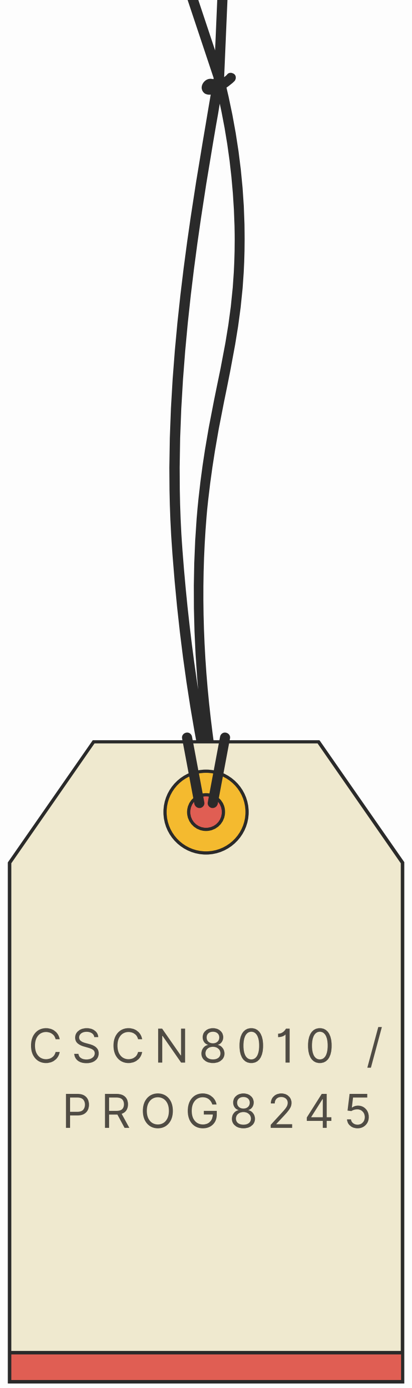
SUSTAINABLE AI – TRANSPARENCY AND ENERGY-EFFICIENT PROMPT/CONTEXT ENGINEERING WITH MACHINE LEARNING

CSCN8010 – FOUNDATIONS OF MACHINE LEARNING FRAMEWORKS
PROG8245 – MACHINE LEARNING PROGRAMMING

PARTH | FENIL | ADHITYA



OVERVIEW

AI workloads consume high compute resources → high energy & CO_2 impact.

Poorly optimized prompts waste resources.

Our system:

1. Analyzes prompts for complexity.
2. Predicts energy & CO_2 usage.
3. Suggests optimizations.
4. Flags anomalies.

Combines NLP + Regression + Anomaly Detection in a Streamlit dashboard.

PROBLEM STATEMENT

01 High Compute Demand: Large Language Models (LLMs) → High cost & Environmental footprint.

02 **Inefficient Prompts:** Poorly structured prompts increase compute time.

03 **Lack of Real-time Insight:** Users can't see energy/CO_2 impact instantly.

04 **Need:** A tool to measure, optimize, and detect inefficiency in AI usage.

