Exploring House Energy Consumption Prediction with Basic Statistical Models

---

Math 748: Theory and Applications of Statistical Machine Learning

Instructure By: Dr. Tao He

Final Project Report

By: Dholakiya Milan Pravinbhai
Id: #923655574

\

Table of Contents

Table of Figures

Table of Tables

Abstract

This project investigates the prediction of house energy consumption, focusing on heating and cooling loads, using architectural and design-related features from the UCI Energy Efficiency dataset [1].A dataset comprising 768 building configurations with ten features, including relative compactness, surface area, and glazing area, was used to train and evaluate five different regression models: Linear Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Regression (SVR). Data preprocessing involved outlier removal and feature scaling for distance-based models. Model training included 10-fold cross-validation for hyperparameter tuning, specifically for KNN and SVR. Performance was assessed using Mean Squared Error (MSE) and R-squared (R^2). Results demonstrated Random Forest's superior predictive accuracy, achieving an MSE of 0.375 and an R^2 of 0.996, significantly outperforming other models. KNN and SVR also demonstrated strong performance, making them suitable alternatives. Feature importance analysis highlighted the influence of Roof_Area, Surface_Area, and Relative_Compactness in predicting heating load. This research underscores the potential of machine learning for optimizing building energy efficiency and informs future research directions, including advanced feature engineering, incorporating additional data sources, exploring Gradient Boosting Machines, and enhancing model interpretability.

1. Introduction

Energy consumption is a critical concern for addressing global sustainability challenges and reducing operational costs in the construction and operation of residential and commercial buildings. With increasing global emphasis on sustainable practices, optimizing energy efficiency in buildings has become a priority (2). Predicting heating and cooling energy requirements is crucial for guiding energy-efficient architectural designs, reducing greenhouse gas emissions, and minimizing financial expenditures (4).

This study utilizes the UCI Energy Efficiency dataset, which includes 768 samples with eight independent variables such as Relative Compactness, Surface Area, and Glazing Area, along with two target variables: Heating_Load and Cooling_Load (1). These variables represent the energy loads for heating and cooling in buildings and are influenced by architectural features and structural designs. The dataset serves as a foundation for understanding how various characteristics impact energy consumption.

This project investigates the application of machine learning techniques to forecast heating loads based on building characteristics. We explore several models, comparing their performance to identify the most suitable approach for this task. This project has implications for architects, engineers, and policymakers seeking to improve building energy efficiency. To address this challenge, the study employs two predictive models:

1. Linear Regression: A statistical model that provides baseline predictions and interpretable coefficients for feature impact analysis.
2. Decision Tree: A non-linear model that captures complex interactions among features and offers interpretability through its hierarchical structure (3).
3. K-Nearest Neighbours (KNN): A non-parametric, instance-based learning algorithm.
4. Random Forest: An ensemble method combining multiple decision trees.
5. Support Vector Regression (SVR): A kernel-based model for flexible regression.

The methodology involves data preprocessing steps, such as renaming features for clarity, detecting and handling outliers, and splitting the dataset into training and testing subsets. These

steps ensure the reliability and accuracy of the models. The goal of this study is to identify the most significant predictors of energy consumption and develop accurate models to forecast heating and cooling loads in buildings.
By providing a comprehensive analysis and leveraging predictive modelling techniques, this report contributes to the broader goals of sustainable building design and operational efficiency.

2. Dataset Description

The UCI Energy Efficiency Dataset provides valuable information about building characteristics and their impact on energy consumption. It includes 768 samples, with the following **independent variables** (features) and **target variables** (outputs):
Independent Variables:

1. Relative Compactness: A measure of the compactness of the building.
2. Surface Area: Total surface area of the building (m^2).
3. Wall Area: Total wall area of the building (m^2).
4. Roof Area: Total roof area of the building (m^2).
5. Height: Overall height of the building (m).
6. Orientation: Orientation of the building (1–5).
7. Glazing Area: Percentage of the building covered by windows (0–0.4).
8. Glazing Area Distribution: Distribution of the glazing area (1–5).

Target Variables:

1. Heating_Load: Energy required for heating the building (kWh/m^2).
2. Cooling_Load: Energy required for cooling the building (kWh/m^2).

Summary Statistics:

Variable	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Relative Compactness	0.62	0.6825	0.75	0.7642	0.83	0.98
Surface Area (m^2)	514.5	606.4	673.8	671.7	741.1	808.5
Wall Area (m^2)	245	294	318.5	318.5	343	416.5
Roof Area (m^2)	110.2	140.9	183.8	176.6	220.5	220.5
Height (m)	3.5	3.5	5.25	5.25	7	7
Heating Load (kWh/m^2)	6.01	12.99	18.95	22.31	31.67	43.1
Cooling Load (kWh/m^2)	10.9	15.62	22.08	24.59	33.13	48.03

Table 1: The table provides summary statistics for the dataset used in project.

The histogram reveals a bimodal distribution, with peaks around 10-15 kWh/m^2 and 30-35 kWh/m^2, indicating two prominent groups of buildings with differing energy requirements for heating. This may reflect variations in structural features like wall or glazing areas. While The Cooling Load exhibits a broader distribution with a peak around 10-20 kWh/m^2, followed by a gradual decline. The spread indicates higher variability in cooling requirements, likely influenced by glazing area and orientation shows in Figure 1.

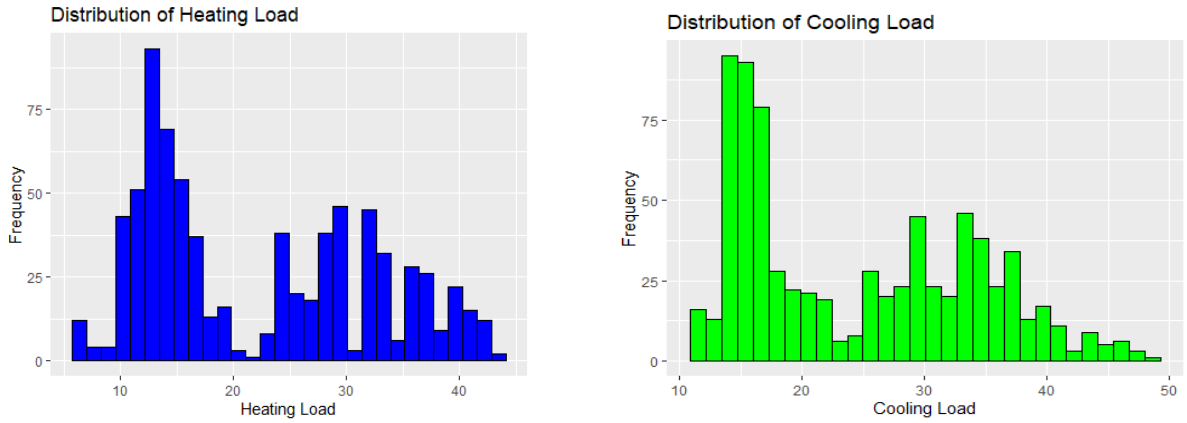

Figure 1: Distribution of Heating Load and cooling load

2.1 Data Cleaning

The original dataset had short column names (e.g., "X1," "X2," "Y1"). These names don't convey much information about the variables they represent. Renaming columns to more descriptive names (e.g., "Relative_Compactness," "Surface_Area," "Heating_Load") greatly improves the report's readability and makes it easier for readers to understand the meaning of the variables and the analyses performed.

Also, Outlier analysis and removal were performed using the Interquartile Range (IQR) method to ensure model robustness. Outliers are data points that are significantly different from the majority of the data. They can be caused by errors in data collection, unusual events, or simply natural variation. Outliers can have a disproportionate influence on some statistical models, especially linear regression, KNN and SVR, potentially leading to inaccurate or misleading results. Removing outliers can improve model robustness and generalization. The Interquartile Range (IQR) is a measure of statistical dispersion. It's calculated as the difference between the 75th percentile (Q3) and the 25th percentile (Q1) of the data. The IQR method identifies outliers as points that fall below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR. No missing values were present in the dataset.

2.2 Data Splitting

To ensure accurate model evaluation, the dataset was split into training and testing subsets using an 80:20 ratio. This is a standard approach that ensures sufficient data for training while preserving an adequate portion for testing model performance on unseen data. The training set, comprising 80% of the data, was used to build predictive models and learn patterns from the features. The testing set, with the remaining 20%, was reserved for evaluating the models' generalization ability. This approach mitigates overfitting, a situation where the model memorizes the training data but performs poorly on new, unseen data.

Random sampling was employed during the split to ensure that the training and testing subsets retained a balanced distribution of variables, particularly Heating Load and Cooling Load. This step ensures the testing set reflects the real-world data distribution, allowing for a fair and meaningful evaluation of the models. To confirm this, histograms (figure 2) were generated for Heating Load distributions in both subsets, which showed similar patterns, reinforcing the integrity of the split.

Figure 2: Heating Load distribution for training and testing data set.

This train-test split allows for robust evaluation of the predictive models. Linear Regression and Decision Tree models were trained on the training data to identify relationships between input features and energy loads. The testing data then provided an unbiased estimate of how well these models could predict energy consumption for unseen samples. The use of this dataset, with its well-distributed and relevant features, enhances the reliability of insights drawn from the analysis. Visualizations such as histograms further support this methodology by demonstrating the balanced distribution of data between training and testing subsets, making the models suitable for real-world application.

3. Modelling Approach & Evaluation

Several machine learning models were trained and evaluated to predict building heating load. These models were selected to represent different learning algorithms and their capabilities in capturing diverse relationships within the data:

- **Linear Regression:** A baseline model assuming a linear relationship between features and heating load. It serves as a benchmark for comparison with more complex models. This basic method helps understand if a complex model is needed at all, or if the simple model works well for the problem. Chosen as a baseline model to assess the assumption of a linear relationship between the features and heating load. If the data exhibits strong linearity, a simple linear model might suffice. It provides good interpretability.
- **Decision Tree:** A tree-based model that can capture non-linear relationships and interactions between features. Its interpretability allows for understanding the factors influencing predictions. The decision making process can be observed from the resulting decision tree. Its main parameters include the cp parameter, which prevents overfitting and also controls the depth of the tree. Selected for its ability to capture non-linear relationships and feature interactions. The resulting tree structure also offers interpretability, allowing us to understand the decision-making process. However, decision trees are prone to overfitting, which is why models like random forests are used as well.
- **K-Nearest Neighbors (KNN):** A non-parametric method that predicts heating load based on the average of the *k* nearest neighbors in the feature space. Scaling features is essential for KNN. In this implementation, we utilized 10-fold cross-validation and evaluated k values from 1 to 20 to determine the optimal *k* that minimizes RMSE on unseen data. Included as a non-parametric method that makes predictions based on local neighborhood information. KNN can be effective when decision boundaries are

irregular or difficult to define with parametric models. Its performance is highly dependent on the distance metric and the chosen value of *k* (the number of neighbors), making hyperparameter tuning essential. We also chose KNN as our earlier analysis showed some dependence on distance between the data points for prediction.

- **Random Forest:** An ensemble method combining multiple decision trees to improve prediction accuracy and reduce overfitting. This robustness makes Random Forest a strong candidate for predictive tasks. The number of trees can be specified in the model and can also be tuned through cross-validation. Chosen as a robust ensemble method that builds upon decision trees. By averaging predictions from multiple trees, random forests mitigate overfitting and generally improve predictive accuracy compared to individual decision trees. The choice of Random Forest is motivated by the complexity of heating load dynamics.
- **Support Vector Regression (SVR):** A kernel-based model that maps data to a higher-dimensional space to perform linear regression. The flexibility of different kernels (linear, polynomial, radial basis function) allows SVR to model complex non-linear relationships. We used a radial basis function (RBF) kernel and employed 10-fold cross-validation to tune hyperparameters (cost, gamma) for optimal performance. Data scaling is essential for better performance using SVR. Selected for its capacity to model complex non-linear relationships using kernel functions. The radial basis function (RBF) kernel was specifically employed due to its flexibility in capturing various data patterns. SVR's ability to handle high-dimensional data and its robustness to outliers made it a suitable candidate for this project. The choice of kernel depends on the nature of data and needs to be fine-tuned accordingly.

3.1 Model Training, Evaluation, and Validation

All models were trained using the training dataset (80% of the original data) and evaluated on a held-out test set (20%). The following steps were taken to ensure robust and reliable model evaluation:

- **Feature Scaling:** For KNN and SVR, features were standardized using z-score normalization (mean = 0, standard deviation = 1) before training, where Centering subtracts the mean of each feature from its values. This centers the data around zero for each feature and Scaling divides the centered values of each feature by its standard deviation. This scales the features to have unit variance (i.e., a standard deviation of 1). This is because these models rely on the distance between data points, which can be heavily influenced by different scales of measurement.
- **Cross-Validation (KNN and SVR):** 10-fold cross-validation was applied during training for the KNN and SVR models to tune hyperparameters and obtain more reliable estimates of their performance on unseen data. For KNN, this involved finding the optimal k. For SVR, this involved selecting the best values of C and gamma.

