***AI\_Phase 5 :-***

***Project Documentation & Submission***

**Measuring Energy Consumption**

**Problem Statement:**

*The measurement of energy consumption is critical in understanding and optimizing energy usage in various sectors, including manufacturing sites, homes, commercial buildings, and transportation. However, the manual collection and analysis of energy consumption data can be time-consuming and error-prone. Therefore, there is a need for an automated approach to collect, analyze and visualize energy consumption data for better decision-making.*

**Abstract:**

*Over the past few decades, energy access has been a major challenge for most American countries due to increasing energy demand because of urban migration, increasing population, and enhanced standard of living. The issue of energy access in Botswana mostly affects rural settlements as the areas are usually far from the national electricity grid. This research was conducted using the Design Thinking (DT) approach to investigate energy availability and management in rural settlements. The DT methodology consists of six phases namely, understand, observe, define a point of view, ideate, prototype, and testing. Through the ‘understand’ phase, an intensive literature review was undertaken to assess the feasible renewable energy resources. In the ‘observe’ phase, the dataset was analysed using different DA tools.A ‘point of view’ (POV) statement was then created from the interview to initiate the ideation stage. The ‘Ideate’ phase involved using viable technologies to generate solutions targeted at the profile defined in the POV. A selection matrix was then used to obtain the best energy solution. Through the ‘prototyping’ phase, a concept model biogas digester was designed for rural settlements. This paper reports on the first 4 stages of DT methodology and the other stages will be presented in future as the final part of the study.*

**The Design Thinking Methodology:**

*The main aim of the study was to identify safe and reliable energy solutions for people living in rural communities in Botswana. The scope of the research is to promote renewable energy solutions to reduce overdependence on solid fuels for activities such as cooking and lighting. The Design Thinking methodology was adopted from Stanford University and Hasso Plattner Institute (HPI), Berlin, Germany. The Design Thinking Process is an unsupervised research methodology that utilizes creative and analytical thinking to solve everyday life problems.*

***Innovation***

*Explore innovative techniques such as time series analysis and machine learning models to predict future energy consumption patterns.*

**Abstract:**

*Predicting future energy consumption patterns is crucial for efficient energy management, resource allocation, and sustainability efforts. Time series analysis and machine learning models offer innovative techniques for achieving accurate energy consumption predictions. Here's an exploration of these methods:*

**1. Time Series Analysis:**

*Time series analysis involves examining historical data to identify patterns and trends. It is a fundamental technique for energy consumption prediction.*

**a. Seasonal Decomposition:**

*Break down the time series data into its constituent components, including trend, seasonality, and noise. This allows for better understanding of recurring patterns and trends in energy consumption.*

**b. Exponential Smoothing:**

*Methods like Holt-Winters exponential smoothing can be used to predict energy consumption by considering the weighted average of past observations, with different weights assigned to recent and older data points.*

**c. ARIMA (AutoRegressive Integrated Moving Average):**

*ARIMA models can capture complex dependencies in time series data, making them suitable for predicting energy consumption when the data exhibits autocorrelation and stationarity.*

**2. Machine Learning Models:**

*Machine learning models can offer more flexibility and accuracy in predicting energy consumption patterns, especially when dealing with large datasets and complex relationships.*

1. **Linear Regression:**

*Simple linear regression can be used when there is a clear linear relationship between energy consumption and factors like temperature, time of day, or historical data.*

1. **Random Forest and Gradient Boosting:**

*Ensemble techniques like Random Forest and Gradient Boosting can capture non-linear relationships and interactions between various features affecting energy consumption.*

1. **Long Short-Term Memory (LSTM) Networks:**

*LSTM is a type of recurrent neural network (RNN) that is effective for time series data. It can model long-range dependencies and is suitable for predicting energy consumption with sequential data.*

1. **Convolutional Neural Networks (CNNs):**

*CNNs, originally designed for image processing, can be adapted for time series data by treating them as one-dimensional sequences. They can capture spatial patterns in energy consumption data.*

1. **Hybrid Models***:*

*Combining multiple models, such as LSTM and CNN, can offer improved predictions by leveraging the strengths of different architectures.*

