**AI-MEASURE ENERGY CONSUMPTION**

**NAME:Harini T**

**REG NO:720421104016**

**INTRODUCTION:**

The concept of energy consumption is directly related to energy efficiency since higher consumption results in lower energy efficiency.

It's estimated that during an hour about 1,000 watts are consumed, so this measure is used to calculate the consumption of homes, businesses, or any other type of building in order to issue the corresponding bills.

There are various factors that directly influence energy consumption such as:

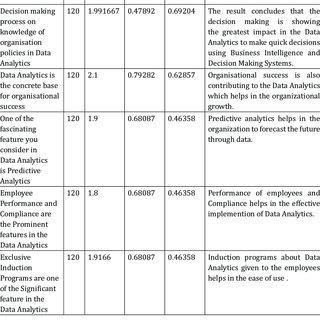
* The activity that takes place in the home or business.
* The number of people in a household or workers.
* The consumption habits of each person.
* The energy performance of household appliances.

With the right information and technology, it's possible to use energy more responsibly and efficiently. This results in a reduction in energy consumption and, therefore, in significant savings on utility bills. **Energy consumption** is the total amount of energy required for a given process and is measured in kilowatt hours (kWh). This includes the use of electricity, gas, diesel, oil, and biomass*.*

**DATASET LINK:**

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

**DATASET:**



**Here's a list of tools and software commonly used in the process:**

**1. Programming Language:**

- Python is the most popular language for machine learning due to its extensive libraries and frameworks. You can use libraries like NumPy, pandas, scikit-learn, and more.

1. **Integrated Development Environment (IDE):** - Choose an IDE for coding and running machine learning experiments. Some popular options include Jupyter Notebook, Google Colab, or traditional IDEs like PyCharm.
2. **Machine Learning Libraries:**

* You'll need various machine learning libraries, including:
* scikit-learn for building and evaluating machine learning models.
* TensorFlow or PyTorch for deep learning, if needed.
* XGBoost, LightGBM, or CatBoost for gradient boosting models.

**4. Data Visualization Tools:**

- Tools like Matplotlib, Seaborn, or Plotly are essential for data exploration and visualization.

**5. Data Preprocessing Tools:**

- Libraries like pandas help with data cleaning, manipulation, and preprocessing.

**6. Data Collection and Storage:**

- Depending on your data source, you might need web scraping tools (e.g., BeautifulSoup or Scrapy) or databases (e.g., SQLite, PostgreSQL) for data storage.

**7. Version Control:**

- Version control systems like Git are valuable for tracking changes in your code and collaborating with others.

**8. Notebooks and Documentation:**

- Tools for documenting your work, such as Jupyter Notebooks or Markdown for creating README files and documentation

**PROGRAM:**

**import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns**

**# customize the style pd.options.display.float\_format = '{:.5f}'.format pd.options.display.max\_rows = 12**

**# load the data filepath = '../input/hourly-energy-consumption/PJME\_hourly.csv' df = pd.read\_csv(filepath)**

**print("Now, you're ready for step one")**

**# turn data to datetime df = df.set\_index('Datetime') df.index = pd.to\_datetime(df.index)**

**# create the plot df.plot(style='.', figsize=(15, 5), title='PJM Energy (in MW) over time') plt.show()**

**MODEL CHOSEN FOR THE PREDICTION**

**SVR ALGORITHM**

1. **import** numpy as nm
2. **import** matplotlib.pyplot as mtp
3. **import** pandas as pd

4.

1. #importing datasets
2. data\_set= pd.read\_csv('user\_data.csv')

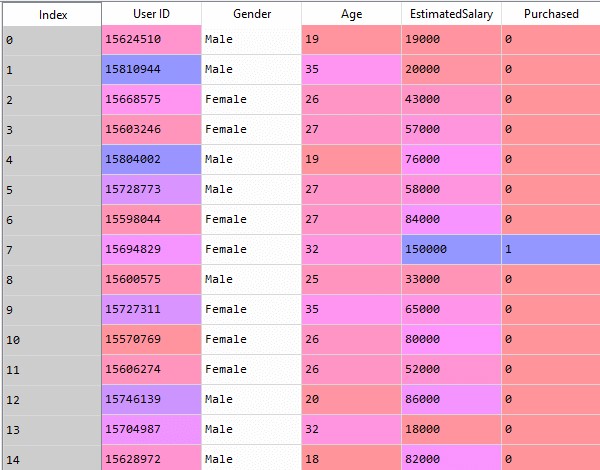
7.

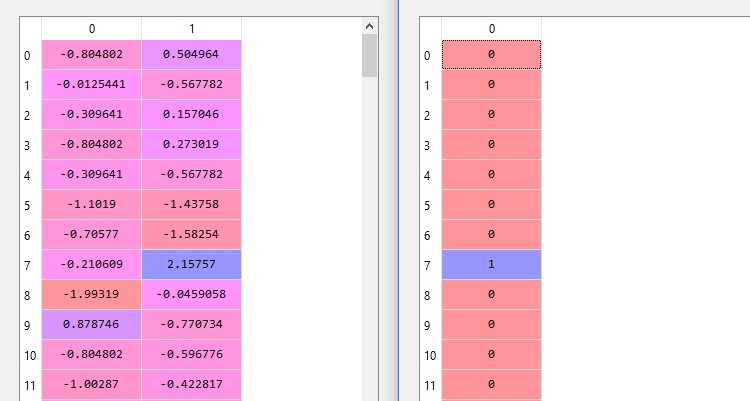
1. #Extracting Independent and dependent Variable
2. x= data\_set.iloc[:, [2,3]].values
3. y= data\_set.iloc[:, 4].values

11.

1. # Splitting the dataset into training and test set.
2. from sklearn.model\_selection **import** train\_test\_split
3. x\_train, x\_test, y\_train, y\_test= train\_test\_split(x, y, test\_size= 0.25, random\_state=0) 15. #feature Scaling
4. from sklearn.preprocessing **import** StandardScaler
5. st\_x= StandardScaler()
6. x\_train= st\_x.fit\_transform(x\_train)
7. x\_test= st\_x.transform(x\_test)

**OUTPUT:**





**Output:**

Out[8]:

SVC(C=1.0, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma='auto\_deprecated',

kernel='linear', max\_iter=-1,

probability=False, random\_state=0, shrinking=True, tol=0.001, verbose=False)

**ADAVANTAGES:**

1. Reduce Living Expenses. Saving energy usually reduces living expenses.
2. Benefits The Environment And Protects Wildlife. Less consumption benefits the environment and protects wildlife.
3. Less Power Plants.
4. Promote Health.
5. Reduce Dependence.
6. Finite Resources.
7. Make A Positive Impact.

**DISADAVANTAGES**

The environmental problems directly related to energy production and consumption include

1.air pollution,

2.climate change,

3.water pollution,

4.thermal pollution,

5.solid waste disposal.

**CONCLUSION:**

**In conclusion, the development of an automated energy consumption monitoring system with data analysis and visualization is a complex and multifaceted task. It encompasses various stages, including data collection, storage, analysis, visualization, and user interface development. Ensuring data security, scalability, and reliable automation further add to the complexity. Regular maintenance and updates are necessary to keep the system accurate and secure. This endeavor demands a multidisciplinary team with expertise in hardware, software, data science, and user interface design.**