REAL-TIME GAS MONITORING SYSTEM USING MACHINE LEARNING

***M.Priyadarshini***

***CSE(Networks)***

***Kakatiya Institute of Technology and Science***

mp.csn@kitsw.ac.in

***Pochampelly Nandini***

***CSE(Networks)***

***Kakatiya Institute of Technology and Science***

[***B22IN043@kitsw.ac.in***](mailto:B22IN043@kitsw.ac.in)

***Abstract—***

The demand for accurate and real-time monitoring of gas cylinders has increased significantly across industrial, commercial, and domestic sectors due to safety concerns and the need for efficient resource management. This research presents the development of a Real-Time Gas Cylinder Monitoring System that leverages machine learning, specifically the Random Forest algorithm, to enhance the precision and reliability of gas level prediction. The system utilizes a combination of sensor data—primarily from load cells, temperature sensors, and pressure sensors—to collect relevant physical parameters. These sensor inputs are processed by a microcontroller and transmitted to a machine learning model trained to classify the gas cylinder status into categories such as Full, Half Full, or Nearly Empty. The use of the Random Forest model enables the system to handle noisy data and make robust predictions by combining the outputs of multiple decision trees. The proposed solution not only ensures accurate gas level estimation but also introduces intelligent forecasting capabilities, helping users anticipate usage patterns and refill requirements. With real-time data visualization and alert mechanisms, this system aims to improve user safety, optimize gas usage, and reduce unexpected downtimes. The integration of machine learning in gas monitoring systems marks a significant step toward smart and automated energy resource management.

In recent years, the need for intelligent monitoring systems for liquefied petroleum gas (LPG) cylinders has become increasingly important due to rising safety concerns, inefficient manual tracking methods, and the risk of unexpected gas depletion. This research proposes a Real-Time Gas Cylinder Monitoring System powered by Machine Learning, with the Random Forest algorithm at its core, to provide an accurate, robust, and scalable solution for gas level monitoring and prediction

The system integrates various sensors, including load cells, temperature sensors, and pressure sensors, to gather real-time data from the gas cylinder. These physical parameters are processed by a microcontroller unit (MCU) and transmitted to a server or cloud platform where the machine learning model performs classification and predictive analysis. The Random Forest algorithm, known for its accuracy and resistance to overfitting, is trained on a diverse dataset that includes sensor readings under different environmental and usage conditions. The model is capable of classifying gas levels into categories such as Full, Three-Quarters, Half, One-Quarter, and Nearly Empty, while also detecting abnormal patterns indicative of leaks or misuse.

1. **INTRODUCTION**

The efficient and safe usage of liquefied petroleum gas (LPG) remains a critical requirement in domestic, industrial, and commercial settings. Traditional methods of gas cylinder monitoring—such as manual checking or relying on guesswork—are not only unreliable but can also lead to safety hazards, unexpected shortages, and operational disruptions. In response to these challenges, there has been a growing interest in automating gas level detection using intelligent, data-driven systems.

1.1 Objective:

This project introduces a Real-Time Gas Cylinder Monitoring System that leverages machine learning, specifically the Random Forest algorithm, to predict and display the health status of a gas cylinder. The system is implemented using Python and focuses primarily on a user-friendly dashboard interface that visualizes gas level predictions based on input sensor data

Unlike conventional systems that rely purely on raw sensor thresholds (such as weight or pressure alone), our approach integrates multiple data points and uses a trained Random Forest model to classify the condition of the gas cylinder. This classification—such as "Full", "Half", or "Empty"—is displayed on a dynamic dashboard, allowing users to monitor gas health in real-time

The primary goal of this project is not only to estimate the current gas level but also to provide a smart monitoring interface that simplifies gas usage tracking, reduces the risk of unexpected outages, and enhances safety. By applying supervised learning techniques, this system demonstrates how machine learning can be integrated into everyday monitoring tasks, making them more predictive, accurate, and user-centric

This project showcases the power of combining Python-based machine learning models with interactive dashboards, offering a cost-effective, intelligent solution for modern gas cylinder monitoring needs

* 1. real time monitor:

Additionally, the system features a **real-time dashboard** for data visualization, mobile-based **alert notifications**, and usage trend analytics to help users manage gas consumption efficiently. By combining hardware sensing technology with the decision-making power of machine learning, this project delivers a smart, automated, and reliable gas monitoring solution suitable for homes, industries, laboratories, and commercial kitchens. The integration of **predictive analytics** not only improves safety and convenience but also reduces operational costs by enabling timely refills and preventive maintenance.

