

https://github.com/pravin962624/EBPL

* College code: 3126
* College name : thangavelu engineering

College

* Department : IT
* Student nm-id: dbd9a4cb58ae73699ff4a12b101d6175
* Roll number : 312623205024
* Date : 14-5-2025

Submitted by,

* Name : PRAvIN KUMAR B
* MOBILE NO: 9629648632

**Phase 4: Performance of the Project Title: Energy Efficiency Optimization System** Objective:

The focus of Phase 4 is to enhance the performance and scalability of the Energy Efficiency Optimization (EEO) system. This involves refining machine learning models for precise energy usage prediction, optimizing real-time control mechanisms, ensuring compatibility with a broader range of energy devices, and enhancing data security. This phase also includes laying the groundwork for predictive maintenance and user-specific energy insights.

**1. Energy Usage Prediction Model Enhancement** Overview:

The machine learning model will be improved using recent consumption patterns and environmental data to more accurately predict energy demands across residential and commercial buildings.

Performance Improvements:

* Dataset Expansion: Integration of diverse data sources, including weather forecasts, occupancy patterns,and seasonal trends.
* Model Tuning: Application of ensemble learning and hyperparameter optimization to increase predictionaccuracy.

Outcome:

The prediction model will offer high-accuracy forecasts of short-term and long-term energy consumption, enabling smarter grid decisions and load balancing.

**2. Real-Time Control System Optimization** Overview:

Optimization of the real-time energy control algorithm will ensure minimal energy waste and improved system responsiveness under dynamic conditions.

Key Enhancements:

* Response Optimization: Faster execution of control decisions in response to changing energy loads andreal-time sensor feedback.
* Algorithm Refinement: Fine-tuning control logic to prioritize energy savings without compromising usercomfort.

Outcome:

A more responsive and efficient energy control system that reduces overall energy consumption while maintaining performance.

**3. IoT and Smart Device Integration** Overview:

This phase enhances compatibility with a wider range of smart appliances, meters, and building management systems.

Key Enhancements:

* Protocol Integration: Support for popular IoT protocols (Zigbee, MQTT, BACnet).
* Data Synchronization: Improved synchronization mechanisms for seamless communication betweenheterogeneous devices.

Outcome:

The system will function reliably with diverse smart devices, ensuring cohesive and comprehensive energy management.

**4. Data Security and Privacy Enhancements** Overview:

With increased user adoption, it's critical to reinforce system security, especially regarding user behavior and energy usage data.

Key Enhancements:

* Encryption Upgrades: Implementation of advanced encryption methods for data in transit and at rest.
* Security Testing: Penetration testing and anomaly detection systems to prevent unauthorized access.

Outcome:

A robust security framework that guarantees user privacy and maintains system integrity under high usage.

**5. Performance Testing and Metrics Collection** Overview:

Performance will be evaluated under high-load scenarios to ensure readiness for commercial deployment.

Implementation:

* Scalability Testing: Simulation of high-density usage across multiple locations.
* Metrics Analysis: Collection of metrics like energy savings, prediction accuracy, and system response times.
* User Feedback: Pilot testing with selected users to refine the UI and UX.

Outcome:

A thoroughly tested system capable of large-scale deployment with optimized energy management features.

**Key Challenges in Phase 4**

1. Scalability:

* Challenge: Handling large-scale deployments with diverse energy profiles.
* Solution: Modular architecture and cloud-based load balancing.

2. Data Privacy:

* Challenge: Managing sensitive consumption data securely.
* Solution: End-to-end encryption and regulatory compliance (e.g., GDPR).

3. Device Interoperability:

* Challenge: Seamless integration across different brands and standards.
* Solution: Use of universal adapters and protocol standardization.

**Outcomes of Phase 4**

1. Accurate Energy Predictions: Data-driven decisions for energy savings.
2. Optimized Real-Time Control: Faster, smarter energy adjustments.
3. Expanded Device Support: Compatibility with more hardware systems.
4. Enhanced Security: User data remains confidential and protected.

