Data Analysis With Python – Module 13 – Case Studies

Case Study #2: Iris Classification

This case study will use data science and machine learning to classify Iris flowers into 3 species:

- Iris-setosa
- Iris-versicolor
- Iris-virginica

You will use the features: petal length/width and sepal length/width to predict which Iris species a flower belongs to.

Please perform the following steps to complete this case study:

1. Create a new empty Jupyter Notebook.
2. Import all the modules required for:
   - numpy
   - pandas
   - matplotlib
   - seaborn
   - KNeighborsClassifier
3. Read the Iris.csv file into a Pandas DataSet called: iris
   - Use the pandas read_csv method.
   - Make sure you only have one index column.
4. Use the describe method to display some stats about the data.
5. Use a Seaborn scatterplot to show x=sepal length, y=sepal width, hue=Species.
   - If you get warnings, use named arguments like:
     - scatterplot(x="x axis column name", y="y axis column name")
   - What does the plot tell you about the data? Be specific. Don't focus on the numbers as much as the visual attributes.
   - Insert a markdown cell and note your observations. Superficial answers will lose marks.
6. Use a Seaborn scatterplot to show x=petal length, y=petal width, hue=Species.
   - If you get warnings, use named arguments like:
     - scatterplot(x="x axis column name", y="y axis column name")
   - What does the plot tell you about the data? Be specific. Don't focus on the numbers as much as the visual attributes.
   - Insert a markdown cell and note your observations. Superficial answers will lose marks.
7. Use a Seaborn pairplot to show the entire DataFrame. Use the hue parameter.
   - What does the plot tell you about the data? Be specific. Don't focus on the numbers as much as the visual attributes.
   - Insert a markdown cell and note your observations. Superficial answers will lose marks.

8. Split the data into training and testing data.
    - Prepare your X and y, using appropriate variable names:
        - X: Drop the Species column.
        - y: Specify the Species column.
    - Use sklearn train_test_split to split the data.
9. Create the model and fit it to the training data.
    - Create an sklearn KNeighborsClassifier model, with k=1.
    - Use the fit method to fit it to the training data.
10. Predict values based on testing data.
    - Use the predict method to predict values with the x testing data and store them in a variable.
11. Print out the classification report for the y test data and the predictions.
12. Use the knn.score method to print out a simplified score for the model.
13. Repeat steps 9 to 12 using k=3, k=5, k=10, k=30, k=50.
14. Insert a markdown cell and explain which k value gives the best results, and why you think that is. Be specific. Superficial answers will lose marks.
15. Choose the k value you believe performed the best, and pick two different sets of numbers directly from the iris dataframe to make predictions using your selected model. See if your model predicts the correct type of iris. For example:
    - pred = knn.predict([[sepallength, sepalwidth, petallength, petalwidth]])
    - Insert a markdown cell and compare your two predictions to the actual values in the iris dataframe.