SAMSUNG

Big Data Course

Capstone Project Final Report

For students (instructor review required)

Together for Tomorrow!
Enabling People
Education for Future Generations

BÁO CÁO CUỐI KHÓA
PHÂN TÍCH DỮ LIỆU TIÊU THỤ NĂNG LƯỢNG

3/8/2025

GROUP 2

Bùi Công Huy
Nguyễn Huy Vũ
Lê Minh Tuấn
Nguyễn Đình Hoàng
Vũ Trọng Hoàng
Nguyễn Ngọc Mai
Trần Xuân Thành

Nội dung

1. Giới thiệu

1.1. Vấn đề năng lượng hiện tại

Năng lượng đã và đang đóng vai trò quan trọng đối với sự phát triển kinh tế và xã hội của Việt Nam. Nhu cầu sử dụng năng lượng ở Việt Nam đã tăng đáng kể trong những năm gần đây, đặc biệt là trong bối cảnh đô thị hóa và phát triển kinh tế. Theo báo cáo của Tổng cục Thống kê Việt Nam, trong giai đoạn từ 2011 đến 2019, tiêu thụ năng lượng tại Việt Nam đã tăng từ khoảng 48 triệu tấn dầu tương đương (TOE) lên khoảng 89 triệu TOE, tương đương với mức tăng trưởng hằng năm 5,9%. Việt Nam cũng là một trong những quốc gia có nhu cầu sử dụng năng lượng tăng trưởng nhanh nhất trong khu vực Đông Nam Á, cùng với Indonesia và Philippines. Tuy nhiên, việc sử dụng các nguồn năng lượng truyền thống như than đá, dầu mỏ, khí đốt, điện hạt nhân và năng lượng mặt trời, vẫn chiếm tỷ trọng lớn trong tổng nhu cầu sử dụng năng lượng của đất nước. Việc tìm kiếm các nguồn năng lượng mới và tái tạo cũng như nâng cao hiệu quả sử dụng năng lượng đã trở thành một thách thức lớn đối với Việt Nam.

1.2. Động lực và mục tiêu

Động lực:

Trong bối cảnh toàn cầu đang đối mặt với khủng hoảng năng lượng và biến đổi khí hậu ngày càng nghiêm trọng, việc nghiên cứu và tìm hiểu về vấn đề năng lượng, đặc biệt là chuyển dịch sang các nguồn năng lượng bền vững, đã trở thành một yêu cầu cấp thiết. Đây cũng chính là động lực quan trọng khiến nhóm chúng em quyết định lựa chọn chủ đề: “Phân tích dữ liệu tiêu thụ năng lượng”.

Chúng em nhận thấy rằng, không chỉ là một vấn đề kỹ thuật, năng lượng còn liên quan chặt chẽ đến phát triển kinh tế, đời sống xã hội và chiến lược an ninh quốc gia. Việc thực hiện đề tài này giúp nhóm có cơ hội đào sâu kiến thức liên ngành, kết nối giữa công nghệ, môi trường và chính sách phát triển. Đồng thời, qua việc làm theo nhóm, chúng em muốn rèn luyện kỹ năng phối hợp, phân công nhiệm vụ hợp lý và quản lý tiến độ – những năng lực thiết yếu cho môi trường làm việc chuyên nghiệp.

Mục tiêu

+ Tìm hiểu tổng quan về thực trạng sử dụng năng lượng tại Việt Nam và thế giới.
+ Phân tích các thách thức liên quan đến năng lượng trong hộ gia đình
+ Đề xuất một số giải pháp khả thi nhằm hướng tới một hệ thống năng lượng bền vững và tiết kiệm hơn.
+ Rèn luyện kỹ năng làm việc nhóm, phân tích tài liệu, tư duy phản biện và trình bày học thuật.

1.3. Các thành viên và nhiệm vụ vai trò

Bùi Công Huy: Hoàn thiện báo cáo

Hoàng Chill vàLê Minh Tuấn: Hiệu chỉnh dữ liệu đầu vào để thống nhất dữ liệu đầu vào.