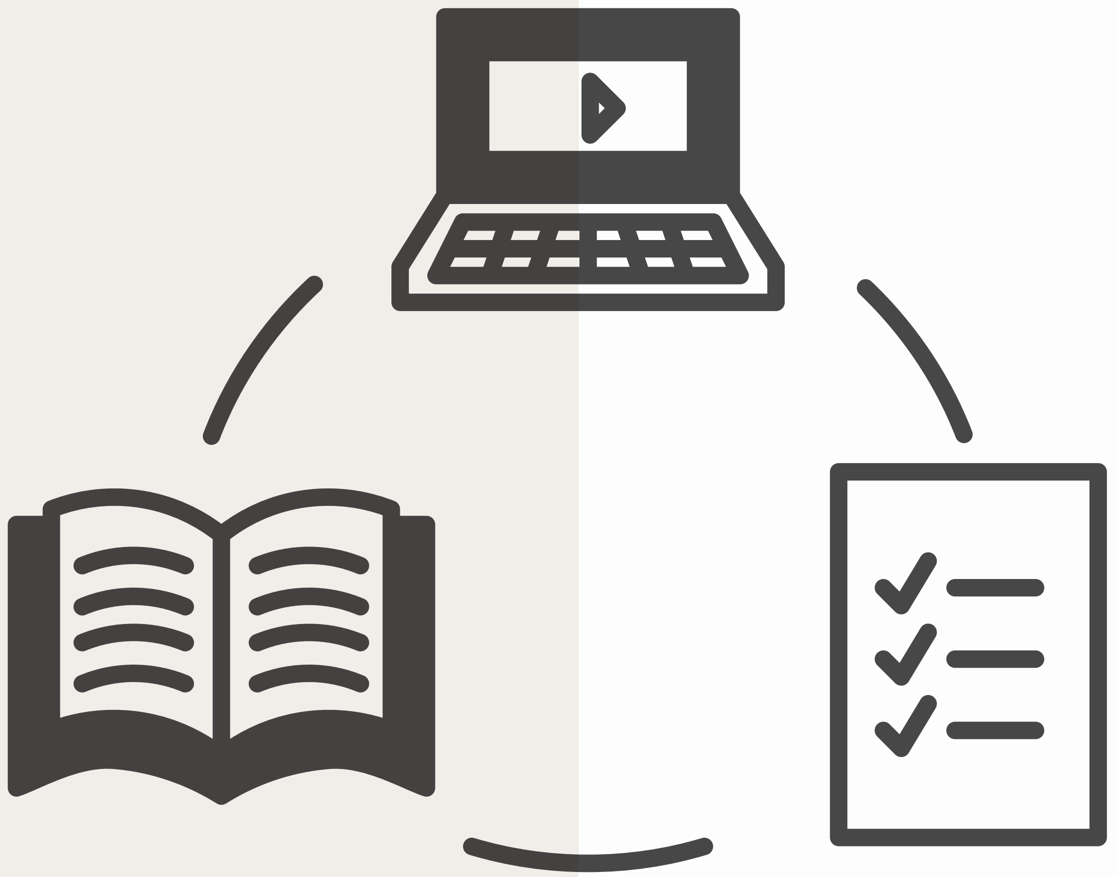
PROJECT OBJECTIVES

Technical:

1. Build NLP complexity scoring system.
2. Train regression model for energy & CO_2.
3. Implement anomaly detection.
4. Deploy with Streamlit dashboard.

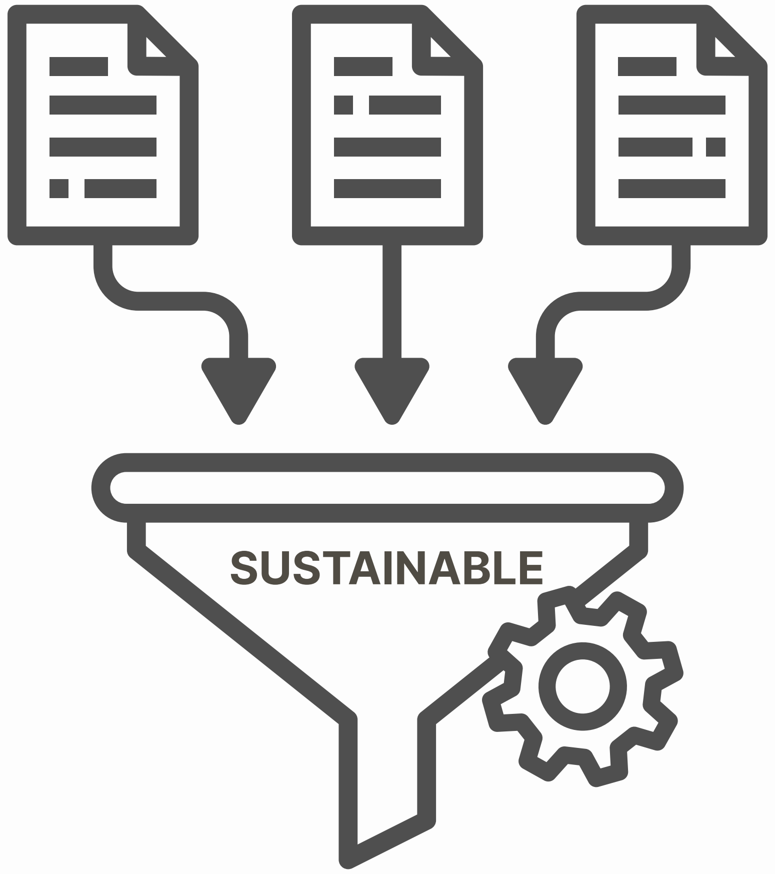
Operational:

