IST 687 – INTRODUCTION TO DATA SCIENCE

Project Report

Energy Consumption Analysis and Conservation Strategies for eSC.

Table of Content:

Introduction:

Our understanding of the production and usage of electricity is changing due to climate change and the need for sustainable energy practices. By assisting our customer, eSC, an energy provider, in managing the growing demand for electricity, particularly during the peak summer season, our initiative directly addresses this issue. With a focus on finding alternatives to costly new power plant construction, eSC, which provides residential services to clients in South Carolina and parts of North Carolina, is dedicated to lowering the danger of blackouts.

Our team is performing an in-depth analysis of data to support the primary goals of eSC. Our goal is to better understand the main causes for household energy use. With this information, eSC will be better equipped to create programs that encourage consumers to save energy, particularly in the summertime.

eSC understands the connection between customer fulfillment and environmental sustainability. We support their aim to minimize their environmental impact while providing reliable electrical supply for their consumers by assisting them in regulating energy use without adding extra infrastructure. This project aligns perfectly with the growing global awareness of the need for climate-conscious energy solutions.

Project Scope:

Our project's goal is to predict South Carolina's county-level energy use in July while considering the climate into consideration. With the assistance of this thorough research, eSC will be able to forecast trends in energy use, manage power output from its present-day plants, and optimize grid allocation. In the end, this strategy seeks to minimize expenses and lower the probability of blackouts while guaranteeing a steady supply of electricity.

Objectives:

Data Preparation:
To ensure high-quality, consistent data for analysis, We have imported and cleaned the dataset carefully from the 3 data resources representing Static houses with the resources available, County data presenting temperature levels, and energy consumed by all the buildings for each of the electrical devices they used.

Visualizations and data exploration:
To discover the underlying patterns and correlations, we have conducted in-depth data research and visualization using R and Shiny app. the use of shiny app helped us present dynamic interpretations.

Build Predictive Energy Cost Models:
Made use of machine learning methods, decision tree, XGBoost and linear regression, to identify characteristics that affect energy usage and forecast energy expenditures.

Create a Shiny Interactive App:
Using Shiny, created an interactive application that lets users interact with the data, see the outcomes, and get insights.

Assist Stakeholders Make Decisions:
Provided decision-makers a strong basis to assist them comprehend the factors that affect energy costs and overcome energy management constraints.

Dataset:

Static House Data:
The dataset comprises details on about 5,000 single-family homes served by eSC, It is stored in Parquet format to manage large volumes of data efficiently. The key elements are:

- **Building ID**: a unique code that links every home to information about how much energy it uses.

- **House Features**: Details regarding the residences, such as size, age, type of construction, and number of bedrooms and bathrooms.

- **Location:** Information on the residences' geographic location, such as the county and ZIP code, which enables geographic analysis.

Energy Usage Data:
Each house in the static data collection has hourly energy use information in this dataset. Each house has a unique file that may be found using the building ID that is in the Parquet data format for ease in storage. The information includes:

- **Energy Consumption:** Total consumption of energy – (kWh).

- **Appliance usage**: Individual appliances usage data.

- **Time and Date:** Event of energy measurement.

Weather Data:
This dataset contains hourly weather information of all mentioned counties from Carolina with its own code. It is in CSV format.

- **Time and Date:** Event of weather measurement.

- **Weather Parameters:** Wind speed, Temperature, Humidity…. etc.

Data Preparation:

In this project, the data import and cleaning process is essential to ensuring the quality, and consistency of the dataset for the subsequent analytical phases. This process involves several important steps to ensure a robust dataset:

Data Filtering:

We start our data preparation by loading a static house information dataset from a Parquet file using arrow::read_parquet(). This dataset contains a wide variety of information for our analysis. To streamline the data, we created a list of columns to remove, which we call columns_to_remove, focusing on technical or redundant fields like climate zones, vehicle information, emissions data, and specific HVAC system details. these are selected because, there is no significant difference in the factor levels of these variables that could potential influence energy consumption. By removing these unnecessary columns, we can reduce clutter and focus on the relevant data. We also create an additional list, excluded_columns, which includes a broader range of columns to exclude, ensuring we filter out everything that doesn't align with our analytical goals. As we remove the unwanted columns from the dataset, we then print the names of the remaining columns to confirm that we've successfully filtered out the irrelevant information. This process helps us prepare a cleaner, more manageable dataset, ready for further analysis and visualization.

Code:

```
library(sfarrow)
# helps read Parquet files
library(sf)
# helps with spatial data
library(arrow)
# URL for the static house info Parquet file
static_house_info_url <- "https://intro-datascience.s3.us-east-2.amazonaws.com/SC-
data/static_house_info.parquet"

# Read the static house info data using the arrow package
static_house_info <- arrow::read_parquet(static_house_info_url)

# Columns to remove
columns_to_remove <- c(
  # List of columns to be removed from the dataset
  "in.cec_climate_zone",
  "in.dehumidifier",
  "in.electric_vehicle",
  "in.emissions_electricity_folders",
```

```
  "in.emissions_electricity_values_or_filepaths",
  "in.geometry_building_horizontal_location_mf",
  "in.geometry_building_horizontal_location_sfa",
  "in.geometry_building_level_mf",
  "in.geometry_building_number_units_mf",
  "in.geometry_building_number_units_sfa",
  "in.geometry_building_type_acs",
  "in.geometry_building_type_height",
  "in.geometry_building_type_recs",
  "in.hot_water_distribution",
  "in.holiday_lighting",
  "in.hot_water_distribution",
  "in.hvac_has_shared_system",
  "in.hvac_secondary_heating_efficiency",
  "in.hvac_secondary_heating_type_and_fuel",
  "in.hvac_shared_efficiencies",
  "in.hvac_system_single_speed_ac_airflow",
  "in.hvac_system_single_speed_ac_charge",
  "in.hvac_system_single_speed_ashp_airflow",
  "in.hvac_system_single_speed_ashp_charge",
  "in.iso_rto_region",
  "in.mechanical_ventilation",
  "in.overhangs",
  "in.simulation_control_run_period_begin_day_of_month",
  "in.simulation_control_run_period_begin_month",
  "in.solar_hot_water",
  "in.units_represented",
  "in.emissions_electricity_folders",
  "in.emissions_electricity_values_or_filepaths",
  "in.emissions_electricity_units",
  "in.emissions_scenario_names",
  "in.geometry_story_bin",
  "in.emissions_electricity_values_or_filepaths",
  "in.electric_vehicle",
  "in.puma_metro_status",
  "in.income_recs_2015",
  "in.income_recs_2020",
  "in.radiant_barrier",
  "in.misc_well_pump"

)
```
```{r}
# Assume df is your data frame
column_names <- colnames(static_house_info)
print(column_names)

