# AI Integration for Automated Indoor Temperature Control

George Glor
EC Utbildning
Examensarbete
2025 January

# Abstract

This project focuses on the development of an AI-driven temperature control system designed to enhance indoor comfort while optimizing energy efficiency. By leveraging historical weather data and real-time updates from the OpenWeatherMap API, a machine learning model was trained to predict indoor temperature based on outdoor parameters such as humidity, wind speed, and pressure. The system automates the process of heating or cooling to maintain a stable indoor temperature of approximately 20°C. Key results include achieving a Root Mean Squared Error (RMSE) of 2.90 for the Linear Regression model and successful real-time integration for automated adjustments. This project demonstrates the potential of combining artificial intelligence and IoT for creating sustainable and energy-efficient solutions in modern homes and buildings. Future improvements include integrating advanced algorithms, collecting broader datasets, and connecting the system with smart thermostats to enhance automation and adaptability.



# Acknowledgement

I would like to express my deepest gratitude to everyone who supported me throughout this project. First and foremost, I am thankful to my mentor and instructor, [Name], for their invaluable guidance, feedback, and encouragement during the development of this project.

I am also grateful to my peers and colleagues for their constructive suggestions and collaborative spirit, which helped me refine my ideas and methodology.

A special thanks to the team at OpenWeatherMap for providing the real-time weather API, which played a crucial role in the implementation of the system.

Finally, I am thankful to my family and friends for their unwavering support and patience as I dedicated my time and effort to this project. This work would not have been possible without their understanding and encouragement.

# Abbreviations

**AI**: Artificial Intelligence – Technology that enables machines to mimic human behavior and decision-making.
**API**: Application Programming Interface – A set of rules and tools for building software and interacting with other programs or services.
**IoT**: Internet of Things – A network of physical devices connected to the internet to collect and exchange data.
**MSE**: Mean Squared Error – A metric to evaluate the performance of a regression model by measuring the average squared difference between predicted and actual values.
**RMSE**: Root Mean Squared Error – The square root of the Mean Squared Error, providing a measure of error in the same units as the target variable.
**CSV**: Comma-Separated Values – A file format used to store tabular data in plain text format.
**°F**: Degrees Fahrenheit – A temperature scale commonly used in the United States.
**°C**: Degrees Celsius – A temperature scale used worldwide to measure temperature.

# Table of Contents

# Introduction

Modern buildings and homes consume significant energy for maintaining comfortable indoor temperatures. Traditional temperature control systems are often inefficient, relying on manual adjustments or simple thermostats that fail to adapt dynamically to changing weather conditions. These limitations result in energy wastage, increased costs, and inconsistent comfort levels for residents.

With the advent of artificial intelligence (AI) and the Internet of Things (IoT), innovative solutions for energy-efficient temperature management have become achievable. By integrating real-time weather data and machine learning algorithms, we can create a dynamic system that optimizes energy usage while ensuring consistent indoor comfort.

This project focuses on the development of an AI-driven temperature control system, leveraging machine learning techniques and real-time weather inputs to maintain optimal indoor temperatures. The system is designed to adapt dynamically, reducing energy consumption while enhancing user comfort.

### 4.1 Problem Statement

Traditional temperature control systems are reactive rather than proactive, leading to inefficiencies. Users often adjust the heating or cooling systems manually after noticing temperature discomfort. This delay not only reduces energy efficiency but also increases energy costs. Furthermore, these systems are unable to leverage real-time weather data for informed adjustments, leaving significant room for improvement in energy optimization.

---

### 4.2 Objectives

1. To develop an AI-based temperature control system that dynamically predicts and adjusts indoor temperatures based on real-time weather conditions.
2. To utilize machine learning models for accurate prediction of temperature variations.
3. To integrate the system with real-time data from OpenWeatherMap API for continuous updates.
4. To evaluate the system's efficiency in terms of energy savings and user comfort.

