In [1]:
```
print('Aim - Implement your own Multiple Linear Regression Class (Use GUI).')
print('Aryan Shaikh 221P008 34 Comps')

import pandas as pd
import numpy as np
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import tkinter as tk

# ------------------ Load dataset ------------------
diabetes = load_diabetes()
X, y = diabetes.data, diabetes.target

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Train using built-in LinearRegression
reg = LinearRegression()
reg.fit(X_train, y_train)
y_pred_builtin = reg.predict(X_test)
score_builtin = r2_score(y_test, y_pred_builtin)

# ------------------ Custom Linear Regression ------------------
class MyLinearRegression:
    def __init__(self):
        self.coef = None
        self.intercept = None

    def fit(self, X_train, y_train):
        X_train = np.c_[np.ones(X_train.shape[0]), X_train]
        # Normal Equation: β = (XᵀX)⁻¹ Xᵀy
        self.coef = np.linalg.inv(X_train.T @ X_train) @ X_train.T @ y_train
        self.intercept = self.coef[0]
        self.coef = self.coef[1:]

    def predict(self, X_test):
        X_test = np.c_[np.ones(X_test.shape[0]), X_test]
        return X_test @ np.concatenate(([self.intercept], self.coef))

# Train custom model
custom_reg = MyLinearRegression()
custom_reg.fit(X_train, y_train)
y_pred_custom = custom_reg.predict(X_test)
score_custom = r2_score(y_test, y_pred_custom)

# ------------------ Tkinter GUI ------------------
root = tk.Tk()
root.title("Diabetes Prediction By Danish")
root.geometry('700x350')

def predict_sum():
    try:
        index = int(entry_index.get())
        input_data = X[index].reshape(1, -1)
        prediction = custom_reg.predict(input_data)

        result_text = (
            f"Custom Model Prediction: {prediction[0]:.2f}\n\n"
            f"Built-in R² Score: {score_builtin:.2f}\n"
            f"Custom R² Score: {score_custom:.2f}\n\n"
            f"Custom Coefficients:\n{custom_reg.coef}\n\n"
            f"Custom Intercept: {custom_reg.intercept:.2f}"
        )
        label_prediction.config(text=result_text, justify="left", anchor="w")
    except ValueError:
        label_prediction.config(text="❌ Please enter a valid index (0-441).")

# Main Frame for Layout
frame_main = tk.Frame(root, padx=20, pady=20)
frame_main.pack(expand=True, fill="both")

# Input Row
label_entry = tk.Label(frame_main, text="Enter index (0-441):", font=("Arial", 11))
label_entry.grid(row=0, column=0, padx=5, pady=5, sticky="e")

entry_index = tk.Entry(frame_main, width=10)
entry_index.grid(row=0, column=1, padx=5, pady=5)

button_predict = tk.Button(frame_main, text="Predict", command=predict_sum, width=1
button_predict.grid(row=0, column=2, padx=10, pady=5)

# Output Label
label_prediction = tk.Label(frame_main, text="", font=("Arial", 11), justify="left'
label_prediction.grid(row=1, column=0, columnspan=3, sticky="w", pady=15)

root.mainloop()
```

```
Aim - Implement your own Multiple Linear Regression Class (Use GUI).
Aryan Shaikh 221P008 34 Comps
```

In [ ]: