

# Low Level Design

Energy Efficiency

Written By	Mayur Rathod
Document Version	0.1
Last Revised Date	14 - August -2021

## Document Control

### Reviews:

Version	Date	Reviewer	Comments

### Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

## Contents

<b>1. Introduction .....</b>	<b>1</b>
<b>1.1. What is Low-Level design document?.....</b>	<b>1</b>
<b>1.2. Scope.....</b>	<b>1</b>
<b>2. Architecture .....</b>	<b>2</b>
<b>3. Architecture Description .....</b>	<b>3</b>
<b>3.1. Data Description .....</b>	<b>3</b>
<b>3.2. Web Scrapping .....</b>	<b>3</b>
<b>3.3. Data Transformation .....</b>	<b>3</b>
<b>3.4. Data Insertion into Database .....</b>	<b>3</b>
<b>3.5. Export Data from Database.....</b>	<b>3</b>
<b>3.6. Data Pre-processing .....</b>	<b>3</b>
<b>3.7. Data Clustering.....</b>	<b>3</b>
<b>3.10. Model Building .....</b>	<b>4</b>
<b>3.11. Data from User .....</b>	<b>4</b>
<b>3.12. Data Validation.....</b>	<b>4</b>
<b>3.13. User Data Inserting into Database .....</b>	<b>4</b>
<b>3.14. Data Clustering .....</b>	<b>4</b>
<b>3.15. Model Call for Specific Cluster .....</b>	<b>4</b>
<b>3.16. Recipe Recommendation &amp; Saving Output in Database .....</b>	<b>4</b>
<b>3.17. Deployment.....</b>	<b>4</b>
<b>4. Unit Test Cases .....</b>	<b>5</b>

## 1. Introduction

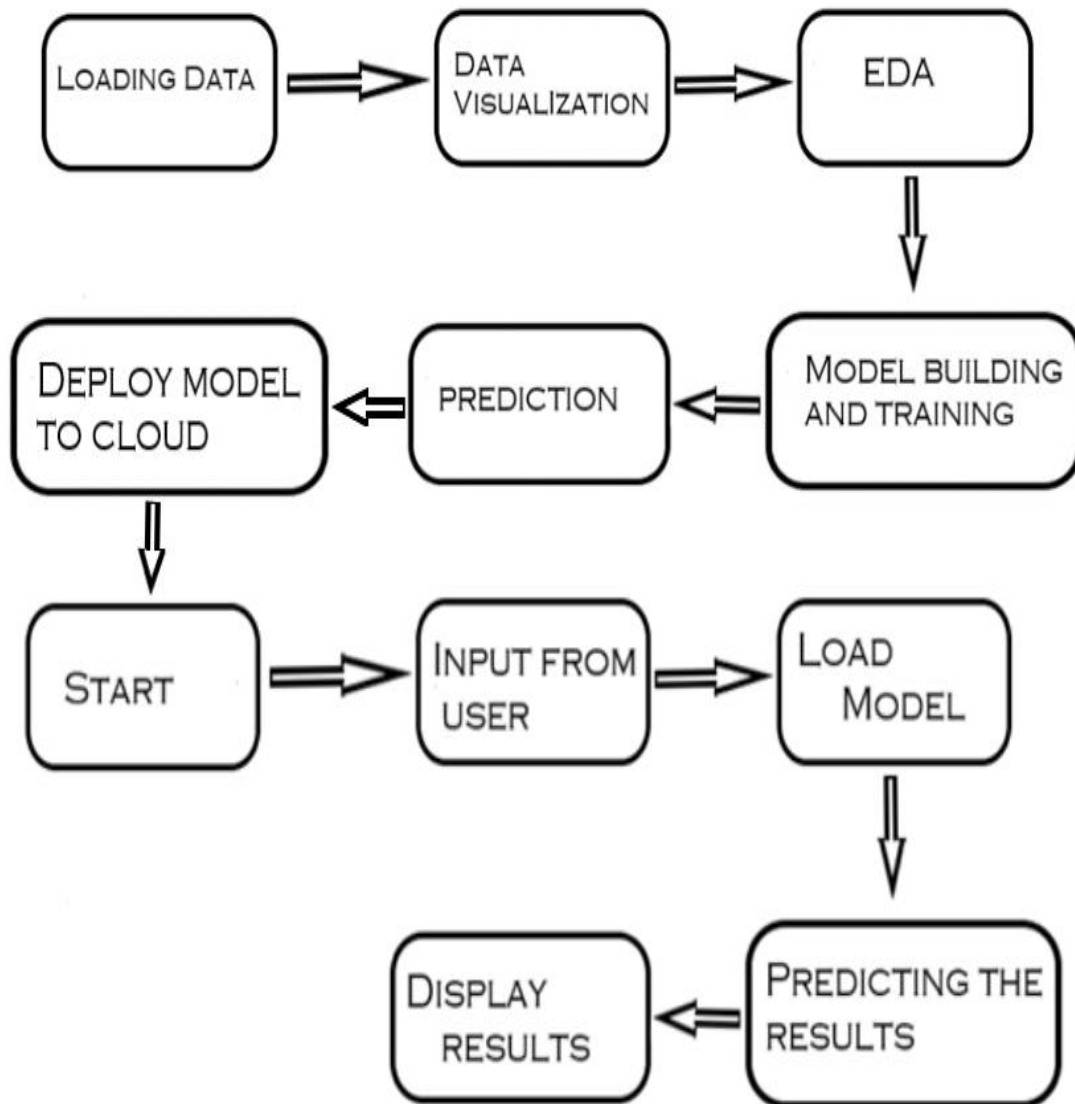
### 1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. 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.