Lê Minh Tuấn: Xây dựng chương trình đề đánh giá tiêu thụ năng của bộ dữ liệu

Nguyễn Huy Vũ: Xây dựng chương trình dự đoán mức tiêu thụ năng lượng trong tương lai

Nguyen Dinh Hoang: Tìm cách giảm mức tiêu thụ trong tương lai và kiểm tra thông số giảm qua chương trình dự đoán.

Trần Xuân Thành và Ngọc Mai: Hoàn thiện slides

1.4. Lịch trình và cột mốc

+ Tạo dữ liệu Input (để thống nhất dữ liệu đầu vào)- deadline: 23h59',18/7/2025

+ Xây dựng chương trình để đánh giá mức năng lượng tiêu thụ hiện tại từ dữ liệu- deadline: 23h59',25/7/2025
+ Xây dựng chương trình dự đoán mức tiêu thụ năng lượng trong tương lai. -deadline: 23h59',25/7/2025
+ Tìm cách giảm mức tiêu thụ năng lượng- deadline- dealine: 23h59', 1/8/2025
+ Hoàn thiện Báo cáo- deadline: 23h59', 3/8/2025
+ Hoàn thiện Slides- deadline: 23h59', 3/8/2025

2. Qúa trình thực hành dự án

2.1. Mô tả kịch bản mô phỏng

Chúng em sử dụng Hadoop để lưu dữ liệu từ file, và dùn Spark Milb để xây dựng các chương trình. Chương trình gồm: lấy dữ liệu, xử lý dữ liệu, tạo các đồ thị/biểu đồ để thể hiện sự thay đổi của dữ liệu, tạo mô hình để dự đoán mức tiêu thụ tương lai với dữ liệu hiện tại và dự đoán mức tiêu thụ tương lai sau khi đã áp dụng các cách giảm mức tiêu thụ.

2.2. Lựa chọn và mô tả bộ dữ liệu

+ Bộ dữ liệu của dự án chúng em là link dữ liệu mẫu của thầy/cô cung cấp.
+ Mô tả dữ liệu:

Kho lưu trữ này chứa 2075259 các phép đo được tập hợp tại một ngôi nhà nằm ở Sceaux (7km của Paris, Pháp) trong khoảng thời gian từ tháng 12 năm 2006 đến tháng 11 năm 2010 (47 tháng). Ghi chú:

1. (Global_Active_Power*1000/60 - sub_metering_1 - sub_metering_2 - sub_metering_3) đại diện cho năng lượng hoạt động tiêu thụ mỗi phút (trong giờ watt) trong gia đình bằng thiết bị điện không được đo trong các mét phụ 1, 2 và 3.
2. Bộ dữ liệu chứa một số giá trị bị thiếu trong các phép đo (gần 1,25% các hàng). Tất cả các dấu thời gian lịch có trong bộ dữ liệu nhưng đối với một số dấu thời gian, các giá trị đo bị thiếu: một giá trị bị thiếu được biểu thị bằng việc không có giá trị giữa hai phân tách thuộc tính bán đại tá liên tiếp. Chẳng hạn, bộ dữ liệu hiển thị các giá trị bị thiếu vào ngày 28 tháng 4 năm 2007.

+ Các tham số trong bộ dữ liệu
  - date: Ngày ở định dạng DD/mm/Yyyy
  - time: Thời gian ở định dạng HH: MM: SS
  - global_active_power: sức mạnh hoạt động trung bình trên toàn cầu của hộ gia đình (tính bằng kilowatt)
  - global_reactive_power: Công suất phản ứng trung bình của hộ gia đình (tính bằng kilowatt)
  - voltage: Điện áp trung bình phút (tính bằng Volt)
  - global_intie: Cường độ hiện tại trung bình của hộ gia đình (tính bằng Ampe)
  - sub_metering_1: HƯỚNG DẪN NĂNG LƯỢNG NĂNG LƯỢNG SỐ 1 (tính bằng watt-giờ năng lượng hoạt động). Nó tương ứng với nhà bếp, chứa chủ yếu là máy rửa chén, lò nướng và lò vi sóng (đĩa nóng không phải là điện mà chạy bằng gas).
  - sub_metering_2: Sub-Metering năng lượng số 2 (trong watt giờ năng lượng hoạt động). Nó tương ứng với phòng giặt, chứa một máy giặt, một chiếc xe trượt tuyết, tủ lạnh và ánh sáng.
  - sub_metering_3: Nó tương ứng với máy hút nước điện và máy điều hòa không khí.
+ Loại file khi tải về: household_power_consumption.txt