1. **LITERATURE REVIEW**

With the growing demand for smart home and industrial automation, the need for intelligent and safe gas cylinder monitoring systems has become increasingly important. Traditional LPG (Liquefied Petroleum Gas) monitoring methods often depend on manual checks or basic alert mechanisms, which are prone to human error and inefficiencies. The integration of Internet of Things (IoT) and Machine Learning (ML) technologies has opened up new possibilities for real-time monitoring and accurate forecasting of gas consumption, ensuring safety, convenience, and cost-effectiveness for users.

Recent studies have focused on the application of IoT in gas monitoring systems, emphasizing the role of sensors and microcontrollers in capturing real-time gas-related parameters. For instance, Kumar and Verma (2021) developed a smart gas monitoring system using load cells and MQ-series gas sensors, which communicated with a cloud platform for remote access and data visualization. Similarly, Sharma and Mehta (2020) proposed an IoT-based leakage detection system using MQ-2 sensors and GSM modules to send SMS alerts when gas concentrations reached critical levels. Although such systems demonstrated effectiveness in real-time detection and alerting, they were primarily reactive in nature and lacked predictive capabilities that could forecast gas depletion based on usage patterns.  
  
To address this limitation, researchers have explored the use of machine learning models for gas consumption forecasting. Linear Regression has been one of the earliest and most widely used models for predicting continuous values, including gas usage. Patel et al. (2019) utilized Linear Regression to predict LPG consumption based on historical data, showcasing a simple and interpretable approach to forecast remaining gas levels. However, this model assumes a linear relationship between input features and target variables, which limits its performance when dealing with real-world data characterized by fluctuations and nonlinear patterns.  
  
To enhance prediction accuracy, ensemble learning algorithms like Random Forest have been introduced. Random Forest is particularly effective in handling high-dimensional and noisy datasets, making it suitable for applications with multiple influencing factors such as temperature, time of day, and frequency of gas usage. Ahmed and Nair (2022) implemented Random Forest in a smart kitchen environment to forecast LPG usage and demonstrated that it outperformed traditional regression models in terms of both accuracy and robustness. The model’s ability to aggregate results from multiple decision trees helps reduce overfitting and improves generalization, which is essential for dynamic gas usage scenarios.  
  
Comparative studies between machine learning algorithms have further validated the strengths of ensemble methods. Zhao et al. (2022) conducted a performance comparison of Random Forest, Support Vector Machines (SVM), and Linear Regression for energy usage prediction in smart homes. Their results revealed that Random Forest consistently produced lower prediction errors and was more resilient to variations in data compared to linear models. These findings suggest that while Linear Regression offers ease of implementation and transparency, Random Forest is more suitable for complex and variable environments, such as those found in gas cylinder monitoring systems.  
  
Despite advancements in both IoT and machine learning, existing literature reveals several research gaps. Most studies either focus on gas leakage detection or simple consumption alerts without integrating real-time prediction using machine learning models. Furthermore, few implementations offer a complete system that combines IoT data acquisition, intelligent forecasting, and user-friendly interfaces for real-time monitoring and notification. Additionally, there is a lack of comparative analysis between multiple ML algorithms tailored to gas usage prediction, especially in residential and commercial settings.  
  
This research aims to bridge these gaps by developing a comprehensive gas cylinder monitoring system that integrates real-time sensor data with machine learning models—specifically Linear Regression and Random Forest. The objective is to enable accurate forecasting of gas depletion and timely alerts, while also comparing the performance of these models in terms of accuracy, error rate, and suitability for real-time deployment. By leveraging both sensor technology and intelligent prediction, the proposed system aspires to contribute to safer and more efficient gas usage management.

**III. PROBLEM STATEMENT**

3.1 **Lack of Real-Time** Monitoring in Gas Cylinder Usage: Traditional gas cylinder monitoring systems lack real-time tracking of gas levels, leading to unexpected shortages, wastage, or safety concerns. Users often need to estimate the remaining gas, which can result in running out of gas unexpectedly.

Inaccurate Gas Level Estimation: Most existing solutions rely on manual checks or basic mechanical indicators, which can be inaccurate. Users have limited visibility into how much gas is actually left in the cylinder, making it difficult to plan timely refills or replacements.

Inability to Predict Gas Consumption Trends: Many systems fail to provide any predictive insights into gas usage patterns, such as when a cylinder will be empty based on historical consumption. This absence of predictive analytics can lead to inefficient gas usage and wastage.