**3. Feature Engineering:**

*Effective feature engineering is crucial for machine learning models. Relevant features might include weather data (temperature, humidity), time of day, historical consumption, holidays, and economic factors.*

**4. Data Preprocessing:**

*Data preprocessing steps like normalization, scaling, and handling missing values are essential to ensure that the input data is suitable for modeling.*

**5. Cross-Validation:**

*Use cross-validation techniques to assess model performance and avoid overfitting. Time-based splitting (e.g., time series cross-validation) is crucial for maintaining temporal order.*

**6. Hyperparameter Tuning:**

*Optimize model hyperparameters to achieve the best predictive performance. Techniques like grid search or Bayesian optimization can help find optimal settings.*

**7. Monitoring and Feedback Loop:**

*Continuously monitor the model's performance and update it as new data becomes available. A feedback loop helps the model adapt to changing consumption patterns.*

**8. Forecast Uncertainty:**

*It's essential to estimate prediction uncertainty, especially for critical decision-making. Techniques like probabilistic forecasting using Bayesian methods can provide uncertainty estimates.*

**9. Interpretability:**

*Consider using interpretable machine learning models or techniques to explain why certain predictions are made, which is especially important for regulatory compliance and decision-making.*

***10. Integration with Energy Management Systems (EMS):***

*Integrate predictive models with EMS to automate responses and optimize energy consumption in real-time.*

**In summary:**

*A combination of time series analysis and machine learning models, along with proper data handling and feature engineering, can help predict future energy consumption patterns accurately. These techniques are valuable for energy providers, businesses, and governments looking to optimize energy usage and reduce costs while contributing to sustainability efforts.*

**Dataset Link:**[**https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption**](https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption)

**Assignment Notebook Submission**

**Energy Consumption:** *Refers To ALL The Energy Used In Day To Day Life. There are many factors at play when it comes to measuring the energy consumption of a country, mainly the consumption difference between a countries industries and its population.*

**Important to know***: Countries population vs industries:*

*• The energy consumption of a population is often measured using Energy intensity per capita, which we looked at a few sections above. In that section we saw that more developed countries tend to have a worse Energy-intensity per Capita, as there population is able to spend more on amenities like technology, appliances, and transportation. We also saw that the countries with the smallest energy intensity per capita were all developing countries, there populations arent able to afford many of the amenities available to more developed nations.*

*• The energy consumption of industries is often measured using Energy intensity of GDP, which we also saw a few sections above. In that section we saw that more developed countries tend to have a smaller energy intensity of GDP, while devolping countries tend to have a larger energy intensity of GDP. The main reason for this is that more developed countries can afford less energy intensive technologies and ways of production, unlike developing countries which cant, therfore leaving them stuck with more energy intensive means of production.*

**Table of Contents**

*• Cleaning Data*

*• Data Structure*

*• Analysis*

*• Conclusion*

**Categorization of different countries:**

*Categorizing countries into distinct groups based on its development is somewhat complex, and when it comes to classifying this, there is no single way (either grounded in theory or based on an objective benchmark) that is gererally accepted. The UN model, or there World Economic Situation and Prospects (WESP) report to be exact classifies every country into one of three broad categories: Developed Economies, Economy in transition, and Developing economy. This is the categorization Ill be using throughout this notebook.*

**Warning:**

*I am not a climate scientist, some things may be inacurate. This is simply just a study on a subject im interested in, allowing me to go deeper into the subject while at the same time imporving my graphing skills. All my sources are at the bottom of the notebook.*

*Column descriptions:*

*• Country - Country in question*

*• Energy\_type - Type of energy source*

*• Year - Year the data was recorded*

*• Energy\_consumption - Amount of Consumption for the specific energy source, measured (quad Btu)*

*• Energy\_production - Amount of Production for the specific energy source, measured (quad Btu)*

*• GDP - Countries GDP at purchasing power parities, measured (Billion 2015$ PPP)*

*• Population - Population of specific Country, measured (Mperson)*

*• Energy\_intensity\_per\_capita - Energy intensity is a measure of the energy inefficiency of an economy. It is calculated as units of energy per unit of capita (capita = individual person), measured (MMBtu/person)*

*• Energy\_intensity\_by\_GDP- Energy intensity is a measure of the energy inefficiency of an economy. It is calculated as units of energy per unit of GDP, measred (1000 Btu/2015$ GDP PPP)*

*• CO2\_emission - The amount of C02 emitted, measured (MMtonnes CO2)*

*# Analysis Tools*

*import numpy as np*

*import pandas as pd*

*from scipy import stats*

*from scipy.stats import norm*

*# Plotting Tools*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*import plotly.express as px*

*import plotly.graph\_objects as go*

*from plotly.subplots import make\_subplots*

*# Extra Plotting Tools Required for Bar Chart Race*

*import matplotlib.ticker as ticker*

*import matplotlib.animation as animation*

*from IPython.display import HTML*

*# Plot Design Settings*

*sns.set\_style("darkgrid", {"axes.facecolor": "#eff2f5", 'grid.color': '#c0ccd8', 'patch.edgecolor': '#B0B0B0', 'font.sans-serif': 'Verdana'})*

*sns.set\_palette('Dark2\_r')*

*plt.rc('font', size=19)*

*plt.rc('axes', titlesize=25)*

*plt.rc('axes', labelsize=20)*

*plt.rc('xtick', labelsize=17)*

*plt.rc('ytick', labelsize=17)*

*plt.rc('figure', titlesize=24)*

*# Other Tools*

*from sklearn.preprocessing import OneHotEncoder*

*# Mute warnings*

*import warnings*

*warnings.filterwarnings('ignore')*

**Cleaning Data**

*df = pd.read\_csv("../input/c02-emission-by-countrys-grouth-and-population/energy.csv")*

*df.shape*

*df.head(6)*

*# Removing extra index column*

*df = df.drop(['Unnamed: 0'], axis=1)*

*Renaming columns for simplicity*

*Some column names are quite long, im just going to shorten them to abreviations*

*df.rename(columns={'Energy\_type' : 'e\_type', 'Energy\_consumption' : 'e\_con', 'Energy\_production' : 'e\_prod'*

*, 'Energy\_intensity\_per\_capita' : 'ei\_capita', 'Energy\_intensity\_by\_GDP' : 'ei\_gdp'}, inplace=True)*

*df.head()*

*Great, much less complicated to write now*

*Renaming e\_type value names*

*Some of the values for the e\_type column are very long, im going to shorten them*

*df['e\_type'] = df['e\_type'].astype('category')*

*df['e\_type'] = df['e\_type'].cat.rename\_categories({'all\_energy\_types': 'all', 'natural\_gas':*

*'nat\_gas','petroleum\_n\_other\_liquids': 'pet/oth',*

*'renewables\_n\_other': 'ren/oth'})*

*df['e\_type'] = df['e\_type'].astype('object')*

**Data Types**

*Converting Year column to datetime dtype*

*df['Year'] = df['Year'].astype('object')*

*df.info()*

*Columns Stats*

*df.describe(include='all')*

*Features Unique Values*

*# Number of unique values in each variable*

*for var in df:*

*print(f'{var}: {df[var].nunique()}')*

*Duplicates*

*print('Number of Duplicates: {}'.format(len(df[df.duplicated()])))Cool, no duplicate values*

**Missing Values**

*There is quite a bit of data thats missing in this dataset. For example, some of the missing data is because the given country no longer exists, leaving only NaN values before/after its creation/collapse, theres is also a lot of data missing not at random (MNAR), as well as data missing at random (MAR). I will be dealing with each of these appropriatlly in this section*

*for var in df:*

*print(f'{var}: {df[var].isnull().sum()}')*

*#Missing data as white lines*

*import missingno as msno*

*msno.matrix(df,color=(0.3,0.36,0.44))*

*# Function ill be using to drop selected coutries*

*def to\_drop(list):*

*for country in list:*

*value = df[df['Country']==country].index*

*df.drop(labels=value, axis=0, inplace=True)*

*Taking care of countries that no longer exist or formed within the time period*

*Quite a bit of the Countries/Territories in this dataset no longer exist or have become another country, leaving only NaN values for the years they didnt exist. Im going to drop all the rows of these years. Below are some examples of some of these countries.*

*df[df['Country']=='Former U.S.S.R.']*

*df[df['Country']=='South Sudan']*

*For all the years that each of the countries didnt exist, a set of NaN values remain for its columns. Luckily, theres a tool we can use to simply drop all rows with a certain amount of NaN values in them, which is dropna().*

**Development Part  - 2**

***Analyzing the energy consumption data Creating visualizations***

*Analyzing energy consumption data and creating visualizations can provide valuable insights and help in decision-making and resource management. Here are the steps you can follow to analyze energy consumption data and create visualizations:*

***1. Data Collection:***

*- Gather energy consumption data from reliable sources, such as smart meters, sensors, or utility bills. Ensure the data is well-structured and includes relevant information like date, time, location, and energy consumption values.*

***2. Data Cleaning and Preprocessing:***

*- Check for missing or erroneous data and handle them appropriately, either by imputing missing values or removing outliers.*

*- Convert data types, ensure consistency, and format dates and times correctly.*

***3. Data Exploration:***

*- Calculate basic statistics to understand the data's characteristics, such as mean, median, standard deviation, and percentiles.*

*- Create summary tables to get an overview of energy consumption patterns.*

***4. Time-Series Analysis (if applicable):***

*- If your data involves time series, perform time-series analysis, including trend analysis, seasonality decomposition, and autocorrelation to identify patterns.*

***5. Data Visualization:***

*- Use data visualization tools and libraries (e.g., Python with Matplotlib, Seaborn, or R with ggplot2) to create various types of charts and plots to represent the data. Some common visualizations for energy consumption data include:*

*- Line charts to show consumption trends over time.*

*- Bar charts for comparing consumption among different locations or periods.*

*- Heatmaps to display consumption patterns across time and locations.*

*- Box plots to identify outliers and distribution characteristics****.***

***6. Geospatial Visualizations (if applicable):***

*- If you have location data, create maps or geospatial visualizations to display energy consumption across different geographic regions.*

***7. Interactive Dashboards (optional):***

*- Consider building interactive dashboards using tools like Tableau, Power BI, or Plotly to allow users to explore and analyze the data themselves.*

***8. Insights and Analysis:***

*- Analyze the visualizations to draw conclusions and gain insights into energy consumption patterns. Look for trends, seasonality, anomalies, and correlations.*

***9. Hypothesis Testing (optional):***

*- If you have specific questions or hypotheses, conduct statistical tests to validate your findings.*

***10. Reporting and Communication:***

*- Create a report or presentation summarizing your findings and insights. Make your visualizations and analysis easily understandable for stakeholders.*

***11. Future Recommendations:***

*- Based on your analysis, provide recommendations for energy conservation, optimization, or other relevant actions.*

***12. Data Updates and Monitoring:***

*- If your data is continuously collected, set up a process for regular updates, monitoring, and real-time visualization if needed.*

*Remember that the choice of visualization types and analysis techniques should be based on the specific goals and nature of your energy consumption data. The process may vary depending on the scale and complexity of the data and the tools available to you****.***

***CODE :***

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*import warnings*

*​warnings.filterwarnings("ignore", category=UserWarning)*

*​from sklearn.model\_selection import train\_test\_split*

*from sklearn.preprocessing import StandardScaler*

*from sklearn.svm import SVR*

*from sklearn.metrics import mean\_squared\_error, r2\_score*

*​*

*RED = "\033[91m"*

*GREEN = "\033[92m"*

*YELLOW = "\033[93m"*

*BLUE = "\033[94m"*

*RESET = "\033[0m"*

*​df = pd.read\_csv("/kaggle/input/hourly-energy-consumption/AEP\_hourly.csv")*

*​df["Datetime"] = pd.to\_datetime(df["Datetime"])*

***​***

***# DATA CLEANING***

*print(BLUE + "\nDATA CLEANING" + RESET)*

*# --- Check for missing values*

*missing\_values = df.isnull().sum()*

*print(GREEN + "Missing Values : " + RESET)*

*print(missing\_values)*

*# --- Handle missing values*

*df.dropna(inplace=True)*

*# --- Check for duplicate values*

*duplicate\_values = df.duplicated().sum()*

*print(GREEN + "Duplicate Values : " + RESET)*

*print(duplicate\_values)*

*# --- Drop duplicate values*

*df.drop\_duplicates(inplace=True)*

***​***

***# DATA ANALYSIS***

*print(BLUE + "\nDATA ANALYSIS" + RESET)*

*# --- Summary Statistics*

*summary\_stats = df.describe()*

*print(GREEN + "Summary Statistics : " + RESET)*

*print(summary\_stats)*

*​*

*# SUPPORT VECTOR MODELLLING*

*print(BLUE + "\nMODELLING" + RESET)*

*# Reduce the dataset size for faster training*

*df = df.sample(frac=0.2, random\_state=42)*

*# Split the data into features (Datetime) and target (AEP\_MW)*

*X = df[["Datetime"]]*

*y = df["AEP\_MW"]*

*# Split the data into training and testing sets*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(*

*X, y, test\_size=0.2, random\_state=42*

*)*

*# Preprocess the features (Datetime) to extract the day of the year*

*X\_train["DayOfYear"] = X\_train["Datetime"].dt.dayofyear*

*X\_test["DayOfYear"] = X\_test["Datetime"].dt.dayofyear*

*# Convert X\_train and X\_test to NumPy arrays*

*X\_train = X\_train["DayOfYear"].values.reshape(-1, 1)*

*X\_test = X\_test["DayOfYear"].values.reshape(-1, 1)*

*# Standardize the data*

*scaler = StandardScaler()*

*X\_train\_scaled = scaler.fit\_transform(X\_train)*

*X\_test\_scaled = scaler.transform(X\_test)*

*# Create an SVR (Support Vector Regression) model with a linear kernel*

*svr = SVR(kernel="linear", C=1.0)*

*# Train the SVR model*

*svr.fit(X\_train\_scaled, y\_train)*

*# Predict on the test set*

*y\_pred = svr.predict(X\_test\_scaled)*

*# Evaluate the model*

*mse = mean\_squared\_error(y\_test, y\_pred)*

*r2 = r2\_score(y\_test, y\_pred)*

*print(f"Mean Squared Error: {mse}")*

*print(f"R-squared: {r2}")*

*# Plot the actual vs. predicted values*

*plt.figure(figsize=(10, 6))*

*plt.scatter(X\_test, y\_test, color="b", label="Actual")*

*plt.scatter(X\_test, y\_pred, color="r", label="Predicted")*

*plt.xlabel("Day of the Year")*

*plt.ylabel("Energy Consumption (MW)")*

*plt.title("SVR Model: Actual vs. Predicted")*

*plt.legend()*

*plt.grid()*

*plt.show()*

***​***

***# DATA VISUALIZATION***

*print(BLUE + "\nDATA VISUALIZATION" + RESET)*

*# --- Line plot*

*print(GREEN + "LinePlot : " + RESET)*

*plt.figure(figsize=(10, 6))*

*sns.lineplot(data=df, x="Datetime", y="AEP\_MW")*

*plt.xlabel("Datetime")*

*plt.ylabel("Energy Consumption (MW)")*

*plt.title("Energy Consumption Over Year")*

*plt.grid()*

*plt.show()*

*# --- Histogram*

*print(GREEN + "Histogram : " + RESET)*

*plt.figure(figsize=(10, 6))*

*plt.hist(*

*df["AEP\_MW"],*

*bins=100,*

*histtype="barstacked",*

*edgecolor="white",*

*)*

*plt.xlabel("AEPMW")*

*plt.ylabel("Frequency")*

*plt.title("Histogram of MEGAWATT USAGE")*

*plt.show()*

***​***

***# SAVING THE FILE***

*df.to\_csv("/kaggle/working/cleaned\_AEP\_hourly.csv", index=False)*

*print(BLUE + "\nDATA ANALYSIS" + RESET)*

*print(GREEN + "Data Cleaned and Saved !" + RESET)*

***DATA CLEANING***

*Missing Values :*

*Datetime 0*

*AEP\_MW 0*

*dtype: int64*

*Duplicate Values :*

*0*

***DATA ANALYSIS***

***Summary Statistics :***

*Datetime AEP\_MW*

*count 121273 121273.000000*

*mean 2011-09-02 03:17:01.553025024 15499.513717*

*min 2004-10-01 01:00:00 9581.000000*

*25% 2008-03-17 15:00:00 13630.000000*

*50% 2011-09-02 04:00:00 15310.000000*

*75% 2015-02-16 17:00:00 17200.000000*

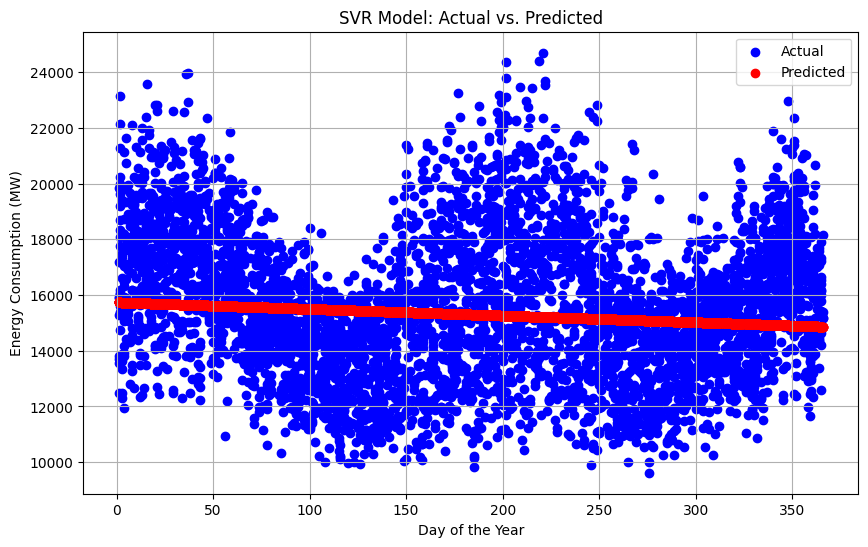
*max 2018-08-03 00:00:00 25695.000000*

*std NaN 2591.399065*

***MODELLING***

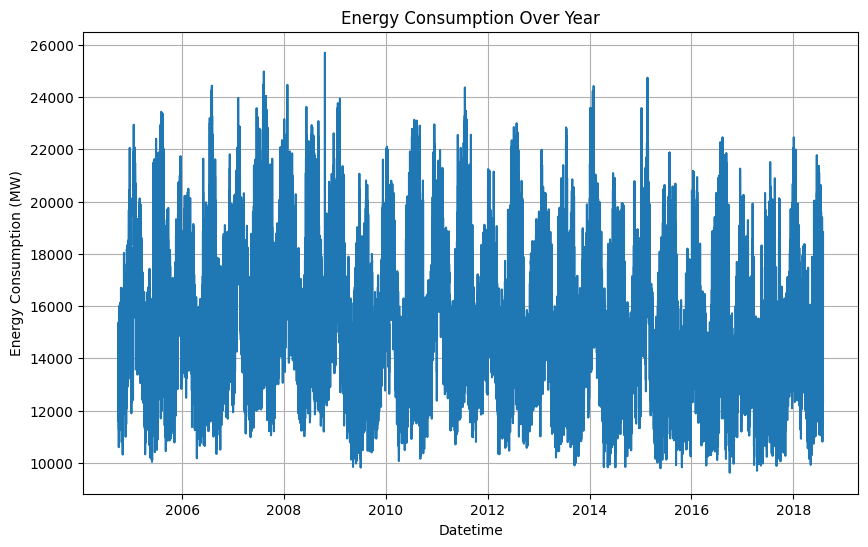
*Mean Squared Error: 6758395.805638685*

*R-squared: 0.00270160624748228*

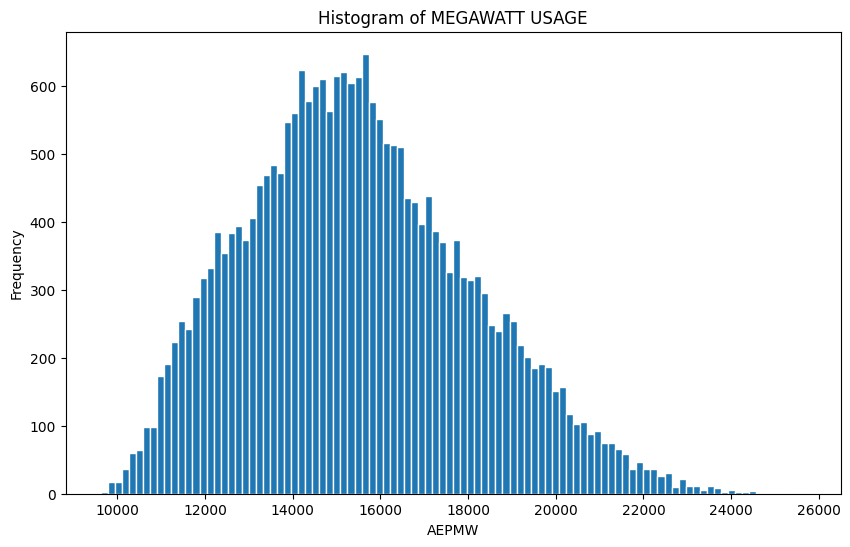


**DATA VISUALIZATION**

***LinePlot :***

******

***Histogram :***

******

***​***

**CODE FILES:-**

****

******

**README File:-**

*README.md*

[*https://github.com/Srikarnan/Srijith007.git*](https://github.com/Srikarnan/Srijith007.git)

[*git@github.com:Srikarnan/Srijith007.git*](mailto:git@github.com:Srikarnan/Srijith007.git)

*gh repo clone Srikarnan/Srijith007*

***About Dataset:-***

**Dataset Link:**[**https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption**](https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption)

**Kaggle:-**

*A subsidiary of Google, it is an online community of data scientists and machine learning engineers.*

*Kaggle allows users to find datasets they want to use in building AI models, publish datasets, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.*

*Kaggle got its start in 2010 by offering machine learning and data science competitions as well as offering a public data and cloud-based business platform for data science and AI education.*

*Most of the core staff were two*

*Anthony Goldblum*

*And*

*Jeremy Howard*

**Kaggle Rankings**

*The Kaggle Rankings page is a live leaderboard of the absolute best data scientists on Kaggle. Each category of expertise has its own leaderboard and point system. A data scientist’s profile will display their current rank, as well as the highest rank they have ever achieved for each category. A data scientist must be a expert tier or higher to be ranked for that category.*

**Points**

*While tiers and medals are permanent representations of a data scientist’s achievements, points are designed to decay over time. This keeps Kaggle’s rankings contemporary and competitive. All points awarded decay in a consistent way using the formula below:*

*In this formula, t is the number of days elapsed since the point was awarded.*

**Competitions**

*Competition points are awarded based on how well a team did in a competition, the number of members on the team, and the number of teams in the competition. Note that Community, Playground, and Getting Started competitions typically do not award points.*

*The algorithm for competition points has not changed since the 13th of May 2015:*

**Datasets**

*Dataset points are awarded based on the popularity of all public datasets a Kaggler has created. Each upvote on a dataset is initially worth 1 point, and decays based on the day the vote was cast.*

**Notebooks**

*Notebook points are awarded based on the popularity of all public notebooks a data scientist has created. Each upvote on a notebook is initially worth 1 point, and decays based on the day the vote was cast.*

**Discussion**

*Discussion points are calculated as the sum of total upvotes minus the sum of total downvotes cast on a data scientist’s topics and comments on Kaggle. Decay is applied to both upvotes and downvotes based on the day the votes were cast.*