

# Home Energy Consumption Prediction using Multi-Linear Regression & Immersive Streamlit UI

---

## 1. Project Overview

This project predicts monthly home energy consumption (kWh) using a Multi-Linear Regression (MLR) model trained on a smart home dataset.

A fully immersive modern Streamlit UI (with themes, gauges, floating labels, glassmorphism, animations) allows:

- Single prediction
  - Batch prediction with CSV upload
  - Interactive gauges reacting to prediction
  - Theme picker
  - Smart preprocessing
  - Animated outputs
- 

## 2. Problem Statement

Household energy consumption varies due to:

- Number of residents
- Usage behavior
- Heating/cooling hours
- Appliance count
- Temperature fluctuations

Users cannot easily estimate upcoming energy bills.

Goal: Build a machine learning system that predicts *monthly energy consumption (kWh)* based on home characteristics.

---

## 3. Project Objectives

### Model Objectives

- Build an accurate MLR-based regression model
- Clean, preprocess, and transform the dataset
- Evaluate performance using R<sup>2</sup> and RMSE
- Save model + scaler as .pkl for deployment

## Application Objectives

- Build fluid, beautiful, theme-based UI
  - Enable real-time energy prediction
  - Provide gauge-based visual indicators
  - Offer batch prediction with CSV download
  - Deliver a professional-grade UX for demos & portfolios
- 

## 4. Dataset Summary

Dataset used: smart\_home\_energy.csv

Columns include:

Feature	Description
house_size	Home size in sq.ft
num_residents	Number of occupants
ac_hours	AC usage per day
heating_hours	Heating usage per day
appliances	Total appliances
avg_temp	Average outside temperature
monthly_energy_kwh	Target variable

---

## 5. Model Development Pipeline

The model training pipeline is defined in your script:

generate\_model

Steps Executed

1. Load dataset
2. Clean missing values
3. One-hot encode categorical columns
4. Numerical feature selection
5. Train-test split → 80/20
6. Scaling using StandardScaler

## 7. Train LinearRegression model

### 8. Save:

- o **model.pkl** → trained model
  - o **scaler.pkl** → preprocessing scaler
- 

## 6. Model Files Produced

✓ **model.pkl**

Includes:

- Trained Linear Regression model
- Selected feature columns

✓ **scaler.pkl**

Used during inference to scale incoming prediction data.

---

## 7. Streamlit Application (Advanced UI)

Your UI file:

UI 2 (advanced)

Key UI Features

✓ Fluid, premium UI design

- Gradient background
- SVG glowing circles
- Glassmorphism cards
- Banschrift font
- Global white text

✓ Dynamic Theme Picker

Themes:

- Ocean
- Sunset
- Midnight
- Forest
- Neon

### ✓ Floating Label Inputs (Material UI)

- Animated labels
- Glowing focus effect
- Clean minimal feel

### ✓ Smart Prediction System

- Auto-loads model.pkl
- Fallback demo mode if model not found
- Animated prediction text
- History tracking

### ✓ Interactive Gauges

- Consumption Gauge: shows predicted kWh
- Efficiency Gauge: reverse-mapped efficiency score

Both update instantly using session\_state.

### ✓ Batch Prediction Mode

- CSV upload
  - Auto-column detection
  - Prediction distribution histogram
  - CSV download with predictions appended
- 

## 8. How to Run the Full Project

Install requirements

```
pip install -r requirements.txt
```

Train the model (optional)

```
python generate_model.py
```

Run the Streamlit UI

```
streamlit run "UI 2 (advanced).py"
```