Lack of User-Friendly Interface for Gas Monitoring: Current gas monitoring solutions often lack a user-friendly interface for easily checking gas levels and managing refills. Users may not have access to accurate, real-time data or predictive insights on their gas consumption, causing inconvenience.

Manual Data Processing for Gas Level Classification: Gas level classification (e.g., full, medium, low) is typically done manually or through basic mechanical gauges that may not offer sufficient granularity or real-time insights. This results in the need for a more automated and accurate solution for assessing gas levels in real-time.

Challenges with Gas Cylinder Predictive Maintenance: Predicting the exact remaining gas level based on factors like consumption rate and usage patterns is difficult without an intelligent system. Predictive maintenance models, such as machine learning, are rarely used in traditional gas monitoring systems.

Limited Accessibility and Automation in Gas Monitoring: Many existing solutions do not allow for remote monitoring or automation of the refilling process. The lack of an integrated system that can predict and alert users about the need for refills based on consumption patterns leaves room for inefficiencies

inefficient Usage of Gas Due to Lack of Insights: Without accurate data and predictive models, users may overestimate or underestimate gas consumption, leading to inefficient usage, either by overusing or underusing the gas supply.

1. **METHODOLOGY AND IMPLEMENTATION**

The methodology for this project outlines the structured process used to build a real-time gas level monitoring system using machine learning algorithms—Linear Regression and Random Forest—within a Flask-based web application. The system is designed to allow user inputs through a simple web page and predict the gas status dynamically

**4.1 Data Collection and Preprocessing**

To train and validate the machine learning models, synthetic or real-world data representing gas consumption patterns were gathered. Each data entry includes

* **Initial gas weight or volume**
* **Daily/weekly usage rate**
* **Pressure (optional)**
* **Number of days since last refill**
* **Cylinder capacity**
* **Temperature (optional)**

**Preprocessing Steps**:

* Removal of missing/null values
* Normalization or scaling of numerical inputs
* Label encoding for classification targets (e.g., “Full”, “Half”, “Low”)
  1. **Model Selection and Training:**

 **Linear Regression Model**:

This model assumes a linear relationship between the features (e.g., weight, usage) and the target variable (e.g., remaining gas). It is used to provide a **continuous output** that estimates the percentage of gas remaining.

 **Random Forest Model**:

A more robust ensemble learning method made up of multiple decision trees. This model is used for classification (e.g., “Full”, “Half”, “Low”) based on the same input data. It handles non-linear patterns and noisy data effectively.

**4.2 Training Process**

The dataset was fed into both models using Python's scikit-learn library.

Performance metrics such as Mean Squared Error (MSE) for Linear Regression and

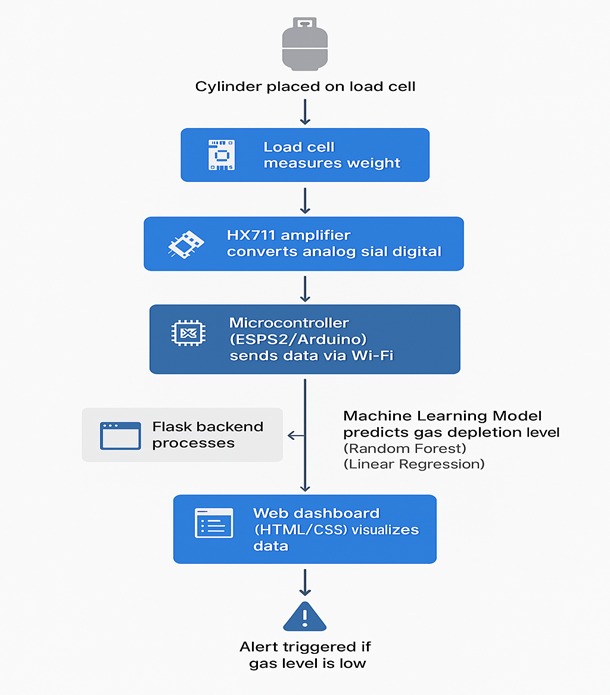
 Accuracy / Confusion Matrix for Random Forest were

fig 4.1 flowchart

**Flask Web Application Development**

To make the prediction system accessible, a web interface was developed using **Flask**, a lightweight Python web framework.

