

DCS 211: In-Class Work – ML Workflow (Iris Dataset)

Names:

1. Block 3: For `pd.read_csv`, what does `header=0` do? Refer to API.
2. Block 3: What does `df.head()` do?
3. Block 5: What information does `df.info()` provide? Refer to API.
4. Block 6: According to the API for `drop`, can it remove rows? Columns? How do you specify what to remove?
5. Block 8: What does `df_clean['irisname'].unique()` do?
6. Block 9: What does the method `dropna` do? Refer to API.
7. Block 10: What code did you write? Justify.
8. Block 11: Explain what the following two lines of code are doing:

```
species_names = df_final['irisname'].unique()
species_name_to_index = { species_names[i]:i for i in range(len(species_names))}
```
9. Block 12: How did you address the CopyOnWarning message from showing up (see link in code block – specifically noting the explanation of using `.loc` at that link)?
10. Block 14: According to the DataFrame API, what is the difference between the following two:
 - a. `A = df_final.values`
 - b. `A = df_final.to_numpy()`
11. Block 16: What does `.shape` on a numpy array return? What data type is it?
12. Block 18: What does `np.linalg.norm` do?
13. Block 18: Inside the `predictiveModel` function, what do `A[i]` and `A[i,0:4]` each represent? Be specific
14. Block 20: What information does `X_all = A[:,0:4]` give you? `y_all = A[:,4]`? How do they relate?
15. Block 22: What would `np.random.permutation([1,2,3,4,5])` do?
16. Block 23: Explain what `X_labeled[:test_size]` and `X_labeled[test_size:]` are doing.
17. Block 25: Explain what `sum(predicted_labels == actual_labels)` does.
18. Blocks 24 & 25: What does `knn.fit` do versus `knn.predict`? Refer to the API.
19. Block 27: What does `knn_model.predict` return, and why do we have to unpack it?
20. Block 27: For the four different example feature sets, what iris was predicted?
21. Block 29: What code did you have to add to the loop, and why?
22. Blocks 30-31: For scikit-learn's kNN approach, there are three primary steps required. What are they?
23. Block 31: How much better did the "best-k model" do? Quantify.
24. Block 32: For at least three different permutations, indicate the best k identified.
25. Block 34: For the four different example feature sets, what iris was predicted?
26. Block 35b: Explain how

```
df_final[ df_final['irisname'] == convertSpecies('versicolor') ]
```

grabs the data corresponding to versicolor.
27. Block 35b: What does `versicolor_data.mean()[:-1]` give back (identify the data type)? Explain how that line of code computes the mean of each different feature.
28. Block 35b: What code did you add to call the predictive model and print a result?

29. Block 35c: What code did you add to automate the process across all species?
30. Block 35d: For each mean-of-features, what species was predicted? Did each match the actual species for the mean-of-features?