1. Modular & scalable architecture.
2. Industry-aligned ML engineering practices.
3. Comprehensive documentation.



BUSINESS CASE

Cost Savings-
Our AI energy prediction and optimization system reduces wasted compute cycles by flagging inefficient prompts before they are run, cutting down on costly cloud inference charges.
Sustainability-
By lowering the energy usage of AI workloads, the system reduces CO_2 emissions, helping organizations meet ESG and sustainability targets.
Competitive Edge-
Optimized prompts deliver results faster while using fewer resources, enabling greener, more efficient AI operations that can be a market differentiator.
Scalability-
The system integrates seamlessly into enterprise AI pipelines, scaling across multiple models and workloads without disrupting existing processes.



DATASET

Source: model_energy_data.csv (synthetic, structured).

Features:

1. Num_Layers
2. FLOPs_in_TFLOPs
3. Training_Hours
4. Complexity (from NLP pipeline)

Targets: Energy_kWh, CO_2_kg.

Preprocessing: Sorting, Cleaning, Feature engineering.

METHODOLOGY

STEP 01

NLP Pipeline → complexity score.
User prompts are tokenized and analyzed for vocabulary richness, sentence length, and readability. This generates a complexity score that helps predict how resource-intensive the prompt might be.

STEP 02

Prediction Model → energy & CO_2 estimate.
Using our trained regression model, we estimate the compute energy (kWh) and resulting CO_2 emissions for the given prompt complexity and model parameters.

