High Level Design
(HLD)
Energy Efficiency Prediction
Model

Document Version Control

ID	Date Issued	Version	Description	Author
01	14/01/2024	0.1	initial HLD - V 1.0	Hasham Akram

Contents

Abstract

The High-Level Design (HLD) for the Structures Energy Efficiency project outlines the conceptual architecture and major components involved in developing a machine learning model for predicting heating and cooling loads in residential buildings. This abstract provides a concise overview of the project's scope, objectives, and key architectural considerations.

The project aims to leverage statistical machine learning techniques to analyze the impact of various input variables on heating load (HL) and cooling load (CL) in residential buildings. By employing classical and non-parametric statistical tools, the project seeks to identify correlations between input variables and output variables to accurately predict energy efficiency factors.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact ata high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:

o Reliability

o Security

o Maintainability

o Portability

o Reusability

o Application compatibility

o Resource utilization

o Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical tomildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
● UGV ● Database ● IDE ● AWS	● Unmanned Ground Vehicle ● Collection of all the information monitored by this system ● Integrated Development Environment ● Amazon Web Service

2 General Description

2.1 Product Perspective:

The Energy Efficiency Prediction is a machine learning-based predictive model which will help us to predict the coefficients of heating and cooling parameters.

2.2 Problem Statement:

To develop an API interface to predict the Energy Efficiency revolves around accurately predicting heating and cooling loads in residential buildings based on a set of independent variables or factors and analyze the following:

Dependent Factors

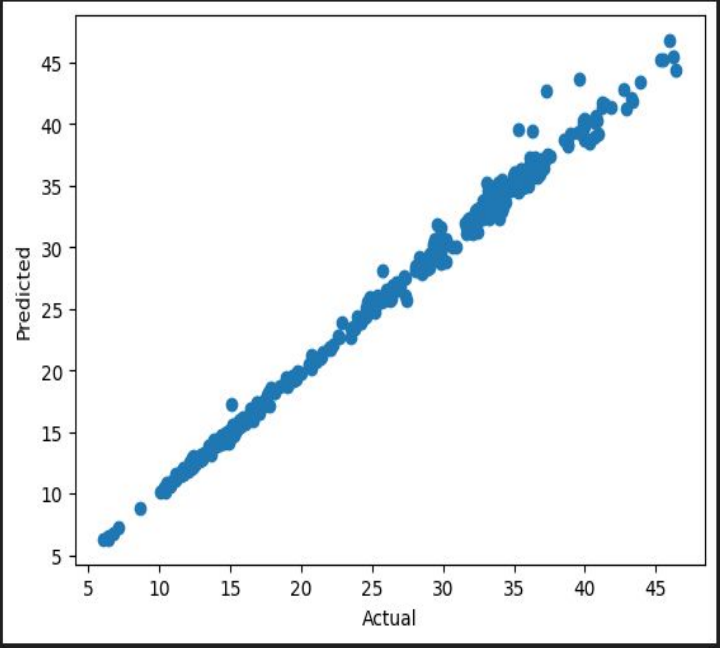
Heating Load (HL): The amount of heat energy required to maintain a comfortable indoor temperature during colder periods.

Cooling Load (CL): The amount of cooling energy required to maintain a comfortable indoor temperature during warmer periods.

Result plots

```
plt.scatter(y_test,y_pred_xgb)
plt.xlabel('Actual')
plt.ylabel('Predicted')
```

```
Text(0, 0.5, 'Predicted')
```




```
import pandas as pd

# Assuming 'Heating Load (kWh)' and 'Cooling Load (kWh)' are your target columns
target_columns = ['Heating Load (kWh)', 'Cooling Load (kWh)']

# Assuming y_test and y_pred are DataFrames with the same structure
pred_df = pd.DataFrame({
    'Actual Value - Heating': y_test[target_columns[0]].values,
    'Predicted Value - Heating': y_pred_xgb[:, 0],  # Assuming the first column corresponds to the first target variable
    'Difference - Heating': y_test[target_columns[0]].values - y_pred_xgb[:, 0],
    'Actual Value - Cooling': y_test[target_columns[1]].values,
    'Predicted Value - Cooling': y_pred_xgb[:, 1],  # Assuming the second column corresponds to the second target variable
    'Difference - Cooling': y_test[target_columns[1]].values - y_pred_xgb[:, 1],
})
pred_df
```

	Actual Value - Heating	Predicted Value - Heating	Difference - Heating	Actual Value - Cooling	Predicted Value - Cooling	Difference - Cooling
0	16.47	16.068878	0.401122	16.90	16.831444	0.068556
1	13.17	13.215577	-0.045577	16.39	16.268087	0.121913
2	32.82	32.337696	0.482304	32.78	32.434685	0.345315
3	41.32	41.572372	-0.252372	46.23	45.482059	0.747941
4	16.69	16.931974	-0.241974	19.76	19.869856	-0.109856
...	...	...	...	...	...	...

2.3 Proposed Solution:

Utilize a machine learning framework to analyze the impact of key independent factors, such as relative compactness, surface area, and glazing area distribution, on heating and cooling loads in residential buildings. Employ advanced regression techniques, including Random Forest, Gradient Boosting, and XGBoost, to develop predictive models. Conduct feature importance analysis to identify the most influential factors and fine-tune model hyperparameters for optimal performance. Deploy the trained models as RESTful APIs on scalable cloud platforms for seamless integration into energy management systems and policy support tools.

2.4 Technical Requirements:

The solution proposed here can be implemented as a cloud-based solution or as an application hosted on an internal server, or even on a local machine. To access this application, the following minimum requirements are necessary:

- A good internet connection.

- A web browser.

For training the model, the following system requirements are preferred:

- 4 GB RAM or more

• An operating system such as Windows, Linux or Mac.

