**2311cs020228-day26**

**Using the same data set of Civil\_Engineering\_Regression\_Dataset.csv**

**Part 5: Advanced Model Optimization & Business Impact**

1. **Feature Selection: Perform backward elimination or Lasso regression to remove less significant variables from the multiple linear regression model. Which variables remain in the final model?**
2. **Residual Analysis: Plot the residuals of the multiple linear regression model. Are they randomly distributed? What does this indicate about model assumptions?**
3. **Outlier Detection: Identify any outliers in the dataset using statistical methods (e.g., box plots, Z-scores). How do they impact the regression model?**
4. **Model Deployment: If this regression model were to be deployed in a construction cost estimation tool, what additional features or real-time data sources could enhance its accuracy?**
5. **Ethical Considerations & Decision Making: How could errors in regression-based cost predictions impact real-world construction projects? Discuss the financial and safety implications of overestimating or underestimating costs.**

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression, LassoCV

from sklearn.metrics import r2\_score, mean\_squared\_error

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from scipy.stats import zscore

# Load dataset

try:

df = pd.read\_csv("Civil\_Engineering\_Regression\_Dataset.csv")

print("CSV file loaded successfully.\n")

except FileNotFoundError:

print("Error: CSV file not found. Check the file path.")

exit()

# Define independent (X) and dependent (y) variables

X = df[["Building Height", "Material Quality", "Labor Cost", "Concrete Strength", "Foundation Depth"]]

y = df["Construction Cost"]

# Convert all values to numeric, handling errors

X = X.apply(pd.to\_numeric, errors="coerce")

y = pd.to\_numeric(y, errors="coerce")

df.dropna(inplace=True)

# Feature Selection using Lasso Regression

lasso = LassoCV(cv=5, random\_state=42).fit(X, y)

selected\_features = X.columns[lasso.coef\_ != 0]

print(f"Selected Features after Lasso: {list(selected\_features)}\n")

# Fit a new regression model with selected features

X\_selected = X[selected\_features]

model = LinearRegression()

model.fit(X\_selected, y)

y\_pred = model.predict(X\_selected)

# Residual Analysis

residuals = y - y\_pred

plt.figure(figsize=(10, 5))

sns.histplot(residuals, bins=30, kde=True)

plt.axvline(0, color='red', linestyle='dashed')

plt.xlabel("Residuals")

plt.ylabel("Frequency")

plt.title("Residual Analysis")

plt.show()

# Outlier Detection using Z-scores

z\_scores = np.abs(zscore(df[selected\_features]))

outliers = np.where(z\_scores > 3)

outlier\_indices = np.unique(outliers[0])

print(f"Outlier Indices: {outlier\_indices}")

print(f"Number of Outliers Detected: {len(outlier\_indices)}\n")

# Box Plot for Outlier Detection

plt.figure(figsize=(10, 6))

sns.boxplot(data=df[selected\_features])

plt.title("Box Plot for Outlier Detection")

plt.xticks(rotation=45)

plt.show()

# Model Evaluation

r2 = r2\_score(y, y\_pred)

mse = mean\_squared\_error(y, y\_pred)

print(f"Final Model R-squared: {r2:.4f}")

print(f"Final Model Mean Squared Error: {mse:.4f}\n")

# Ethical Considerations

print("Ethical Considerations & Business Impact:")

print("1. Overestimating construction costs may result in lost business opportunities.")

print("2. Underestimating costs can lead to budget overruns and financial losses.")

print("3. Safety risks arise if cost constraints compromise material or labor quality.")

print("4. Incorporating real-time data such as market price fluctuations can improve accuracy.")