

In [21]:

```
# Create a DataFrame from the iris data
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
iris_df['target'] = iris.target
```

In [22]:

```
# Visualize the data
plt.figure(figsize=(10, 6))
plt.scatter(iris_df['sepal length (cm)'], iris_df['sepal width (cm)'],
            c=iris_df['target'], cmap='viridis')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.title('Iris Sepal Length vs. Sepal Width')
plt.colorbar(ticks=[0, 1, 2], label='Species',
             format=plt.FuncFormatter(lambda val, _: iris.target_names[int(val)]))
plt.show()
```

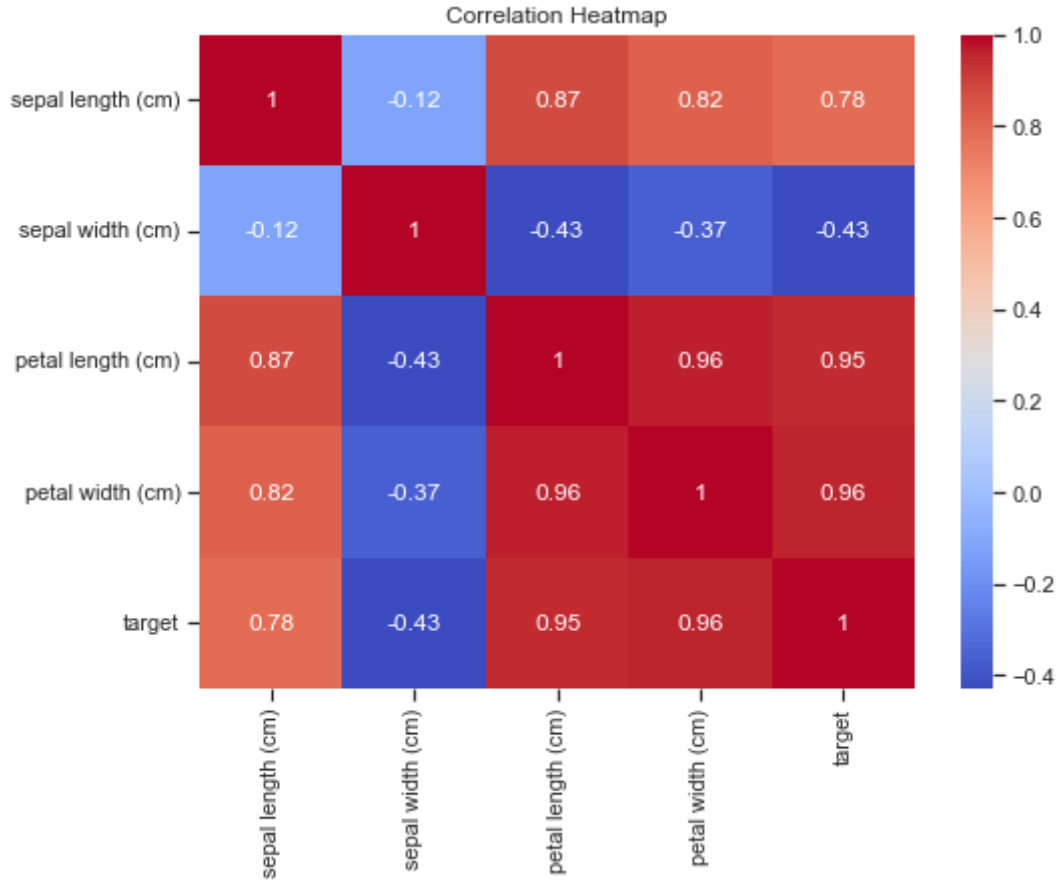


In [23]:

```
# Calculate the correlation matrix
corr_matrix = iris_df.corr()
```