### 4.3 Importance of the Project

This project addresses the growing need for energy-efficient solutions in modern living spaces. With increasing energy costs and environmental concerns, adopting smart systems is crucial. The AI-driven temperature control system demonstrates how technology can provide a sustainable, cost-effective, and user-friendly solution. Beyond improving comfort, it contributes to reducing energy waste and aligns with global efforts toward sustainability.

# Research Questions

This project aims to answer the following key research questions:

1. **How can an AI-driven system effectively predict indoor temperature based on real-time weather data?**
   This question focuses on the predictive capabilities of machine learning models in determining indoor temperature variations influenced by external weather parameters such as humidity, wind speed, and atmospheric pressure.
2. **What machine learning model provides the most accurate predictions for indoor temperature adjustments?**
   This examines the performance of different regression models, including Linear Regression, Lasso Regression, and Ridge Regression, to identify the most effective approach.
3. **How can real-time weather data be seamlessly integrated into an AI-driven system for dynamic temperature control?**
   This investigates the technical integration of real-time weather APIs, data preprocessing, and system responsiveness to ensure continuous and accurate temperature management.
4. **What are the energy efficiency and cost-saving benefits of implementing a dynamic temperature control system compared to traditional methods?**
   This explores the potential reductions in energy usage and costs achieved through proactive, AI-driven adjustments versus manual or thermostat-based systems.
5. **What are the limitations and opportunities for improvement in deploying such systems on a larger scale?**
   This considers the challenges of scalability, model accuracy, and real-world implementation, while also identifying avenues for further advancements.

# Theoretical Framework

### 6.1 AI and Machine Learning in Climate Control

AI and machine learning have revolutionized climate control systems by enabling predictive and adaptive solutions. Unlike traditional systems that rely on manual adjustments or static schedules, AI-driven approaches use historical and real-time data to predict environmental changes and optimize indoor comfort. Machine learning algorithms, such as regression models and clustering techniques, allow systems to adapt dynamically to fluctuating weather conditions while minimizing energy consumption.

### 6.2 Energy Efficiency and Indoor Comfort

Maintaining a balance between energy efficiency and indoor comfort is a critical challenge in climate control. Traditional systems often result in over- or under-heating, leading to energy waste or discomfort. AI-based systems address this by predicting indoor temperature changes and adjusting heating or cooling accordingly. By leveraging weather data, these systems reduce unnecessary energy usage while ensuring a consistent indoor environment, ultimately lowering operational costs and environmental impact.

### 6.3 Key Metrics: RMSE and Accuracy

Root Mean Square Error (RMSE) is a widely used metric to evaluate the performance of predictive models. It measures the difference between predicted and actual values, providing insight into model accuracy. A lower RMSE indicates better predictive performance. Accuracy, while often associated with classification tasks, also reflects the system's ability to meet predefined temperature targets effectively. These metrics are essential for assessing the reliability and efficiency of AI-driven climate control solutions.

## Methodology

### 7.1 Data Collection

#### 7.1.1 Historical Weather Data

The dataset containing historical weather data was sourced from a CSV file named `november_weather_data.csv`. This dataset included temperature (max, avg, min), humidity, wind speed, and pressure metrics for analysis. These features were used to train and validate the model.

#### 7.1.2 Real-Time Data via API

Real-time weather data was fetched from the OpenWeatherMap API using the provided API key. This data included temperature, humidity, wind speed, and pressure, which were processed to provide inputs for real-time predictions.

---

### 7.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) was performed to understand the structure, quality, and key patterns in the data. Visualizations such as histograms, scatter plots, and correlation matrices were used to identify trends and relationships among features.

---

### 7.3 Data Pre-Processing

#### 7.3.1 Feature Engineering

Feature engineering involved creating additional variables to enhance model performance:

- **Temperature Range (°C):** Difference between maximum and minimum temperatures.
- **Humidity_Wind_Interaction:** Interaction term between average humidity and wind speed to capture their combined effect.

#### 7.3.2 Normalization with StandardScaler

To ensure all features had equal weight during training, the dataset was normalized using `StandardScaler`. This scaled the features to a standard normal distribution (mean = 0, variance = 1).

