**Overview**

This report provides a comprehensive walkthrough of the code for an Intelligent Decision Support (IDS) system for energy data analysis. The code is divided into several sections, each serving a specific purpose. The main functionalities include:

1. **Data Merging**: Integration of static house information, energy data, and weather data.
2. **Data Summarization by Day**: Aggregation of energy and weather data on a daily basis.
3. **Data Cleaning**: Preprocessing steps to handle missing values, convert data types, and create new features.
4. **Data Exploration**: Visualizations to explore patterns and relationships in the data.
5. **Data Modelling**: Building a linear regression model to predict energy consumption.
6. **Energy Consumption Prediction for 5 Degree Warmer Temperature**: Predicting energy consumption with increased temperatures.
7. **Conclusion and Recommendations**

## Section 1: Data Merging

### Purpose:

This section aims to merge static house information, energy data, and weather data. It iterates through each house ID, reads the corresponding data, performs necessary manipulations, and merges datasets.

### Steps:

1. \*\*Loading Libraries:\*\* Essential libraries such as `arrow`, `dplyr`, and `ggplot2` are loaded to facilitate data manipulation, visualization, and analysis.

2. \*\*Reading Static House Information:\*\* The static house information, containing details about each house, is loaded from a Parquet file stored on Amazon S3.

3. \*\*Creating a List of House IDs:\*\* The unique building IDs are extracted from the static house information.

4. \*\*Data Merging Loop:\*\* A loop iterates through each house ID:

- Constructs the URL for the energy data.

- Reads the energy data from the specified URL.

- Filters the data for July 2018.

- Calculates total energy consumed.

- Adds the house ID to the data.

- Merges the energy data with static house information.

- Merges the resulting dataframe with weather data.

- Appends the merged dataframe to a list (`full\_df`).

5. \*\*Combining Dataframes:\*\* The list of dataframes is combined into one large dataframe (`data\_merged`).

6. \*\*Saving Data to Parquet:\*\* The final merged dataframe is saved as a Parquet file.

### Challenges and Solutions:

- \*\*Dynamic Data Sources:\*\* The use of dynamic URLs and data retrieval introduces potential errors. The `tryCatch` block handles errors, ensuring that the code continues execution even if an issue arises during data fetching.

- \*\*Progress Tracking:\*\* The use of a counter (`n`) and print statements helps track progress and identify potential problems.

## Section 2: Data Summarization by Day

### Purpose:

This section aims to summarize the merged data on a daily basis, creating a new dataframe called `data\_summarized`.

### Steps:

1. \*\*Reading Merged Data:\*\* The merged data is read from the Parquet file.

2. \*\*Converting Time Column:\*\* The `time` column is converted to a Date object, and a new column (`day`) is created.

3. \*\*Grouping and Summarizing Data:\*\* The data is grouped by building ID, time, and county. The code then calculates various summary statistics for energy consumption, energy production, and weather variables (e.g., temperature, wind speed).

4. \*\*Merging with Static House Information:\*\* The summarized data is merged with static house information based on the building ID.

5. \*\*Saving Summarized Data:\*\* The final summarized dataframe is saved as a Parquet file.

### Challenges and Solutions:

- \*\*Data Volume:\*\* The efficient use of the `dplyr` package for grouping and summarizing minimizes the risk of performance bottlenecks.

## Section 3: Data Cleaning

### Purpose:

This section focuses on preprocessing steps to handle missing values, convert data types, and create new features.

### Steps:

1. \*\*Reading Summarized Data:\*\* The summarized data is read from the Parquet file.

2. \*\*Creating a Copy:\*\* A copy of the data is created for further cleaning.

3. \*\*Removing Columns with Zero Variance:\*\* Columns with constant values are removed.

4. \*\*Converting Income Range Strings:\*\* A custom function (`range\_to\_mean`) is applied to convert income range strings to mean values.

5. \*\*Converting Variables to Numeric:\*\* Various columns representing numeric values are converted from factor to numeric.

6. \*\*Creating New Classification Columns:\*\* New columns (`energy\_usage\_group` and `building\_size`) are created based on specified thresholds for classification.

7. \*\*Imputing Missing Values:\*\* Missing values in numeric columns are imputed with their mean, and missing values in categorical columns are imputed with the mode.

8. \*\*Checking and Handling Non-Standard Missing Values:\*\* The code checks for non-standard missing values and replaces them with NA.

9. \*\*Calculating Percentage of Missing Data:\*\* The percentage of missing data for each column is calculated.

10. \*\*Removing Columns with High Missing Percentage:\*\* Columns with more than 70% missing values are removed.

11. \*\*Replacing NA with Mode:\*\* NA values in each column are replaced with their respective mode.

12. \*\*Saving Cleaned Data:\*\* The final cleaned dataframe is saved as a Parquet file.

### Challenges and Solutions:

- \*\*Data Heterogeneity:\*\* The code handles various data types and missing values with custom functions and careful imputation strategies.

## Section 4: Data Exploration

### Purpose:

This section focuses on creating visualizations to explore patterns and relationships in the data.

### Steps:

1. \*\*Reading Cleaned Data:\*\* The cleaned data is read from the Parquet file.

2. \*\*Fuel Types Distribution Across Cities:\*\* A bar plot visualizes the distribution of fuel types across different cities.

3. \*\*Energy Consumption in One and Two Story Buildings:\*\* A boxplot visualizes the distribution of energy consumption in one and two-story buildings.

4. \*\*Correlation Heatmap:\*\* A heatmap is created to visualize correlations among weather and energy consumption variables.

5. \*\*Average Temperature and Electricity Over Time:\*\* A line plot with dual axes shows the average temperature and electricity consumption over time.

6. \*\*Energy Consumption and Production by City:\*\* A bar plot visualizes the net energy consumption (consumption minus production) in different cities.

### Challenges and Solutions:

- \*\*Visualization Complexity:\*\* The code focuses on specific aspects, providing targeted visualizations to explore complex relationships.

## Section 5: Data Modelling

### Purpose:

This section involves building a linear regression model to predict energy consumption.

### Steps:

1. \*\*Reading Cleaned Data:\*\* The cleaned data is read from the Parquet file.

2. \*\*Handling Negative Values:\*\* Negative energy consumption values are assumed to occur when a house generates more energy than it consumes. These values are converted to zero.

3. \*\*Removing Unnecessary Columns:\*\* Columns related to building IDs and energy variables are removed.

4. \*\*Removing Columns with Zero Variance:\*\* Columns with low variability are removed.

5. \*\*Splitting Data:\*\* The data is split into training and testing sets using `createDataPartition` from the `caret` package.

6. \*\*Building Linear Regression Model:\*\* A linear regression model (`lm`) is built using the training dataset.

7. \*\*Predicting and Evaluating:\*\* The model is used to predict energy consumption on the test set, and metrics such as MAE, MSE, RMSE, and R-squared are calculated.

### Challenges and Solutions:

- \*\*Model Selection:\*\* The code selects a linear regression model as a suitable choice for understanding linear relationships in the data.

## Section 6: Energy Consumption Prediction for 5 Degree Warmer Temperature

### Purpose:

This section simulates a scenario where the temperature increases by 5 degrees and predicts the corresponding impact on energy consumption.

### Steps:

1. \*\*Reading Cleaned Data:\*\* The cleaned data is read from the Parquet file.

2. \*\*Creating a New Dataset with Increased Temperatures:\*\* The code creates a new dataset with temperatures increased by 5 units.

3. \*\*Predicting Energy Consumption:\*\* The linear regression model is used to predict energy consumption with the modified dataset.

### Challenges and Solutions:

- \*\*Simulating Temperature Increase:\*\* The code effectively simulates a temperature increase by modifying the dataset and predicting energy consumption, demonstrating adaptability to changing scenarios.

## Conclusion

The

IDS code showcases a comprehensive approach to handling dynamic data, performing summarization, cleaning, exploration, and modeling. Each section addresses specific challenges, and the code exhibits flexibility and resilience in the face of potential issues. By leveraging libraries such as `dplyr`, `ggplot2`, and `caret`, the code streamlines complex data manipulation and modeling tasks. The result is a robust IDS system that provides valuable insights into energy consumption patterns and enables predictive modeling for various scenarios.

##Recomendations

To recommend strategies for reducing energy consumption in the coming year, we leveraged insights gained from the exploratory data analysis (EDA) and modeling conducted in your IDS project. Here are some tailored recommendations based on the findings:

1. \*\*Focus on High Energy Consumers:\*\*

Input: We identified buildings in the "Very High" energy usage group

Aciton: Prioritize energy efficiency measures for these buildings to maximize impact.

2. \*\*Target Cities with High Consumption:\*\*

- We analysed cities with consistently high net energy consumption.

- Implement energy-saving initiatives and awareness campaigns in these areas.

3. \*\*Building Size Classification:\*\*

- We analysed energy consumption patterns based on building size.

- Tailor energy reduction strategies for small, medium, and large buildings.

4. \*\*Weather-Related Strategies:\*\*

- Considered the impact of weather variables on energy consumption.

- Implement weather-specific energy-saving measures, especially during extreme conditions like: adjusting the thermostat to optimal temperature, not leaving the doors open for long time

5. \*\*Renewable Energy Production:\*\*

- Identify buildings that are effective renewable energy producers.

- Explore incentives for expanding renewable energy systems in these buildings.

6. \*\*Temperature Impact Analysis:\*\*

- Investigate the relationship between temperature and energy consumption.

- Develop strategies to optimize heating and cooling systems based on temperature patterns.

7. \*\*Building-Specific Recommendations:\*\*

- Conduct detailed assessments for specific buildings with high energy consumption.

- Implement targeted improvements, such as upgrading insulation, optimizing HVAC systems, or installing energy-efficient appliances.

8. \*\*Behavioral Interventions:\*\*

- Educate occupants about energy-efficient practices.

- Encourage energy-conscious behaviors to reduce unnecessary consumption.

9. \*\*Predictive Maintenance:\*\*

- Implement predictive maintenance strategies for energy-intensive equipment.

- Proactively address issues to maintain optimal system efficiency.

10. \*\*Continuous Monitoring:\*\*

- Establish a continuous monitoring system for energy consumption.

- Use real-time data to identify anomalies and address inefficiencies promptly.

11. \*\*Collaboration and Community Engagement:\*\*

- Collaborate with local communities, businesses, and government agencies.

- Foster a collective effort to promote energy conservation practices.

12. \*\*Incentive Programs:\*\*

- Introduce incentive programs for buildings that consistently demonstrate energy efficiency.

- Recognize and reward energy-conscious behavior.

Remember, the effectiveness of these recommendations depends on the specific characteristics of the buildings, local regulations, and the willingness of occupants to adopt energy-saving practices. Continuous monitoring and periodic reassessment will be key to optimizing energy consumption reduction efforts.



