**import tkinter as tk**

**import pandas as pd**

**data=pd.read\_csv('add.csv')**

**x=data[['x','y']]**

**y=data['sum']**

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

**x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.33,random\_state=42)**

**from sklearn.linear\_model import LinearRegression**

**model=LinearRegression()**

**model.fit(x\_train,y\_train)**

**def display\_inputs():**

**# Retrieve inputs from the entry widgets**

**input1 = float(entry1.get())**

**input2 = float(entry2.get())**

**y\_pred=model.predict([[input1,input2]])[0]**

**# Display the inputs in the label widget**

**result\_label.config(text=f"Input 1: {input1}\nInput 2: {input2}\n {y\_pred:.2f}")**

**# Create the main window**

**root = tk.Tk()**

**root.title("Input Display")**

**# Create and place the first label and entry widget**

**label1 = tk.Label(root, text="Enter Input 1:")**

**label1.pack(pady=5)**

**entry1 = tk.Entry(root)**

**entry1.pack(pady=5)**

**# Create and place the second label and entry widget**

**label2 = tk.Label(root, text="Enter Input 2:")**

**label2.pack(pady=5)**

**entry2 = tk.Entry(root)**

**entry2.pack(pady=5)**

**# Create and place the display button**

**display\_button = tk.Button(root, text="Display", command=display\_inputs)**

**display\_button.pack(pady=10)**

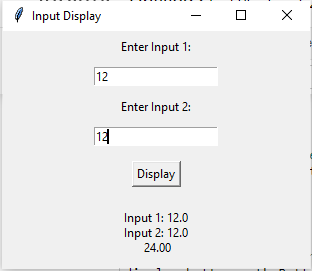
**# Create and place the result label**

**result\_label = tk.Label(root, text="")**

**result\_label.pack(pady=10)**

**# Start the Tkinter event loop**

**root.mainloop()**

****

**import numpy as np**

**import tkinter as tk**

**from tkinter import ttk**

**from sklearn.datasets import load\_diabetes**

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

**from sklearn.metrics import r2\_score**

**# 222P005**

**data = load\_diabetes()**

**X = data.data[:, :2]**

**y = data.target**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42)**

**class MyLinearRegression:**

**def \_\_init\_\_(self):**

**self.intercept\_ = None**

**self.coef\_ = None**

**def fit(self, X\_train, y\_train):**

**X\_b = np.c\_[np.ones((X\_train.shape[0], 1)), X\_train]**

**self.coef\_ = np.linalg.inv(X\_b.T @ X\_b) @ X\_b.T @ y\_train**

**self.intercept\_ = self.coef\_[0]**

**self.coef\_ = self.coef\_[1:]**

**def predict(self, X\_test):**

**X\_b = np.c\_[np.ones((X\_test.shape[0], 1)), X\_test]**

**return X\_b @ np.concatenate(([self.intercept\_], self.coef\_))**

**my\_model = MyLinearRegression()**

**my\_model.fit(X\_train, y\_train)**

**y\_pred\_my = my\_model.predict(X\_test)**

**r2\_my = r2\_score(y\_test, y\_pred\_my)**

**intercept\_my = my\_model.intercept\_**

**coefficients\_my = my\_model.coef\_**

**class RegressionApp:**

**def \_\_init\_\_(self, root):**

**self.root = root**

**self.root.title("Multiple Linear Regression By 222P005")**

**self.create\_widgets()**

**def create\_widgets(self):**

**ttk.Label(self.root, text="Feature 1:").grid(column=0, row=0, padx=10, pady=10)**

**self.feature1\_var = tk.DoubleVar()**

**self.feature1\_entry = ttk.Entry(self.root, textvariable=self.feature1\_var)**

**self.feature1\_entry.grid(column=1, row=0, padx=10, pady=10)**

**ttk.Label(self.root, text="Feature 2:").grid(column=0, row=1, padx=10, pady=10)**

**self.feature2\_var = tk.DoubleVar()**

**self.feature2\_entry = ttk.Entry(self.root, textvariable=self.feature2\_var)**

**self.feature2\_entry.grid(column=1, row=1, padx=10, pady=10)**

**self.calc\_button = ttk.Button(self.root, text="Calculate", command=self.calculate)**

**self.calc\_button.grid(column=0, row=4, columnspan=2, pady=10)**

**# Result Labels**

**ttk.Label(self.root, text="Prediction:").grid(column=0, row=5, padx=10, pady=10)**

**self.result\_var = tk.StringVar()**

**self.result\_label = ttk.Label(self.root, textvariable=self.result\_var)**

**self.result\_label.grid(column=1, row=5, padx=10, pady=10)**

**# Custom Model Info**

**ttk.Label(self.root, text="R² Score:").grid(column=0, row=6, padx=10, pady=10)**

**self.r2\_var = tk.StringVar(value=f"{r2\_my:.4f}")**

**self.r2\_label = ttk.Label(self.root, textvariable=self.r2\_var)**

**self.r2\_label.grid(column=1, row=6, padx=10, pady=10)**

**ttk.Label(self.root, text="Intercept:").grid(column=0, row=7, padx=10, pady=10)**

**self.intercept\_var = tk.StringVar(value=f"{intercept\_my:.4f}")**

**self.intercept\_label = ttk.Label(self.root, textvariable=self.intercept\_var)**

**self.intercept\_label.grid(column=1, row=7, padx=10, pady=10)**

**ttk.Label(self.root, text="Coefficients:").grid(column=0, row=8, padx=10, pady=10)**

**self.coef\_var = tk.StringVar(value=str(coefficients\_my))**

**self.coef\_label = ttk.Label(self.root, textvariable=self.coef\_var)**

**self.coef\_label.grid(column=1, row=8, padx=10, pady=10)**

**def calculate(self):**

**feature1 = self.feature1\_var.get()**

**feature2 = self.feature2\_var.get()**

**features = np.array([[feature1, feature2]])**

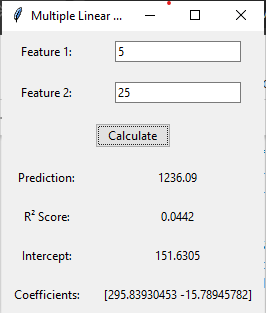
**prediction = my\_model.predict(features)[0]**

**self.result\_var.set(f"{prediction:.2f}")**

**root = tk.Tk()**

**app = RegressionApp(root)**

**root.mainloop()**

****