### 1.2. 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

## 2. Architecture



## 3. Architecture Description

### 3.1. Data Description

The dataset is of Energy efficiency for building. Our project performs energy analysis using 12 different building shapes simulated in Ecotect. The buildings differ with respect to the glazing area, the glazing area distribution, and the orientation, amongst other parameters. The dataset comprises 768 samples and 8 features. Our aim is to predict two real valued responses.

The dataset contains eight attributes and two responses.

X1=Relative Compactness

X2=Surface Area

X3=Wall Area

X4=Roof Area

X5=Overall Height

X6=Orientation

X7=Glazing Area

X8=Glazing Area Distribution

Y1=Heating Load

Y2=Cooling Load

### 3.2. Data Loading

Our Dataset is in .xlsx format so we can import it using “read\_excel” with help of pandas library.

### 3.3. Data Visualization

- We use matplotlib and seaborn for data visualization.
- We may need to check whether our attributes contain outliers or not. Because outliers result in a poor fitting and lower predictive modeling performance.
- So, with the help of boxplot we see the outliers. Outliers i.e the values below 25% and above 75% in plot.

### 3.4. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, description of data, values counts and types of data, selection of features and targets etc.

### 3.5. Data Preparing

We split data into training and testing set. Separating data into training and testing sets is an important part of evaluating data mining models. Typically, when we separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing. Analysis Services randomly samples the data to help ensure that the testing and training sets are similar. By using similar data for training and testing, we can minimize the effects of data discrepancies and better understand the characteristics of the model.

### 3.6 Model Building

After features are created, we will find the best model for our data. Different algorithms will be passed with the best parameters. We need to choose variables that we think we'll be good predictors for the dependent variable — that can be done by checking the correlation(s) between variables, by plotting the data and searching visually for relationship, by conducting preliminary research on what variables are good predictors of y etc. We will calculate the RMSE and R2 scores for models and select the model with the best score.

### 3.7 Data from User

Here we will collect data from user such Relative Compactness, Surface Area, Wall Area, Roof Area, Overall Height, Orientation, Glazing Area, Glazing Area Distribution;

### 3.8 Data Validation

Here Data Validation will be done, given by the user

### 3.9 Model Call for Specific Cluster

The model having highest accuracy for a given data is called and use for prediction.

### 3.10 Results

After calling model result will be displayed with the values of heating and cooling load on the basis of data provided by user.

### 3.11 Deployment

We will be deploying the model to Heroku.

## 4. 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 on display	1. Application is accessible	User should be able to see input fields on interface.
Verify whether user is able to edit all input fields	1. Application is editable	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. User get submit button	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. User get the results	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application shows results.	The recommended results should be in accordance to the selections user made