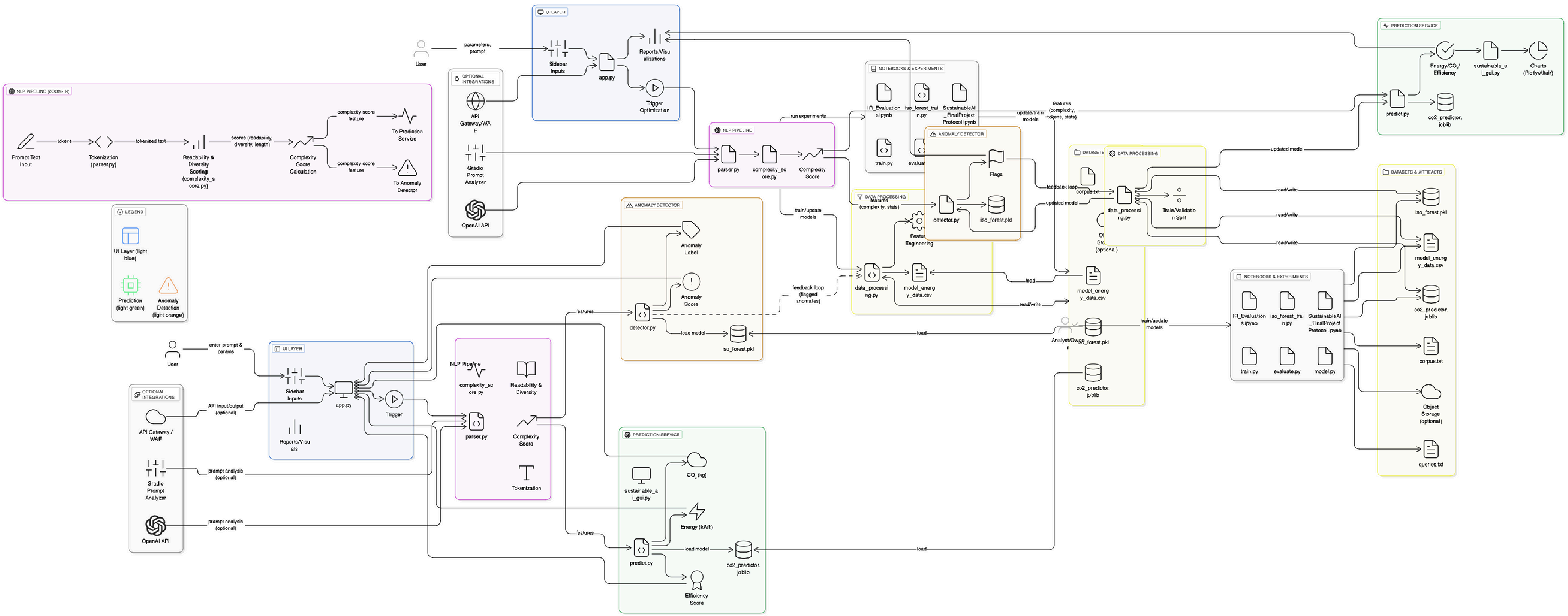
STEP 03

Anomaly Detection → inefficiency alerts.
Isolation Forest flags outlier prompts that consume unusually high energy for their complexity, triggering inefficiency alerts.

STEP 04

Visualization & recommendations via dashboard.
Our Streamlit dashboard visualizes energy use, CO_2 footprint, and efficiency scores while offering optimization tips to improve prompt efficiency.

