**Linear Regression Web App using NumPy & Flask**

**Objective**

The objective of this project was to implement a linear regression model **from scratch using NumPy**, train it on a dataset with **8 numerical features**, and create a **web application using Flask** to take user input and return a prediction.  
The final trained model was also hosted on **Hugging Face** for accessibility.

**Repository Link**

GitHub Repository  
🔗 <https://github.com/tahaSawan/Machine-Learning-A3/tree/main>

**Model Hosting**

Hugging Face Model  
🔗 [tahasawan/linear-regression-numpy at main](https://huggingface.co/tahasawan/linear-regression-numpy/tree/main)

**Approach and Steps Performed**

**1. Data Preprocessing**

* Handled dataset with **8 input features** and a target variable.
* Used StandardScaler from sklearn to **scale the features** before training the model.

**2. Model Implementation**

* Implemented **Linear Regression from scratch** using **NumPy only** (no LinearRegression from sklearn).
* Training was done using **Gradient Descent**.
* Loss function: **Mean Squared Error (MSE)**

**3. Model Evaluation**

* Evaluated performance using:
  + **MSE** (Mean Squared Error)
  + **R² Score**
* Exported:
  + Trained weights: model\_weights.joblib
  + Scaler: scaler.joblib

**4. Model Hosting on Hugging Face**

* Uploaded model and scaler files to:  
  🔗 [tahasawan/linear-regression-numpy at main](https://huggingface.co/tahasawan/linear-regression-numpy/tree/main)

**5. Web App Creation using Flask**

* Built a Flask web app with:
  + A simple HTML form (index.html)
  + A Flask backend (app.py)
  + Accepts **8 comma-separated inputs**
  + Returns the predicted output

| **Tool/Library** |  |
| --- | --- |

|  |  |
| --- | --- |
| **NumPy** | * Linear regression calculations |

|  |  |
| --- | --- |
| **Flask** | * Web app backend |

|  |  |
| --- | --- |
| **HTML (Jinja)** | * Frontend user input interface |

|  |  |
| --- | --- |
| **Joblib** | * Model saving/loading |

|  |  |
| --- | --- |
| **scikit-learn** | * Feature scaling |

|  |  |
| --- | --- |
| **Hugging Face** | * Model hosting |

|  |  |
| --- | --- |
| **GitHub** | * Version control & documentation |

**Future Improvements**

* Implement **Regularization (L1/L2)** for better generalization
* Add **Error Handling** in the Flask app
* Compare performance with sklearn.LinearRegression
* Deploy the web app using **Render** or **Heroku**
* Add CSV upload functionality