

The Tree of Meowledge

Regression Tree

Prompts



- Create a Regression Tree using the preprocessed dataset 'data', with 'star_rating' as the target.
- Scale the features before training
- Plot the tree

Strategy



- Prepare the data
- Train the Regression Tree
- Check the Accuracy
- Visualise the tree

Code & Results

1. Prepare the data

This code 1) separates features and target, 2) scales the features, and 3) splits train and test sets.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

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

# 2. Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 3. Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

2. Train the Model

This code creates a regression tree model and trains it using the training set (Note max_depth=3).

```
from sklearn.tree import DecisionTreeRegressor, plot_tree

# 3. Train a Regression Tree
tree_model = DecisionTreeRegressor(max_depth=3, random_state=42) # you can tune max_depth
tree_model.fit(X_train, y_train)

# 4. Predictions
y_pred = tree_model.predict(X_test)
```

3. Check the Accuracy

This code evaluates model accuracy.

```
# 5. Evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.4f}")
print(f"R² Score: {r2:.4f}")
```

Mean Squared Error: 0.2777
R² Score: -0.3484

✅ The MSE helps to see if your model is good enough. The smaller the MSE, the closer the model predictions are to the actual values.

❌ Negative R² means that the tree predicts worse than the average!

4. Visualize the tree

This code plots the regression tree.

```
plt.figure(figsize=(16,8))
plot_tree(tree_model, feature_names=X.columns, filled=True, rounded=True, fontsize=10)
plt.show()
```