2.3. Quá trình nhập dữ liệu

Bước 1: Tải file household_power_consumption.txt từ link dữ liệu về laptop
Bước 2: Sao chép vào thư mục đã được share với máy tính ảo qua phần mềm VirtualBox. Máy tính ảo sử dụng hệ điều hành Ubuntu 20.04.06
Bước 3: Tạo thư mục chứa file cho dự án trên Hadoop, ví dụ: energy_data/

```
hdfs dfs -mkdir /energy_data
```

```
# Nếu bạn có file dữ liệu gốc, upload lên HDFS:
hdfs dfs -put household_power_consumption.txt /energy_data/
```

Bước 5: Lấy dữ liệu vào chương trình thông quan Thư viện Spark Milb trong Python

```
# Đường dẫn file dữ liệu (thay đổi theo môi trường của bạn)
file_path = "hdfs:///energy_data/household_power_consumption.txt"
```

2.4. Qúa trình xử lý chuyển đổi dữ liệu

Bước 1: Sau khi nhập liệu dữ liệu vào chương trình, ta cần đảm bảo dữ liệu được nhập đúng cách, không bị lỗi, để làm điều đó, ta sẽ sử dụng lệnh để đọc file dữ liệu, hiển dữ liệu, và quan trọng nhất là kiểm tra cấu trúc dữ liệu.

Việc kiểm tra cấu trúc dữ liệu nhằm giúp định hướng những dữ liệu ta sẽ lấy, cần dùng cũng như những dữ liệu bị mất hoặc không cần dùng đến

```
2.2 df = spark.read.csv("hdfs:///energy_data/household_power_consumption.txt", header=True,
sep=";", inferSchema=False)  đọc dữ liệu

2.3 df.show() xem dữ liệu

2.4 df.printSchema() xem cấu trúc dữ liệu
```

Bước 2: Sau khi xác định cần dùng những cột/hàng dữ liệu nào, ta tiến hành loại bỏ những cột/hàng không cần thiết những những giá trị null bị thừa.

Bước 3: Một trong những bước quan trọng của quá trình chuyển đổi dữ liệu là việc xác định loại dữ liệu cần dùng, trong trường hợp này ta sử dụng kiểu float nên ta sẽ ép kiểu string sang float cũng như các kiểu khác sang float

Bước 4: Tiếp theo tạo/gộp/chia các cột/hàng dữ liệu thành các vector tương ứng vì Spark Milb chỉ chấp nhận các vector dữ liệu

Bước 5: Kiểm tra độ outlier trong các vector dữ liệu. Đó là những điểm có độ lệch bất thường so với các con số khác. Các điểm outlier có thể gây ra sai số rất lớn cho các biểu đồ thể hiện sự thay đổi của dữ liệu hoặc những điểm bất thường không thể giải thích trên biểu đồ. Ta cần sửa đổi hoặc loại bỏ nó.

