Final Project-Class 687 Intro to Data Science

Group 4

# By: Eric Lee, Jade Matrone, Jeet Sanjay Bardapurkar,

# Nithish Kumar Senthil Kumar, Sandeep Kanaram Parmar

Our group worked together seamlessly to perform all necessary steps to strategize recommendations to the CEO of eSC energy. We met together several times in person to clean data, share ideas, run code and make graphs. Everyone was willing to do their part and always willing to give constructive feedback.

In-depth Exploratory Data Analysis (EDA) was at the heart of the analytical process. The team structured the raw data to unveil underlying structures and relationships. Through exploratory techniques, significant trends and outlier behaviors were identified, providing a robust foundation for subsequent analysis. Visualization played a crucial role in this phase, with the team creating a series of graphs and charts that not only captured the nuanced intricacies of the data but also facilitated an intuitive understanding of complex energy consumption patterns.

**Cleaning**:

This section details the systematic approach and methodologies employed in the data-cleaning phase of the project. The objective is to outline each step taken to refine the dataset, ensuring its reliability and suitability for creating accurate predictive models. The process encompasses the identification and elimination of redundant data, management of missing values, conversion of data types, variable significance analysis, and strategic data merging.

* All the columns in the data for houses were checked using table() function to find columns which contain only a single value.
* Following are the removed columns due to having a single value:

weight, upgrade, applicability, in.ahs\_region, in.ashrae\_iecc\_climate\_zone\_2004, in.ashrae\_iecc\_climate\_zone\_2004\_2\_a\_split, in.cec\_climate\_zone, in.census\_division, in.census\_division\_recs, in.census\_region, in.clothes\_washer\_presence, in.corridor, in.dehumidifier, in.door\_area, in.doors, in.eaves, in.electric\_vehicle, in.emissions\_electricity\_folders, in.emissions\_electricity\_units, in.emissions\_electricity\_values\_or\_filepaths, in.emissions\_fossil\_fuel\_units, in.emissions\_fuel\_oil\_values, in.emissions\_natural\_gas\_values, in.emissions\_propane\_values, in.emissions\_scenario\_names, in.emissions\_types, in.emissions\_wood\_values, in.generation\_and\_emissions\_assessment\_region, 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.geometry\_stories\_low\_rise, in.geometry\_story\_bin, in.heating\_setpoint\_has\_offset, 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\_is\_faulted, 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.interior\_shading, in.iso\_rto\_region, in.lighting\_interior\_use, in.lighting\_other\_use, in.location\_region, in.mechanical\_ventilation, in.natural\_ventilation, in.neighbors, in.overhangs, in.plug\_loads, in.radiant\_barrier, in.schedules, in.simulation\_control\_run\_period\_begin\_day\_of\_month, in.simulation\_control\_run\_period\_begin\_month, in.simulation\_control\_run\_period\_calendar\_year, in.simulation\_control\_run\_period\_end\_day\_of\_month, in.simulation\_control\_run\_period\_end\_month, in.simulation\_control\_timestep, in.solar\_hot\_water, in.state, in.units\_represented, in.water\_heater\_in\_unit, upgrade.hvac\_cooling\_efficiency

* clothes\_washer\_presence was removed because it can be inferred from the in.clothes\_washer column.
* There is redundant data between in.county and in.county\_and\_puma, so dropped in.county
* in.geometry\_stories and in.geometry\_stories\_low\_rise are columns with the same data, soin.geometry\_stories\_low\_rise was dropped.
* in.heating\_setpoint\_has\_offset is a useless column because in.heating\_setpoint\_offset\_magnitude contains same info in a more detailed manner
* in.infiltration was transformed to a numerical column because it is the air leakage rates and all values are numbers suffixed with 'ACH50' which is the unit
* in.occupants is a character column, so it was changed to int
* Iterate through the df\_houses dataframe, and capture bldg\_id and county.
* Download dataframes for both, filter out for July and join them on timestamp
* Add the data from df\_houses and append them to a large dataframe.
* After iteration, there should be a giant dataframe containing the hour by hour energy usage and weather data for all houses.
* Problems:

1. Does daylight time affect these timestamps?
2. What exactly does peak energy demand mean?
3. Can’t get arrow to access url parquet file because the link is not working.

Identifying Significant Variables for 28 different models

* Method Used: Utilized the sapply function to identify single-valued columns in the dataset.
* Findings: Detected 14 single-valued columns, find out each column containing only the value 0 with table function.

A screenshot of a computer program

Description automatically generated

* Action Taken: Removed these 14 columns, opting to retain only those columns with at least two unique values, to ensure meaningful variability in the dataset.

Checking Null Value

* Method Used: Employed the is.null function to check for null values in the dataset.
* Outcome: Checked there is no null value.

Handling NA Values

* Method Used: Applied the is.na and sum function to identify NA values.
* Findings: Uncovered 16,594,176 NA values, predominantly in columns beginning with "upgrade."
* Action Taken: Replaced NA values with the term "not installed" to preserve the integrity of these columns, as they are deemed crucial for the model.

Conversion of Character Columns to Numeric Types

* Method: Used str function to check the data types of columns and sapply function to subtract columns with character type.
* Initial Conversion: Transformed all character columns into factors using the as.factor() function.
* Final Conversion: Converted these factors into numeric values with as.numeric() function, facilitating data prediction.

**Correlation algorithm:**

The columns in the dataframe are divided into 2 different lists, one for the input variables and the other for the output variables. Following were the columns in the 2 variables:

Input columns: "in.bathroom\_spot\_vent\_hour","in.bedrooms", "in.building\_america\_climate\_zone","in.ceiling\_fan" , "in.city","in.clothes\_dryer", "in.clothes\_washer","in.cooking\_range", "in.cooling\_setpoint","in.cooling\_setpoint\_has\_offset", "in.cooling\_setpoint\_offset\_magnitude","in.cooling\_setpoint\_offset\_period", "in.county","in.county\_and\_puma", "in.dishwasher","in.ducts", "in.federal\_poverty\_level","in.geometry\_attic\_type", “in.geometry\_floor\_area","in.geometry\_floor\_area\_bin", "in.geometry\_foundation\_type","in.geometry\_garage", "in.geometry\_stories","in.geometry\_wall\_exterior\_finish", "in.geometry\_wall\_type","in.has\_pv", "in.heating\_fuel","in.heating\_setpoint", "in.heating\_setpoint\_offset\_magnitude","in.heating\_setpoint\_offset\_period", "in.hot\_water\_fixtures","in.hvac\_cooling\_efficiency", "in.hvac\_cooling\_partial\_space\_conditioning","in.hvac\_cooling\_type", "in.hvac\_has\_ducts","in.hvac\_has\_zonal\_electric\_heating", "in.hvac\_heating\_efficiency","in.hvac\_heating\_type", "in.hvac\_heating\_type\_and\_fuel","in.income", "in.income\_recs\_2015","in.income\_recs\_2020", "in.infiltration","in.insulation\_ceiling", "in.insulation\_floor","in.insulation\_foundation\_wall", "in.insulation\_rim\_joist","in.insulation\_roof", "in.insulation\_slab","in.insulation\_wall", "in.lighting" ,"in.misc\_extra\_refrigerator", "in.misc\_freezer","in.misc\_gas\_fireplace" , "in.misc\_gas\_grill","in.misc\_gas\_lighting", "in.misc\_hot\_tub\_spa","in.misc\_pool", "in.misc\_pool\_heater","in.misc\_pool\_pump", "in.misc\_well\_pump","in.occupants", "in.orientation","in.plug\_load\_diversity" , "in.puma","in.puma\_metro\_status" , "in.pv\_orientation","in.pv\_system\_size", "in.range\_spot\_vent\_hour","in.reeds\_balancing\_area" , "in.refrigerator","in.roof\_material" , "in.tenure","in.usage\_level" , "in.vacancy\_status","in.vintage" , "in.vintage\_acs","in.water\_heater\_efficiency", "in.water\_heater\_fuel","in.weather\_file\_city", "in.weather\_file\_latitude","in.weather\_file\_longitude" , "in.window\_areas","in.windows" , "upgrade.insulation\_roof","upgrade.water\_heater\_efficiency", "upgrade.infiltration\_reduction","upgrade.geometry\_foundation\_type", "upgrade.clothes\_dryer","upgrade.insulation\_ceiling", "upgrade.ducts","upgrade.hvac\_heating\_type" , "upgrade.insulation\_wall","upgrade.insulation\_foundation\_wall", "upgrade.hvac\_heating\_efficiency","upgrade.cooking\_range".

Output Columns: "out.electricity.ceiling\_fan.energy\_consumption" , "out.electricity.clothes\_dryer.energy\_consumption","out.electricity.clothes\_washer.energy\_consumption" , "out.electricity.cooling\_fans\_pumps.energy\_consumption", "out.electricity.cooling.energy\_consumption", "out.electricity.dishwasher.energy\_consumption","out.electricity.freezer.energy\_consumption" ,"out.electricity.heating\_fans\_pumps.energy\_consumption", "out.electricity.heating.energy\_consumption" ,"out.electricity.hot\_tub\_heater.energy\_consumption","out.electricity.hot\_tub\_pump.energy\_consumption" ,"out.electricity.hot\_water.energy\_consumption","out.electricity.lighting\_exterior.energy\_consumption" ,"out.electricity.lighting\_garage.energy\_consumption","out.electricity.lighting\_interior.energy\_consumption" ,"out.electricity.mech\_vent.energy\_consumption","out.electricity.plug\_loads.energy\_consumption" ,"out.electricity.pool\_heater.energy\_consumption","out.electricity.pool\_pump.energy\_consumption" ,"out.electricity.pv.energy\_consumption","out.electricity.range\_oven.energy\_consumption" ,"out.electricity.refrigerator.energy\_consumption","out.electricity.well\_pump.energy\_consumption" ,"out.natural\_gas.fireplace.energy\_consumption","out.natural\_gas.grill.energy\_consumption" , "out.natural\_gas.hot\_tub\_heater.energy\_consumption","out.natural\_gas.lighting.energy\_consumption" ,"out.natural\_gas.pool\_heater.energy\_consumption".

Variables with strong Correlation

4-

out.electricity.cooling\_fans\_pumps.energy\_consumption

[1] 0.2600942729 0.2598243833

"in.geometry\_floor\_area\_bin" "in.cooling\_setpoint"

5-

"out.electricity.cooling.energy\_consumption"

0.3227622603 0.2684728583

"in.geometry\_floor\_area\_bin" "in.cooling\_setpoint"

12

"out.electricity.hot\_water.energy\_consumption"

[1] 0.1585674880 0.1167456729 0.0966067948 0.0924045623

[1] "in.occupants"

[2] "in.vacancy\_status"

[3] "in.usage\_level"

[4] "upgrade.water\_heater\_efficiency"

13

out.electricity.lighting\_exterior.energy\_consumption

[1] 0.3523722676

[1] "in.vacancy\_status"

15

"out.electricity.lighting\_interior.energy\_consumption

[1] 0.2828608229 0.2534481429 0.1997934861

[1] "in.geometry\_floor\_area\_bin"

[2] "in.vacancy\_status"

[3] "in.heating\_setpoint"

17

out.electricity.plug\_loads.energy\_consumption

[1] 0.5247357059 0.4436238436 0.4171564158 0.2532280610

[1] "in.vacancy\_status"

[2] "in.usage\_level"

[3] "in.heating\_setpoint"

[4] "in.occupants"

21

"out.electricity.range\_oven.energy\_consumption"

[1] 0.0821182623

[1] "in.vacancy\_status"

22

"out.electricity.refrigerator.energy\_consumption"

[1] 0.6298420483 0.0361706471

[1] "in.misc\_extra\_refrigerator"

[2] "in.refrigerator"

24

"out.natural\_gas.fireplace.energy\_consumption

1] 0.7876880057 0.0564590954 0.0547139274

[1] "in.misc\_gas\_fireplace"

[2] "in.heating\_setpoint"

[3] "in.vacancy\_status"

Choosing significant variables:

* At this point, it is decided to use linear models in this project. So, a linear model is trained for each output variable with all the input variables.
* The significance code is seen for each of the input variables and the input variables with 3 stars are chosen for the final model training.
* Initially, this is done manually, but it is too time consuming and cumbersome. So, a programmatic way to parse through the model training output data was used to get a list of important inputs for each output variable.

Final data processing before training:

* The time column is used to create 2 new columns: date and hour. The date is the date of the observation in the month of July and the hour is the hour between 1 and 24 on that day.
* Hot\_tub\_heater appears twice, once for electricity and once for natural gas. So those 2 columns are added up to get a single hot\_tub\_heater column. This combined variable is then used to select the significant inputs.
* Pool heater also appears twice, once for electricity and once for natural gas. So those 2 columns are added up to get a single pool heater column. This combined variable is then used to select the significant inputs.
* The number of output variables is now 26.

Data Merging Based on Time and Date

* Method: Merged total sum of energy consumption by time and date using mutate, group by, summarise function.
* Findings: The analysis revealed distinct differences in energy consumption when comparing scenarios of original recorded temperatures with those simulating a 5-degree temperature increase, demonstrating the impact of temperature fluctuations on energy usage through comparative analysis.

**Training:**

* A linear model is created for each output variable using the lm() function.
* This took a lot of time because the system kept running out of memory. To work through this, it was required to train 3-5 models at a time, save them using the save function and clear out the memory.
* Then, the data needed to be loaded again using to train the next set of 3-5 models.
* This was repeated until all 26 models were trained.

A screenshot of a computer program

Description automatically generated

**Creating input data:**

The input data to the models was almost the same as the data used to train them. The only difference was that the Dry.Bulb.Temperature...C. was increased by 5 for all observations. This increased the average temperature by 5.

**Generating output:**

* The input data was fed into all 26 models . All 26 models generated 1 column each for the 4248240 observations. This created a dataframe of dimensions 4248240x26.
* The time column was added back into the final output.
* The output dataframe also contained many negative values which was unusual because the input values did not consist of any negative values. To deal with this, all negative numbers in the data were converted to positive.

The rows were grouped either by days and by hours depending on what kind of data was required.

Visualization Output: Ensuring the final visualizations were both accurate was paramount. The team conducted reviews of each output, refining, and adjusting as necessary to achieve the highest standards of clarity and reliability.

A graph on a black background

Description automatically generated

Showing top appliance use by day of the month.

A graph with colorful lines

Description automatically generated

Here we removed the top two appliances, cooling and plug loads so we could see more variations in the other top variables(#3-7 in our overall list). Comparing energy consumption of all 5 appliances, refrigerator and interior lighting alone consumes more energy than the others. Also, an interesting fact is that the energy consumption of cooling fan pumps and interior lighting is more volatile whereas, the energy consumption for heating water and refrigerator is more stable.

A graph with colorful lines

Description automatically generated

Energy consumption fluctuates based on daily temperatures. Around 7th of July, the energy consumption of all the appliances is significantly lower and on 11th of July you can see it spiking.

A graph with lines and lines on a black background

Description automatically generated

This graph shows the energy consumption of given variables for particular hours of the day.

A graph with lines and lines on a black background

Description automatically generated

This graph has one distinct peak, which correlates to that being the average time of day during the month of July when electricity is used at its peak.

A graph with lines and numbers

Description automatically generated

(Above and Below)These graph show the top energy consuming appliances by hour of the day. And you can see there is energy spike between the hours of 3-7pm when people are generally coming home from school and work, cooling their house and using extra appliances.

A graph with lines and colors

Description automatically generated

A graph with red and blue lines

Description automatically generated

A graph with lines and numbers

Description automatically generated

A grid of white lines on a black background

Description automatically generatedA grid of white lines on a black background

Description automatically generated

A graph showing a graph of water energy

Description automatically generated with medium confidenceA graph of a graph showing a graph of electricity

Description automatically generated with medium confidence

CODE:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Shiny App:**

**Link to the Shiny APP:** [**https://idsgrp2.shinyapps.io/Group2\_ShinyApp/**](https://idsgrp2.shinyapps.io/Group2_ShinyApp/)

**Approach:**

The Shiny App was created using the data and modeling that was prepared by our group to give an informative and visual understanding of the energy consumption for the future month of July(5 degrees warmer). We totaled hour during the day to see the daily fluctuations and by total day to see how energy usage can very based on daily weather conditions.

**Execution:**

1. Libraries

library(shiny)

library(shinyWidgets)

library(shinyjs)

These lines load the necessary libraries for building Shiny applications. Shiny is the core package, shinyWidgets provides additional UI elements, and shinyjs allows for the execution of JavaScript code within Shiny apps.

1. UI Definition

ui <- fluidPage(

)

The fluidPage function creates the user interface (UI) of the Shiny app. It is a fluid layout, which means that the UI adjusts dynamically based on the user's screen size.

1. Header Section

headerPanel(

"Energy Usage Comparison",

windowTitle = "Energy Usage Comparison App"

)

The headerPanel function defines the header section of the app, displaying the title "Energy Usage Comparison" and setting the browser window title to "Energy Usage Comparison App."

1. Navbar with Tabs

navbarPage(

"Energy App",

)

The navbarPage function creates a navigation bar with tabs. Three tabs are defined for different sections of the app: "Total Energy consumption Stats," "Total Energy consumption Stats of Significant Appliances," and "Energy consumption Stats of All Appliances."

1. Sidebar Panels and Main Panels in Each Tab

# (Similar structure for each of the first three tabs)

tabPanel("Total Energy consumption Stats",

sidebarPanel(

# Description section

p("This Shiny app compares energy usage with a 5-degree increase in temperature over a year's time based on different criteria."),

p("The images below show the comparison by hour and by day of the month.")

),

mainPanel(

# Main panel with images side by side

fluidRow(

column(6, imageOutput("image\_hour")),

column(6, imageOutput("image\_day"))

)

)

)

Each tab consists of a tabPanel with a sidebarPanel for descriptions and a mainPanel for the main content. The first three tabs display images representing energy consumption statistics, and the fourth tab includes a slider for selecting a date and a section to display daily energy consumption.

1. Footer

tags$footer(

style = "text-align: center; padding: 10px; background-color: #f0f0f0;",

"Made by IDS group 4 | Data source: Professor Saltz and Professor Anderson | All rights reserved"

)

A footer is added to the bottom of the UI using the tags$footer function. It contains information about the creators of the app, the data source, and a copyright notice. Styling is applied to center the text, add padding, and set a background color.

1. Server Logic

server <- function(input, output) {

}

The server function contains the server logic of the Shiny app. It includes reactive functions (renderImage) that specify how the images should be displayed based on the inputs.

1. Reactive Image Rendering

output$image\_hour <- renderImage({

list(src = "Energy Comparison by Hour.png",

alt = "Comparison By Hour",

width = 450, height = 400)

}, deleteFile = FALSE)

# (Similar renderImage functions for other images)

These reactive functions determine how the images are rendered in the UI. The images are specified with file paths, alt text, and dimensions. The deleteFile = FALSE argument ensures that the image files are not deleted after rendering.

1. Reactive Value for Selected Date

selected\_date <- reactive({

input$selected\_date

})

This reactive value (selected\_date) stores the selected date from the slider input in the "Daily Energy Consumption" tab.

1. Reactive Function to Display Daily Energy Consumption

output$daily\_energy\_output <- renderPrint({

daily\_energy\_data <- read.csv("daily\_total\_energy.csv")

date\_index <- as.numeric(selected\_date())

selected\_data <- subset(daily\_energy\_data, date == date\_index)

if (nrow(selected\_data) > 0) {

cat("Energy consumption for Date", daily\_energy\_data[date\_index, 1], ":",

daily\_energy\_data[date\_index, 2], "kWH", "\n")

} else {

cat("No data available for Date", date\_index, "\n")

}

})

This reactive function (renderPrint) reads the "daily\_total\_energy.csv" file, filters data for the selected date, and prints the corresponding energy consumption. It uses the verbatimTextOutput in the UI to display the result.

1. Run the Shiny App

Finally, the shinyApp function is called to launch the Shiny application, combining the UI and server logic.

**Theory:**

Integrating a Shiny App into our final project represents a pivotal step towards providing a dynamic and user-friendly interface for exploring and visualizing our comprehensive analysis of power consumption trends. Shiny, a web application framework for R, facilitates the seamless hosting of our R app on the web, offering an interactive and accessible platform for users to delve into the nuances of our research.

Our Shiny App serves as a gateway to our insights derived from an intricate analysis that combines diverse datasets, including appliance usage patterns and weather data. The integration of these datasets into our predictive models empowers us to scrutinize and predict power consumption trends with a heightened level of accuracy. This multifaceted approach allows us to discern how variations in appliance usage impact overall power consumption, especially in the context of a 5-degree temperature increase.

The visual manifestation of our findings is a key feature of the Shiny App, as it enables users to effortlessly navigate through graphical representations of the data. Graphs, charts, and other visual elements serve as powerful tools for conveying complex information in an easily digestible format. Users can dynamically interact with these visuals, adjusting parameters and inputs to observe real-time changes in power consumption predictions.

By harnessing the capabilities of Shiny, we transcend the traditional boundaries of static data presentation, fostering a truly engaging and informative user experience. The interactive nature of the Shiny App empowers users, whether they are stakeholders, policymakers, or the general public, to explore the intricate interplay between appliance usage and power consumption under varying temperature scenarios.

Furthermore, our Shiny App acts as a bridge between the intricacies of our analytical models and the practical implications of our research. Users can gain valuable insights into the potential ramifications of a 5-degree temperature increase on power consumption, allowing for informed decision-making and strategic planning.

In conclusion, the integration of our R app into Shiny not only elevates the accessibility of our research but also transforms it into a powerful tool for understanding and interpreting the intricate dynamics of power consumption trends. The Shiny App stands as a testament to our commitment to delivering actionable insights and fostering a data-driven understanding of the critical relationship between appliance usage and power consumption in the face of climate-induced temperature changes.

**Recommendations**:

Global warming is here, therefore, using past energy consumption data we have made these suggestions to help prevent future blackouts. We believe that reducing energy consumption between peak hours of the day, 3-7 PM, holding off on washing machine, dryer, dishwasher and pool pump usage can help to spread out the energy use so that everyone can remain cool during the forthcoming heat. eSC can encourage customers by advertising and social media but another way to encourage people to stop consuming as much energy during those peak periods is to add a rate increase during the hours of 3-7 PM. A second recommendation, is to have a social media campaign that explains the money LED lightbulbs can save households on a yearly basis and how they can contribute to the overall goal of helping the environment. eSC can have free giveaways of LED lighting as well. And our last recommendation, is to send out a survey to all the homes in their network that ask questions about their plug in electrical items during the summer months to so we can further make suggestions on how to reduce plug loads, which is the second highest energy draw to homes in their network. If eSC is to put these measures into place this will help to reduce their peak energy usage times so that they can continue to supply energy to all households in their network.