**Next Steps for Finalization**

In the next phase, the system will enter full deployment mode with continuous monitoring, real-time analytics dashboards, and preparation for integration with renewable energy sources such as solar and wind.

# Page 1: Source Code for Bar and Scatter Plot

import pandas as pd import matplotlib.pyplot as plt

# Load CSV data df = pd.read\_csv('global-data-on-sustainable-energy.csv')

# Filter data for Afghanistan (2000-2005) df\_afghanistan = df[(df['Entity'] == 'Afghanistan') & (df['Year'] >= 2000) & (df['Year'] <= 2005)]

# Bar Chart: Access to electricity plt.figure() plt.bar(df\_afghanistan['Year'], df\_afghanistan['Access to electricity (% of population)'], color='red') plt.title('Access to Electricity (2000-2005)') plt.xlabel('Year') plt.ylabel('% of Population') plt.savefig('bar\_chart.png')

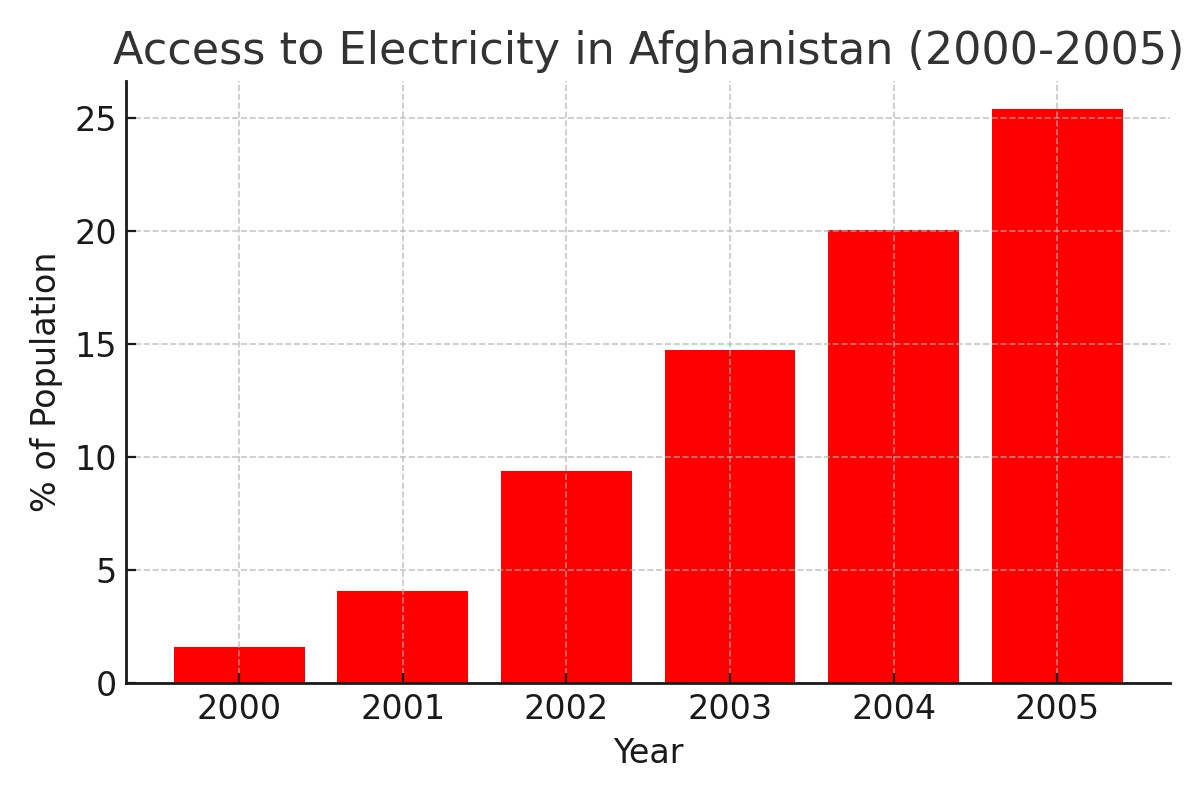
# Scatter Plot: GDP per capita vs Access to electricity plt.figure()

plt.scatter(df\_afghanistan['gdp\_per\_capita'], df\_afghanistan['Access to electricity (% of population)']) plt.title('GDP vs Electricity Access') plt.xlabel('GDP per Capita') plt.ylabel('Access to Electricity (%)')

