datascience-task3

January 5, 2024

1 Importing necessary libraries

[31]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix
```

2 Load the Iris dataset

[32]:
```
df=pd.read_excel("E:\\codsoft Data science\\IRIS.xlsx")
```

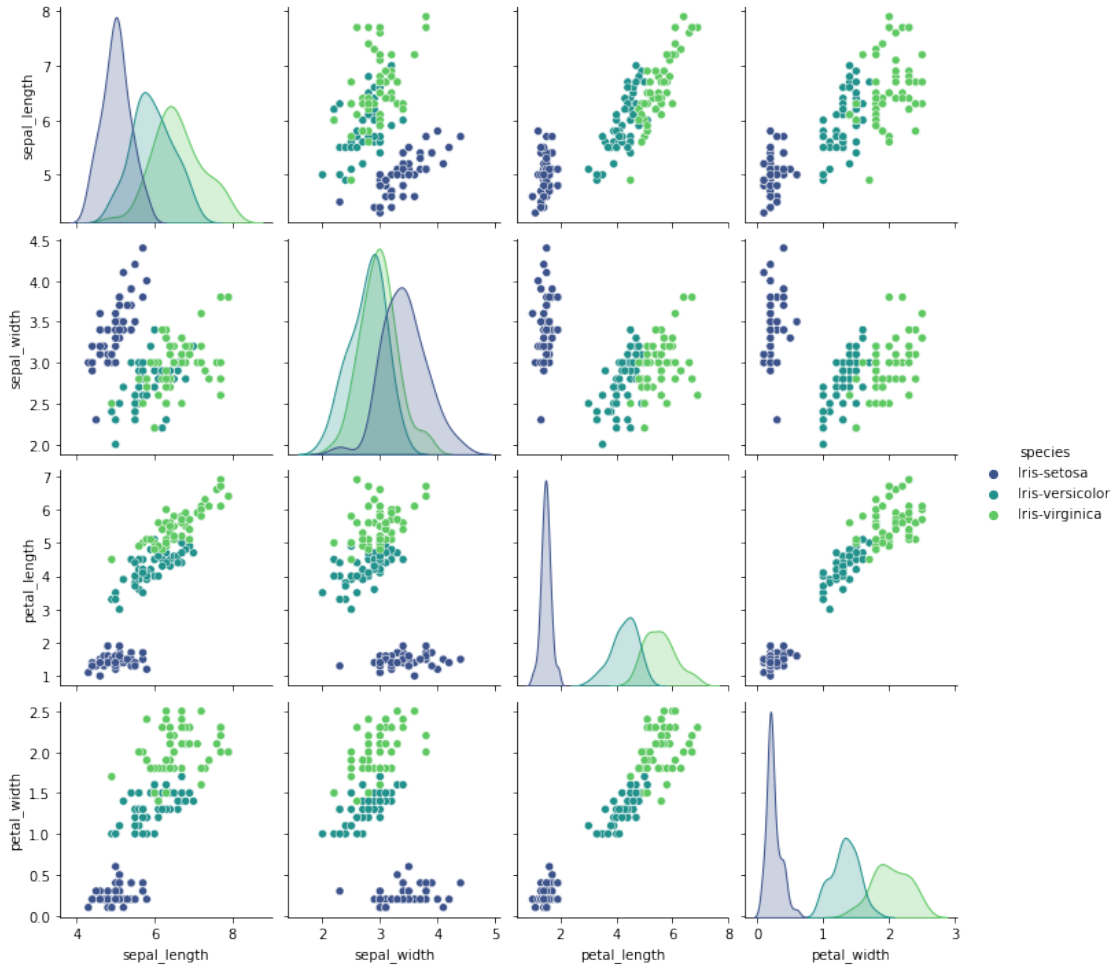
[33]:
```
# Display the first few rows of the dataset

df.head()
```

[33]:

	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

3 Visualization

[34]:
```
sns.pairplot(df, hue='species', palette='viridis')
plt.show()
```



[35]:
```
# Define features (X) and target variable (y)
X = df.drop('species', axis=1)
y = df['species']
```

[36]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[37]:
```
# Initialize and train the K-Nearest Neighbors classifier
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
```

[37]:
```
KNeighborsClassifier(n_neighbors=3)
```

[38]:
```
# Make predictions on the test set
predictions = model.predict(X_test)
```

[39]:
```
# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
report = classification_report(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)

print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:\n", report)
print("Confusion Matrix:\n", conf_matrix)
```

```
Accuracy: 1.00
Classification Report:
                  precision    recall  f1-score   support

     Iris-setosa       1.00      1.00      1.00        10
 Iris-versicolor       1.00      1.00      1.00         9
  Iris-virginica       1.00      1.00      1.00        11

        accuracy                           1.00        30
       macro avg       1.00      1.00      1.00        30
    weighted avg       1.00      1.00      1.00        30

Confusion Matrix:
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

[ ]: