**TITLE: Servo Response Time Prediction Using Linear Regression**

Objective:

The objective of this project is to predict the response time of a servo system based on features like motor input, screw, gain, and feedback delay using a linear regression model.

Data Source:

- UCI Machine Learning Repository (Servo Data Set)

- Custom sensor or actuator data from servo systems

Import Library:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

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

Import Data:

servo\_data = pd.read\_csv('servo\_data.csv') # Replace with actual file path

Describe Data:

print(servo\_data.info())

print(servo\_data.describe())

print(servo\_data.head())

print(servo\_data.isnull().sum())

Explanation:

The dataset typically includes features like motor input, screw, gain, feedback delay, and response time. Understanding the structure and any missing data is crucial.

Data Visualization:

# Visualizing the relationship between features and response time

sns.pairplot(servo\_data)

plt.title('Feature Relationships with Response Time')

plt.show()

# Correlation heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(servo\_data.corr(), annot=True, cmap='coolwarm')

plt.title('Feature Correlation Heatmap')

plt.show()

Data Preprocessing:

# Handling missing values (if any)

servo\_data.fillna(method='ffill', inplace=True)

# Converting categorical variables to numerical (if applicable)

servo\_data = pd.get\_dummies(servo\_data, drop\_first=True)

Define Target Variable (y) and Feature Variables (X):

# Defining features and target variable

X = servo\_data.drop('response\_time', axis=1) # Features (motor, screw, gain, etc.)

y = servo\_data['response\_time'] # Target variable (Servo response time)

Train Test Split:

# Splitting the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

Modeling (Linear Regression):

# Linear Regression Model

lr\_model = LinearRegression()

lr\_model.fit(X\_train, y\_train)

# Predictions on test set

y\_pred\_lr = lr\_model.predict(X\_test)

Model Evaluation:

# Mean Squared Error and R-squared Score for Linear Regression

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

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

print(f"Linear Regression MSE: {mse\_lr}")

print(f"Linear Regression R2 Score: {r2\_lr}")

Prediction:

# Predict the response time for a new servo system with specific features

new\_servo = np.array([[3, 5, 2, 4]]) # Example features (motor, screw, gain, feedback delay)

predicted\_response\_time = lr\_model.predict(new\_servo)

print(f"Predicted Response Time: {predicted\_response\_time[0]}")

Explanation:

1. Linear Regression: This model assumes a linear relationship between the features and the response time. It works well for understanding the general trend of the data, but it may not capture complex interactions.

2. Model Evaluation: Metrics such as Mean Squared Error (MSE) and R-squared (R2) scores help assess the performance of the model. A lower MSE indicates a better fit, while a higher R2 score indicates the percentage of variance explained by the model.

This outline provides the basic structure for developing a ‘Servo Response Time Prediction’ system using ‘Linear Regression’ in Python. You can further customize this structure depending on the dataset and project requirements.