SOLUTION ARCHITECTURE

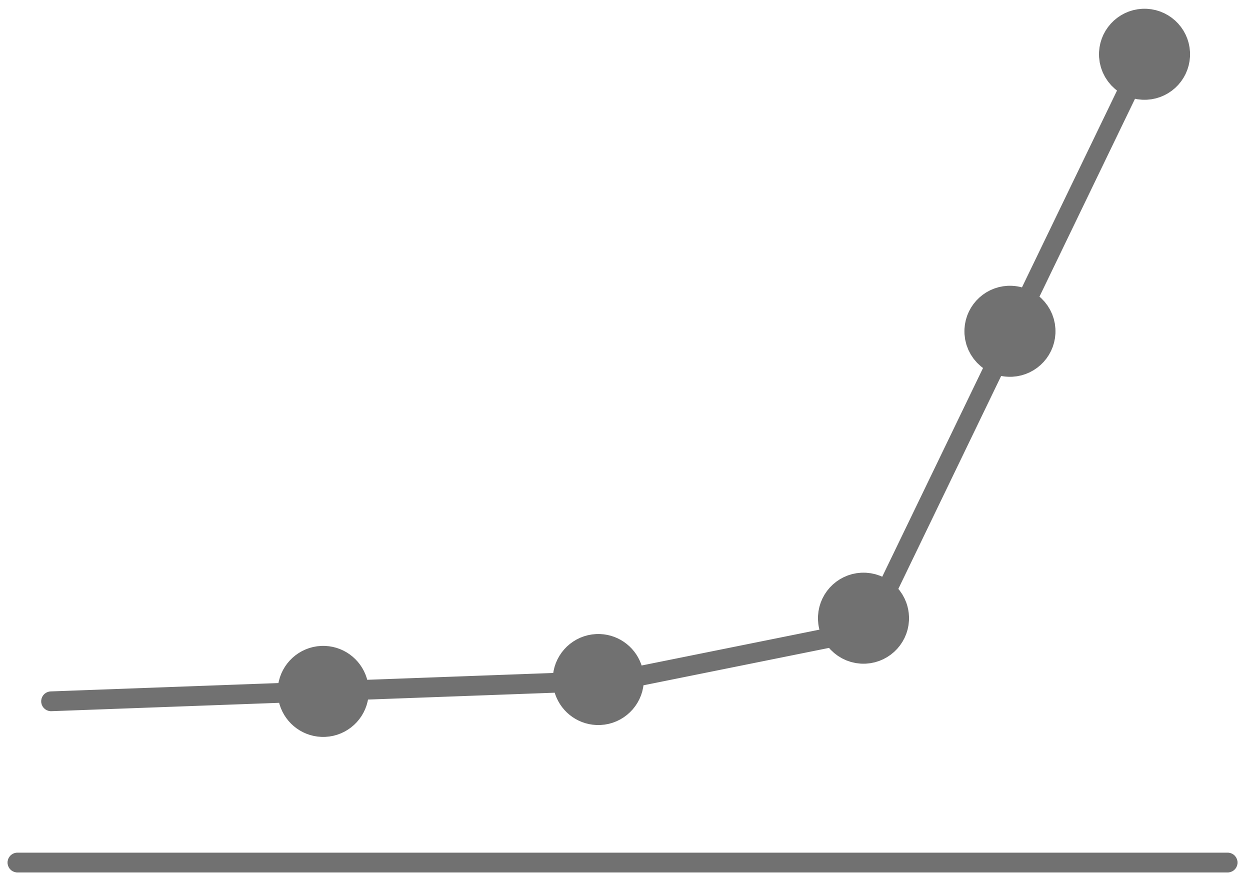


NLP Pipeline

- ✓ TOKENIZATION → BREAK TEXT INTO WORDS.
- ✓ READABILITY SCORE (FLESCH-KINCAID).
- ✓ VOCABULARY DIVERSITY → WORD VARIATION MEASURE.
- ✓ COMPLEXITY SCORE → WEIGHTED FINAL METRIC
- ✓ OUTPUT: NUMERIC FEATURE FOR PREDICTION & ANOMALY DETECTION

Regression Model

1. Algorithm: Linear Regression.
2. Variables: Num_Layers, FLOPs, Hours, Complexity.
3. Outputs: Energy (kWh) & CO_2 (kg).
4. Loss Function: Mean Squared Error (MSE).



Formula

$$\hat{y} = w1 \cdot Layers + w2 \cdot FLOPs + w3 \cdot Hours + w4 \cdot Complexity + b$$

Anomaly Detection

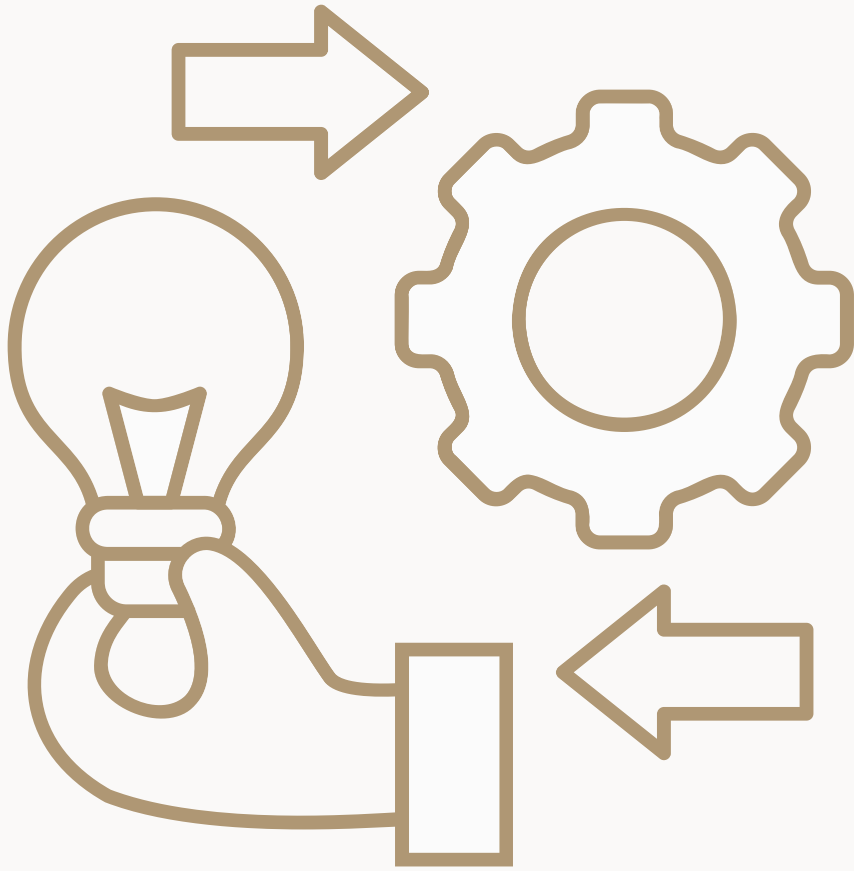
1. Algorithm: Isolation Forest.
2. Detects outlier configurations.
3. Alerts user if setup is unusually inefficient.
4. Integrated into dashboard in real-time.

