Big Data Technologies DA 2

Suryakumar P 21MIS1146

**Dataset – Sensor Readings (1.4 Million + Rows), Algorithm: Linear Regression – Finding MSE**

1. Python – joblib algorithm for Parallel Processing:

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

from joblib import Parallel, delayed

import numpy as np

*# Load the CSV file into a pandas DataFrame*

df = pd.read\_csv("sensor\_data.csv", header=None, names=["SensorReading"])

*# Extract the input feature (X) and create the target variable (y) from row indices*

X = df[["SensorReading"]]

y = np.arange(len(df))

*# 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)

*# Initialize the linear regression model*

model = LinearRegression()

*# Function to fit the model and make predictions*

def fit\_model(X\_train, y\_train, X\_test):

*# Train the model*

    model.fit(X\_train, y\_train)

*# Make predictions*

    return model.predict(X\_test)

*# Perform parallel processing using joblib*

predictions = Parallel(n\_jobs=-1)(delayed(fit\_model)(X\_train, y\_train, X\_test) for \_ in range(10))

*# Calculate mean squared error for each set of predictions*

mse\_values = [mean\_squared\_error(y\_test, pred) for pred in predictions]

*# Print mean squared error for each set of predictions*

for i, mse in enumerate(mse\_values):

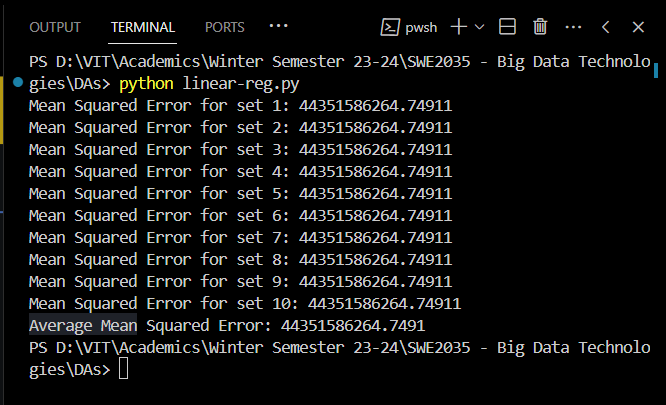
    print(f"Mean Squared Error for set {i+1}: {mse}")

*# Average mean squared error across all sets*

average\_mse = sum(mse\_values) / len(mse\_values)

print("Average Mean Squared Error:", average\_mse)

**Output:**

****

**Hadoop:**