In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:
```
data = pd.read_csv('fitness_dataset.csv')
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]:
```
data_cleaned = data.drop(columns=['Unnamed: 0', 'Title', 'Desc', 'RatingDesc']).dropna()
```

In [6]:
```
# Encode categorical variables
data_encoded = pd.get_dummies(data_cleaned, columns=['Type', 'Equipment', 'Level', 'BodyPart'])
```

In [7]:
```
# Separate features and target
X = data_encoded.drop(columns=['Rating'])
y = data_encoded['Rating']
```

In [8]:
```
# 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 [9]:
```
# Train a Random Forest Regressor
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)
```

Out[9]:
```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [10]:
```
# Make predictions
y_pred = model.predict(X_test)
```

In [11]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R² Score: {r2:.2f}")
```

```
Mean Squared Error: 10.33
R² Score: 0.16
```

In [ ]: