Architecture Design

EcoComfort: Intelligent Energy Efficiency for Residential Buildings

Document Control

Version	Date	Author	Comments
1	15/07/2023	Shailendra Sahu	

Index

Content	Page No
Abstract	4
1. Introduction	4
1.1 What is Architecture Design?	4
1.2 Scope	4
1.3 Constraints	4
2. Technical Specification	5
2.1 Dataset	5
2.2 Logging	6
2.3 Deployment	6
3. Technology Stack	7
4. Proposed Solution	7
5. Architecture	8
5.1 Architecture Description	9
6. User Input/Output Workflow	10

Abstract

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization's target. In this project, we will estimate the heating and cooling load required for the building on the basis of the various informations provided to us. Taking various aspects of a dataset collected from client, and the methodology followed for building a predictive model.

1. Introduction

1.1 What is Architecture Design?

The goal of Architecture Design (AD) is to give the internal design of the actual program code for the ` EcoComfort: Intelligent Energy Efficiency for Residential Buildings`. AD describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Architecture Design (AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

1.3 Constraints

We only predict the heating load and cooling load based on the information provided by client.

2. Technical Specification

2.1 Dataset

The dataset containing verified data, having complete details of the different measurements of the building like 'relative compactness', 'surface area', 'wall area', 'roof area', 'overall height', 'orientation', glazing area', 'glazing area distribution', 'heating load (HL)', 'cooling load (CL)'.

The objective is to find a way to estimate the values in the 'heating load (HL)' and 'cooling load (CL)'. column using the values in the other columns of the dataset.

The dataset looks like as follow:

```
In [3]: df
```

Out[3]:

	relative compactness	surface area	wall area	roof area	overall height	orientation	glazing area	glazing area distribution	heating load (HL)	cooling load (CL)
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	20.84	28.28
...	...	...	...	...	...	...	...	...	...	...
763	0.64	784.0	343.0	220.50	3.5	5	0.4	5	17.88	21.40
764	0.62	808.5	367.5	220.50	3.5	2	0.4	5	16.54	16.88
765	0.62	808.5	367.5	220.50	3.5	3	0.4	5	16.44	17.11
766	0.62	808.5	367.5	220.50	3.5	4	0.4	5	16.48	16.61
767	0.62	808.5	367.5	220.50	3.5	5	0.4	5	16.64	16.03

768 rows × 10 columns

The data set consists of various data types from integer to floating as shown in Fig.

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   relative compactness       768 non-null    float64
 1   surface area               768 non-null    float64
 2   wall area                  768 non-null    float64
 3   roof area                  768 non-null    float64
 4   overall height             768 non-null    float64
 5   orientation                768 non-null    int64
 6   glazing area               768 non-null    float64
 7   glazing area distribution  768 non-null    int64
 8   heating load (HL)          768 non-null    float64
 9   cooling load (CL)          768 non-null    float64
dtypes: float64(8), int64(2)
memory usage: 60.1 KB
```

Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes.

```
df.describe()
```

	relative compactness	surface area	wall area	roof area	overall height	orientation	glazing area	glazing area distribution	heating load (HL)	cooling load (CL)
count	768.000000	768.000000	768.000000	768.000000	768.00000	768.000000	768.000000	768.00000	768.000000	768.000000
mean	0.764167	671.708333	318.500000	176.604167	5.25000	3.500000	0.234375	2.81250	22.307201	24.587760
std	0.105777	88.086116	43.626481	45.165950	1.75114	1.118763	0.133221	1.55096	10.090196	9.513306
min	0.620000	514.500000	245.000000	110.250000	3.50000	2.000000	0.000000	0.00000	6.010000	10.900000
25%	0.682500	606.375000	294.000000	140.875000	3.50000	2.750000	0.100000	1.75000	12.992500	15.620000
50%	0.750000	673.750000	318.500000	183.750000	5.25000	3.500000	0.250000	3.00000	18.950000	22.080000
75%	0.830000	741.125000	343.000000	220.500000	7.00000	4.250000	0.400000	4.00000	31.667500	33.132500
max	0.980000	808.500000	416.500000	220.500000	7.00000	5.000000	0.400000	5.00000	43.100000	48.030000

Preprocessing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns and model values for categorical columns. Maximum and minimum values in numerical columns, along with their percentile values for median, play an important factor in deciding which value to be chosen at priority for further exploration tasks and analysis. Data types of different columns are used further in label processing.

2.2 Logging

We should be able to log every activity done by the user

- The system identifies at which step logging require.
- The system should be able to log each and every system flow.
- The system should be not be hung even after using so much logging.
  Logging is done so that we can easily debug issues, so logging is mandatory to do.

2.3 Deployment

For the hosting of the project, we will use GCP (Google Cloud Platform).

3. Technology Stack

Front End	HTML/CSS
Backend	Python/ Flask
Deployment	GCP

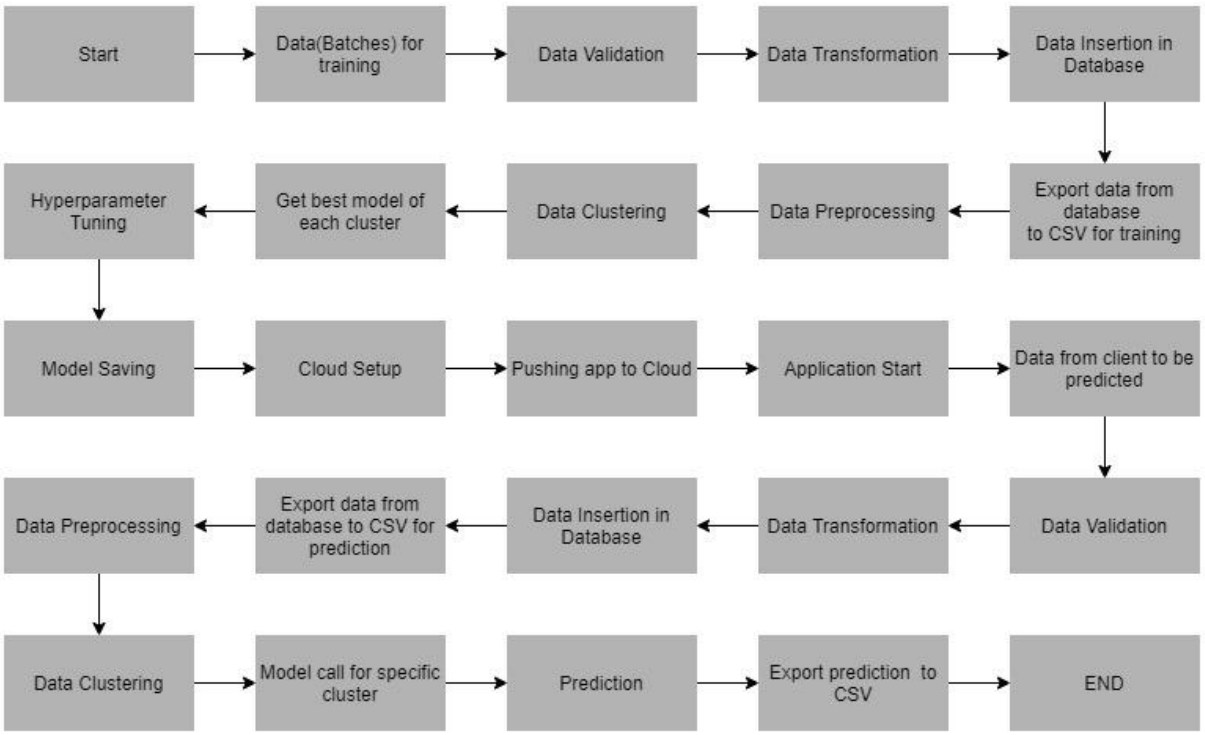
4. Proposed Solution

To enhance energy efficiency in residential buildings and accurately predict the heating load and cooling load, a machine learning regression model will be developed. The model will utilize comprehensive data on various measurements within the building to accurately estimate the specific heating and cooling requirements.

This approach will enable the optimization of HVAC systems to match the actual requirements, resulting in reduced energy waste and improved comfort levels for occupants.

We will perform EDA to find the important relation between different attributes and willuse a machine-learning algorithm to estimate the heating and cooling load. The client will give the input with required features and will get results through the web application. The system will get features and it will be passed into the backend where the features will be validated and preprocessed and then it will be passed to a hyperparameter tuned machine learning model to predict the final outcome.

5. Architecture



5.1 Architecture Description

5.1.1 Data Gathering

Data source: Data is provided by client.
Dataset is stored in .csv format.

5.1.2 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because the attributes which contain these informations are of no use and it will not play role in contributing to the estimating of the the heating and cooling load.

5.1.3 Exploratory Data Analysis

Visualized the relationship between the dependent and independent features. Also checked relationship between independent features to get more insights about the data.

5.1.4 Feature Engineering

After pre-processing, log transformations and standard scalar is performed to scale down all the numeric features. For this process, pipeline is created to log transform and to scale numerical features.

5.1.5 Model Building

After doing all kinds of pre-processing operations mention above and performing scaling and transformation, the data set is passed through a clustering algorithm (kmeans) to divide the dataset into different clusters, so that we can apply different algorithms in different clusters of the dataset. The dataset of each clusters are passed through a pipeline to both the models, Linear Regression and Random Forest. Then performance score is calculted for both the model and the model with the best score is selected for prediction for that perticular cluster.

5.1.6 Model Saving

The best model for each cluster is saved. Model is saved using pickle library in pickle` format.

5.1.7 Flask Setup for Web Application

After saving the model, the API building process started using Flask. Web application

creation was created in Flask for testing purpose. Whenever the user will enter the data in the path and give this path as an input in the web application then that data will be extracted by the model to predict the heating and cooling load, this isperformed in this stage.

5.1.8 GitHub

https://github.com/shailendra24sahu/EcoComfort-Intelligent-Energy-Efficiency-for-Residential-Buildings
The whole project directory is pushed into the GitHub repository.

5.1.9 Deployment

The project was deployed into the GCP platform.

6. User Input / Output Workflow.