

```
In [12]: import pandas as pd
```

```
In [2]: pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from seaborn) (2.3.3)
Collecting matplotlib!=3.6.1,>=3.4 (from seaborn)
  Downloading matplotlib-3.10.8-cp311-cp311-win_amd64.whl.metadata (52 kB)
Collecting contourpy>=1.0.1 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading contourpy-1.3.3-cp311-cp311-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading fonttools-4.61.1-cp311-cp311-win_amd64.whl.metadata (116 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading kiwisolver-1.4.9-cp311-cp311-win_amd64.whl.metadata (6.4 kB)
Requirement already satisfied: packaging>=20.0 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (25.0)
Collecting pillow>=8 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading pillow-12.1.0-cp311-cp311-win_amd64.whl.metadata (9.0 kB)
Collecting pyparsing>=3 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading pyparsing-3.3.2-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from pandas>=1.2->seaborn) (2025.3)
Requirement already satisfied: six>=1.5 in c:\users\oknuy\anaconda3\envs\midterm311\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Downloading matplotlib-3.10.8-cp311-cp311-win_amd64.whl (8.1 MB)
----- 0.0/8.1 MB ? eta -:-:--
----- 1.0/8.1 MB 5.6 MB/s eta 0:00:02
----- 2.9/8.1 MB 7.3 MB/s eta 0:00:01
----- 5.8/8.1 MB 9.3 MB/s eta 0:00:01
----- 8.1/8.1 MB 10.1 MB/s 0:00:00
Downloading contourpy-1.3.3-cp311-cp311-win_amd64.whl (225 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.61.1-cp311-cp311-win_amd64.whl (2.3 MB)
----- 0.0/2.3 MB ? eta -:-:--
----- 2.3/2.3 MB 19.0 MB/s 0:00:00
Downloading kiwisolver-1.4.9-cp311-cp311-win_amd64.whl (73 kB)
Downloading pillow-12.1.0-cp311-cp311-win_amd64.whl (7.0 MB)
----- 0.0/7.0 MB ? eta -:-:--
----- 3.9/7.0 MB 18.1 MB/s eta 0:00:01
----- 7.0/7.0 MB 16.7 MB/s 0:00:00
Downloading pyparsing-3.3.2-py3-none-any.whl (122 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler, contourpy, matplotlib, seaborn
----- 0/8 [pyparsing]
----- 1/8 [pillow]
----- 1/8 [pillow]
----- 1/8 [pillow]
```



```
----- 6/8 [matplotlib]
----- 7/8 [seaborn]
----- 8/8 [seaborn]
```

Successfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.61.1 kiwisolver-1.4.9 matplotlib-3.10.8 pillow-12.1.0 pyparsing-3.3.2 seaborn-0.13.2

Note: you may need to restart the kernel to use updated packages.

WARNING: The scripts fonttools.exe, pyftmerge.exe, pyftsubset.exe and ttx.exe are installed in 'C:\Users\oknuy\anaconda3\envs\midterm311\Scripts' which is not on PATH.

Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

```
In [3]: import seaborn as sns
df = sns.load_dataset("penguins")
df.head()
```

```
Out[3]:   species    island  bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g
          0   Adelie  Torgersen        39.1           18.7            181.0       3750.0   I
          1   Adelie  Torgersen        39.5           17.4            186.0       3800.0   Fe
          2   Adelie  Torgersen        40.3           18.0            195.0       3250.0   Fe
          3   Adelie  Torgersen        NaN             NaN             NaN             NaN
          4   Adelie  Torgersen        36.7           19.3            193.0       3450.0   Fe
```

```
In [4]: df.isnull().sum()
```

```
Out[4]: species      0
island        0
bill_length_mm  2
bill_depth_mm   2
flipper_length_mm  2
body_mass_g     2
sex           11
dtype: int64
```

```
In [5]: df = df.dropna(subset=["species"])
```

```
In [6]: numeric_cols = df.select_dtypes(include=["float64", "int64"]).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].median())
```

```
In [7]: df["sex"] = df["sex"].fillna("Unknown")
```

```
In [8]: df.isnull().sum()
```

```
Out[8]: species          0  
island           0  
bill_length_mm   0  
bill_depth_mm    0  
flipper_length_mm 0  
body_mass_g      0  
sex              0  
dtype: int64
```

```
In [9]: df_model = df[["bill_depth_mm", "island", "species"]].copy()
```

```
In [10]: from sklearn.preprocessing import LabelEncoder  
  
le = LabelEncoder()  
df_model["species_label"] = le.fit_transform(df_model["species"])
```

```
In [13]: df_model = pd.get_dummies(df_model, columns=["island"], drop_first=True)
```

```
In [15]: X = df_model.drop(["species", "species_label"], axis=1)  
y = df_model["species_label"]
```

```
In [16]: from sklearn.model_selection import train_test_split  
  
X_train, X_temp, y_train, y_temp = train_test_split(  
    X, y, test_size=0.30, random_state=42, stratify=y  
)  
  
X_val, X_test, y_val, y_test = train_test_split(  
    X_temp, y_temp, test_size=0.50, random_state=42, stratify=y_temp  
)
```

```
In [17]: from sklearn.ensemble import RandomForestClassifier  
from sklearn.metrics import accuracy_score, classification_report  
  
#Model  
rf = RandomForestClassifier(  
    n_estimators=200,  
    random_state=42  
)  
  
#Fit  
rf.fit(X_train, y_train)  
  
# Validate  
y_val_pred = rf.predict(X_val)  
val_accuracy = accuracy_score(y_val, y_val_pred)  
  
val_accuracy
```

```
Out[17]: 0.7307692307692307
```

```
In [18]: y_test_pred = rf.predict(X_test)
test_accuracy = accuracy_score(y_test, y_test_pred)

print("Validation Accuracy:", val_accuracy)
print("Test Accuracy:", test_accuracy)
print("\nClassification Report:\n", classification_report(y_test, y_test_pred))
```

```
Validation Accuracy: 0.7307692307692307
```

```
Test Accuracy: 0.75
```

```
Classification Report:
precision    recall   f1-score   support
          0       0.71      0.74      0.72       23
          1       0.64      0.64      0.64       11
          2       0.88      0.83      0.86       18

accuracy                           0.75      52
macro avg       0.74      0.74      0.74      52
weighted avg    0.75      0.75      0.75      52
```

```
In [20]: import pickle

with open("penguin_model.pkl", "wb") as f:
    pickle.dump(rf, f)
```

```
In [ ]:
```