Guess with your neighbors!

k-Nearest Neighbors

Prompts

- Build a kNN classifier with "yammy" as the target

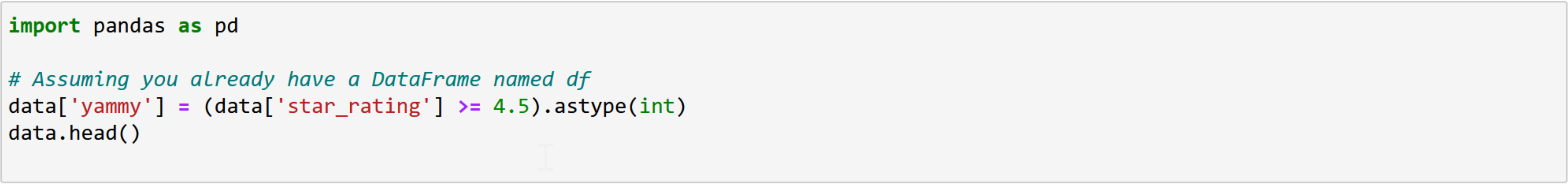
Strategy

- Prepare the data
- Train the kNN Model
- Check the Accuracy and the Confusion Matrix

Code & Results

1. Prepare the data

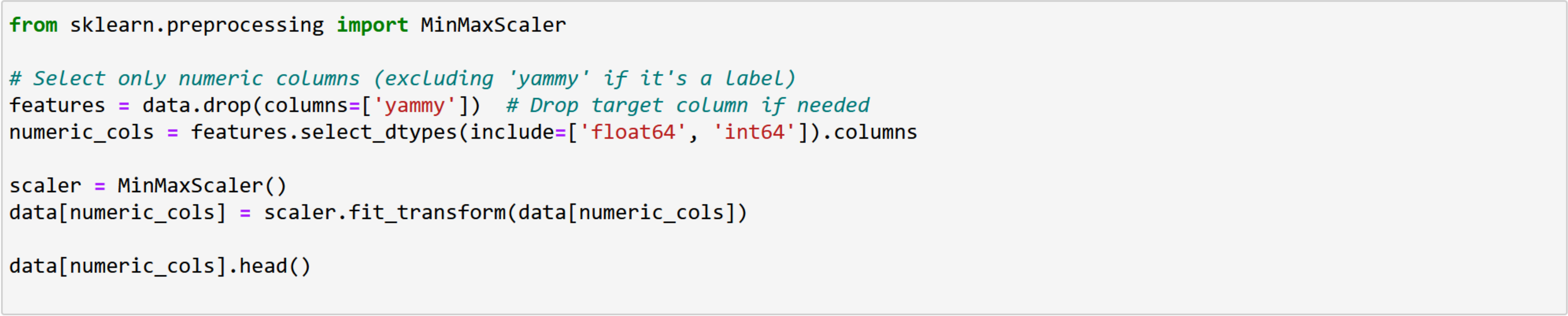
This code checks the "star_rating" and puts a 1 in the new "yammy" column if it is greater than or equal to 4.5.



```python
import pandas as pd

# Assuming you already have a DataFrame named df
data['yammy'] = (data['star_rating'] >= 4.5).astype(int)
data.head()
```

This code puts features between 0 and 1.



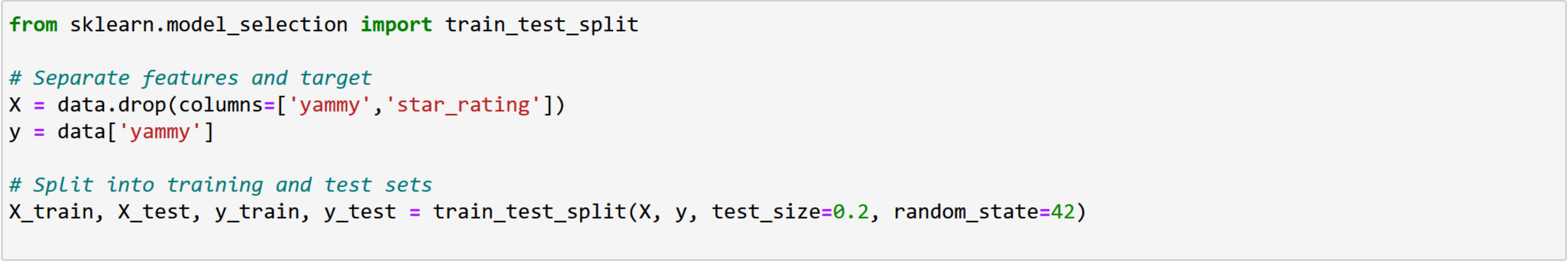
```python
from sklearn.preprocessing import MinMaxScaler

# Select only numeric columns (excluding 'yammy' if it's a label)
features = data.drop(columns=['yammy'])  # Drop target column if needed
numeric_cols = features.select_dtypes(include=['float64', 'int64']).columns

scaler = MinMaxScaler()
data[numeric_cols] = scaler.fit_transform(data[numeric_cols])

data[numeric_cols].head()
```

