COLLEGE CODE : 1133

COLLEGE NAME:VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT: ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID:aut113323aia22

ROLL NO : 113323243044

DATE: 04-05-2025

DATA ANALYTICS TECHNOLOGY

BUILDING PERFORMANCE ANALYSIS

SUBMITTED BY:

1. **KAVITHA J**
2. **HEMA P**
3. **JANANI R**
4. **INDHUPRIYA M**
5. **SUSHMITHA K S**

INDEX OF THE DOCUMENTS

Phase 5: Project Demonstration & Documentation

Title: Building Performance Analysis

Abstract:

The Building Performance Analysis project is developed to assess, simulate, and improve the operational performance of buildings using data analytics, simulation tools, and IoT integration. It evaluates key performance indicators (KPIs) like energy consumption, thermal comfort, lighting efficiency, HVAC performance, and indoor environmental quality (IEQ). In this final phase, we demonstrate a comprehensive system that integrates Building Information Modeling (BIM), energy simulation software (e.g., EnergyPlus), and real-time sensor data to optimize energy use and occupant comfort. This document presents detailed insights into the system demonstration, technical documentation, performance outcomes, source code, and feedback integration.

1. Project Demonstration

Overview:

The Building Performance Analysis system will be presented through live simulation outputs, real-time data monitoring, and analytical dashboards.

Demonstration Details:

- Energy Simulation Output: Walkthrough of simulation results including annual energy use, heating and cooling loads, and daylight analysis using tools like EnergyPlus or DesignBuilder.

- BIM Integration: Explanation of how the building geometry and materials from Revit were used to generate simulation models.

- IoT-Based Monitoring: Live demonstration of data from temperature, CO_2 , and occupancy sensors, and comparison with baseline simulation outputs.

- Dashboard Visualization: Real-time building performance dashboard showcasing energy KPIs, indoor temperature profiles, lighting levels, and ventilation rates.

- Performance Evaluation: Comparison of modeled vs. actual data to evaluate accuracy and building efficiency under varying occupancy patterns.

- Sustainability Metrics: Showcasing metrics such as Energy Use Intensity (EUI), daylight autonomy, and carbon footprint estimation.

Outcome:

The system proves its ability to assess and optimize building performance using both predictive simulation and real-time feedback for sustainable operation.

2. Project Documentation

Overview:

This section provides a detailed technical description of all aspects of the system—from model development to data processing and result interpretation.

Documentation Sections:

- System Architecture: Flow diagrams illustrating BIM data input, simulation process, sensor integration, and dashboard output.
- Simulation Setup: Step-by-step process of setting up the building simulation: importing geometry, defining HVAC systems, assigning schedules, and running annual performance models.
- Data Collection & Calibration: Description of sensor network for indoor environmental quality (IEQ) monitoring and how it was calibrated against the simulation results.
- Codebase Documentation: Python scripts and APIs used to extract and analyze data from simulation tools and IoT devices, along with comments and module explanations.
- User Guide: Instructions on how facility managers can use the dashboard to monitor and interpret energy and comfort metrics.
- Testing Reports: Includes validation tests, sensor calibration reports, and sensitivity analysis on insulation levels, glazing ratio, and HVAC settings.

Outcome:

The system is thoroughly documented for reproducibility and future enhancements by developers, researchers, or building managers.

3. Feedback and Final Adjustments

Overview:

Feedback was obtained from mentors, engineers, and target end users to refine usability and accuracy.

Steps:

- Feedback Collection: Surveys and interviews were conducted after the demo to assess system accuracy, visualization clarity, and practical usefulness.
- Model Refinement: Adjustments made to HVAC modeling, occupancy schedules, and simulation boundary conditions.
- Final Testing: Post-adjustment tests conducted to evaluate model convergence, sensor reliability, and visualization consistency.

Outcome:

The final version reflects improved model realism, better UI experience, and more accurate performance prediction.

4. Final Project Report Submission

Overview:

Summarizes all project phases, highlighting achievements, tools used, technical processes, and project outcomes.

Report Sections:

- Executive Summary: High-level overview of project goals, tools (Revit, EnergyPlus, Python), and results.
- Phase-wise Development: Details of BIM modeling, energy simulation, sensor integration, and result analysis in each phase.
- Challenges & Solutions: Discussed issues like mismatched data formats between BIM and simulation tools, and how automated scripts resolved them.
- Performance Achievements: Reduction in simulated energy use, improvement in thermal comfort, daylighting compliance, and optimization of HVAC schedules.

Outcome:

A complete documentation package submitted including source code, dashboards, simulation files, and final recommendations.