```
```{r}
# Assuming static_house_info is your data frame
all_columns <- colnames(static_house_info)

# List of columns to exclude
excluded_columns <- c('in.weather_file_city','bldg_id', 'in.sqft',
'in.county','in.heating_fuel','in.income','in.insulation_ceiling','in.lighting','in.usage_level',
'in.weather_file_longitude','in.weather_file_latitude','in.bathroom_spot_vent_hour','in.building_america_cl
```

```
imate_zone','in.ceiling_fan','in.clothes_dryer','in.clothes_washer','in.clothes_washer_presence','in.cooking_
range', 'in.cooling_setpoint','in.cooling_setpoint_has_offset','in.cooling_setpoint_offset_magnitude',
'in.cooling_setpoint_offset_period','in.corridor')

# Filter out the excluded columns
filtered_columns <- all_columns[!all_columns %in% excluded_columns]

# Print the remaining columns
print(filtered_columns)
```

Retrieving and Processing Energy Data for Buildings:

The “obtain_energy” function retrieves energy consumption data for a specific building, identified by its building ID (“bldg_id”). It starts by creating the URL to access the corresponding Parquet file and then reads the data using arrow::read_parquet(). The function filters the data to include only records from July, using the month() function to ensure the correct time frame. It identifies numeric columns and corrects any negative values by converting them to positive, ensuring energy consumption is always represented as a positive number. To calculate hourly energy consumption, the function sums all columns except for time to create a “total_energy_hour” column. The data is then converted to a daily format by extracting the date from time and aggregating energy usage by building ID and date. The final output is a dataframe with the total energy consumption for each day in July, providing a comprehensive view of daily energy use. This approach is useful for analyzing energy consumption patterns and understanding the energy usage behavior of different buildings

Code:

```
# Function to obtain energy related data using building id
library(purrr)

obtain_energy <- function(bldg_id) {

  bldg_url <- paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/2023-
houseData/",bldg_id,".parquet")

  df <- arrow::read_parquet(bldg_url)

  #Choosing the data of energy consumption in month of July
  df <-  df %>% filter(month(time) == 7)

  # Checking the numeric variables for negative values and converting them to positive values
  numerical_cols <- sapply(df, is.numeric)
  df[, numerical_cols] <- lapply(df[, numerical_cols], function(x) ifelse(x < 0, abs(x), x))
```

```
  # Obtaining total energy consumption for hour
  df$total_energy_hour <- rowSums(df[,-which(names(df) == 'time')])

  df <- df[, c('time', 'total_energy_hour')]

  df$time <- as.POSIXct(df$time, tz= 'EST', origin = '1970-01-01')

  # Aggregating by day, obtaining daily total energy consumption for month of july
  daily_df <- df %>%
          mutate(date = as.Date(time)) %>%
          mutate(bldg_id = bldg_id) %>%
          group_by(bldg_id, date) %>%
          summarize(total_energy = sum(total_energy_hour, na.rm = TRUE), .groups = 'drop')

  return(daily_df)
}
```

Extracting and Summarizing Weather Data for All Counties in July:

We started by loading weather data from a CSV file into weather_data and retrieving building IDs from house_data. To fetch weather data for specific counties, we create a function, fetch_county_weather, that constructs a URL based on the county's name, reads the corresponding CSV, and converts the date_time column into a Date format to ensure the extraction and analysis to be performed only on July month. This function then filters the weather data for July and adds a new column, in.county, to indicate which county the data corresponds to. Next, we loop over the unique county IDs from house_data and use the fetch_county_weather function to collect weather data for each county, combining the results into a single dataframe, all_county_weather_df, using do.call(rbind, all_county_weather). To summarize the weather data, we group it by in.county and use summarize(across(where(is.numeric), sum, na.rm = TRUE)) to calculate the total values for each numeric column in July. This comprehensive weather summary provides a useful dataset for analyzing weather conditions across counties, aiding further studies and comparisons with energy consumption data.

Code:

```
library(sfarrow)
# helps read Parquet files
library(sf)
# helps with spatial data
library(arrow)
library(readr)
```

```
library(dplyr)
library(lubridate)
# helps in working with dates an times

# Read house data
bldg_id <- house_data$bldg_id

# Function to fetch weather data for a specific county
fetch_county_weather <- function(in.county) {
  # Construct URL for weather data of the county
  weather_url <- paste0("https://intro-datascience.s3.us-east-2.amazonaws.com/SC-data/weather/2023-
weather-data/", in.county, ".csv")

  # Read weather data
  weather_df <- read_csv(weather_url)

  # Convert date column to Date type
  weather_df$date <- as.Date(weather_df$date_time)

  # Filter weather data for July
  july_weather <- weather_df %>%
    filter(month(date) == 7)

  #adds a new column to the weather data called in.county, which tells us which county the weather data is
for
  july_weather$in.county <- in.county

  return(july_weather)
}

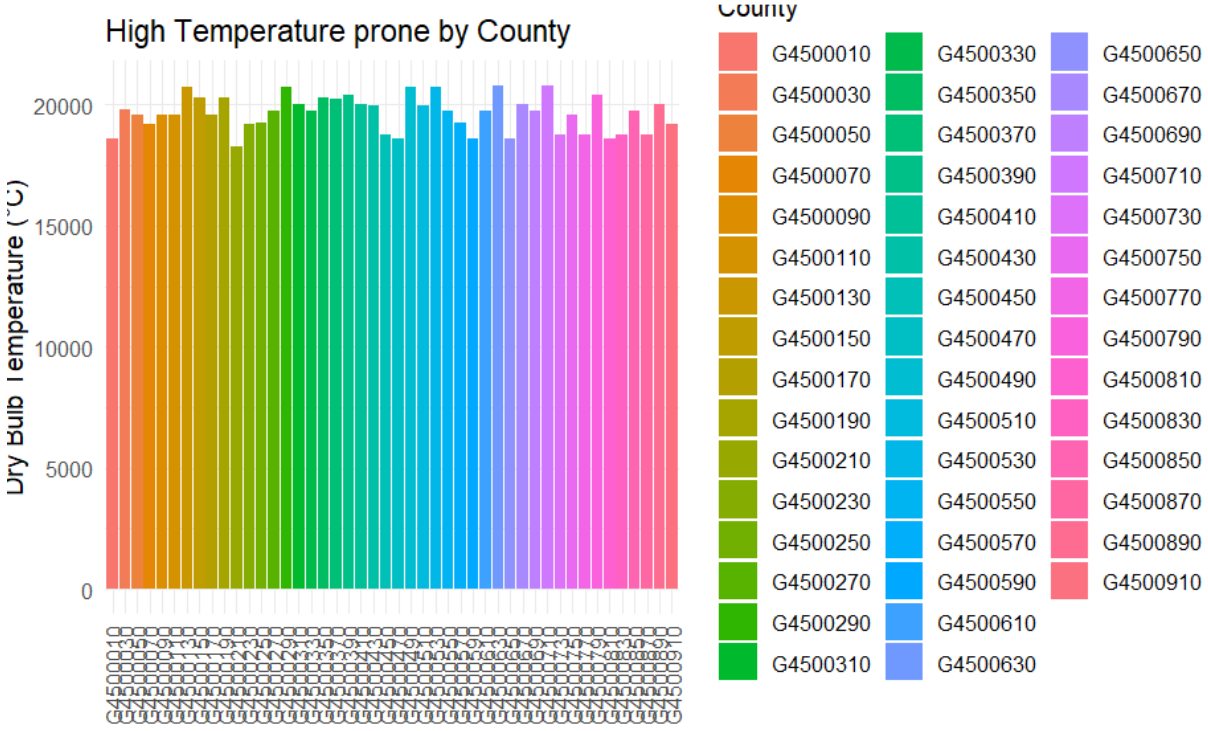
# Loop over unique county IDs in houses_df and fetch weather data for each county
all_county_weather <- lapply(unique(house_data$in.county), fetch_county_weather)

# Combine weather data for all counties into a single dataframe
all_county_weather_df <- do.call(rbind, all_county_weather)

all_county_weather_df <- all_county_weather_df %>%group_by(in.county, )
%>%summarise(across(where(is.numeric), sum, na.rm = TRUE))
```

Analyzing Counties with Peak Temperatures and Merging Building and Energy Data:

To identify which counties experience the highest temperatures, we created a bar plot using ggplot2, with in.county on the x-axis and dry bulb temperatures (°C) on the y-axis. This plot, derived from all_county_weather_dataset, provides a visual representation of temperature variations across counties, helping us determine which regions are prone to extreme heat. We use a minimal theme for clarity and rotate the x-axis labels for better readability.



Additionally, we merge building information with energy consumption data to understand the relationship between building attributes and energy use. We use the merge() function to combine static_house_info with Energy_consumed on the bldg_id key which is uniquely identified and acts as a foreign key , creating a new dataset called building_energy_data. This merged dataset allows us to analyze energy consumption in relation to building characteristics, providing insights into how factors like size, age, or location might affect energy demand. This comprehensive dataset is valuable for deeper analysis and guiding energy management strategies.

Code:

```
#understanding counties with peak temperature

library(ggplot2)

# Assuming your dataframe is named 'data' and contains a column named 'Dry.Bulb.Temperature...C.'
# and 'in.county' for the county identifier
temperature_plot <- ggplot(all_county_weather_dataset, aes(x = in.county, y =
all_county_weather_dataset$Dry.Bulb.Temperature...C., fill = in.county)) +
  geom_bar(stat = "identity") +
  labs(
    title = "High Temperature prone by County",
    x = "County",
    y = "Dry Bulb Temperature (°C)"
  ) +
  theme_minimal() +
```

```
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  guides(fill = guide_legend(title = "County"))

# Print the plot
print(temperature_plot)


#MERGE static_house_info with total energy based on buidling ID

building_energy_data <- merge(static_house_info, Energy_consumed, by = "bldg_id")
#write.csv('C:/Users/HP/Desktop/INTRO TO DS/Final/building_energy_data.csv')
```

Data Transformation and Integration for Energy and Weather Analysis:

We begin by creating the obtain_mean function to convert values with ranges into single values. To further clean the data, we use mappings to convert categorical values to numeric ones, such as in_geometry_floor_area_mapping for floor area and in_vacancy_status_mapping for occupancy status.

Then, We merged building_energy_data with county-level weather data to integrate energy consumption with weather conditions.. Additionally, we converted certain text-based values to numeric by removing the word "Hour" from columns like in.bathroom_spot_vent_hour.

The final merged dataset, final_data, is saved to a CSV file, offering a unified dataset for analyzing energy consumption concerning building characteristics and weather conditions. This helps us understand the factors influencing energy usage across different regions.

