

```
In [1]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
In [2]: # Load the data
data = pd.read_csv('fitness_dataset.csv') # Replace with your dataset
data
```

Out[2]:

	Unnamed: 0		Title	Desc	Type	BodyPart	Equipment	Level	Rating	RatingDesc
0	0	Partner plank band row	The partner plank band row is an abdominal exe...	Strength	Abdominals	Bands	Intermediate	0.0	NaN	
1	1	Banded crunch isometric hold	The banded crunch isometric hold is an exercis...	Strength	Abdominals	Bands	Intermediate	NaN	NaN	
2	2	FYR Banded Plank Jack	The banded plank jack is a variation on the pl...	Strength	Abdominals	Bands	Intermediate	NaN	NaN	
3	3	Banded crunch	The banded crunch is an exercise targeting the...	Strength	Abdominals	Bands	Intermediate	NaN	NaN	
4	4	Crunch	The crunch is a popular core exercise targetin...	Strength	Abdominals	Bands	Intermediate	NaN	NaN	
...
2913	2913	EZ-bar skullcrusher-	The EZ-bar skullcrusher is a popular exercise ...	Strength	Triceps	E-Z Curl Bar	Intermediate	8.1	Average	
2914	2914	Lying Close-Grip Barbell Triceps Press To Chin		NaN	Strength	Triceps	E-Z Curl Bar	Beginner	8.1	Average
2915	2915	EZ-Bar Skullcrusher - Gethin Variation	The EZ-bar skullcrusher is a popular exercise ...	Strength	Triceps	E-Z Curl Bar	Intermediate	NaN	NaN	
2916	2916	TBS Skullcrusher	The EZ-bar skullcrusher is a popular exercise ...	Strength	Triceps	E-Z Curl Bar	Intermediate	NaN	NaN	
2917	2917	30 Arms EZ-Bar Skullcrusher		NaN	Strength	Triceps	E-Z Curl Bar	Intermediate	NaN	NaN

2918 rows × 9 columns

```
In [4]: # Preprocessing
data_cleaned = data.drop(columns=['Unnamed: 0', 'Title', 'Desc', 'RatingDesc']).dropna()
```

```
In [5]: # Map BodyPart into binary labels
target_classes = ['Abdominals', 'Chest', 'Quadriceps', 'Shoulders']
data_cleaned['BodyPartBinary'] = data_cleaned['BodyPart'].apply(
    lambda x: 1 if x in target_classes else 0
)
```

```
In [6]: # Drop original BodyPart column
data_cleaned = data_cleaned.drop(columns=['BodyPart'])
```

```
In [8]: # Encode categorical variables
data_encoded = pd.get_dummies(data_cleaned, columns=['Type', 'Equipment', 'Level'])
```

```
In [10]: # Separate features and target
X = data_encoded.drop(columns=['BodyPartBinary'])
y = data_encoded['BodyPartBinary']
```

```
In [11]: # 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)
```

```
In [12]: # Train a RandomForestClassifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Out[12]:

RandomForestClassifier

RandomForestClassifier(random_state=42)

```
In [13]: # Make predictions
y_pred = model.predict(X_test)
```

```
In [14]: # Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

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

```
Accuracy: 0.59
Classification Report:
      precision    recall  f1-score   support

     0       0.51      0.52      0.51        85
     1       0.66      0.65      0.65       122

 accuracy          0.59          207
  macro avg       0.58      0.58      0.58          207
 weighted avg     0.60      0.59      0.59          207
```

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js