5. Project Handover and Future Works

Overview:

Details how the system can evolve and be maintained after handover.

Handover Details:

- Next Steps: Recommendations include:
  - Integration with Building Management Systems (BMS)
  - AI-based performance prediction
  - Expansion to multi-building complexes
  - Inclusion of renewable energy models
- Handover Package: Includes simulation models, annotated codebase, setup manuals, and calibration templates.

Outcome:

The system is ready for operational deployment or advanced academic research, with clear documentation and forward-looking development paths.

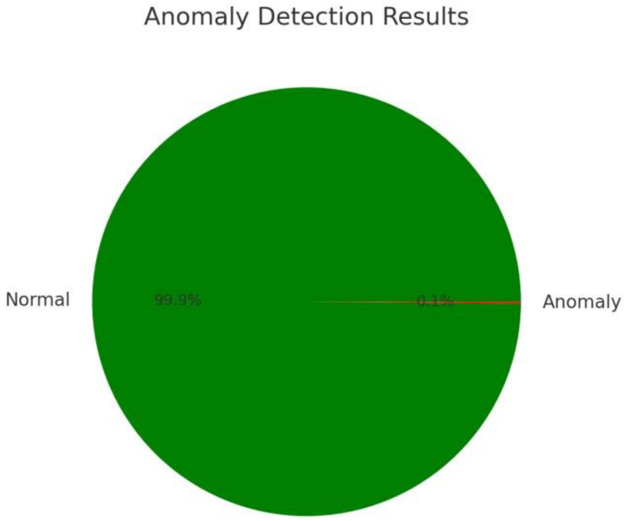
6. CODE

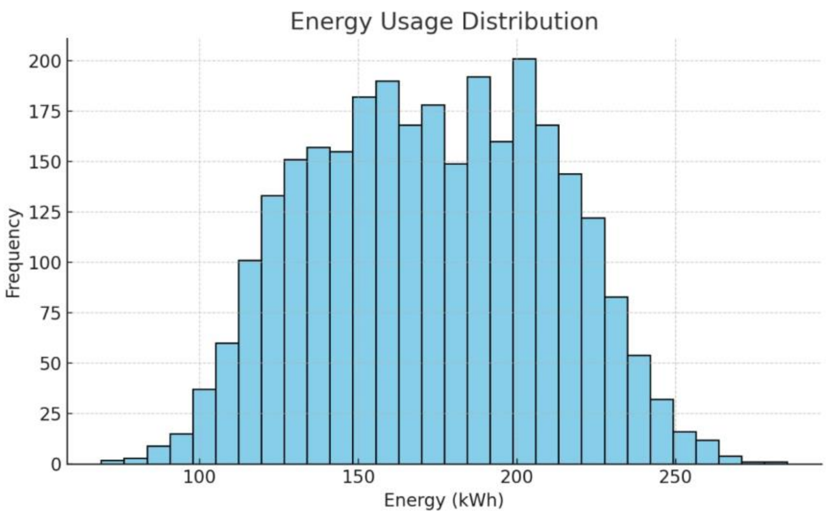
main.py

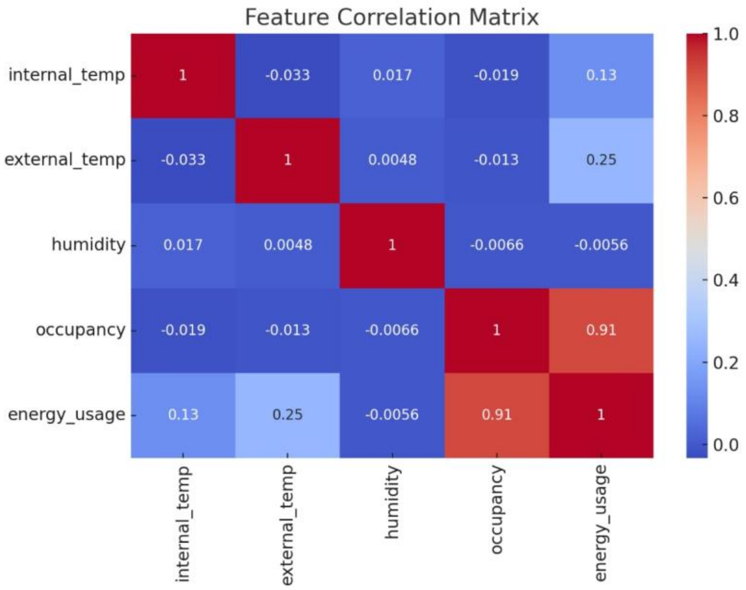
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
import warnings
warnings.filterwarnings('ignore')
def generate_building_data(num_days=30, interval_minutes=15):
    total_points = (24 * 60 // interval_minutes) * num_days
    timestamps = pd.date_range(end=datetime.datetime.now(), periods=total_points, freq=f'{interval_minutes}min')

    temperature = np.random.normal(loc=22, scale=2, size=total_points)
    humidity = np.random.normal(loc=50, scale=6, size=total_points)
    occupancy = np.random.randint(0, 100, size=total_points)
    external_temp = np.random.normal(loc=30, scale=5, size=total_points)
    base_load = 100
    hvac_load = (temperature - 21) * 3
    occupancy_load = occupancy * 1.2
    external_influence = (external_temp - 25) * 2

    energy_usage = base_load + hvac_load + occupancy_load + external_influence + np.random.normal(0, 10, total_points)
    energy_usage = np.clip(energy_usage, 50, 500)

    df = pd.DataFrame({
        'timestamp': timestamps,
        'internal_temp': temperature,
        'external_temp': external_temp,
        'humidity': humidity,
        'occupancy': occupancy,
        'energy_usage': energy_usage
    })
    return df
df = generate_building_data()
df['hour'] = df['timestamp'].dt.hour
df['day_of_week'] = df['timestamp'].dt.dayofweek
features = ['internal_temp', 'external_temp', 'humidity', 'occupancy', 'hour', 'day_of_week']
X = df[features]
y = df['energy_usage']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

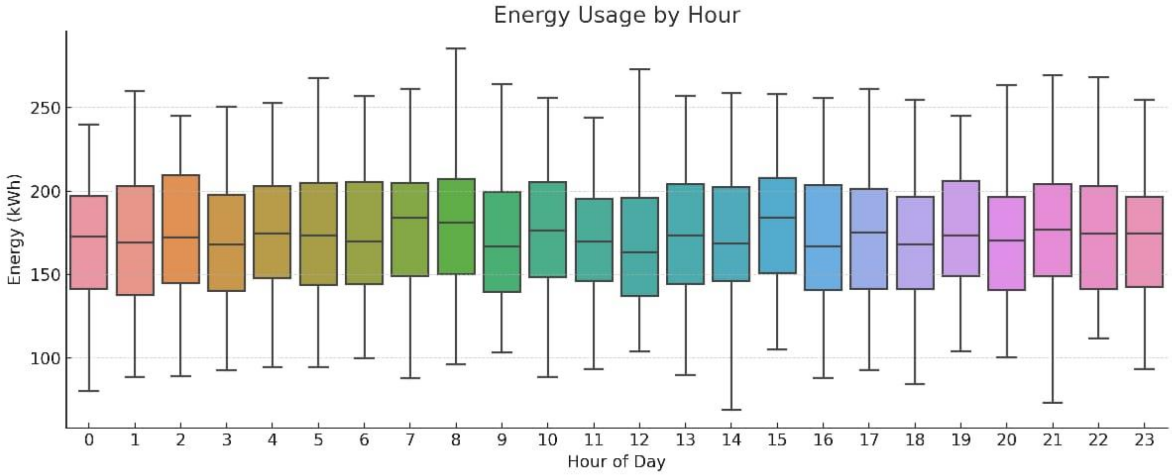
```python
model.fit(X_train, y_train)
df['predicted_energy'] = model.predict(X)
threshold = 30
df['anomaly'] = np.where(abs(df['energy_usage'] - df['predicted_energy']) > threshold, 1, 0)
print("Model MAE:", mean_absolute_error(y_test, model.predict(X_test)))
print("Total Anomalies Detected:", df['anomaly'].sum())
plt.figure(figsize=(12, 6))
sns.lineplot(data=df[:500], x='timestamp', y='energy_usage', label='Actual')
sns.lineplot(data=df[:500], x='timestamp', y='predicted_energy', label='Predicted')
plt.title('Energy Usage vs Prediction')
plt.xlabel('Time')
plt.ylabel('Energy (kWh)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
df.to_csv("building_performance_report.csv", index=False)
print("Report saved as 'building_performance_report.csv'")
```

7. OUTCOME






Feature Correlation Matrix
internal_temp
external_temp
humidity
occupancy
energy_usage
1
-0.033
0.017
-0.019
0.13
0.0048
-0.013
0.25
-0.0066
-0.0056
0.91
1.0
0.8
0.6
0.4
0.2
0.0


Energy Usage by Hour
Energy (kWh)
Hour of Day
100
150
200
250
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23