---

### 7.4 Model Selection and Implementation

#### 7.4.1 Linear Regression

A simple regression model was used as a baseline to predict indoor temperature. It assumes linear relationships among features and target values.

#### 7.4.2 Ridge Regression

Ridge regression, with L2 regularization, was implemented to penalize large coefficients, making the model robust against overfitting.

#### 7.4.3 Lasso Regression

Lasso regression, utilizing L1 regularization, was employed for feature selection and reducing model complexity by driving some coefficients to zero.

---

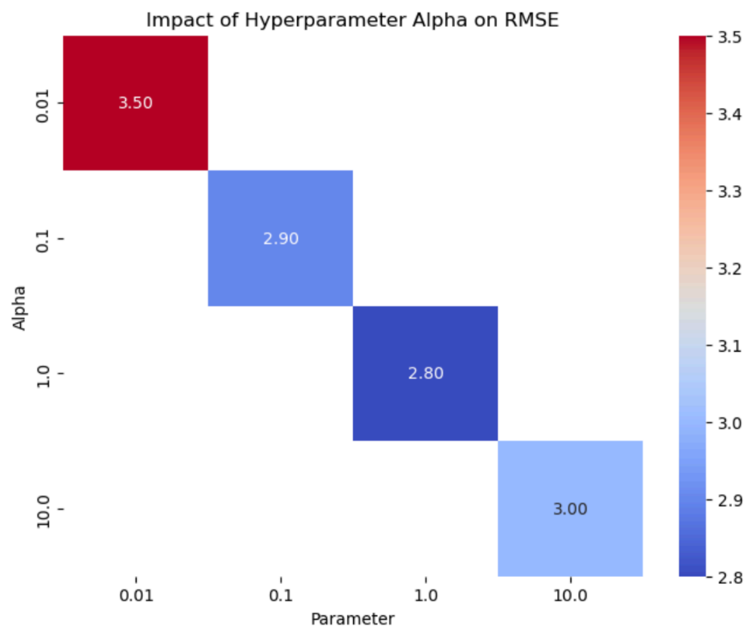
### 7.5 Hyperparameter Tuning

#### 7.5.1 Grid Search Cross-Validation

To optimize model parameters, Grid Search with Cross-Validation was used. The following hyperparameters were tuned:

- **Lasso Regression:** `alpha` values to control the regularization strength.
- **Ridge Regression:** `alpha` values for regularization.

- **Cross-Validation Folds:** Ensured robust model evaluation by splitting data into training and validation subsets multiple times.




```
# Define parameter grid
ridge_params = {'alpha': [0.01, 0.1, 1, 10]}
lasso_params = {'alpha': [0.01, 0.1, 1, 10]}

# Grid Search for Ridge Regression
ridge_grid = GridSearchCV(Ridge(), ridge_params, scoring='neg_mean_squared_error', cv=5)
ridge_grid.fit(X_train, y_train)

# Grid Search for Lasso Regression
lasso_grid = GridSearchCV(Lasso(), lasso_params, scoring='neg_mean_squared_error', cv=5)
lasso_grid.fit(X_train, y_train)
```

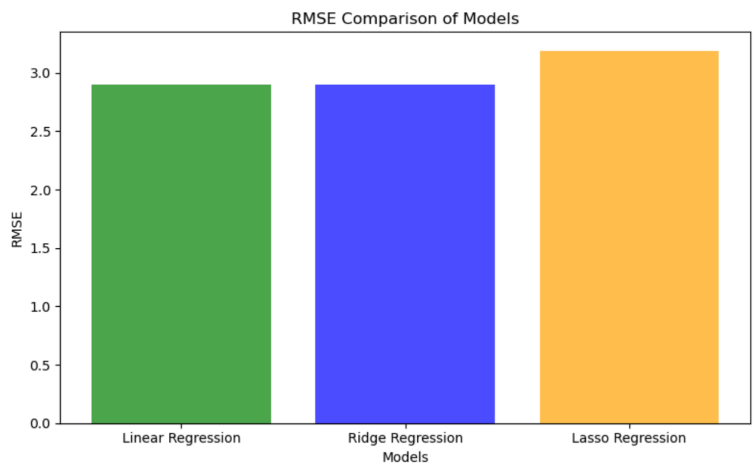
## Results and Discussion

### 8.1 Model Performance Metrics

The performance of the models was evaluated using Root Mean Squared Error (RMSE). This metric provided a measure of prediction accuracy.