This code first splits the teacher from the features and then splits the data into training and testing sets.



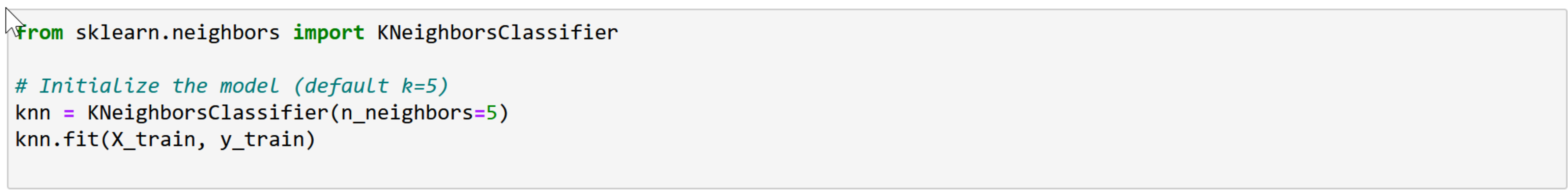
```python
from sklearn.model_selection import train_test_split

# Separate features and target
X = data.drop(columns=['yammy','star_rating'])
y = data['yammy']

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

2. Train the Model

This code creates a kNN model and trains it using the training set.

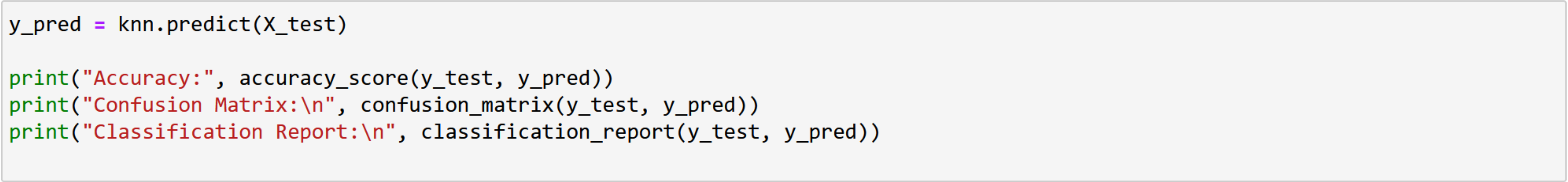


```python
from sklearn.neighbors import KNeighborsClassifier

# Initialize the model (default k=5)
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

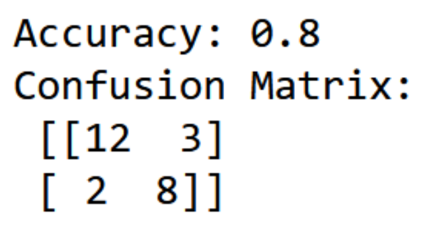
3. Check the Accuracy & Confusion Matrix

This code evaluates model accuracy and calculates a confusion matrix.



```python
y_pred = knn.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```



```
Accuracy: 0.8
Confusion Matrix:
 [[12  3]
 [ 2  8]]
```

✅ The model got 80% of the tins right.

❌ It made some mistakes:

- It mislabeled 3 out of 15 meh tins as yammy
- It mislabeled 2 out of 10 yammy tins as meh

Made by: @katerynakononova
Website: https://katerynakononova.github.io/meowlearning/
Book: https://www.amazon.com/dp/B0CW9SFYXF