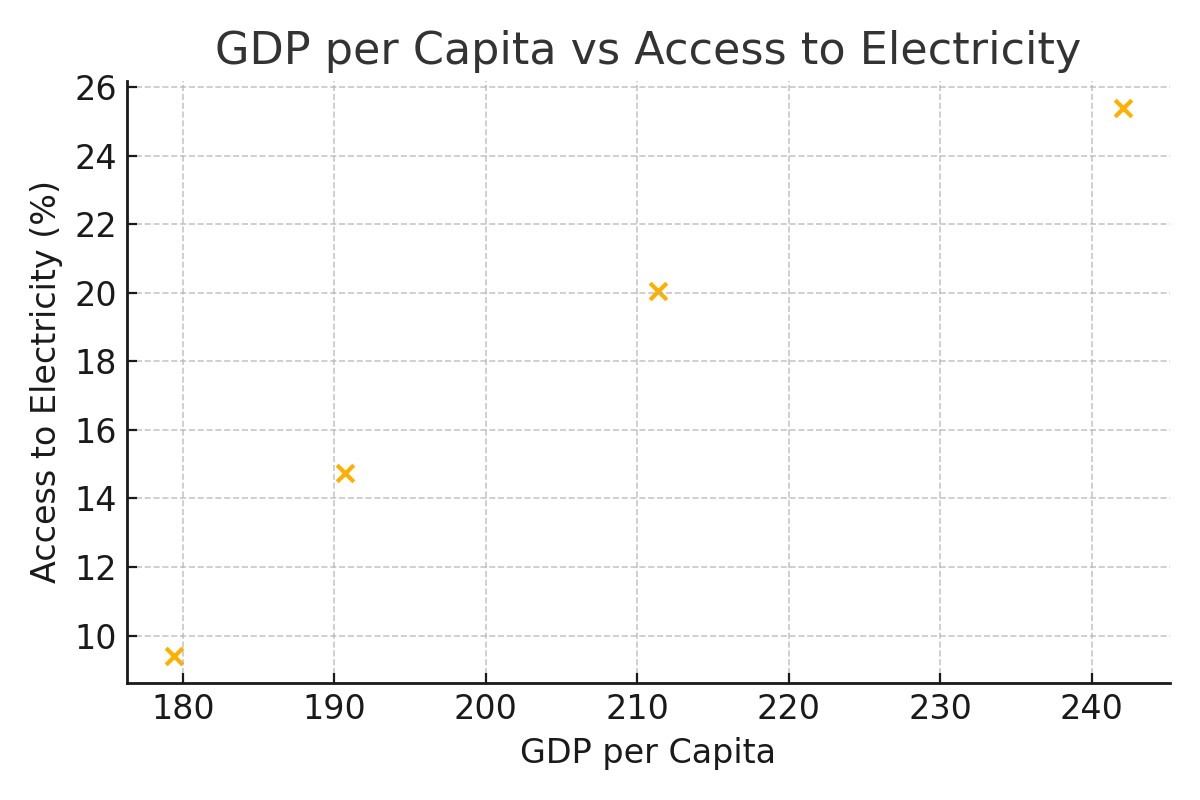
plt.savefig('scatter\_plot.png')

# Page 2: Output of Source Code

Bar Chart (Access to Electricity in Afghanistan):



Scatter Plot (GDP per Capita vs Access to Electricity):



# Page 3: Software Requirements

To run the above code and generate plots, you need the following software and libraries:

1. Python 3.x
2. pandas library (for data handling)
3. matplotlib library (for plotting charts)
4. Microsoft Word or any DOCX viewer to view this report
5. Jupyter Notebook or any Python IDE (like VS Code, PyCharm, etc.) for running code