# Report on Cooling Load Prediction AI Model

**1. Objective**

The primary goal of the project is to predict the future cooling load for two buildings, Building A and Building B, based on historical operational data.

**2. Exploratory Data Analysis (EDA)**

The project began with an exploratory data analysis phase. It was noted that the dataset for Building A was significantly larger (10,000+ rows) than for Building B (1,000+ rows), leading to a strategy of focusing on Building A first.

An interactive dashboard was built using Plotly and Dash to visualize the data. This tool was crucial for:

* Plotting time-series data against all parameters
* Comparing the performance of different chillers within a building
* Analyzing correlations between different parameters through heatmaps and scatter plots.
* Identifying key patterns, such as the daily operating hours of the chillers.

**3. Model Selection**

The task was identified as a **supervised learning, time-series regression problem**, as the goal was to predict a continuous value (cooling load).

* **Initial Model:** A `RandomForestRegressor` was chosen for the first attempt due to its ease of use and strong performance on tabular data out-of-the-box.
* **Final Model:** The project later transitioned to a **Gradient Boosting Machine (GBM)**, specifically XGBoost, for its superior performance and flexibility in fine-tuning.

The reasoning for selecting these tree-based models was their ability to capture complex non-linear relationships, handle a mix of feature types, and their training efficiency.

**4. Initial Workflow & Pipeline**

The development philosophy was to **“Build, review, iterate, repeat,”** focusing on rapidly creating a baseline model and improving it.

The data workflow structure can be found in **Annex A**.

* **Data Preparation:** The target variable, Total\_Cooling\_Load, was calculated for the training set by applying the given formula to the raw chiller sensor data.
* **Feature Engineering:** The only feature available for the prediction data was prediction\_time. Therefore, the initial feature set for both training and testing was derived from this timestamp: hour\_of\_day, day\_of\_week, month, and is\_weekend.

**5. Iterative Model Improvement**

The core of the project involved a highly iterative fine-tuning process, meticulously documented across 24 attempts in a changelog **(Annex B)**. This allowed for systematic tracking of how changes in features and model parameters affected the Normalized Root Mean Square Error (NRMSE).

**Key Iteration Highlights:**

* **Attempt 4:** Switched from Random Forest to XGBoost, resulting in a significant NRMSE improvement.
* **Attempt 9:** Introduced is\_holiday as a feature, which proved to be one of the most impactful changes, dropping the NRMSE score substantially.
* **Attempts 10-15:** Focused on integrating weather data. However, this consistently degraded model performance, indicating an overfitting problem. Techniques like early stopping and parameter tuning were introduced to combat this.
* **Attempt 16:** Implemented a solution for Building B, which had very little data.
* **Attempts 18-22:** Used feature importance plots from XGBoost to debug the model, leading to the removal of all weather features and a focus on time features.

**6. Key Feature Engineering Insights**

The project demonstrated that thoughtful feature engineering based on domain context was more effective than simply adding more raw data.

* **High-Impact Features:** The best results came from features that captured human behavior and operational schedules:
  + is\_business\_hour: A binary feature created after observing the chiller’s daily start and stop times (7:30 am - 6:00 pm) on the EDA dashboard.
  + is\_holiday & is\_weekend: Binary features that captured non-working days, which strongly correlate with lower cooling demand.
* **Weather Data Challenges:** The model struggled to generalize from the weather data. The high variance and potential noise from this data led to overfitting, where the model performed well on training data but poorly on unseen data. The final model removed all weather features.

**7. Final Model & Architecture**

The final, most successful approach used a refined XGBoost model and a transfer learning strategy.

* **Model:** XGBoost Regressor.
* **Features:** hour\_of\_day, is\_weekend, season, is\_business\_hour, and is\_holiday.
* **Training Strategy:**
  + **Pre-training on Building A:** An XGBoost model was trained on the extensive dataset for Building A. Early stopping was used with a validation set to find the optimal number of training rounds and prevent overfitting.
  + **Transfer Learning for Building B:** For Building B, instead of training a new model from scratch on its small dataset, the pre-trained model from Building A was used as a starting point. This **fine-tuning** (using the xgb\_model parameter in XGBoost) allows the model to adapt its existing knowledge to the specifics of Building B, leading to much better performance than training on limited data alone.

Final workflow diagram can be found in **Annex C**.

**8. Conclusion**

The final model was trained exclusively on time features (i.e, hour\_of\_day, season, is\_business\_hour, is\_workday,is\_weekend, is\_holiday) after model iteration and analysis revealed that including raw weather data without corresponding building-specific parameters degraded predictive performance. This outcome underscores a critical insight: cooling load is not driven by external weather in isolation, but by the **interaction** between external conditions and the building's internal state and systems.

Other key datasets that could improve the model.

- **Live chiller operating state**

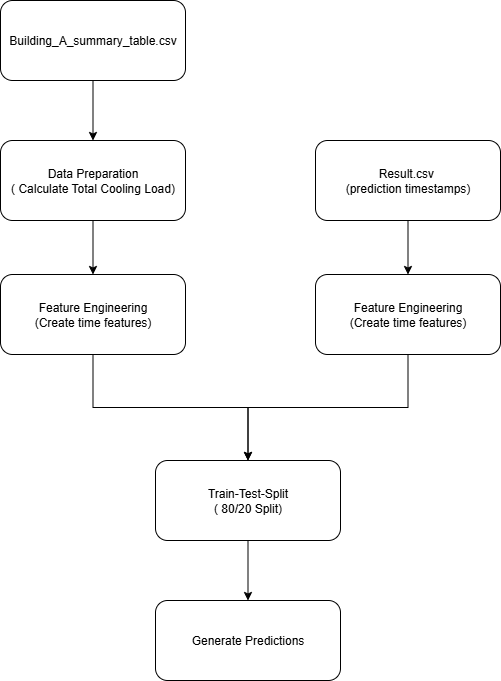
While we were given chiller power consumption, flow rates, and supply/return temperatures, we were not able to fully utilise these training data since it was not available in the test set, which could be the key to predicting more accurate cooling loads.

**- Indoor temperature/ setpoints, occupancy levels**

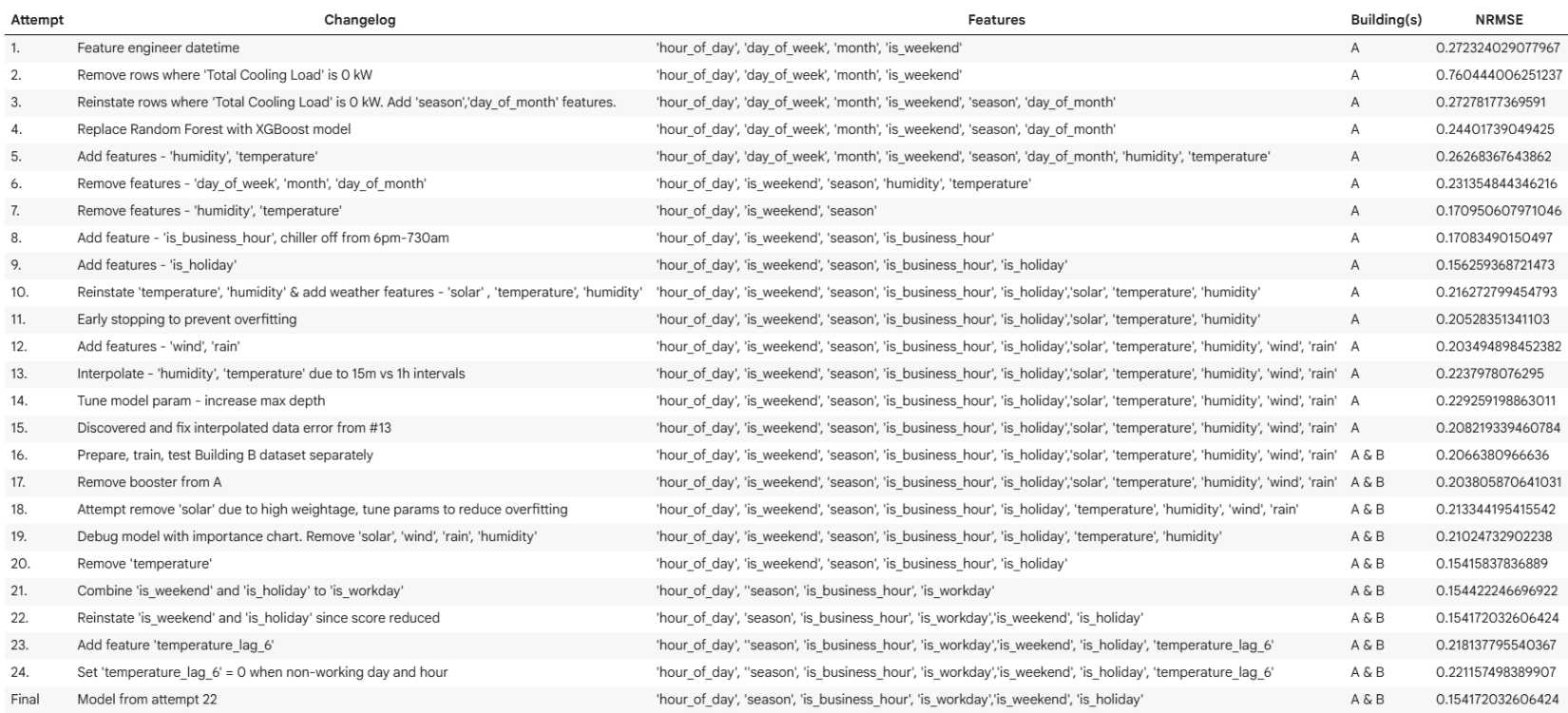
To create relationships between a building cooling load and environmental factors, we need to anchor these to meaningful features, i.e. heat gain through building envelope, or by building occupants/ users. The above attempts to introduce weather data without relationships reiterates this point.

In summary, while a model based on time patterns provides a baseline, accurate cooling load prediction is fundamentally a physics-informed problem that requires data reflecting the building's internal response to external conditions.

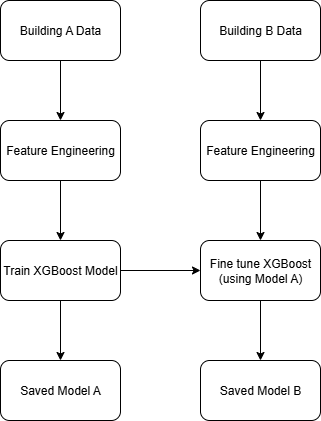
**Annex A : Initial workflow**



**Annex B: Model Iteration Changelog**



**Annex C: Final Workflow**

****