```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
```

```
df = pd.read_csv("/content/marketing_data.csv")
df.head()
```

	User_ID	Likes	Shares	Comments	Clicks	Engagement_with_Ads	Time_Spent_on_Platform	Purchase_
0	1	0.006024	0.768844	0.44	0.26	0.733333	0.919463	
1	2	0.841365	0.482412	0.53	0.74	0.466667	0.697987	
2	3	0.847390	0.351759	0.43	0.12	0.000000	0.187919	
3	4	0.698795	0.291457	0.27	0.76	0.500000	0.885906	
4	5	0.548193	0.130653	0.86	0.38	0.700000	0.885906	

5 rows × 22 columns

```
df = df.drop('User_ID', axis=1)
```

```
# Handle missing values
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = df[col].fillna(df[col].mode()[0])
    else:
        df[col] = df[col].fillna(df[col].mean())
```

```
# Encode categorical features
le = LabelEncoder()
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = le.fit_transform(df[col])
```

```
# Define features (X) and target (y)
X = df.drop('shopping', axis=1)
y = df['shopping']
```

```
# Scale numerical features
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

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

```
# RandomForestRegressor for continuous target
classifier = RandomForestRegressor(random_state=42) # Instantiate the regressor
```

```
# Train the model
classifier.fit(X_train, y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

```
# Evaluate the model using metrics appropriate for regression
mse = mean_squared_error(y_test, y_pred) # Calculate Mean Squared Error
print(f"Mean Squared Error: {mse}")
```

```
Mean Squared Error: 0.05044997351643971
```