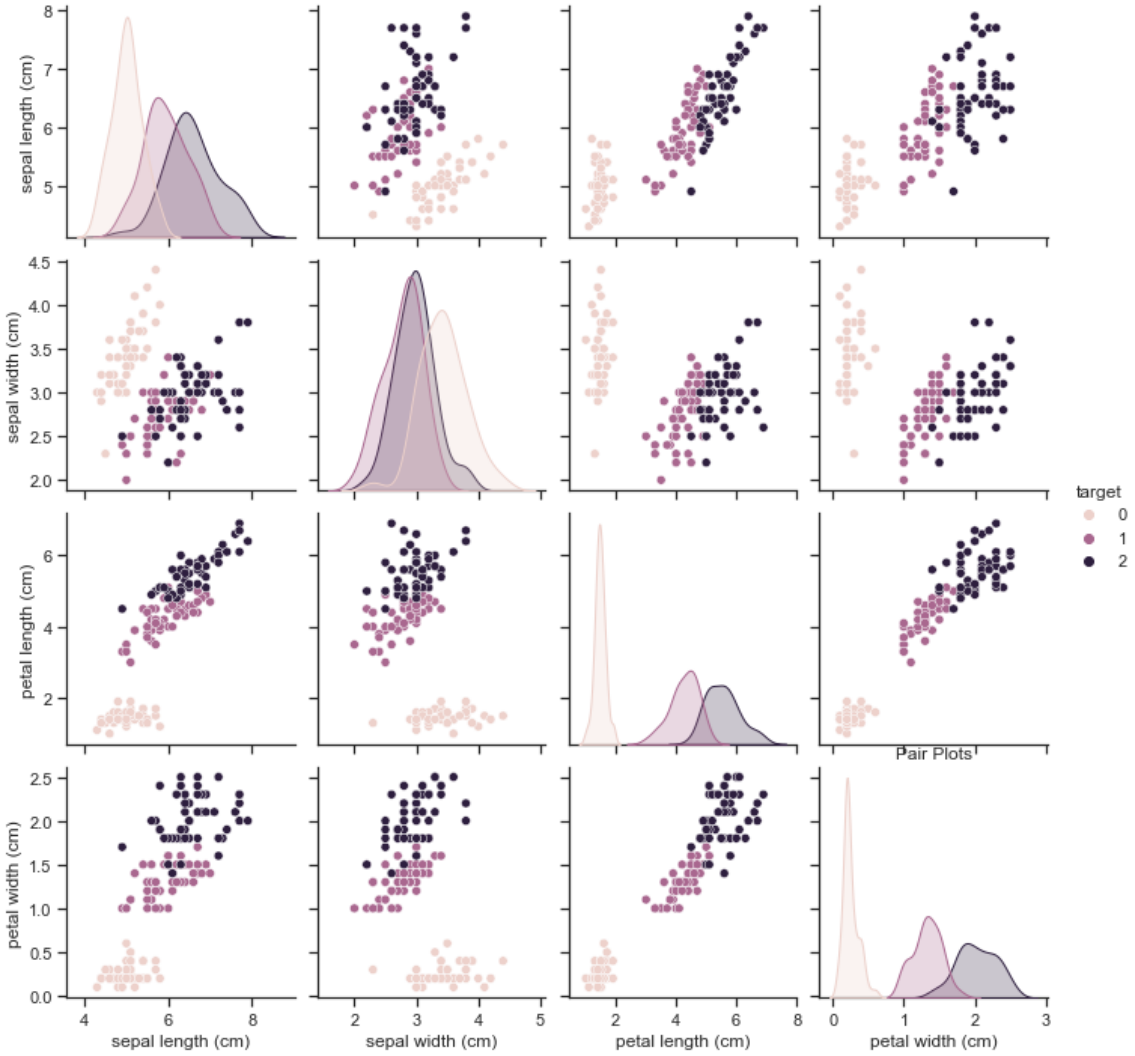
In [24]:

```
# Plot the correlation heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



In [25]:

```
# Plot pair plots
sns.set(style="ticks")
sns.pairplot(iris_df, hue="target")
plt.title('Pair Plots')
plt.show()
```



In [26]:

```
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(
    iris_df[iris.feature_names], iris_df['target'], test_size=0.2, random_state=42)
```

In [27]:

```
# Create a logistic regression model
model = LogisticRegression()
```

In [32]:

```
log_reg = LogisticRegression(solver='lbfgs',class_weight='balanced', max_iter=10000)
```

In [33]:

```
# Train the model
model.fit(X_train, y_train)
```

Out[33]:

```
LogisticRegression()
```

In [45]:

```
# Make predictions on the test set
y_pred = model.predict(X_test)
y_pred
```

Out[45]:

```
array([0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 1, 0])
```

In [42]:

```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

In [57]:

```
classification_report = classification_report(y_test, y_pred)
print("Classification Report:\n", classification_report)
```

```
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       1.00      1.00      1.00        10

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25
```