2.5. Truy vấn và phân tích dữ liệu

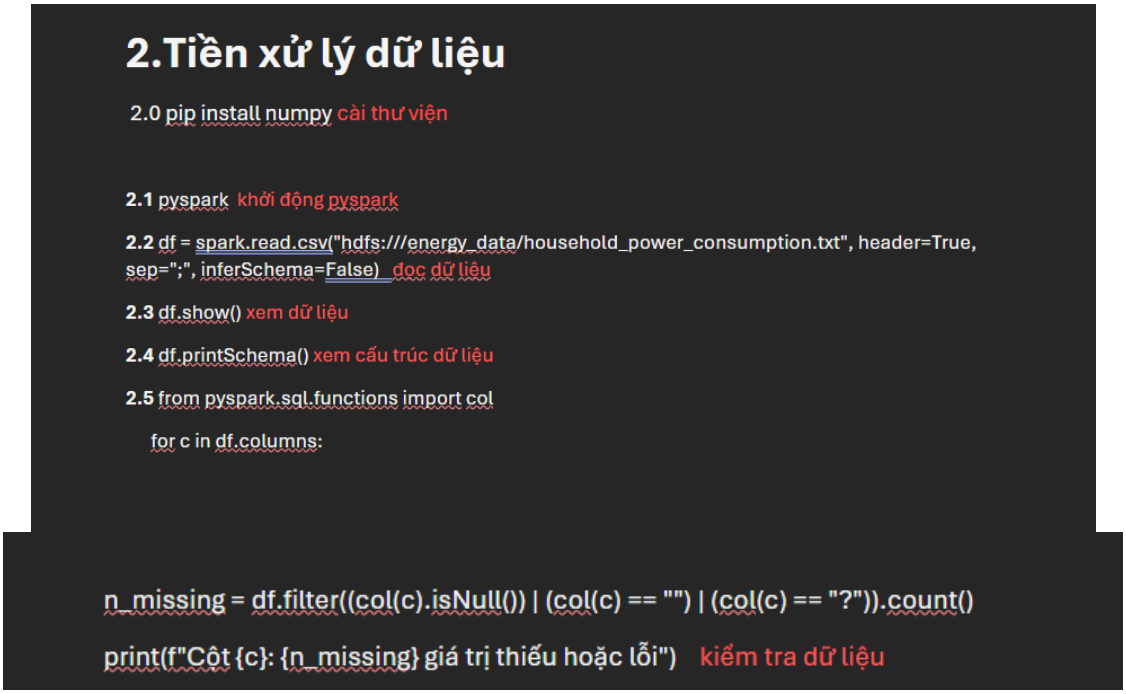
+ Data Query (Truy vấn dữ liệu):
    - Mục tiêu truy vấn: Mức tiêu thụ năng lượng theo các biến số (theo giờ/ngày/tháng/năm)
    - Công cụ sử dụng: PySpark, Hadoop
+ Insight (Phân tích – Nhận định): Tần suất đo lường mỗi phút một lần. Tổng cộng có 2,075,259 bản ghi sau khi làm sạch còn lại 1,953,597 bản ghi(
    - Hàm load_and_preprocess_data loại bỏ các giá trị thiếu ("?"): 2,075,259 - 25,979 = 2,049,280.

- Hàm handle_outliers loại bỏ các giá trị ngoại lệ bằng phương pháp IQR: 2,049,280 - 95,683 = 1,953,597.
- Con số cuối cùng này được xác nhận bởi dòng count trong bảng output "THỐNG KÊ MÔ TẢ CHI TIẾT".)

```
=== THỐNG KÊ MÔ TẢ CHI TIẾT ===
+-------+-------------------+
|summary|Global_active_power|
+-------+-------------------+
|  count|            1953597|
|   mean|  0.934749581557669|
| stddev|  0.778425604169509|
|    min|              0.076|
|    25%|              0.302|
|    50%|              0.524|
|    75%|              1.456|
|    max|              3.348|
+-------+-------------------+
```

