Energy Efficiency

Project Members:
1. Subhransu Sekhar Mallick
2. Itishree Samal

Objective

To create an AI solution implemented through a web application for following use cases:

- To determine the heating and cooling load of a building
- To retrain a new model based on new data to have up to date predictions

Proposed Solution

- The solution here is a web application through which a machine learning model can be trained as well as accessed for prediction of heating load and cooling load.

- In order to train the model the data will first be subjected to classical and non parametric tests to obtain features highly correlated to the output labels.

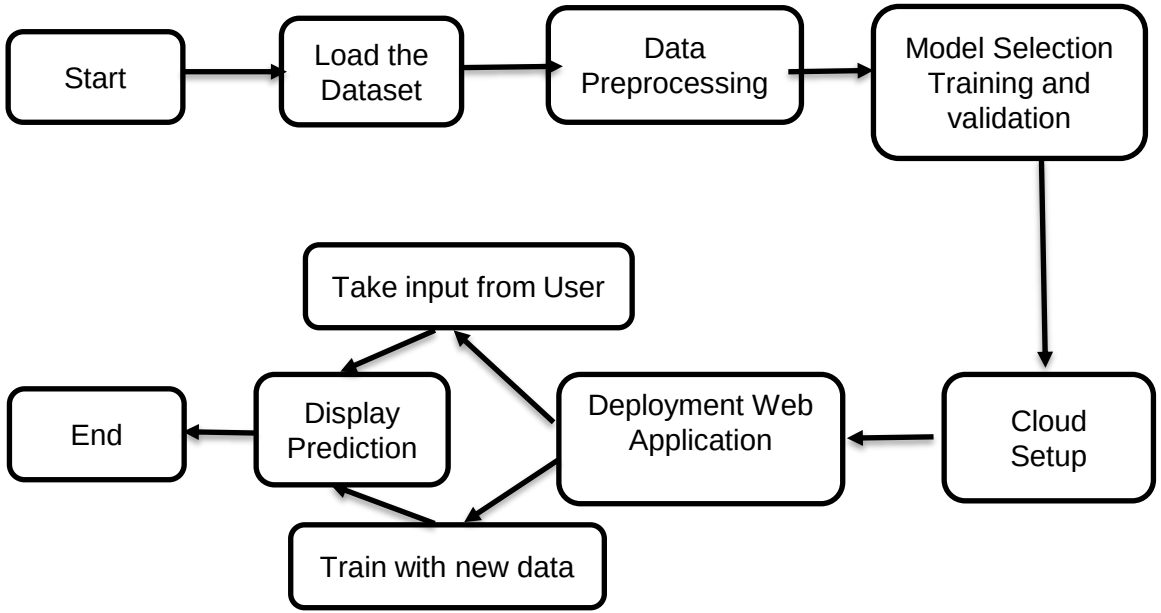
Data Sharing Agreement

1. The data must be provided in 10 columns for training. The description of the columns are stated below:
   a. “X1” (input) - Relative Compactness
   b. “X2” (input) - Surface Area
   c. “X3” (input) - Wall Area
   d. “X4”(input) - Roof Area
   e. “X5” (input) - Overall Height
   f. “X6” (input) - Orientation
   g. “X7” (input) - Glazing Area
   h. “X8” (input) - Glazing Area Distribution
   i. “Y1” (output) - Heating Load
   j. “Y2” (output)- Cooling Load

   The terms in double quotes are the column names of data that should be provided.

4. For prediction the data should have only the inputs.

Architecture



Model Training:

- **Data Export from Db** :
  - The accumulated data from db is exported in csv format for model training
- **Data Preprocessing:**
  - Performing EDA to get insight of data like identifying distribution , outliers ,trend among data etc.
  - Check for null values in the columns. If present impute the null values.
  - Perform Standard Scalar to scale down the values.

Algorithm Selection

After data preprocessing, the model was initially trained using baseline model Linear Regression on 67% of the training data, and its testing score was assessed with the remaining 33% of testing data.

- During the training process, cross-validation techniques were employed to enhance the model's robustness.
- However, in pursuit of further improving regressor performance, the Random Forest Regressor was subsequently employed.
- This transition led to a significant increase in the model's overall performance.
- Consequently, the model's final evaluation was conducted using the Random Forest Regressor to ensure its effectiveness in predicting the desired outcomes.

Cloud Setup

For our cloud setup and deployment, we utilize GitHub and Streamlit to seamlessly deploy our web application to the web.

- This deployment strategy not only ensures accessibility but also establishes a continuous integration and continuous deployment (CI/CD) pipeline.
- As we push code changes to GitHub, Streamlit facilitates the automatic deployment of our application, keeping it up-to-date with any modifications made to the codebase.
- This streamlined process not only enhances the efficiency of our development workflow but also ensures that our users always have access to the latest version of our web application, promoting a seamless and responsive user experience.

Prediction

- The input from the user is taken using the Streamlit user interface.
- The saved model loaded in Github shares code to the stream lit where data pre-processing techniques is performed on it.
- The Random Forest Regressor algorithm is chosen for its ability to handle complex relationships and provide accurate predictions for continuous variables like heating and cooling loads.
- Once the user provides input data, the trained model takes this input and applies the learned patterns to make predictions for both heating and cooling loads.
- The Random Forest Regressor employs an ensemble of decision trees to make predictions. It aggregates the results from multiple decision trees to reduce overfitting and improve prediction accuracy

Q&A

1. **What's the source of data?**
   Users can provide data through text input fields. This data is typically collected and processed within the app. The app uses the trained model to make predictions

2. **What was the type of data?**
   The data comprised only of non negative nnumerical type with all of them being discrete in nature.

3. **What's the complete flow you followed in this Project?**
   Refer slide 4th for better Understanding.

4. **What techniques were you using for data pre-processing?**
   - Cleaning data and imputing if null values are present.
   - Visualizing relation of independent variables with each other and output variables
   - Removing unwanted attributes
   - Scaling the data

5. How training was done or what models were used?

- Before divding the data in training and validation set we performed cross validation with size of 6.

.

- Initially the model is trained with baseline model, linear regression to check R square value and error

- Random Forest Regression is employed to compare the error and R square value obtained by the baseline model.

- Finally the outcome prediction is done by the Random Forest Regressor and saved as pickle for upload to the streamlit.

6. How logs are managed?

- Logs are managed in the code by incorporating error messages for each input field, checking for missing values, non numeric data types and ensuring that negative values are not accepted during user input validation, enhancing the reliability and user-friendliness of the application.

7. **How was Prediction done?**
   - User input is taken through the Web Application.
   - Then on the trained model data is feeded for processing.
   - In the end we get the accumulated data of predictions which are presented in the web app.

8. **How the Web Application Deployed?**

   The Web Application was deployed to Streamlit and displayed using its user interface. The code file uploaded in Github is shared to Streamlit using standard deployment procedure.

   The web Application is deployed at:
   https://energy-efficient-building-load-calculation-app-3ukurqwqpujw46k.streamlit.app/

9. **What is the latency of the model response?**

   The model takes a maximum of 1 seconds to display the results from the point of time when the user input is submitted.