- **Performance Metrics:** Model performance was assessed using Mean Squared Error (MSE) and the coefficient of determination (R-squared or R^2):
    - **MSE:** Measures the average squared difference between predicted and actual heating loads. Lower MSE indicates better predictive accuracy.
    - **R^2:** Represents the proportion of variance in heating load explained by the model. Higher R^2 (closer to 1) indicates a better fit. This metric is similar to MSE, however has the same units as the target variable.

3.2 Results and Discussion

This section presents the performance of the trained machine learning models in predicting building heating load. The primary evaluation metrics used are Mean Squared Error (MSE) and R-squared (R^2). Lower MSE values indicate better predictive accuracy, while higher R^2 values (closer to 1) signify a better fit to the data. The results are summarized in the table and visualized in the accompanying bar charts.

Table 2 Results of the Models

Model	MSE	R-Squared
Random Forest	0.3749	0.996
KNN	2.1815	0.979
SVR	3.7804	0.964
Linear Regression	8.5916	0.917
Decision Tree	6.2996	0.939

- **Random Forest**: As anticipated, the Random Forest model significantly outperformed all other models, boasting the lowest MSE (0.3749) and the highest R-squared (0.9964). This exceptional performance underscores its effectiveness in capturing the complex relationships between building characteristics and heating load, generalizing well to unseen data. Also, from the feature importance plot, we find that the Roof-Area feature is the most important feature for Heating load.

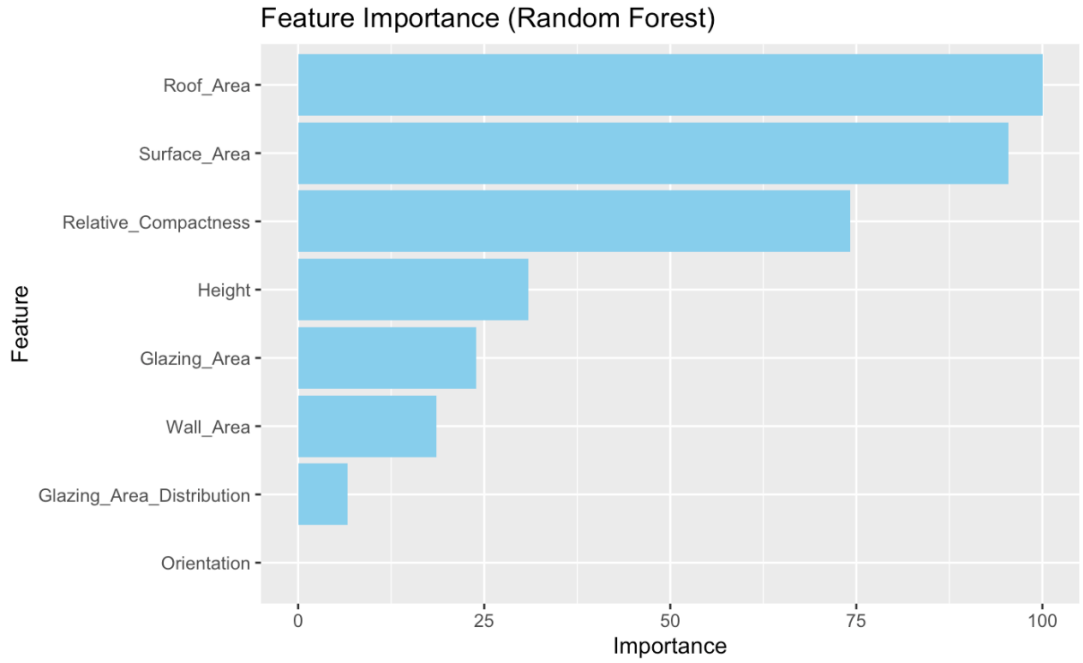

Figure 2 Feature Importance for Random Forest

The feature importance plot for the Random Forest model provides a more robust assessment of feature influence compared to the single Decision Tree. In contrast to the Decision Tree where Glazing_Area was dominant, the Random Forest identifies Roof_Area as the most important predictor, followed by Surface_Area and Relative_Compactness. This shift in feature ranking highlights the ensemble method's ability to capture different aspects of the data and improve generalization. It could be indicative of complex interactions between features that a single decision tree may miss. The consistency of Orientation and Glazing_Area_Distribution as having low

importance across both models suggests these features may not be strong drivers of heating load on their own. This does not mean that these features are unimportant by themselves. It is important to see how these features interact with the other features. It is important to look at other models as well to see if this trend still prevails, or if these features become more important for other models.

- **KNN**: The KNN model demonstrated strong predictive capabilities with an MSE of 2.1815 and an R-squared of 0.9793. This result reinforces the importance of feature scaling and appropriate k-value selection. The optimal k determined through cross-validation was 6, that is the k value for which RMSE is the lowest.

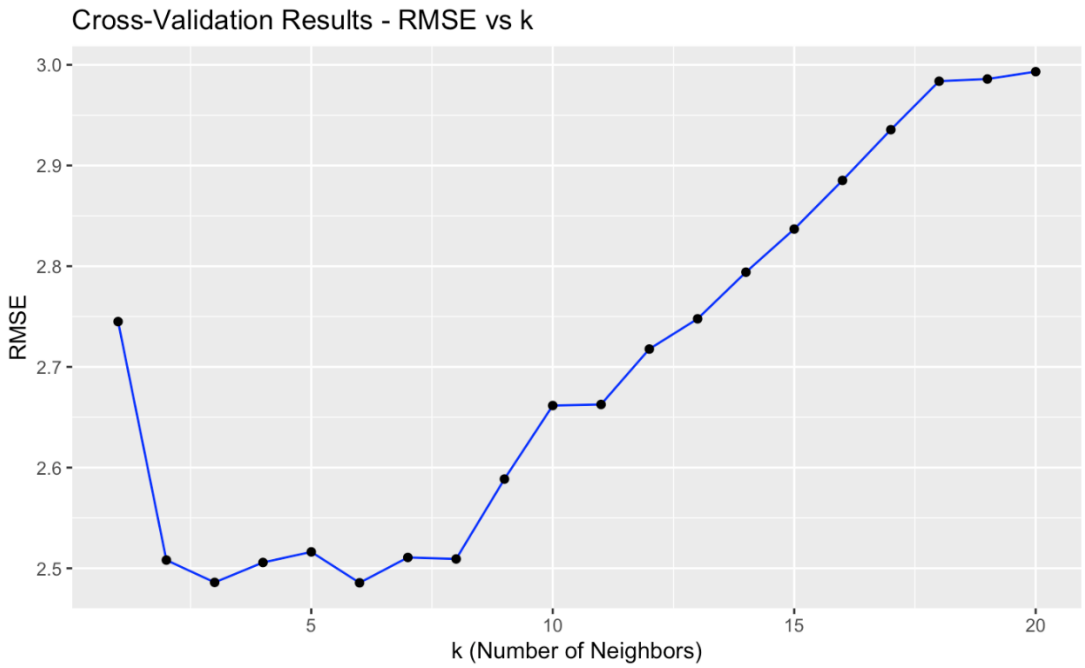

Figure 3 Optimal K-value

- **SVR**: The SVR model, employing a radial basis function (RBF) kernel, achieved a good performance, with an MSE of 3.7804 and an R-squared of 0.9642,which supports its ability to generalize well enough. However, there is a difference in the performance of SVR with respect to Random Forest and KNN. Using a different kernel and also hyperparameter tuning across different combinations of parameters like `cost`, `gamma`, and `epsilon` might provide better results.