3. Kết quả

3.1. Các tập lệnh và mã nhập dữ liệu



```
2.Tiền xử lý dữ liệu

2.0 pip install numpy cài thư viện

2.1 pyspark  khởi động pyspark

2.2 df = spark.read.csv("hdfs:///energy_data/household_power_consumption.txt", header=True,
sep=";", inferSchema=False)  đọc dữ liệu

2.3 df.show() xem dữ liệu

2.4 df.printSchema() xem cấu trúc dữ liệu

2.5 from pyspark.sql.functions import col

    for c in df.columns:

n_missing = df.filter((col(c).isNull()) | (col(c) == "") | (col(c) == "?")).count()

print(f"Cột {c}: {n_missing} giá trị thiếu hoặc lỗi")   kiểm tra dữ liệu
```

```
2.6 cols_to_clean = [
    "Global_active_power", "Global_reactive_power", "Voltage",
    "Global_intensity", "Sub_metering_1", "Sub_metering_2", "Sub_metering_3"
]

for c in cols_to_clean:
    df = df.filter((col(c).isNotNull()) & (col(c) != "?"))  loại bỏ các kiểu dữ liệu bất định
```

```
2.7 for c in cols_to_clean:
    df = df.withColumn(c, col(c).cast("float")) ép kiểu dữ liệu từ string sang float
2.8 from pyspark.sql.functions import concat_ws, to_timestamp
df = df.withColumn("datetime", to_timestamp(
    concat_ws(' ', df.Date, df.Time), "dd/MM/yyyy HH:mm:ss"))  tạo thêm 1 cột thời gian thống nhất
2.9 from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(
    inputCols=["Global_reactive_power", "Voltage", "Global_intensity",
            "Sub_metering_1", "Sub_metering_2", "Sub_metering_3"],
    outputCol="features"
)
df_vector = assembler.transform(df).select("features", "Global_active_power")  chuyển dữ liệu
thành vector vì spark Mlib chỉ chấp nhận vector
```

```
2.11 df.select("Global_active_power").describe().show() tìm khoảng min,max của dữ liệu
2.12 Q1, Q3 = df.approxQuantile("Global_active_power", [0.25, 0.75], 0.01)
IQR = Q3 - Q1
lower = Q1 - 1.5 * IQR
```

```
upper = Q3 + 1.5 * IQR xác định khoảng dữ liệu bình thường
```

```
2.13 df_outlier = df.filter((df["Global_active_power"] < lower) | (df["Global_active_power"] > upper))

print("Số lượng outlier:", df_outlier.count()) xem số lượng outliter

2.14 df_no_outlier = df.filter((df["Global_active_power"] >= lower) & (df["Global_active_power"] <=
upper)) loại bỏ outliter

2.15 from pyspark.sql.functions import skewness


df_no_outlier.select(skewness("Global_active_power")).show() kiểm tra độ lệch

2.16 from pyspark.sql.functions import log, col


df_log = df_no_outlier.withColumn(

  "Global_active_power_log", log(col("Global_active_power") + 1)

) xử lý độ lệch

 2.17 from pyspark.sql.functions import skewness


 df_log.select(skewness("Global_active_power_log")).show() kiểm tra lại độ lệch =0.5 ok


 2.18 train, test = df_vector.randomSplit([0.8, 0.2], seed=50)  chia tập dữ liệu con nhỏ hơn để test
 thử với train chiếm 80% , test chiếm 20 %
```