Code:

```
# Display the dimensions (number of rows and columns) of merge_static_house_info_df
dim(building_energy_data)

in_geometry_floor_area_mapping <- c("0-499"=1 ,"500-749"=2,"750-999"=3,"1000-1499"=4,"1500-
1999"=5,"2000-2499"=6,"2500-2999"=7,"3000-3999"=8,"4000+"=9)
in_hot_water_fixtures_mapping <- c("100% Usage"=1, "50% Usage"=0, "200% Usage"=2)
upgrade_cooking_range_mapping <- c("Electric, Induction, 100% Usage"=1, "Electric, Induction, 80%
Usage"=0,  "Electric, Induction, 120% Usage"=3)
in_occupants_mapping <- c("1"=1  , "2"=2,"3"=3,"4"=4,"5"=5,"8"=8,"6"=6,"7"=7,"10+"=10,"9"=9)
in_vacancy_status_mapping <- c("Occupied"=1, "Vacant"=0 )
income_mapping <- c("<10000"=1, "10000-14999"=2, "15000-19999"=3, "20000-24999"=4, "25000-
29999"=5, "30000-34999"=6, "35000-39999"=7, "40000-44999"=8, "45000-49999"=9, "50000-59999"=10,
"60000-69999"=11, "70000-79999"=12, "80000-99999"=13, "100000-119999"=14, "120000-139999"=15,
"140000-159999"=16, "160000-179999"=17, "180000-199999"=18, "200000+"=19)


# For the columns that have ranges, we are generally taking mean for them.
```

```
obtain_mean <- function(values) {
 if (grepl(">", values)) {
  greater_than_value <- as.numeric(gsub(">", "", values))
  return(greater_than_value + 1)
 }

 if (grepl("<", values)) {
  less_than_value <- as.numeric(gsub("<", "", values))
  return(less_than_value - 1)
 }

 split <- strsplit(values, "-")[[1]]

 split <- as.numeric(gsub("\\+", "", split))

 if (length(split) == 2) {
  return(mean(split))
 } else {
  return(split[1])
 }
}

building_energy_data$in.income <- sapply(building_energy_data$in.income, obtain_mean)
```

```
#Convert all necessary columns to numerical form


building_energy_data$in.geometry_floor_area <-
as.numeric(in_geometry_floor_area_mapping[building_energy_data$in.geometry_floor_area])


building_energy_data$in.hot_water_fixtures <-
as.numeric(in_hot_water_fixtures_mapping[building_energy_data$in.hot_water_fixtures])


building_energy_data$upgrade.cooking_range <-
as.numeric(upgrade_cooking_range_mapping[building_energy_data$upgrade.cooking_range])
```

Further, we converted data into factor levels for those required to ensure the removal of all factors less than 2 thus preparing data ready for model training especially for linear regression.

```
final_data_4<-final_data_2
final_data_4_column_names <- colnames(final_data_4)
final_data_4_column_names

# Assuming factor_columns has been defined as:
factor_columns <- sapply(final_data_4, is.factor)

# Loop through the dataframe and convert columns to factor
for (col in names(final_data_4)[!factor_columns & !sapply(final_data_4, is.numeric)]) {
```

```
  final_data_4[[col]] <- as.factor(final_data_4[[col]])
}

factor_columns <- sapply(final_data_4, is.factor)

# Check levels in each factor column
levels_count <- sapply(final_data_4[factor_columns], function(x) length(levels(x)))

# Identify columns with fewer than two levels
problematic_columns <- names(levels_count[levels_count < 2])

problematic_columns

final_data_6 <- final_data_4[, -which(names(final_data_4) %in% problematic_columns)]
str(final_data_6)

#Merge county column with energy consumed dataframe based on county ID

```{r}
final_merged_data <- merge(building_energy_data_to_merge,all_county_weather_dataset , by =
"in.county")
```
```

Prediction Models:

Preparation for Modeling:

First, we start by removing unnecessary columns from the dataset to keep the data relevant, resulting in a cleaner dataset with only the essential information. Next, we convert non-numeric columns to factors, ensuring that all categorical data is properly formatted for modeling. After that, we identify and remove problematic columns with fewer than two levels, as these lack the necessary variability for effective analysis. Once the data is cleaned, we split it into training and testing sets for building our regression model.

We allocate 80% of the data to the training set and 20% to the testing set. This setup allows us to build the regression model using the training data and then assess its accuracy with the test data. This approach creates a robust foundation for analyzing energy consumption and constructing effective predictive models, providing reliable insights for further analysis and decision-making.

XGBoost Model:

To predict “total_energy”, we use the XGBoost algorithm. This model is set for 100 boosting rounds to fit the regression model.

Code:

```
library(dplyr)

# Building the model
model_xgb <- xgboost(
  data = as.matrix(train_data_set2[, -which(names(train_data_set2) == "total_energy")] %>%
select_if(is.numeric)),
  label = train_data_set2$total_energy,
  objective = "reg:squarederror",  # Use squared error for regression
  nrounds = 100  # Adjust the number of boosting rounds
)

# Make predictions on the test set
predictions <- predict(model_xgb, as.matrix(test_data_set2[, -which(names(test_data_set2) ==
"total_energy")] %>% select_if(is.numeric)))

# Obtain the R-Squared error.
rsquared <- 1 - (sum((predictions - test_data_set2$total_energy)^2) /
sum((mean(test_data_set2$total_energy) - test_data_set2$total_energy)^2))
cat("R-squared:", rsquared, "\n")
cat("Accuracy of XG BOOST model is : ", rsquared * 100, "\n")
```

To evaluate the model's accuracy, we calculate the R-Squared value, which represents the proportion of variance in total_energy explained by the model.

Model Accuracy: The model explains over half of the variance in energy consumption, showing moderate accuracy, with an R-Squared Value of approximately “49.41%”. This finding implies that the model might be enhanced by modifying its parameters, enhancing the quality of the data, or considering alternative machine learning strategies.

Decision Tree:

The training dataset (train_data_set2) contains a variety of predictors that are used in the development of this Decision Tree model to forecast total_energy. The model makes use of the R rpart package, which makes the CART (Classification and Regression Trees) technique possible. Regression tasks are the only use for it, as the method="anova" parameter indicatesThe train_data_set2 dataset is used to train the model, while test_data_set2 is used to generate predictions. Together with other predictor factors, the total_energy column needs to be present in both datasets.

Code:

```
library(rpart)

# Building the Decision Tree model
lm_model <- rpart(total_energy ~ ., data = train_data_set2, method = "anova",
                  control = rpart.control(maxdepth = 8))

# Predicting with the Decision Tree model
predictions_dt <- predict(lm_model, newdata = test_data_set2, type = "vector")

# Calculate R-squared for Decision Tree
actual_values_dt <- test_data_set2$total_energy
rsquared_dt <- 1 - (sum((actual_values_dt - predictions_dt)^2) / sum((actual_values_dt -
mean(actual_values_dt))^2))

# Print the R-squared and accuracy
cat("R-squared for Decision Tree:", rsquared_dt, "\n")
cat("Accuracy of Decision Tree model is:", rsquared_dt * 100, "%\n")
```

The rpart function is used to develop the Decision Tree model:
equation: total_energy ~. (The dependent variable in the model is total energy; the predictors are all the other columns in train_data_set2.)

Approach: Anova (This indicates that a regression tree is the model.) Anova is used to compare make a prediction for 2 or more dependable variables. this delivering better fit of the model.
Controlling variables: maximum depth of 8 (This controls the model's complexity by setting the tree's maximum depth to eight.) Predictions are made using the trained model on the test_data_set2. The predictions are computed as a vector of numerical values representing the predicted total_energy. The R-squared metric, which measures the percentage of variance in the dependent variable that is predicted from the independent variables, is used to assess the model's performance. This metric assesses how well the regression model fits the data.

Model Accuracy: An improved fit is indicated by a higher R-squared value. The Decision Tree model's R-squared was roughly 0.563, which indicates that it can account for roughly 56.28% of the variation in total energy.

Linear Model:

To determine the relationships within the dataset, specifically between the dependent variable, total energy costs, and a set of independent factors or features, we used linear regression. To begin, the dataset is split into training and testing sets. This makes it easier for the model to be trained on one subset and tested on another to see how well it can generalize.

The training dataset is then used to train the linear regression model, which finds the coefficients for each independent variable to best predict the dependent variable.

Code:

```
# Splitting 80% of data into training set, and 20% into testing set.

lmoutSet2<- lm(total_energy ~ ., data = train_data_set2)
summary(lmoutSet2)


new_merge_static_house_info_df4<- final_data_6
new_merge_static_house_info_df4$Dry.Bulb.Temperature...C.<- final_data_6$Dry.Bulb.Temperature...C.+5
lmoutNew <- predict(lm_model, newdata = new_merge_static_house_info_df4)


summary(lmoutNew)
```

The model predicts total energy values for a test dataset, which are then compared to actual values to assess accuracy. R-squared, a measure of the model's goodness of fit, quantifies the proportion of variance in total energy consumption explained by the independent variables. A higher R-squared value indicates a better fit, meaning the model can predict total energy consumption more accurately based on the selected features.

Model Accuracy: The adjusted R-squared value of 0.9439 indicates that approximately 94.39% of the variance in total energy consumption can be explained by the independent variables in the linear regression model. This high value suggests that the model is highly accurate in predicting total energy consumption based on the provided dataset.

The R-squared metric is a valuable tool for understanding and forecasting energy usage patterns because it provides a quantitative measure of how well the model fits the data. A high R-squared value indicates that the model can capture a large portion of the variability in total energy consumption, which is essential for accurate predictions.

Overall, the high adjusted R-squared value demonstrates the effectiveness of the linear regression model in predicting total energy consumption and highlights its potential as a reliable tool for energy usage forecasting.

Shiny App:

An interactive platform for analyzing consumption of energy data is developed by the Shiny app. It combines a user interface (UI) function that establishes the layout with a server function that performs the application's logic.

We used ggplot2 for plotting and read the data from a CSV file. The plots are generated based on different aspects of energy consumption.

User Interface (UI) Function: This function determines the app's layout using a “fluidPage”. The layout is designed with a “tabsetPanel”, allowing users to switch between different types of analyses by clicking on various tabs.

The Shiny app allows users to interact with the data, explore visualizations, and analyze energy consumption trends. The user-friendly interface provides flexibility in switching between different plot types.

The structure of the Shiny app enables easy customization and extension.

Code:

```
library(shiny)
library(ggplot2)
library(readr)

predictedDF <- read.csv("predicted_df.csv")

predictedDF$date_time <- as.Date(predictedDF$date_time)

server <- function(input, output) {

  # Time Series Plot of Energy Consumption
  output$timeseriesPlot <- renderPlot({
    ggplot(data = predictedDF, aes(x = date_time, y = total_energy)) +
      geom_line(aes(color = "Actual")) +
      geom_line(aes(y = new_total_energy, color = "Predicted")) +
      labs(x = "Date", y = "Energy Consumption (kWh)", color = "Legend") +
      theme_minimal()
  })

  # Temperature vs. Energy Consumption Scatter Plot
  output$tempEnergyScatter <- renderPlot({
    ggplot(data = predictedDF, aes(x = Dry_Bulb_Temperature, y = total_energy)) +
      geom_point() +
      labs(x = "Temperature (°C)", y = "Energy Consumption (kWh)") +
      theme_minimal()
  })
```

```
  # Heatmap of Energy Consumption by Building Characteristics
  output$heatmapBuilding <- renderPlot({
   ggplot(data = predictedDF, aes(x = factor(in.sqft), y = factor(in.bedrooms), fill = total_energy)) +
     geom_tile() +
     scale_fill_gradient(low = "blue", high = "red") +
     labs(x = "Square Feet", y = "Bedrooms", fill = "Energy (kWh)") +
     theme_minimal()
  })

  # Histogram of Energy Consumption by Building Age
  output$histogramBuildingAge <- renderPlot({
   ggplot(data = predictedDF, aes(x = in.vintage, y = total_energy)) +
     geom_histogram(stat = "identity", position = "dodge") +
     labs(x = "Building Vintage", y = "Energy Consumption (kWh)") +
     theme_minimal()
  })

  # Boxplot of Energy Consumption by Insulation Quality

  output$boxplotInsulation <- renderPlot({
    ggplot(data = predictedDF, aes(x = reorder(in.insulation_ceiling, total_energy, FUN = median), y =
total_energy)) +
     geom_boxplot() +
     labs(x = "Ceiling Insulation Quality", y = "Energy Consumption (kWh)") +
     theme_minimal() +
     coord_flip() # Optional: Flip the coordinates if you want horizontal boxplots
  })
}

ui <- fluidPage(
  titlePanel("Energy Consumption Analysis for eSC"),

    mainPanel(
     tabsetPanel(
       tabPanel("Time Series", plotOutput("timeseriesPlot")),
       tabPanel("Temp vs Energy", plotOutput("tempEnergyScatter")),
       tabPanel("Heatmap", plotOutput("heatmapBuilding")),
       tabPanel("Histogram by Age", plotOutput("histogramBuildingAge")),
       tabPanel("Boxplot Insulation", plotOutput("boxplotInsulation"))

    )
  )
)

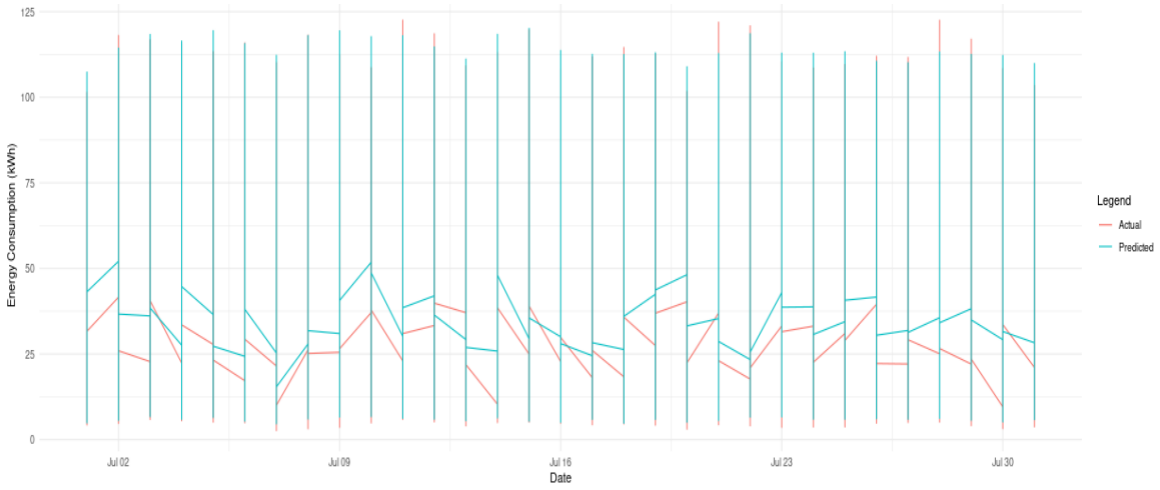
shinyApp(ui = ui, server = server)
```

Shiny App Link:

https://yvishwak.shinyapps.io/IDS_Final/

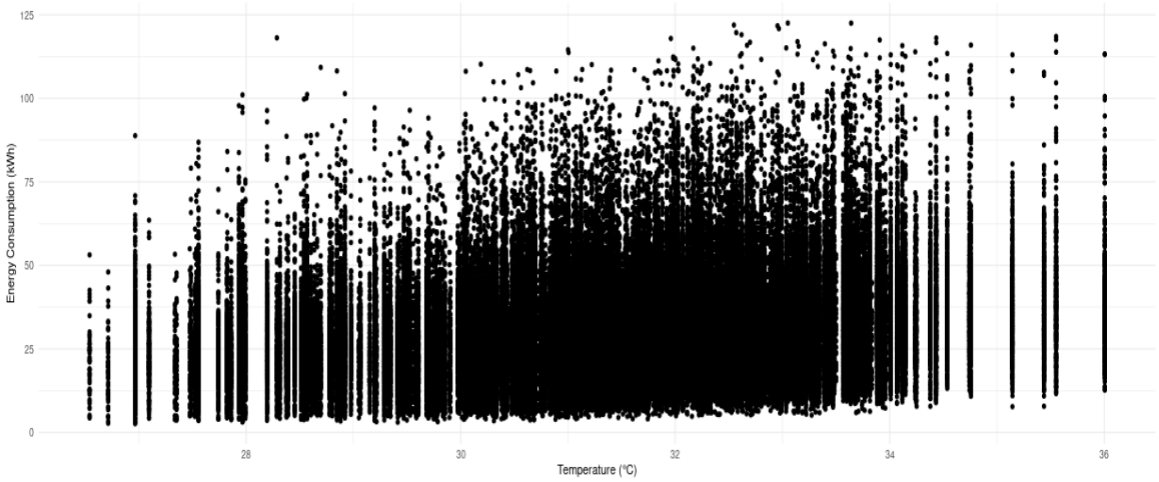
Visualizations:

1. Daily Energy Consumption: “Actual vs Predicted”



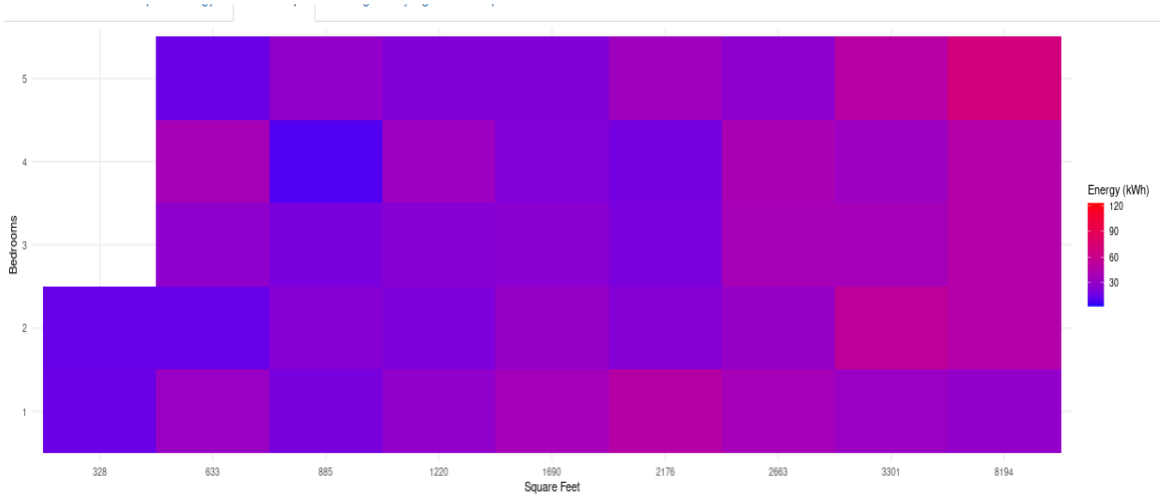
The graph shows daily energy consumption over a month, revealing a pattern where energy use peaks periodically, likely on weekends when people are home and use more electricity. The blue line represents actual energy use, while the red line shows predicted usage, closely following the actual trends. This indicates the predictions are quite accurate but could be improved to better match actual peaks, helping in planning energy production and encouraging efficient energy use during high-demand periods.

2.Temperature vs Energy Consumption



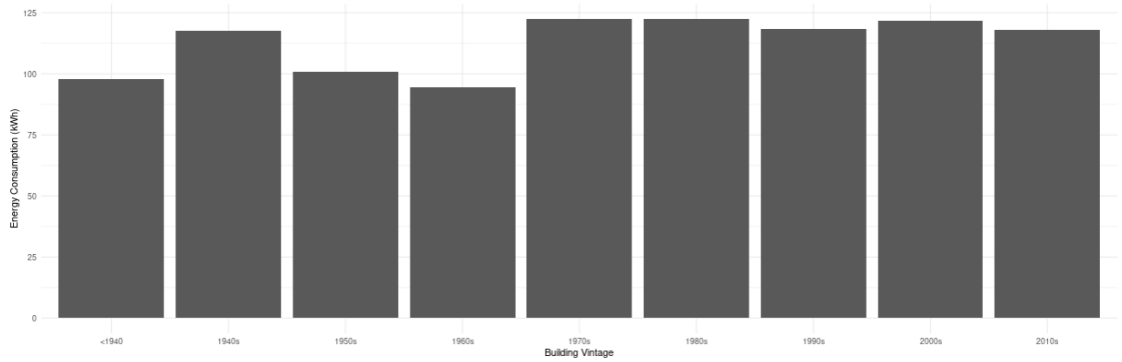
The Scatter plot represents the relationship between temperature and energy usage. The data points, which are denser around and above 30°C, suggest a significant positive correlation where higher temperatures lead to increased energy consumption. This pattern is likely attributed to greater use of cooling appliances such as air conditioners on hotter days. The plot reveals that as the temperature approaches and exceeds 30°C, the concentration of data points becomes more pronounced, illustrating a sharp increase in energy usage in response to rising temperatures. This insight highlights the impact of temperature on energy demand and can be crucial for energy management strategies, especially in designing systems to cope with peak load conditions during warm weather periods.

3. Heatmap of Energy Consumption by Square Feet and Bedrooms



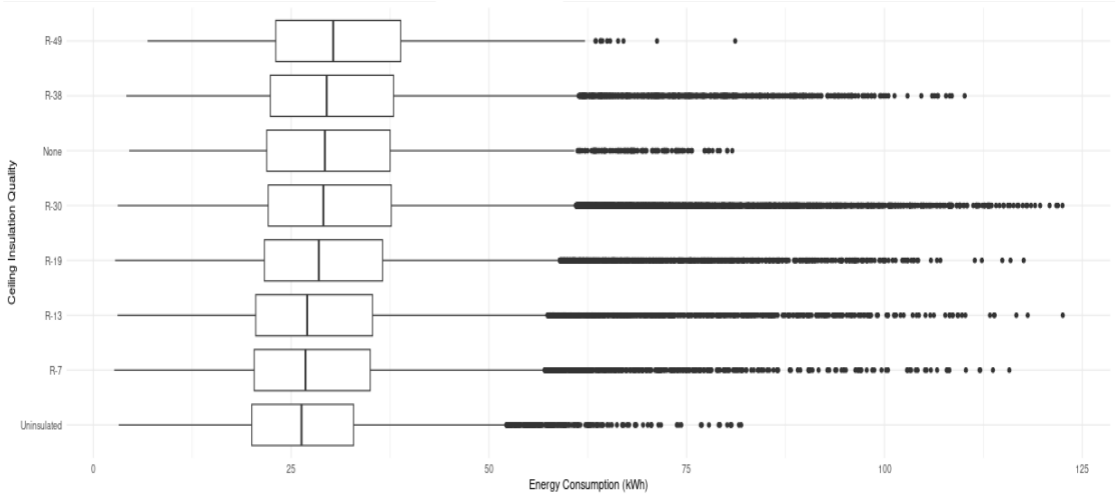
The heatmap visualizes energy consumption based on property size and number of bedrooms, revealing that larger properties and those with more bedrooms tend to consume more energy. As shown, energy usage increases from lighter purple (30 kWh) to darker red (120 kWh) as both the square footage and the number of bedrooms increase. Notably, the highest energy consumption is observed in the largest properties with the most bedrooms, located in the upper right corner of the heatmap, whereas the smallest and least bedroomed properties in the lower left corner exhibit the lowest energy usage. This pattern underscores a clear relationship between increased property size and bedroom count with higher energy consumption.

4. **Energy Consumption Trends in Buildings by Decade of Construction**



The bar graph displays the energy consumption of buildings, measured in kilowatt-hours (kWh), segmented by the decade in which they were constructed, ranging from pre-1940 to the 2010s. It reveals a trend where pre-1940 buildings start with relatively high energy consumption, which slightly increases in the 1940s. A noticeable decrease in energy usage is observed in buildings from the 1950s and 1960s, possibly reflecting advancements in building technology and improvements in construction standards during these periods. However, from the 1970s onward, there is a gradual increase in energy consumption that continues through the 2000s, suggesting changes in building size, insulation standards, or increased incorporation of energy-intensive technologies. By the 2010s, energy consumption levels off, potentially indicating the stabilization of energy consumption due to the effective implementation of energy-efficient technologies and stricter building codes. This graph provides valuable insights into how construction practices and building technologies have evolved over the decades and their impact on energy consumption patterns.

5. Energy Consumption by Insulation Quality:



The boxplot graph effectively illustrates the relationship between ceiling insulation quality, represented by R-values, and residential energy consumption in kilowatt-hours (kWh). R-values, which measure the insulation's resistance to heat flow, span from uninsulated to R-49, where higher values signify better insulation. The graph reveals a clear trend: as insulation quality improves (higher R-values), there is a noticeable decrease in median energy consumption and a more compact distribution of data points, suggesting more consistent and efficient energy use. Specifically, properties with higher-quality insulation, such as those with R-38 and R-49, demonstrate significantly lower and more uniform energy usage compared to properties with no or poor insulation, which exhibit a wider range and higher extremes of energy consumption. This variability in poorly insulated homes is likely due to increased heating and cooling demands needed to compensate for inadequate thermal retention. Overall, the data emphasizes the critical role of good insulation in enhancing a home’s energy efficiency, highlighting how superior insulation can lead to reduced energy consumption and lower variability in energy use across different residential properties.

Work Contribution:

Name	**Contribution**
Mrudu Lahari Malayanur	Data cleaning and merging and Data Analysis
Uday Suhas Nakkapalli	Data Analysis and Report Work
Yash Sumit Vishwakarma	Building Shiny App and Visualizations
Shruti Kamath	Data cleaning and making PowerPoint presentation.
Aman Kumar	Data analysis and merging

Recommendations to Reduce the Energy Demand:

Energy-Saving Campaigns:

eSC should start running advertisements emphasizing efficiency and temperature control, particularly in the summer. Peak energy demands could be reduced by providing subsidies for thermal curtains, which reduce indoor temperatures and energy use.

Smart Home Energy Management Systems:

eSC could stand for smart home technologies like solar panels and smart thermostats in order to maximize energy use in larger homes. These houses would gain the most from this technology, which would lower overall energy use.

Implement Incentive Programs for Energy-Efficient Retrofits:

There could be significant energy savings using an incentive scheme that targets energy-efficient retrofits for homes of all ages. To address the larger energy use sometimes observed in older homes, this might include support for updating heating, cooling, and insulation systems.

Dynamic Pricing Models:

In order to promote energy use during off-peak hours and assist balance the demand on the grid, eSC might use dynamic pricing. This approach lowers the possibility of energy surges during peak times.

Insulation Upgrades:

By decreasing total demand, improved insulation lowers energy prices and helps to create a more stable grid. This can be achieved by offering subsidies for insulation upgrades.

Challenges faced.

- Extraction of data: The data files were large and was time consuming to load and execute the program. We tried to use Parquet file format throughout as needed to overcome this challenge.

- Merging data: with over 300 columns from all 3 files, understanding the necessary columns to work on for the scope of the project was challenging. We used data cleaning process to remove and format the data with values as required.

- Interpretation: Finding correlations to determine and explore the dataset was challenging. We tried to merge the data and remove unnecessary columns at the initial stages of the analysis helping us maintain the analysis necessary for the goal to restrict the scope of the project.

Conclusion:

In conclusion, this project has significantly advanced our understanding of energy consumption patterns and has provided actionable insights for eSC to enhance energy management and sustainability. Through meticulous data preparation, innovative visualizations, and the application of advanced predictive models, our team has highlighted key factors influencing residential energy use across South Carolina. The Shiny app developed as part of this project has proved to be a vital tool for stakeholders to interactively explore data and derive insights, which is crucial for making informed decisions.

The integration of weather data with energy usage information allowed us to pinpoint the effects of climatic conditions on energy consumption, particularly during the peak month of July. This integration is vital for eSC's operational strategy, especially in optimizing grid performance and preventing blackouts during high-demand periods. Moreover, our predictive models have identified trends and patterns that will aid eSC in forecasting future energy needs accurately, ensuring the efficient allocation of resources and the promotion of energy conservation measures.

Our collaboration has not only fostered a deeper understanding of energy dynamics but has also underscored the importance of data-driven decision-making in the energy sector. As we move forward, the insights gained from this project will play a crucial role in helping eSC achieve its goals of reducing costs, enhancing service reliability, and contributing to environmental sustainability. The success of this project serves as a testament to the power of teamwork and innovation in tackling complex challenges and making a positive impact on society.