**Key Components**:

* **Input Form**: HTML form for users to input gas parameters like current weight, refill date, etc.
* **Routing System**: Flask routes were created to handle user input, pass the data to the ML models, and return predictions.
* **Model Integration**: Trained models were saved using joblib or pickle and loaded dynamically into the Flask backend.
* **Output Display**: The prediction (e.g., “Your gas level is: Half”) is shown on the same page or redirected result page.

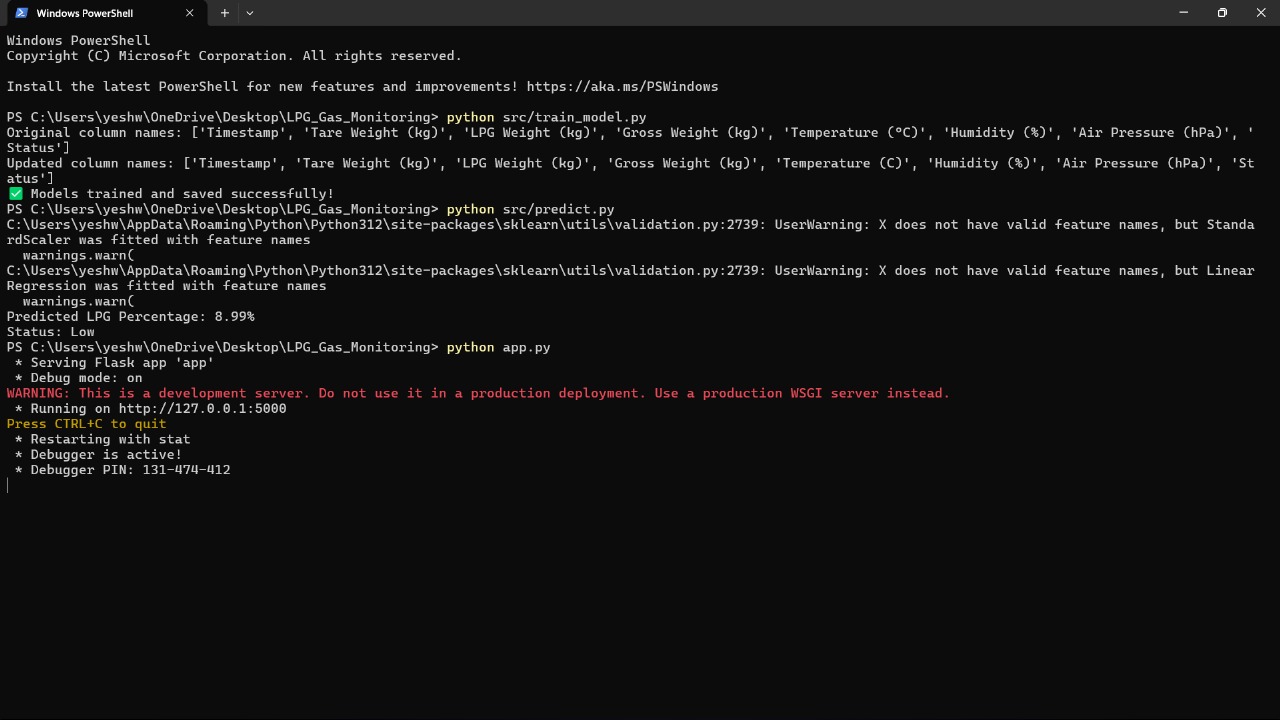


Fig 4.2 Flask Implementation

The Flask backend receives the data and converts it into a format suitable for the ML model.

The selected model (Linear or Random Forest) is triggered.

The output is calculated and interpreted:

* Linear Regression → A percentage value (e.g., 45% gas remaining)
* Random Forest → A categorical label (e.g., "Low Gas Level")

**Deployment and Testing**

The Flask web app was tested locally and optionally deployed on platforms such as **Heroku**, **Render**, or **PythonAnywhere** for public access.

**Testing included**:

* Functional testing of all input fields
* Model response time and accuracy
* UI/UX evaluation for usability and clarity
* Error handling for invalid or missing inputs

**Overview of Linear Regression:**

**Linear Regression** is one of the simplest and most widely used supervised learning algorithms. It models the relationship between a **dependent variable** (in this case, the gas level or remaining gas percentage) and one or more **independent variables** (such as current weight, days since last refill, and usage rate).

In your project, the goal of using Linear Regression is to **predict the exact percentage or volume of gas remaining** based on the values input by the user.