• Visual Studio Code or Jupyter notebook.

2.5 Data Requirements:

The data requirements for this project will depend on the specific problem statement. A CSV file will be used as the input file, and the feature/field names and sequence should be followed as decided. It's important to have a clear understanding of the problem statement and the data that is required to solve it, to design a suitable data pipeline, and to train the model effectively.

2.6 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit Learn are used to build the whole model.

- Pandas is an open-source Python package that is widely used for data analysis and machine-learning tasks.
- NumPy is the most commonly used package for scientific computing in Python.
- Plotly is an open-source data visualization library used to create interactive and quality charts/graphs.
- Scikit-learn is used for machine learning.

- Flask is used to build API.
- VS Code is used as an IDE (Integrated Development Environment)
- GitHub is used as a version control system.
- Front-end development is done using HTML and CSS.
- AWS EC2 instance is used for the deployment of the model.

2.7 Constraints

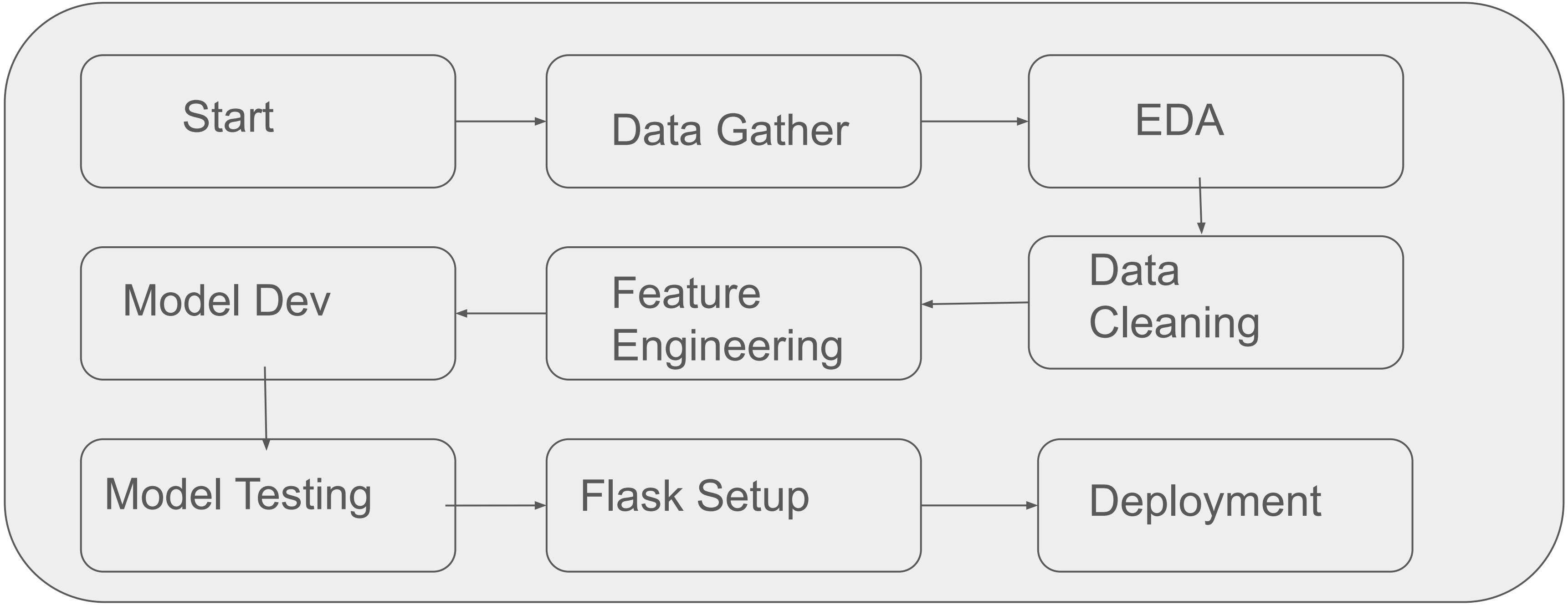
Data quality and availability pose constraints, impacting model accuracy and generalization. Computational resources and time are also limiting factors for training and deployment.

2.8 Assumptions

The primary goal of the project is to construct a machine learning model for predicting heating and cooling loads in residential buildings. It assumes the availability of a comprehensive dataset containing relevant features such as building characteristics and environmental factors.

3 Design Details

3.1 Process Flow



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developer can choose logging method. You can choose database logging / File logging as well.
- System should not hang even after using so many loggings

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An Error will be defined as anything that falls outside the normal and intended usage.

4 Performance.

4.1 Reusability

The entire solution will be done in a modular fashion and will be API oriented. So, in the case of scaling the application, the components are completely reusable.

4.2 Application Compatibility

The interaction with the application is done through the designed user interface, which the end user can access through any web browser.

4.3 Deployment

5 Conclusion

This project illuminates the intricacies of predicting heating and cooling loads in residential buildings, considering a myriad of influential factors. It discerns nuanced variations in load estimations across diverse building attributes and environmental parameters. Remarkably, through meticulous evaluation, Gradient Boosting emerges as the optimal model for precise predictions. These insights empower stakeholders to make informed decisions, facilitating efficient energy management strategies tailored to specific building requirements and environmental conditions.

Results

```python
pd.DataFrame(list(zip(model_list, r2_list)), columns=['Model Name', 'R2_Score']).sort_values(by=["R2_Score"],ascending=False)
```

Python

	Model Name	R2_Score
6	XGBRegressor	0.994589
5	Random Forest Regressor	0.983311
4	Decision Tree	0.977667
7	AdaBoost Regressor	0.948286
3	K-Neighbors Regressor	0.929521
0	Linear Regression	0.899394
2	Ridge	0.899159
1	Lasso	0.865189