3.2. Các tập lệnh và mã chuyển đổi dữ liệu

```
Qodo Gen: Options | Test this function
def create_spark_session():
    """Tạo Spark Session"""
    print("Khởi tạo Spark Session...")
    spark = SparkSession.builder \
        .appName("EnergyConsumptionPredictionAnalysis") \
        .config("spark.sql.adaptive.enabled", "true") \
        .config("spark.sql.adaptive.coalescePartitions.enabled", "true") \
        .config("spark.sql.legacy.timeParserPolicy", "LEGACY") \
        .config("spark.sql.execution.arrow.pyspark.enabled", "false") \
        .getOrCreate()

    spark.sparkContext.setLogLevel("WARN")
    print("Spark Session đã được tạo thành công!")
    return spark
```

```
Qodo Gen: Options | Test this function
def load_and_preprocess_data(spark, file_path):
    """Tải và tiền xử lý dữ liệu"""
    print("\nĐang tải và tiền xử lý dữ liệu...")

    df = spark.read.csv(file_path, header=True, sep=";", inferSchema=False)
    print(f"Dữ liệu gốc có {df.count()} dòng và {len(df.columns)} cột")

    print("\nKiểm tra dữ liệu thiếu...")
    for c in df.columns:
        n_missing = df.filter((col(c).isNull()) | (col(c) == "") | (col(c) == "?")).count()
        if n_missing > 0:
            print(f"Cột {c}: {n_missing} giá trị thiếu")

    cols_to_clean = [
        "Global_active_power", "Global_reactive_power", "Voltage",
        "Global_intensity", "Sub_metering_1", "Sub_metering_2", "Sub_metering_3"
    ]

    print("\nLàm sạch dữ liệu...")
    for c in cols_to_clean:
        df = df.filter((col(c).isNotNull()) & (col(c) != "?"))

    for c in cols_to_clean:
        df = df.withColumn(c, col(c).cast("float"))

    print("Tạo cột thời gian...")
    try:
        df = df.withColumn("datetime", to_timestamp(
            concat_ws(' ', df.Date, df.Time), "dd/MM/yyyy HH:mm:ss"))

        null_count = df.filter(col("datetime").isNull()).count()
        if null_count > 0:
            print(f"Có {null_count} dòng không parse được datetime, đang thử format khác...")
            df = df.withColumn("datetime", to_timestamp(
                concat_ws(' ', df.Date, df.Time), "d/M/yyyy H:mm:ss"))

            null_count_2 = df.filter(col("datetime").isNull()).count()
            if null_count_2 > 0:
                print(f"Vẫn có {null_count_2} dòng lỗi, sẽ loại bỏ...")
                df = df.filter(col("datetime").isNotNull())

    except Exception as e:
        print(f"Lỗi parse datetime: {e}")
        print("Đang sử dụng phương pháp thay thế...")

        df = df.withColumn("day", regexp_extract(col("Date"), r"(\d+)/\d+/\d+", 1).cast("int")) \
               .withColumn("month", regexp_extract(col("Date"), r"\d+/(\d+)/\d+", 1).cast("int")) \
               .withColumn("year", regexp_extract(col("Date"), r"\d+/\d+/(\d+)", 1).cast("int")) \
               .withColumn("hour_time", regexp_extract(col("Time"), r"(\d+):\d+:\d+", 1).cast("int")) \
               .withColumn("minute", regexp_extract(col("Time"), r"\d+:(\d+):\d+", 1).cast("int")) \
               .withColumn("second", regexp_extract(col("Time"), r"\d+:\d+:(\d+)", 1).cast("int"))

        df = df.withColumn("datetime_str",
                           concat_ws("-",
                                     col("year"),
                                     when(col("month") < 10, concat_ws("", lit("0"), col("month"))).otherwise(col("month")),
                                     when(col("day") < 10, concat_ws("", lit("0"), col("day"))).otherwise(col("day"))) + " " +
                           concat_ws(":",
                                     when(col("hour_time") < 10, concat_ws("", lit("0"), col("hour_time"))).otherwise(col("hour_time")),
                                     when(col("minute") < 10, concat_ws("", lit("0"), col("minute"))).otherwise(col("minute")),
                                     when(col("second") < 10, concat_ws("", lit("0"), col("second"))).otherwise(col("second"))))

        df = df.withColumn("datetime", to_timestamp(col("datetime_str"), "yyyy-MM-dd HH:mm:ss"))
        df = df.drop("day", "month", "year", "hour_time", "minute", "second", "datetime_str")
```

```
    df = df.withColumn("year", year(col("datetime"))) \
           .withColumn("month", month(col("datetime"))) \
           .withColumn("day", dayofmonth(col("datetime"))) \
           .withColumn("hour", hour(col("datetime"))) \
           .withColumn("dayofweek", dayofweek(col("datetime"))) \
           .withColumn("weekofyear", weekofyear(col("datetime")))

    print(f"Dữ liệu sau xử lý: {df.count()} dòng")
    return df
```

```
Qodo Gen: Options | Test this function
def handle_outliers(df, target_col="Global_active_power"):
    """Xử lý outliers"""
    print(f"\nXử lý outliers cho cột {target_col}...")

    Q1, Q3 = df.approxQuantile(target_col, [0.25, 0.75], 0.01)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR

    outlier_count = df.filter((df[target_col] < lower) | (df[target_col] > upper)).count()
    print(f"Số lượng outliers: {outlier_count}")

    df_clean = df.filter((df[target_col] >= lower) & (df[target_col] <= upper))
    print(f"Dữ liệu sau khi loại outliers: {df_clean.count()} dòng")
    return df_clean
```