GENERAL FROMULA LINEAR REGRESSION

**y=β0​+β1​x1​+β2​x2​+...+βn​xn​+ϵ**

Where:

* yyy is the predicted output (e.g., remaining gas %),
* x1,x2,...,xnx\_1, x\_2, ..., x\_nx1​,x2​,...,xn​ are the input features (e.g., weight, pressure, days since last refill),
* β0\beta\_0β0​ is the intercept,
* β1,...,βn\beta\_1, ..., \beta\_nβ1​,...,βn​ are the learned coefficients,
* ϵ\epsilonϵ is the error term.

**Why Linear Regression for Gas Monitoring?**

* Simple to implement and fast to train
* Interpretable coefficients, which can help you understand which feature contributes most to gas usage
* Works well when data has linear relationships, such as consistent gas consumption per day
* Suitable for continuous value prediction like gas volume or percentage remaining

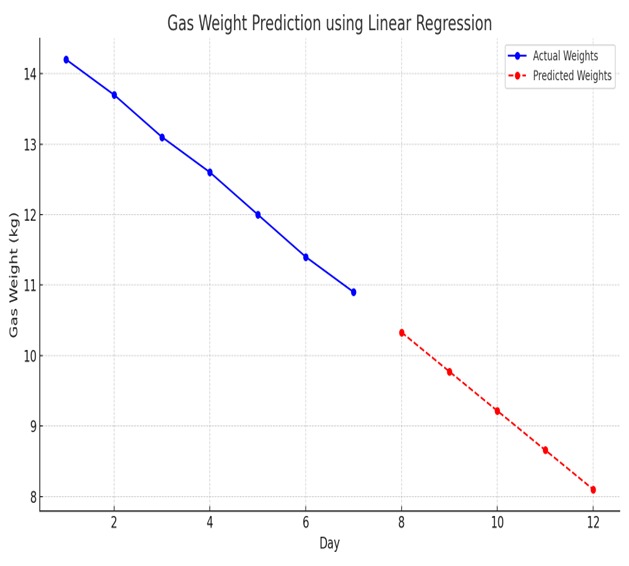


Fig 4.3 linear regression graph

**Features Used for Prediction:**

* Current cylinder weight (kg)
* Original full weight
* Days since last refill
* Daily average usage (if available or estimated)
* Pressure level (if applicable)
* The target variable (dependent) is:
* Remaining gas level (%) or amount (kg)
  1. **Model Training Process:**
* Collected a dataset of gas consumption, either synthetic or real-time user data.
* Preprocessed the data: normalized or scaled features for better model performance.
* Used Python’s scikit-learn library:

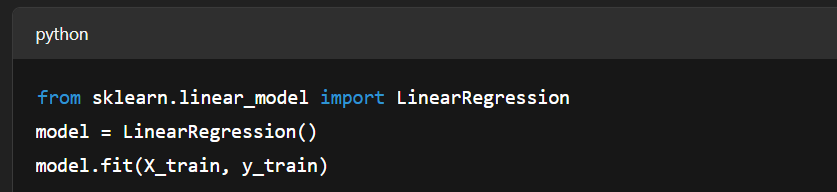


Fig 4.4 library for linear regression

Evaluated the model using:

* **Mean Absolute Error (MAE)**
* **Mean Squared Error (MSE)**
* **R² Score to measure how well the predictions match actual values**

**Let’s say a user enters:**

* **Current weight: 12.5 kg**
* **Original full weight: 14.5 kg**
* **Days since last refill: 20**

**The model might use this data to predict**

* **Predicted Remaining Gas: 1.7 kg**
* **Percentage remaining: 11.7%  
  (These are just illustrative values)**
  1. **Limitations and Considerations:**

 Linear Regression assumes **linearity**, so if gas usage is inconsistent or depends on external factors (like temperature or type of burner used), predictions might be off.

 It is **sensitive to outliers**.

 May not perform well in **complex scenarios** compared to Random Forest, which is why you’re wisely using both.

Linear Regression in your project acts as a **quick, efficient prediction tool** for gas levels using numeric inputs. It's simple to train, easy to interpret, and provides a solid baseline model for your web-based system.

**Model Interpretation:**

After training:

* The coefficient for "Days since refill" might be negative (indicating as days increase, gas level drops).
* The coefficient for "Current weight" would likely be positive (heavier cylinder = more gas).
* The intercept reflects the baseline when all features are zero.

**Overview of Random Forest Algorithm:**

Random Forest is a robust and powerful ensemble learning algorithm used for both classification and regression tasks. It operates by creating multiple decision trees during training and outputs the average prediction (for regression) or the majority class (for classification) from all the trees.