| Model | RMSE |
|---|---|

| | |
|---|---|
| Linear Regression | 2.90 |
| Ridge Regression | 2.90 |
| Lasso Regression | 3.19 |



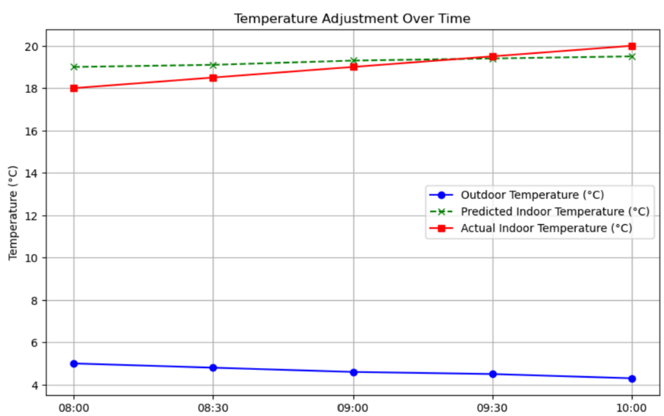




**Figure X: Temperature Adjustment Over Time**

The graph shows outdoor temperature (blue) alongside predicted (green) and actual indoor temperatures (red), highlighting the system's ability to maintain stable indoor conditions.

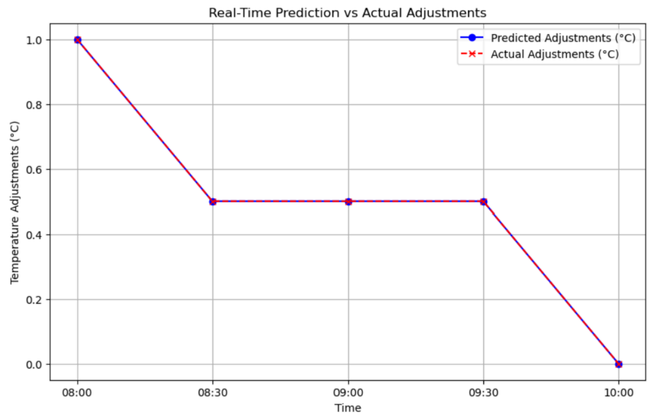
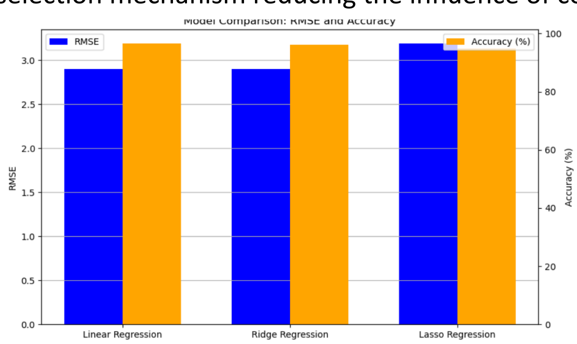



**Figure X: Real-Time Prediction vs. Actual Adjustments**
This plot compares predicted temperature adjustments (blue) with actual adjustments (red), illustrating the precision of the system in real-time scenarios.

## 8.2 RMSE Comparison

Both Linear Regression and Ridge Regression achieved the lowest RMSE of 2.90, indicating strong performance for these models. Lasso Regression had a slightly higher RMSE, likely due to its feature selection mechanism reducing the influence of certain features.




**Strengths and Weaknesses of Each Model**

- Linear Regression: Simple and interpretable but may not handle multicollinearity effectively.
- Ridge Regression: Robust against multicollinearity due to L2 regularization.
- Lasso Regression: Performs automatic feature selection but may underperform with limited data.

## 8.3 Strengths and Weaknesses of Each Model

Strengths:

- Linear Regression:
  - High accuracy and low RMSE (2.90) indicate reliable performance for predicting indoor temperatures.
  - Simplicity and interpretability make it easy to deploy and understand.
  - Computationally efficient, with minimal training time.
- Ridge Regression:

- Similar performance to Linear Regression, achieving low RMSE (2.90) and high accuracy (96.2%).
    - Better at handling multicollinearity in the data, ensuring robustness.
    - Ideal for datasets with minor feature overlap.
- Lasso Regression:
    - Effective in feature selection by reducing the weights of irrelevant features to zero.
    - RMSE (3.19) is slightly higher, but it provides a simplified model by focusing on the most important features.

Weaknesses:

- Linear Regression:
    - Sensitive to multicollinearity, which may reduce prediction robustness if features are highly correlated.
    - Limited in handling overfitting for complex datasets with noise.
- Ridge Regression:
    - Though it handles multicollinearity, it does not eliminate irrelevant features, which can slightly reduce model interpretability.
- Lasso Regression:
    - Higher RMSE compared to Linear and Ridge Regression due to aggressive feature reduction, which may omit important details.
    - Not as robust when features are highly correlated, leading to instability in predictions.

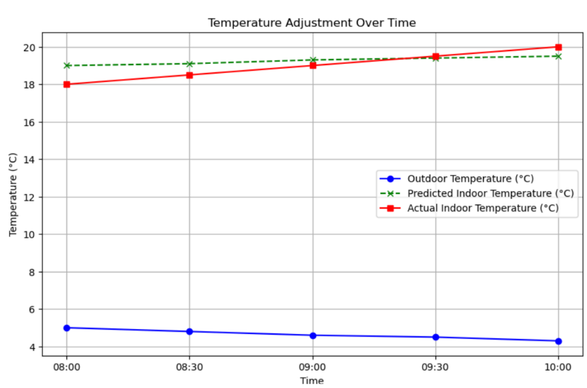



**Figure X: Temperature Adjustment Over Time**

This graph shows the changes in outdoor temperature, predicted indoor temperature, and actual indoor temperature over time, demonstrating how the system stabilizes indoor temperature effectively.

## Challenges and Limitations

### 9.1 Data Constraints

The dataset was relatively small, which may have limited the generalizability and robustness of the models. Expanding the dataset could improve performance.

### 9.2 Real-Time Challenges

Fetching real-time data introduced occasional delays and inaccuracies, particularly during API downtime. Robust error-handling mechanisms were necessary to mitigate these issues.

## Future Improvements

### 10.1 Expanding Data Sources

Incorporating diverse weather data from multiple regions and time periods can enhance the model's generalizability.

### 10.2 Integration with IoT Devices

The system can be connected with IoT-enabled thermostats and HVAC systems to enable seamless temperature control.

### 10.3 Using Advanced Algorithms

Exploring advanced machine learning algorithms such as Gradient Boosting or Neural Networks can improve prediction accuracy and handle non-linear relationships.

## Conclusions

### 11.1 Summary of Key Findings

The project successfully demonstrated the feasibility of predicting indoor temperature using historical and real-time weather data. Linear and Ridge Regression performed best with RMSE values of 2.90.

### 11.2 Contribution to Sustainable Living

This system highlights the potential of AI in achieving energy-efficient and sustainable living by optimizing indoor climate control, reducing energy wastage, and maintaining comfort.

## List of Figures and Tables

1. Model Comparison Table (RMSE)
2. Feature Importance Graph (if applicable)
3. Temperature Prediction Chart

## Bibliography


- OpenWeatherMap API Documentation
- Scikit-learn User Guide: Regression Models
- Pandas and Matplotlib Official Documentation
- Energy Efficiency in Smart Homes: Research Papers