```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score
import xgboost as xgb

# Load dataset
file_path = r"/content/15.eq_building_damage.csv"  # Added r to fix path
df = pd.read_csv(file_path)

# Drop unnecessary index column if present
if "Unnamed: 0" in df.columns:
    df = df.drop(columns=["Unnamed: 0"])

# Fix erroneous 'year_built' values (assume values >2025 are incorrect)
df["year_built"] = df["year_built"].apply(lambda x: x if x < 2025 else 1925)

# Convert categorical variables to numerical using one-hot encoding
df = pd.get_dummies(df, columns=["struct_typ", "occ_type"], drop_first=True)

# Split data into features and target
X = df.drop(columns=["meandamage"])
y = df["meandamage"]
```

```
# Split 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)

# Train an XGBoost regression model
model = xgb.XGBRegressor(n_estimators=100, random_state=42)

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

# Predictions
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Absolute Error: {mae}")
print(f"R-squared: {r2}")

# Plot Feature Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()

# Plot Actual vs. Predicted Damage
plt.figure(figsize=(8, 6))
```

```
plt.scatter(y_test, y_pred, alpha=0.5, color='blue')

plt.xlabel("Actual Damage")

plt.ylabel("Predicted Damage")

plt.title("Actual vs. Predicted Building Damage")

plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')  # Corrected red line

plt.show()


# Plot Feature Importance

feature_importances = pd.Series(model.feature_importances_, index=X.columns)

feature_importances.nlargest(10).plot(kind='barh', color='teal')

plt.xlabel("Feature Importance Score")

plt.ylabel("Feature")

plt.title("Top 10 Important Features")

plt.show()
```