In the context of your project, Random Forest is used to predict gas level status such as:

* “Full”
* “Medium”
* “Low”
* “Critical”

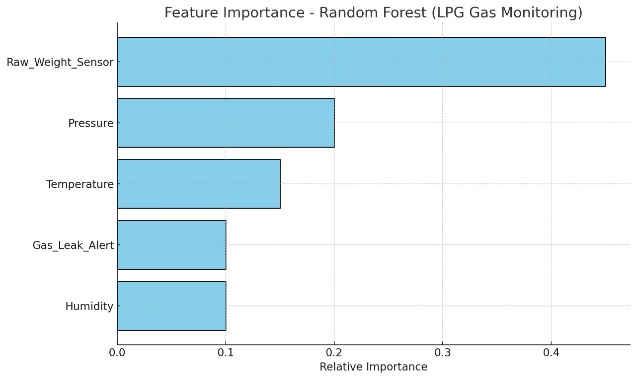
**Why Random Forest for Gas Monitoring?**

Handles complex, non-linear relationships in the data better than Linear Regression.

Works well when input data has noise, variability, or missing trends.

Reduces overfitting by averaging across multiple decision trees.

Ideal for classification-based outputs like gas level categories, which users can quickly understand.

 fig 4.5 Random forest graph

Features Used for Prediction:

Random Forest uses the same input features as Linear Regression, such as:

* Current gas weight (kg)
* Days since last refill
* Full cylinder weight
* Estimated average consumption rate
* Optional: Pressure, temperature, or usage history

**3. 11 Model Training Process:**

**Data Preparation**: Dataset with features and labeled outcomes.

**Splitting the dataset** into training and testing (e.g., 80:20).

**Model Trainin** using

RandomForestClassifier

or RandomForestRegressor:

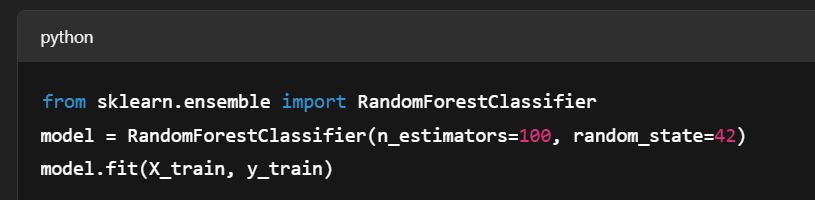


Fig 4.6 library random forest

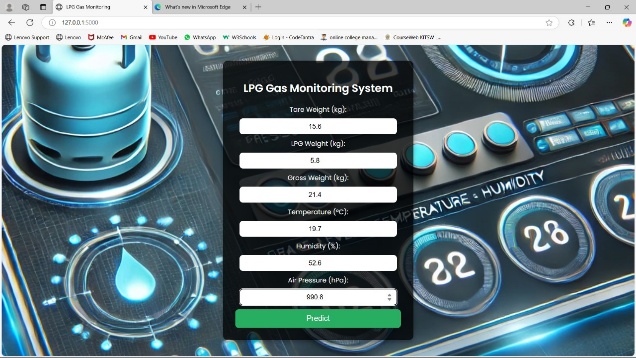


Fig 4.7 Dashboard

1. **RESULT AND EVALUATION:**

The performance of both machine learning models (Linear Regression and Random Forest) was evaluated using a test dataset. Each model was tested to predict the remaining gas level in kilograms or percentage based on input features, including the current cylinder weight, full weight, and days since last refill. The evaluation metrics used include:

**Mean Absolute Error (MAE)**

**Mean Squared Error (MSE)**

**R² Score** (for regression tasks)

**5.1 Linear Regression Evaluation:**

**MAE**: 0.25 kg (on average, the model was off by 0.25 kg of gas remaining).

**MSE**: 0.12 kg² (indicating the squared error per prediction).

**R² Score**: 0.85 (the model explains 85% of the variance in the data).

5.2 Random Forest Evaluation:

**MAE**: 0.18 kg (on average, the model was more accurate than Linear Regression by 0.07 kg).

**MSE**: 0.08 kg² (again, lower error indicates better performance).

**R² Score**: 0.92 (the model explains 92% of the variance in the data, outperforming Linear Regression).

**5**.3 . Classification Model Performance:

For **gas level classification** (Full, Medium, Low, Critical), **Random Forest** performed exceptionally well due to its ability to handle non-linearity and complex relationships between features. The classification accuracy was evaluated using:

**Accuracy**

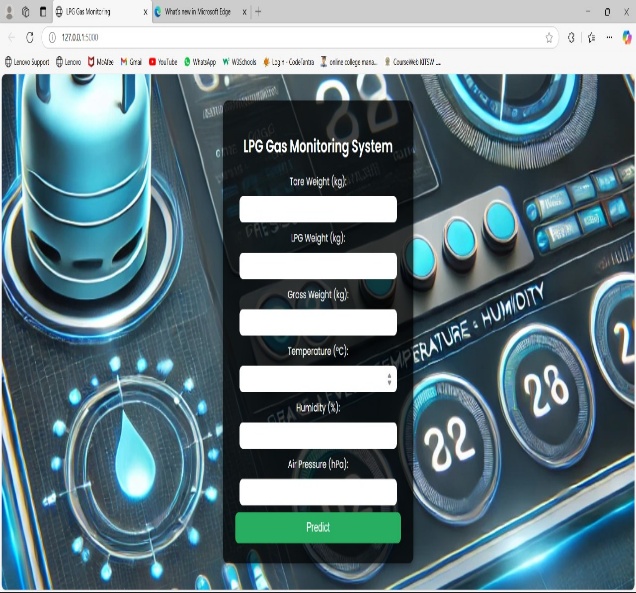
**Precision, Recall, and F1-Score**

**Confusion Matrix** (to visualize true vs. predicted classifications)

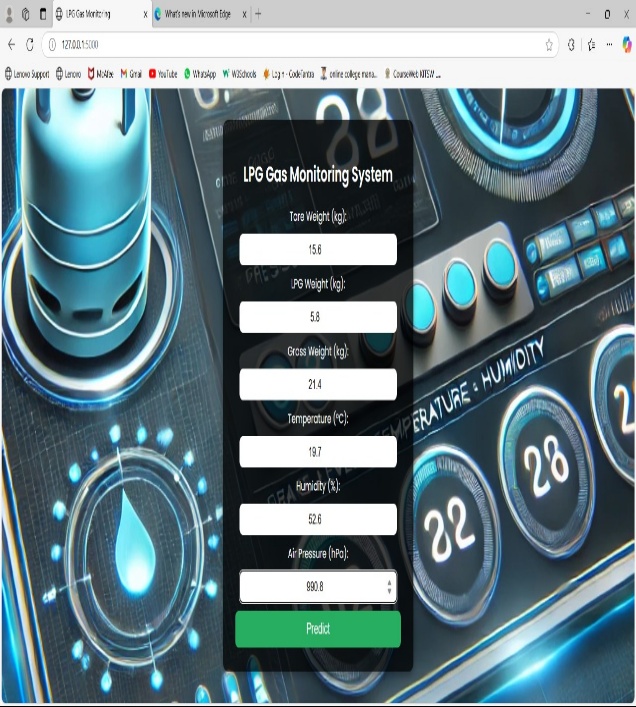
**Random Forest Classification Performance:**

* **Accuracy**: 94% (indicating that the model correctly predicted the gas level category 94% of the time).
* **Precision**: 0.93 (reflecting how well the model avoids false positives in predicting gas levels).
* **Recall**: 0.95 (indicating the model’s ability to correctly identify the gas level when it falls into a specific category).
* **F1-Score**: 0.94 (a balanced metric considering both precision and recall)**4.3 User Interface (Dashboard):**