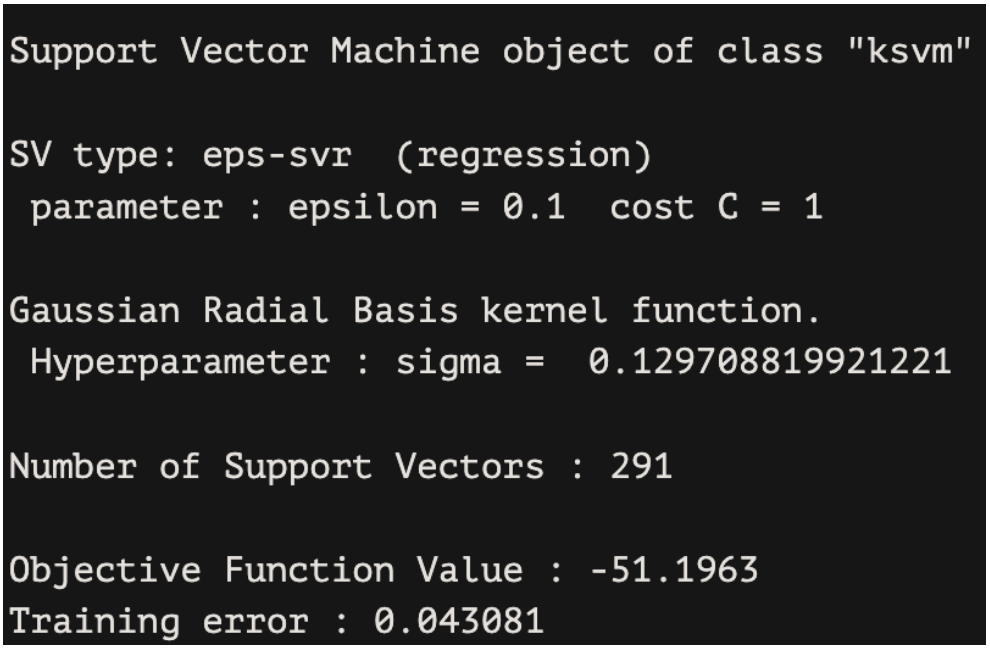

Figure 4 SVR Parameters

The Support Vector Regression model employed a Gaussian Radial Basis Function (RBF) kernel. The parameters used to train the model were a cost of 1, epsilon of 0.1 and the kernel width, sigma, was automatically tuned to an optimal value of 0.13 (approximately) during the 10-fold cross-validation process. The model utilized 291 support vectors, which indicates a relatively complex model, and hence has a risk of overfitting and requires further analysis. The training error was found to be approximately 0.04, which indicates a good fit for the training data. However, it is important to look at the performance of the model on unseen test data to truly estimate its generalization capabilities.

- **Linear Regression**: Serving as a baseline, the Linear Regression model performed reasonably well, with an MSE of 8.5917 and an R-squared of 0.9176. This suggests a moderate linear relationship between features and heating load. The residual plots provide a view to examine if the underlying assumptions of linear regression are met.

Figure 5 Residual Plot

The residual plot for the linear regression model (shown above) displays a mostly random scatter of points around the zero line, suggesting that the assumptions of linearity and homoscedasticity (constant variance) are reasonably well met. However, there is a hint of a funnel shape at higher predicted values, indicating possible mild heteroscedasticity. This means that there might be more variation in the model predictions at higher predicted heating load values. A few potential outliers are also observed, although their influence appears limited. Overall, the residual analysis suggests that while the linear model is a reasonable fit, some deviations from ideal assumptions exist, which could mean potential for improvement using more advanced models like Random Forest or SVR that can capture non-linear patterns and are less sensitive to variations in data.

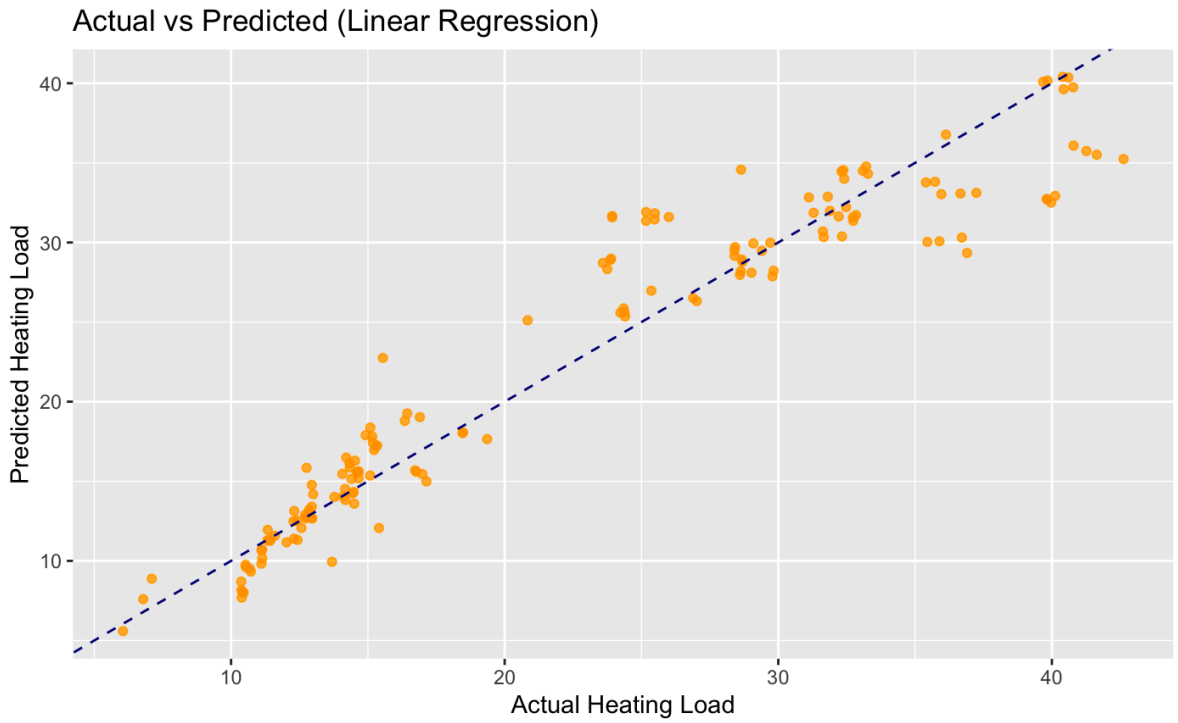

Figure 6 Actual vs Predicted Linear Regression

This scatter plot compares the actual heating load values from your test set against the heating load values predicted by your linear regression model. While most predictions align reasonably well with the actual values, there is noticeable scatter around the diagonal line, indicating prediction errors. Some curvature is apparent in the relationship, which suggests the possibility of non-linearity in the data. This implies that models like Random Forest and SVR might provide even better predictive performance. A few data points appear farther from the diagonal, warranting further investigation to determine if they are outliers or simply examples of higher prediction errors by the linear model.

- **Decision Tree**: While offering interpretability, the Decision Tree model showed the weakest performance with an MSE of 6.2997 and an R-squared of 0.9398. This may indicate overfitting on the training data. Despite using cross-validation for pruning (by tuning the `cp` parameter), it appears the model struggles to generalize effectively to new data. Tuning the `cp` parameter across different ranges might help, or a different complexity parameter needs to be used to improve performance.

  The decision tree model, while relatively simple, offers valuable insights into the factors influencing heating load. The initial split on 'Relative_Compactness' < 0.75 suggests that the feature plays a significant role in dete rmining heating load. Subsequent splits on features like Glazing_Area, highlight its importance in the decision-making process. For example, for approximately 19% of the data, the model predicts a heating load of approximately 11 if Relative_Compactness is less than 0.75 and Glazing Area is less than 0.18. For about 11% of the data, the predicted heating load is 39 when Relative_Compactness is >=0.75 and >=0.81 and Glazing Area < 0.18. This can help energy experts focus on these key attributes for making accurate predictions.

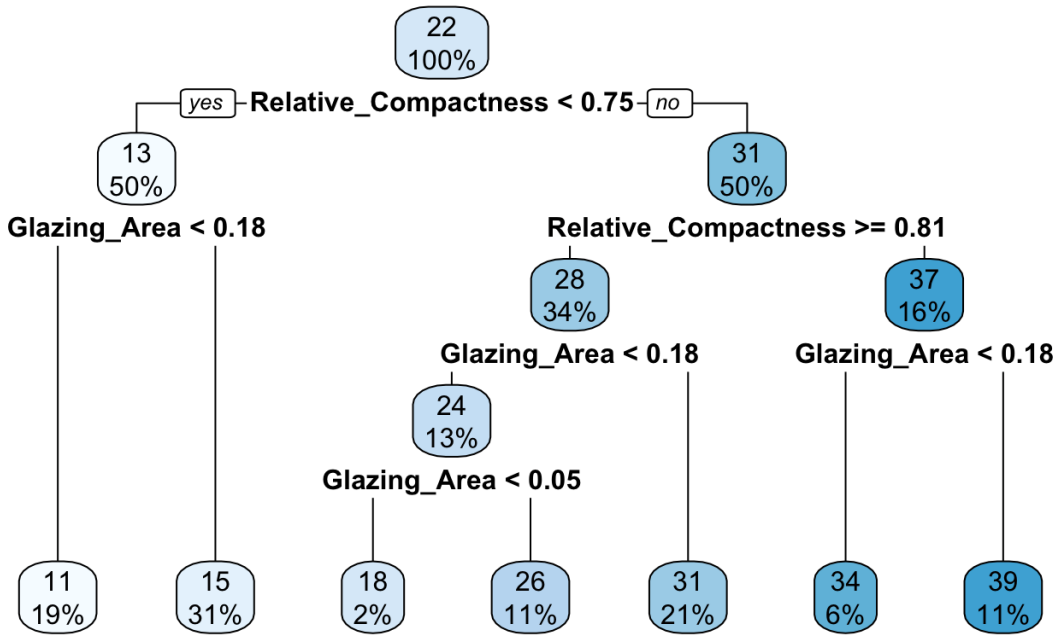

Figure 7 Decision Tree

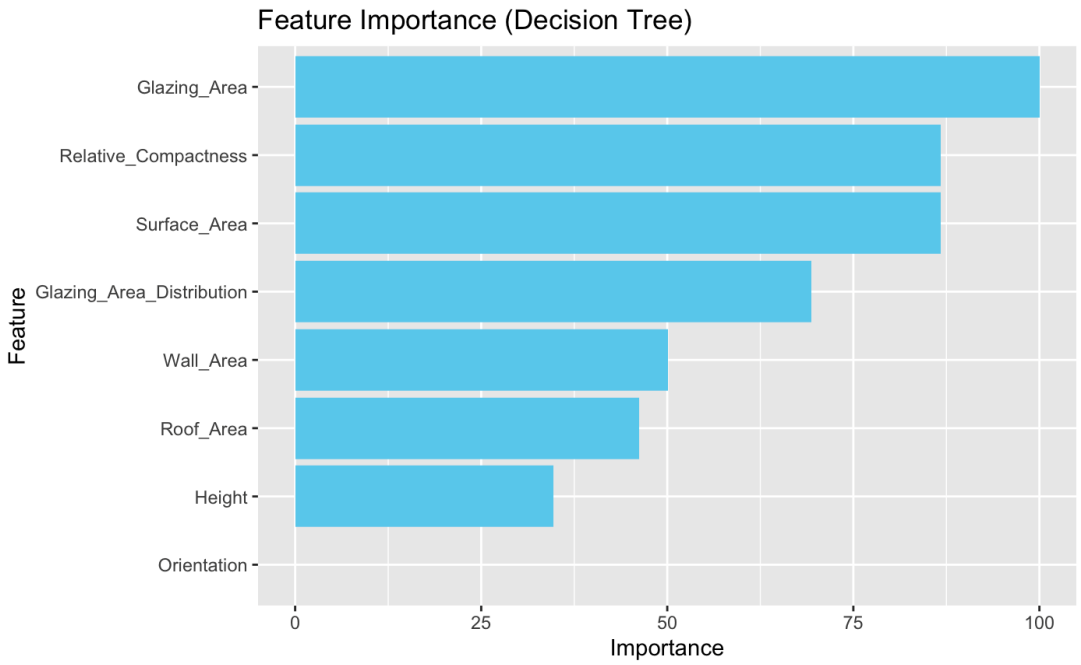

Figure 8 Feature Importance Decision Tree

The feature importance plot for the Decision Tree model reveals that Glazing_Area is the most influential predictor of heating load, followed by Relative_Compactness and Surface_Area. These features likely capture crucial aspects of a building's thermal characteristics, strongly influencing its heating requirements. Glazing_Area being the most important aligns with the expectation that larger glazing areas can significantly impact heat loss or gain. Other features such as Relative_Compactness, Surface_Area also indicate that the structure of the building, which is captured through these features, is highly important for calculating the heating load. The relatively low importance of Orientation suggests it has minimal impact on heating load

prediction in this specific decision tree model. However, orientation might be a crucial factor in relation with other features, for example, large Glazing_Area and South-facing Orientation might lead to more heat intake in winter, hence decreasing the heating load.

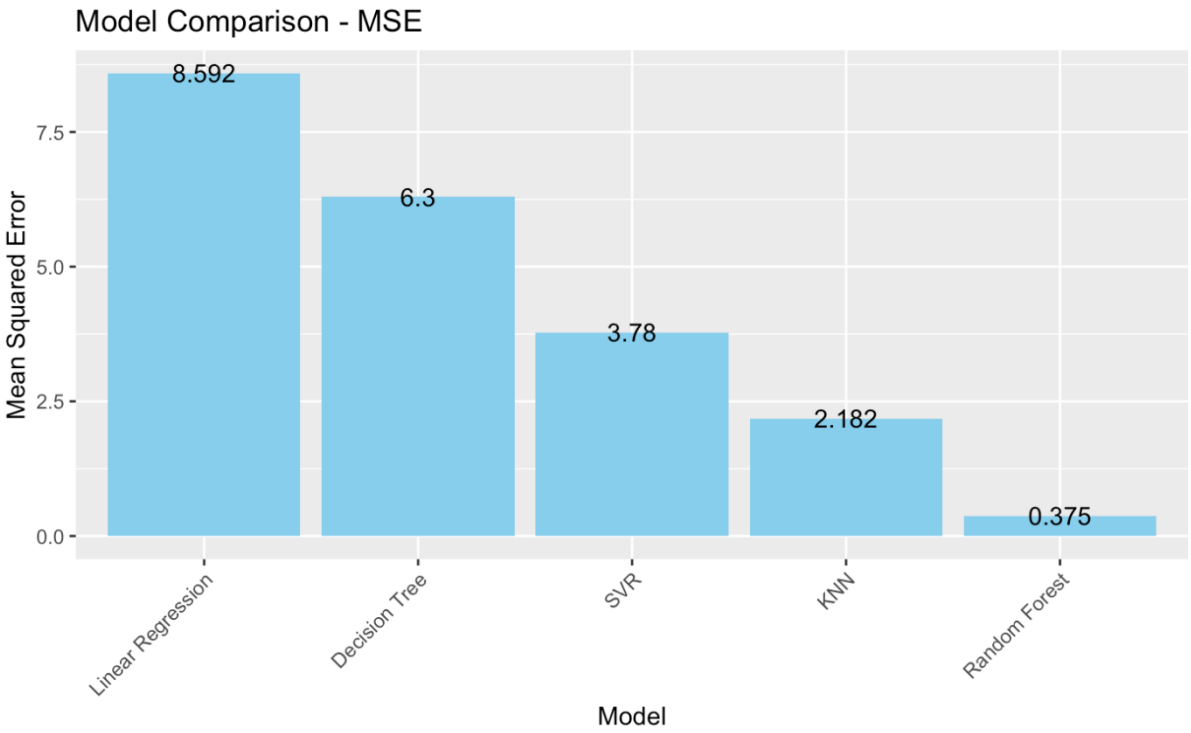

Figure 9 Model Comparison with MSE

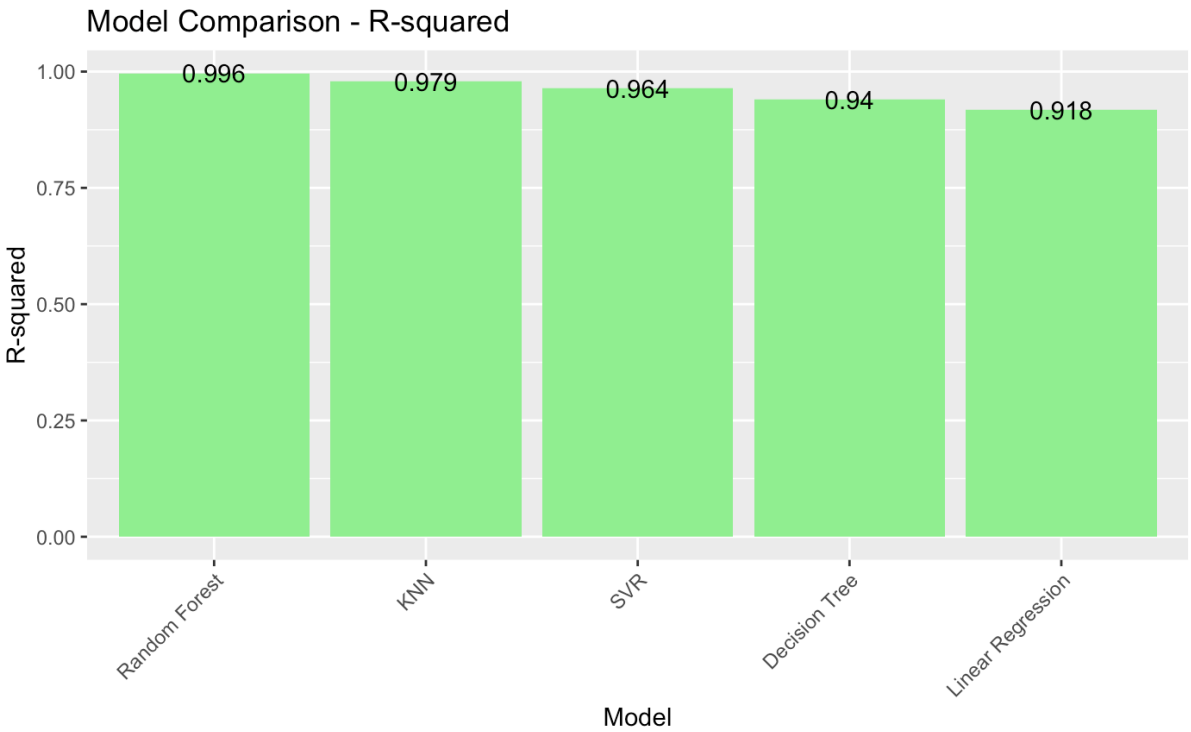

Figure 10 Model Comparison with R^2

The model comparison charts clearly demonstrate the superior performance of the Random Forest model. It achieves the lowest MSE (0.375) and the highest R-squared (0.996), indicating both high accuracy and excellent explanatory power. KNN and SVR also exhibit strong performance, with R-squared values close to 0.98, and MSE of 2.182 and 3.78 respectively, highlighting their ability to capture non-linear relationships in the data. The Linear Regression model produced a moderate fit (R-squared = 0.918), but its higher MSE (8.592) suggests that the assumption of linearity might not fully hold, as suggested by the residual plots, but it is still a reasonable model. The Decision Tree, while interpretable, has a good R-squared (0.94), though its high MSE (6.3) indicates susceptibility to overfitting and a reduced ability to generalize to unseen data.

4. Conclusion & Future Work

4.1 Conclusion

This project successfully explored and compared several machine learning models for predicting building heating load based on architectural and environmental features. The results demonstrate the effectiveness of machine learning in this domain, with Random Forest achieving exceptional predictive accuracy (MSE = 0.375, R-squared = 0.996). KNN and SVR also performed well, providing viable alternatives, especially when interpretability is a priority. The relatively poorer performance of the Decision Tree highlights the limitations of simpler models in capturing the complex relationships present in the data. Linear Regression, while a reasonable baseline, indicated the presence of non-linearity in the data, making a strong case for models like Random Forest and SVR.

The findings of this study can be directly applied to improve building design and energy management strategies. By accurately predicting heating load, architects, engineers, and building managers can optimize energy consumption, reduce costs, and minimize environmental impact. This project showcases that data-driven approaches are highly valuable for improving energy efficiency and supporting sustainability goals in the building sector.

4.2 Future Directions

Several promising avenues exist for extending this research and further improving heating load prediction models. One key area is advanced feature engineering, where creating new features based on domain expertise or exploring non-linear transformations of existing features could significantly enhance model accuracy. For instance, incorporating ratios of building dimensions or interaction terms between glazing area and orientation could capture more nuanced relationships impacting heat transfer. Additionally, gathering more extensive data, such as historical weather patterns, building occupancy schedules, and detailed information about construction materials, would provide a richer context for model training and likely improve predictive power.

Exploring more sophisticated modeling techniques also holds potential. Specifically, implementing Gradient Boosting Machines (GBM), renowned for their high accuracy in various prediction tasks, could yield superior results compared to Random Forest. Fine-tuning the hyperparameters of the best-performing models (Random Forest, KNN, SVR) using more advanced optimization methods like Bayesian optimization or genetic algorithms could uncover even better parameter settings and further enhance prediction accuracy.

Finally, addressing the interpretability challenge of complex models like Random Forest is crucial for practical application. Employing techniques like SHAP values or LIME would enable us to understand the model's decision-making process, identify the most influential features in specific predictions, and gain valuable insights into the factors driving heating load

variations. Combining predictions from multiple top-performing models through ensemble methods could further improve robustness and accuracy. Validating the chosen model on real-world building data will be essential to assess its practical effectiveness and refine it for real-world deployment.

5. References

(1) UCI Machine Learning Repository: Energy Efficiency Dataset. Available at: https://archive.ics.uci.edu/ml/datasets/Energy+efficiency

(2) Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques.* Elsevier

(3) Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

(4) Energy Information Administration (EIA). (2020). Annual Energy Outlook 2020 with projections to 2050. U.S. Department of Energy.

6. Appendix

```{r setup}
# Load libraries
library(ggplot2)
library(dplyr)
library(caret)
library(lattice)
library(readxl)
library(rpart)
library(rpart.plot)
library(randomForest)
library(e1071)
library(corrplot)
```

```{r setup 1}
# Step 1: Load the dataset
data <- read_excel("PRoject Data.xlsx")
str(data)

# Step 2: Rename columns for better understanding
colnames(data) <- c("Relative_Compactness", "Surface_Area", "Wall_Area", "Roof_Area", "Height",
                    "Orientation", "Glazing_Area", "Glazing_Area_Distribution", "Heating_Load", "Cooling_Load")
```

```{r setup 2}
# Step 3: Basic Data Exploration
summary(data)
```

```{r setup 3}
# Step 4: Visualize the distribution of Heating Load
ggplot(data, aes(x = Heating_Load)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  labs(title = "Distribution of Heating Load", x = "Heating Load", y = "Frequency")
```

```{r setup 4}
# Visualize the distribution of Cooling Load
ggplot(data, aes(x = Cooling_Load)) +
  geom_histogram(bins = 30, fill = "green", color = "black") +
  labs(title = "Distribution of Cooling Load", x = "Cooling Load", y = "Frequency")
```

```{r setup 5}
# Step 5: Outlier Detection and Removal for Heating_Load and Cooling_Load
Q1_heating <- quantile(data$Heating_Load, 0.25)
Q3_heating <- quantile(data$Heating_Load, 0.75)
IQR_heating <- Q3_heating - Q1_heating

Q1_cooling <- quantile(data$Cooling_Load, 0.25)
Q3_cooling <- quantile(data$Cooling_Load, 0.75)
IQR_cooling <- Q3_cooling - Q1_cooling
```

```{r setup 6}
# Define bounds for detecting outliers
lower_bound_heating <- Q1_heating - 1.5 * IQR_heating
upper_bound_heating <- Q3_heating + 1.5 * IQR_heating

lower_bound_cooling <- Q1_cooling - 1.5 * IQR_cooling
upper_bound_cooling <- Q3_cooling + 1.5 * IQR_cooling
```

```{r setup 7}
# Remove outliers
data_cleaned <- data %>%
  filter(Heating_Load >= lower_bound_heating & Heating_Load <= upper_bound_heating)
%>%
  filter(Cooling_Load >= lower_bound_cooling & Cooling_Load <= upper_bound_cooling)

cat("Outliers removed:", nrow(data) - nrow(data_cleaned), "\n")
```

```{r setup 8}
```

```
# Step 6: Split the data into training and testing sets
set.seed(123)  # Ensure reproducibility
trainIndex <- createDataPartition(data_cleaned$Heating_Load, p = 0.8, list = FALSE)
train <- data_cleaned[trainIndex, ]
test <- data_cleaned[-trainIndex, ]

cat("Training Set Size:", nrow(train), "\n")
cat("Testing Set Size:", nrow(test), "\n")
```

```{r setup 8a}
# Visualize the distribution of Heating_Load in training and testing datasets
ggplot(train, aes(x = Heating_Load)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Heating Load (Training Data)", x = "Heating Load", y = "Frequency")

ggplot(test, aes(x = Heating_Load)) +
  geom_histogram(bins = 30, fill = "brown", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Heating Load (Testing Data)", x = "Heating Load", y = "Frequency")
```
```{r}
models <- list()
predictions <- list()
results <- data.frame(Model = character(), MSE = numeric(), R_squared = numeric(), stringsAsFactors = FALSE)
```

```{r setup 9}
# Step 7: Train a Linear Regression Model for Heating Load
lm_model <- lm(Heating_Load ~ .-Cooling_Load, data = train)
summary(lm_model)
```

```{r setup 10}
# Predict on the test data
predictions_lm <- predict(lm_model, newdata = test)

# Evaluate Linear Regression Model
mse_lm <- mean((predictions_lm - test$Heating_Load)^2)
r_squared_lm <- R2(predictions_lm, test$Heating_Load)

cat("Linear Regression - Mean Squared Error (MSE):", mse_lm, "\n")
cat("Linear Regression - R-Squared:", r_squared_lm, "\n")
results <- rbind(results, data.frame(Model = "Linear Regression",
                                     MSE = mse_lm,
```

```
                          R_squared = r_squared_lm))
```

```{r setup 10a}
# Actual vs Predicted (Linear Regression)
ggplot(data.frame(Actual = test$Heating_Load, Predicted = predictions_lm), aes(x = Actual, y = Predicted)) +
  geom_point(color = "orange", alpha = 0.7) +
  geom_abline(slope = 1, intercept = 0, color = "navy", linetype = "dashed") +
  labs(title = "Actual vs Predicted (Linear Regression)", x = "Actual Heating Load", y = "Predicted Heating Load")
```

```{r setup 10b}
# Residual Plot (Linear Regression)
residuals_lm <- test$Heating_Load - predictions_lm
ggplot(data.frame(Residuals = residuals_lm, Predicted = predictions_lm), aes(x = Predicted, y = Residuals)) +
  geom_point(color = "purple", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "brown", linetype = "dashed") +
  labs(title = "Residual Plot (Linear Regression)", x = "Predicted Heating Load", y = "Residuals")
```

```{r setup 11}
# Step 8: Train a Decision Tree Model for Heating Load
ctrl <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation
tuneGrid <- expand.grid(cp = seq(0.01, 0.1, 0.01)) # Example cp values to try

tree_model_tuned <- train(Heating_Load ~ .-Cooling_Load,
                          data = train,
                          method = "rpart",
                          trControl = ctrl,
                          tuneGrid = tuneGrid)
rpart.plot(tree_model_tuned$finalModel)

predictions_tree_tuned <- predict(tree_model_tuned, newdata = test)
```

```{r setup 12}
# Evaluate Decision Tree Model
mse_tree <- mean((predictions_tree_tuned - test$Heating_Load)^2)
r_squared_tree <- R2(predictions_tree_tuned, test$Heating_Load)

cat("Decision Tree - Mean Squared Error (MSE):", mse_tree, "\n")
cat("Decision Tree - R-Squared:", r_squared_tree, "\n")
```

```
results <- rbind(results, data.frame(Model = "Decision Tree",
                                     MSE = mse_tree,
                                     R_squared = r_squared_tree))
```

```{r setup 12a}
#Feature Importance (Decision Tree)
importance_rf <- varImp(tree_model_tuned)$importance
importance_rf <- data.frame(Feature = rownames(importance_rf), Importance = importance_rf$Overall)

 ggplot(importance_rf, aes(x = reorder(Feature, Importance), y = Importance)) +
   geom_bar(stat = "identity", fill = "skyblue") +
   coord_flip() +
   labs(title = "Feature Importance (Decision Tree)", x = "Feature", y = "Importance")
```
```{r}
set.seed(0)
k_values <- seq(1, 20, by = 1)
knn_model_tuned <- train(Heating_Load ~ .-Cooling_Load, data = train, method = "knn",
                 preProcess = c("center", "scale"),
                 trControl = trainControl(method = "cv", number = 5),
                 tuneGrid = expand.grid(k = k_values))
cv_results <- knn_model_tuned$results
ggplot(cv_results, aes(x = k, y = RMSE)) +
  geom_line(color = "blue") +
  geom_point() +
  labs(title = "Cross-Validation Results - RMSE vs k", x = "k (Number of Neighbors)", y = "RMSE")
best_k <- knn_model_tuned$bestTune$k
cat("Optimal k value:", best_k, "\n")

knn_model <- train(Heating_Load ~ .-Cooling_Load, data = train, method = "knn",
                 preProcess = c("center", "scale"),trControl = trainControl(method = "cv", number = 5), tuneLength = 10,tuneGrid = expand.grid(k = best_k))
predictions$knn <- predict(knn_model, newdata = test)
mse_knn = mean((predictions$knn - test$Heating_Load)^2)
r2_knn = R2(predictions$knn, test$Heating_Load)
cat("KNN - Mean Squared Error (MSE):", mse_knn, "\n")
cat("KNN - R-Squared:", r2_knn, "\n")
results <- rbind(results, data.frame(Model = "KNN",MSE=mse_knn,R_squared=r2_knn))
```

```{r}
rf_model <- train(Heating_Load ~ .-Cooling_Load, data = train, method = "rf", trControl = trainControl(method = "cv", number = 10), tuneLength = 5)
predictions$rf <- predict(rf_model, newdata = test)
```

```
results <- rbind(results, data.frame(Model = "Random Forest",
                                     MSE = mean((predictions$rf - test$Heating_Load)^2),
                                     R_squared = R2(predictions$rf, test$Heating_Load)))
```

```{r}
importance_rf <- varImp(rf_model)$importance
 importance_rf <- data.frame(Feature = rownames(importance_rf), Importance = importance_rf$Overall)

 ggplot(importance_rf, aes(x = reorder(Feature, Importance), y = Importance)) +
   geom_bar(stat = "identity", fill = "skyblue") +
   coord_flip() +
   labs(title = "Feature Importance (Random Forest)", x = "Feature", y = "Importance")
```

```{r}
svr_model <- train(Heating_Load ~ .-Cooling_Load, data = train, method = "svmRadial", trControl = trainControl(method = "cv", number = 10), preProcess = c("center", "scale"), tuneLength = 3) #tuneLength to reduce execution time, can be increased
predictions$svr <- predict(svr_model, newdata = test)
results <- rbind(results, data.frame(Model = "SVR",
                                     MSE = mean((predictions$svr - test$Heating_Load)^2),
                                     R_squared = R2(predictions$svr, test$Heating_Load)))
```

```{r}
support_vectors <- svr_model$finalModel
support_vectors
```

```{r setup 13}
# Step 9: Compare Models
ggplot(results, aes(x = reorder(Model, -MSE), y = MSE)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Model Comparison - MSE", x = "Model", y = "Mean Squared Error") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+geom_text(label=round(results$MSE,3),check_overlap=T)

ggplot(results, aes(x = reorder(Model, -R_squared), y = R_squared)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  labs(title = "Model Comparison - R-squared", x = "Model", y = "R-squared") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+geom_text(label=round(results$R_squared,3),check_overlap=T)
```