Implementation

LANGUAGE: PYTHON 3.X.
LIBRARIES: SCIKIT-LEARN, PANDAS, NUMPY, NLTK, STREAMLIT, PLOTLY.

FILES:

1. PARSER.PY, COMPLEXITY_SCORE.PY – NLP.
2. TRAIN.PY – REGRESSION TRAINING.
3. ISO_FOREST_TRAIN.PY – ANOMALY DETECTION.
4. APP.PY – STREAMLIT UI.



RESULTS & DASHBOARD

Outputs

Energy estimate (kWh)

CO_2 estimate (kg)

Efficiency score

Optimization suggestions

Visuals

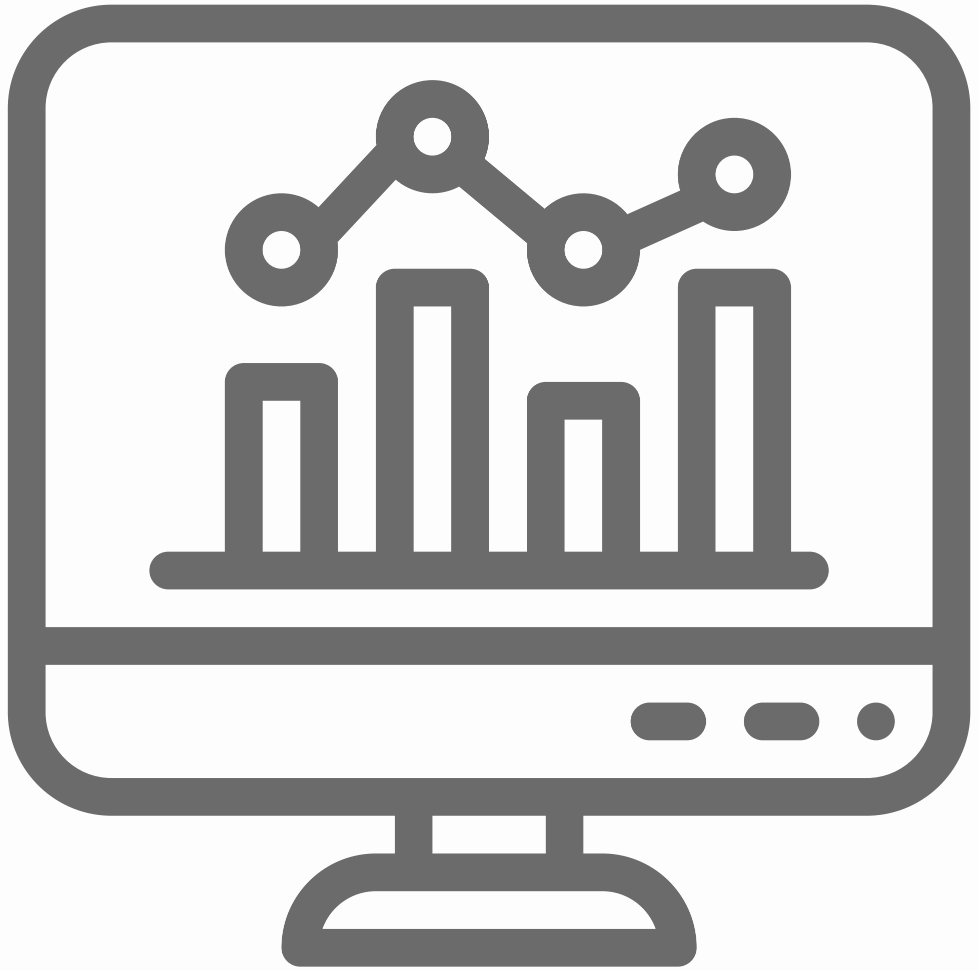
Gauge chart – Efficiency rating

Bar graph – Energy & CO_2 comparison

Pie chart – Compute contribution breakdown

Anomaly Alerts

Red flag indicators for inefficiency



EVALUATION

Regression Metrics

1. Mean Squared Error (MSE)
2. Mean Absolute Error (MAE)
3. R^2 (Coefficient of Determination)

Usability Testing

1. Real-time predictions
2. Minimal delay for live use

Code Quality

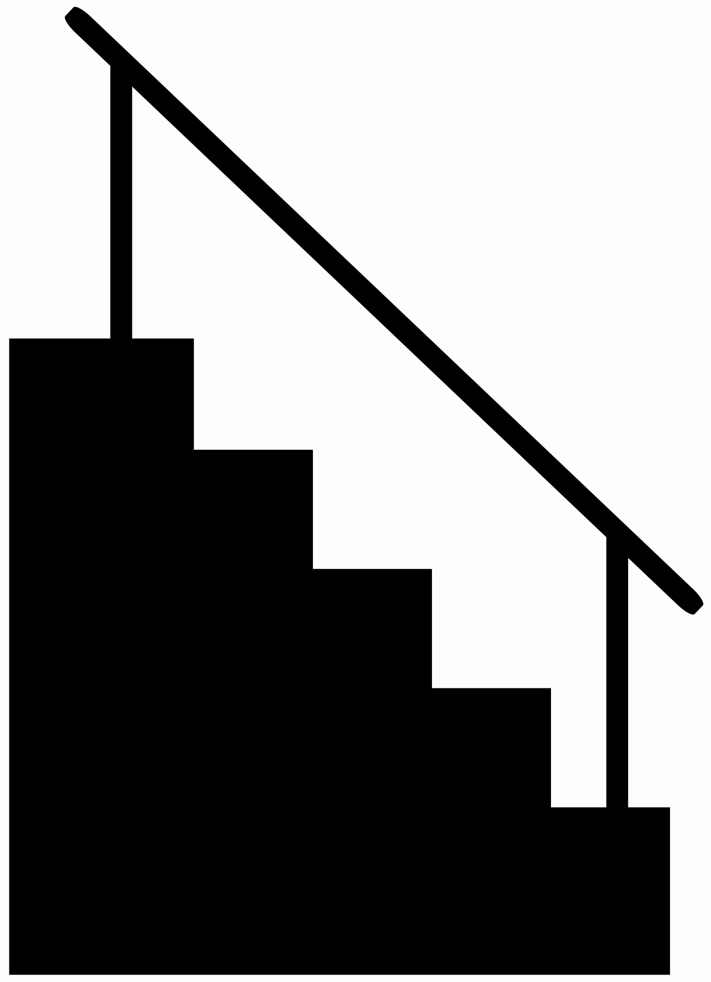
1. Modular structure
2. Well-documented
3. Maintainable for future updates

CONCLUSION

AI Efficiency Gains
Optimized prompts significantly improve AI efficiency, reducing energy usage and CO_2 emissions while maintaining output quality. Our model proved that even small prompt adjustments can lead to measurable sustainability impact.

Sustainability + Machine Learning
The project integrates sustainability goals directly into the AI development workflow, ensuring that environmental responsibility is built into the decision-making process, not added as an afterthought.

Technical and Business Value
By combining NLP, regression predictions, and anomaly detection in a single dashboard, we deliver both operational savings for developers and strategic ESG benefits for organizations.



FUTURE WORK

Real-Time Carbon Data-
Integrate live APIs to measure CO_2 intensity based on local grid conditions for more precise environmental tracking.

Multi-Model Support-
Extend compatibility to various AI architectures, ensuring flexibility as LLM technology evolves.

Automated Model Retraining-
Deploy continuous learning pipelines to keep regression models updated with the latest data and maintain accuracy.

Historical Trend Insights-
Introduce long-term tracking to measure efficiency progress over months or years, providing stakeholders with clear performance trends.