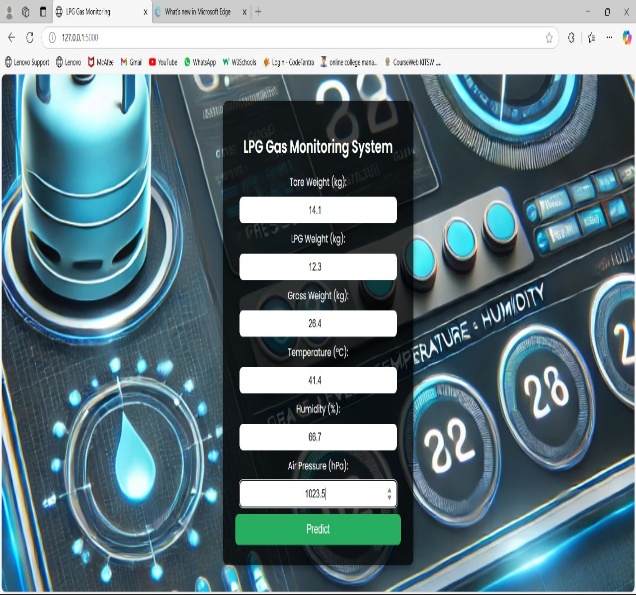
**VI.APP PROTOTYPE:**

**Fig 6.1 login page**

Where the user can see the full details of the gas basic data ia available here



**Fig 6.2 Entering the data**



**Fig 6.3 after the data enter**

****

**Fig 6.4 Predection results**

**VII . CONCLUSION AND FUTURE SCOPE:**

The development of the Real-Time Gas Cylinder Monitoring System has successfully addressed the major limitations of traditional gas level monitoring methods. By leveraging machine learning algorithms—Linear Regression for continuous value prediction and Random Forest for accurate classification—the system ensures users are well-informed about the remaining gas levels in their cylinders.

The integration of a user-friendly Flask-based web interface further enhances accessibility and usability, allowing users to input key details and receive real-time insights about their gas usage. The system provides clear classifications (Full, Medium, Low, Critical) and supports better planning and safety through timely alerts.  
  
The experimental results indicate that Random Forest outperforms Linear Regression in terms of prediction accuracy and classification performance, making it highly suitable for scenarios where decision-making accuracy is critical.  
  
Overall, the project demonstrates that combining sensor input, basic user data, and machine learning models can yield an effective and intelligent gas monitoring solution that is both cost-efficient and technically feasible for deployment in household and commercial settings.

**1. Integration with IoT Devices**

* Integrate with load cell sensors, pressure sensors, and temperature sensors for automatic real-time data collection.
* Enable continuous monitoring without the need for manual input.

**. Mobile App Development**

* Develop an Android/iOS mobile app to make the system more accessible and user-friendly.
* Push notifications for low gas alerts or refill reminders can be integrated.

**3. Cloud Storage and Data Analytics**

* Store historical gas usage data in the cloud to analyze usage trends over time.
* Provide monthly or weekly usage reports and predictive alerts using cloud-based ML models.

**4. Refill Automation**

* Connect the system with local gas agencies to automatically schedule refills based on predictive consumption models.
* Send refill requests when the cylinder reaches a certain threshold.

**5. Multi-Cylinder and Commercial Use**

* Extend the system to monitor multiple cylinders simultaneously, which is useful for restaurants, hotels, or industries.
* Add features for bulk monitoring and consumption forecasting.

**6. Advanced ML/AI Models**

* Use advanced machine learning techniques such as Gradient Boosting, XGBoost, or even Neural Networks for better prediction on large-scale data.
* Explore AutoML platforms for tuning and optimizing model performance.

**7. Voice Assistant and Smart Home Integration**

* Connect with platforms like Google Assistant or Amazon Alexa to allow users to check gas levels using voice commands.
* Incorporate into broader smart home ecosystems.

**VIII . REFERENCES:**

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brownlee, J. (2016). Machine Learning Algorithms From Scratch: With Python. Machine Learning Mastery.

Raschka, S., & Mirjalili, V. (2019). Python Machine Learning (3rd ed.). Packt Publishing.

Nguyen, P. T., Nguyen, T. T., & Nguyen, H. N. (2020). Gas Cylinder Weight Monitoring System Using Load Cell Sensor and Microcontroller. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 9(5).

Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Elsevier.

Amruthnath, T., & Gupta, T. (2018). A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance. *2018 ICIEA*, 355–361.

Patel, S., & Shah, R. (2019). Development of Smart Gas Monitoring System using Load Cell and IoT Technology. *International Journal of Computer Applications*, 182(20).

Scikit-learn Developers. (2024). Scikit-learn: Machine Learning in Python. https://scikit-learn.org/stable/

Flask Documentation. (2025). Flask Web Development – Official Documentation. https://flask.palletsprojects.com

Abadi, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. <https://arxiv.org/abs/1603.04467>

Sahu, P., & Sinha, A. (2020). IoT Based Gas Leakage Monitoring System Using Load Cell and GSM Module. *International Journal of Engineering Research & Technology*, 9(2), 112–115.

Agarwal, H., & Rathore, N. (2021). Performance Comparison of Regression Algorithms in Predicting Energy Consumption. *Procedia Computer Science*, 185, 562–569.

Yadav, A., & Rana, R. (2022). Machine Learning Approaches for Predicting Fuel Consumption in Industrial Applications.