Low Level Design (LLD)

EcoComfort: Intelligent Energy Efficiency for Residential Buildings

Written by	Shailendra Sahu
Revision Number	1.0
Last date of revision	14/07/2023

Document Control

Change Record:

Version	Date	Author	Comments
0.1	13/07/2023	Shailendra Sahu	Introduction & Architecture defined
0.2	14/07/2023	Shailendra Sahu	Architecture & Architecture Description appended and updated

Reviews:

Version	Date	Reviewer	Comments

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

1. Contents

2. Introduction

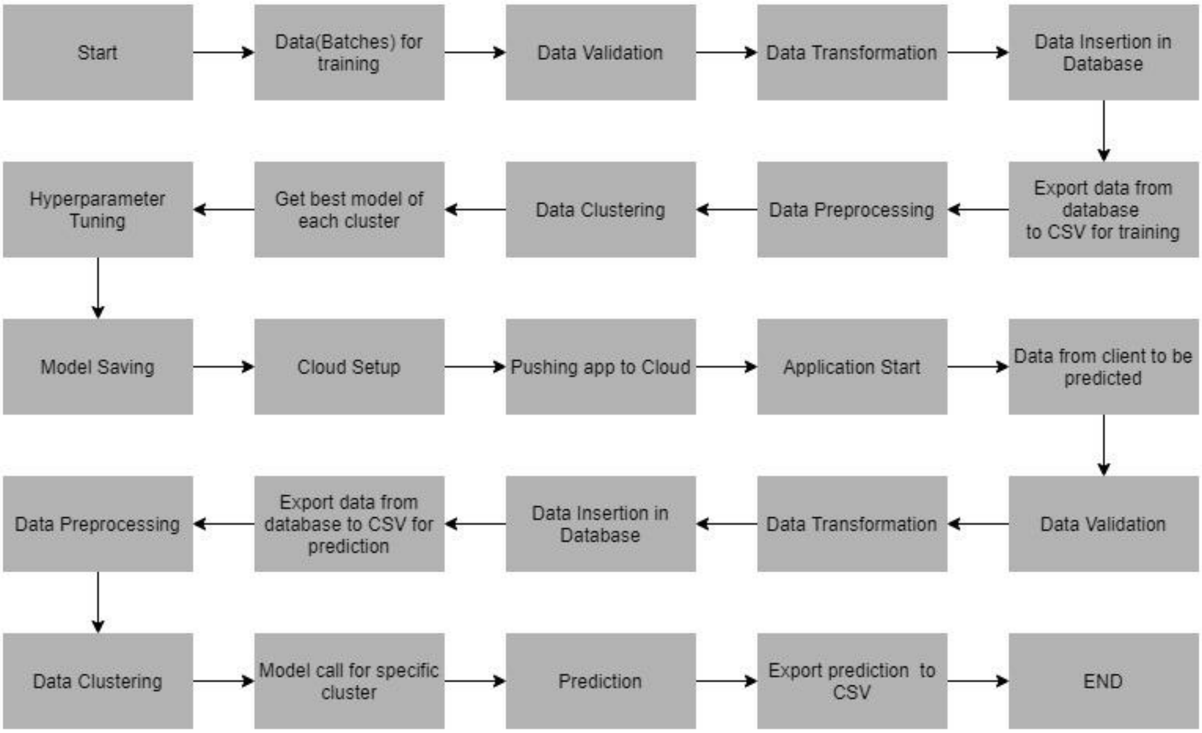
2.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for EcoComfort: Intelligent Energy Efficiency for Residential Buildings model. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

2.2. Scope

Low-level design (LLD) 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

3. Architecture


Start
Data(Batches) for training
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for training
Data Preprocessing
Data Clustering
Get best model of each cluster
Hyperparameter Tuning
Model Saving
Cloud Setup
Pushing app to Cloud
Application Start
Data from client to be predicted
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for prediction
Data Preprocessing
Data Clustering
Model call for specific cluster
Prediction
Export prediction to CSV
END

4. Architecture Description

4.1. Data Batches

Data will be provided to us by client in a perticular format as discussed in data sharing agreement. And the data will be stored in a specific directory in batch files.

4.2. Data Validation

In this step, we perform different sets of validation on the given set of training files as File Name Validation, Number of columns, Name of columns, Datatype of columns and Null values in columns.

4.3. Data Transformation

In the Transformation Process, we convert all the columns with string datatype such that each value for that column is enclosed in quotes. This is done to avoid the error while inserting string values in table as varchar.

4.4. Data Insertion into Database

a. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
b. Table creation in the database.
c. Insertion of the data from files into the table.

4.5. Export Data from Database

The data from the stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

4.6. Data Pre-processing

In Data Pre-processing steps we check for null values in the columns. If present, impute the null values using the KNN imputer, log transformation of the features, Scaling the training and test data separately, applying principal component analysis.

4.7. Data Clustering

K-Means algorithm will be used to create clusters in the pre-processed data. The optimum number of clusters is selected by plotting the elbow plot and for the dynamic selection of the number of clusters, we are using "KneeLocator" class. The idea behind clustering is to implement different algorithms to train data in different clusters. The K-means model is trained over preprocessed data and the model is saved for further use in prediction.

4.8. Get Best Model For Each Cluster and Model Tuning

After clusters are created, we find the best model for each cluster. We are using two algorithms, "Random forest Regressor" and “Linear Regression”. For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the Rsquared scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

4.9. Model Saving

Models for each clusters are saved in a directory for using in prediction.

4.10. Cloud Setup

Setup in the project folder is done for deploying the model in the cloud platform.

4.11. Pushing App to Cloud

After setting up the required setups, the app is pushed in the cloud platform to make it usable by client. Then we provide our client the application link to use it for prediction.

4.12. Application Start

Application is started by deploying the model in cloud platform.

4.13. Data from User

Client will send the data in multiple set of files in batches at a given location in a perticular format as discussed in data sharing agreement.

4.14. Data Validation

In this step, we perform different sets of validation on the given set of files as File Name Validation, Number of columns, Name of columns, Datatype of columns and Null values in columns.

4.15. Data Transformation

In the Transformation Process, we convert all the columns with string datatype such that each value for that column is enclosed in quotes. This is done to avoid the error while inserting string values in table as varchar.

4.16. User Data Inserting into Database

a. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
b. Table creation in the database.

c. Insertion of the data from files into the table.

4.17. Export Data from Database

The data from the stored database is exported as a CSV file to be used for Data Pre-processing and prediction.

4.18. Data Preprocessing

In Data Pre-processing steps we check for null values in the columns. If present, impute the null values using the KNN imputer, log transformation of the features, and Scaling the data.

4.19. Data Clustering

The Kmeans model created during training will be loaded, and clusters for the user data will be predicted.

4.20. Model Call for Specific Cluster

Based on the cluster number, the respective model will be loaded and will be used to predict the data for that cluster.

4.21. Prediction

The loaded model will do the prediction over the prediction dataset.

4.22. Export Predictions to CSV

All the predictions are stored in a CSV file and is exported to a directory.

4.23. Deployment

We will be deploying the model to GCP (Google Cloud Platform).

1. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to see input fields after loading the application	1. Application is accessible 2. Application is deployed	User should be able to see input fields on loading the application
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. Application is deployed	User should get Submit button to submit the inputs
Verify whether user gets the results on clicking submit	1. Application is accessible 2. Application is deployed	User should be able to get the results on clicking submit

7 LOW LEVEL DESIGN (LLD)