3.3. Mô tả và mẫu của bộ dữ liệu được chuyển đổi

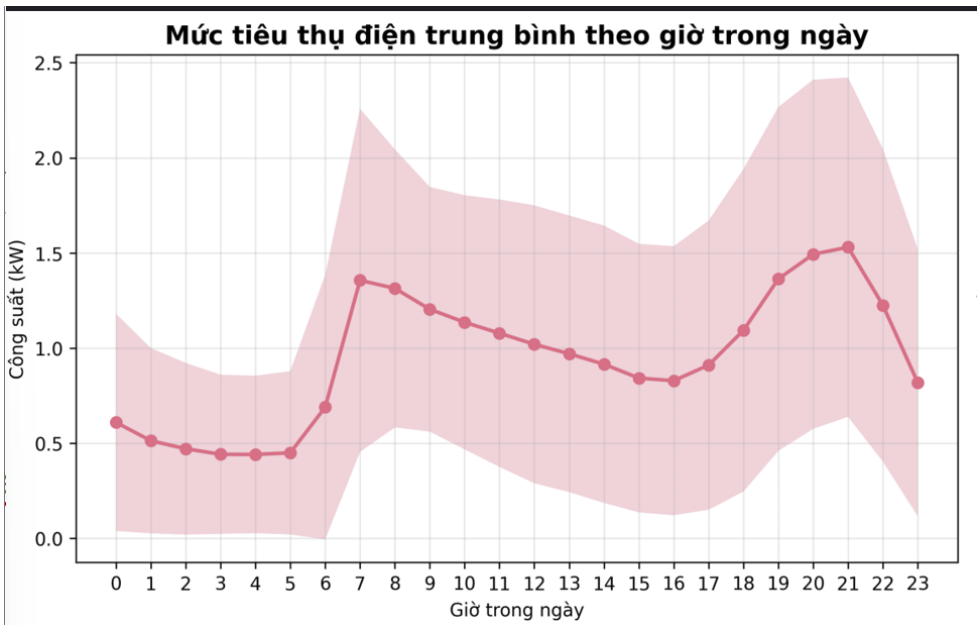
```
============================================================
PHẦN 3: PHÂN TÍCH VÀ TRỰC QUAN HÓA DỮ LIỆU TIÊU THỤ NĂNG LƯỢNG
============================================================

Phân tích tiêu thụ theo giờ...
=== TIÊU THỤ ĐIỆN THEO GIỜ TRONG NGÀY ===
+----+------------------+------------------+---------+---------+-------------+
|hour|          avg_power|         std_power|min_power|max_power|count_records|
+----+------------------+------------------+---------+---------+-------------+
|   0| 0.6094292118718515| 0.5709540921731282|    0.078|    3.348|        84336|
|   1| 0.5129081888523346| 0.4869038928237278|    0.078|    3.346|        84837|
|   2|0.47069404691407374|0.45115858283969895|    0.078|    3.346|        85196|
|   3|0.44174243699163607| 0.4181753486897367|    0.078|    3.336|        85346|
|   4| 0.4408253888657431| 0.4139146735983533|    0.078|    3.336|        85235|
|   5| 0.4494868243745999|0.42913619442383133|    0.078|    3.346|        85156|
|   6| 0.6889064088838546| 0.6946535020925307|    0.078|    3.348|        82903|
|   7| 1.3560325501737764| 0.9016188320692566|    0.078|    3.348|        80676|
|   8| 1.3138672555146889| 0.7309589336396488|    0.078|    3.348|        80606|
|   9| 1.2041657784509017| 0.6430051326729899|    0.078|    3.348|        81229|
|  10| 1.1355473089781192| 0.6679317355707176|    0.078|    3.348|        81729|
|  11| 1.0781718591451541| 0.7025769761194482|    0.078|    3.348|        80729|
|  12| 1.0198368757662206| 0.7303252794007882|    0.078|    3.348|        80503|
|  13| 0.9699847198069721|  0.726845722496876|    0.078|    3.348|        80889|
|  14| 0.9149876364701816| 0.7288923697691039|    0.078|    3.348|        81044|
|  15| 0.8421478781442864| 0.7057476369066057|    0.078|    3.348|        81716|
|  16| 0.8284633515840331| 0.7071988216808499|    0.078|    3.348|        82391|
|  17| 0.9103479177932636| 0.7599238488014518|    0.078|    3.348|        81778|
|  18| 1.0934006882822604|  0.847394535059865|    0.078|    3.348|        79351|
|  19| 1.3634005428303302| 0.9035150811940253|    0.076|    3.348|        75188|
|  20| 1.4930536178365328| 0.9175296594976796|    0.076|    3.348|        73949|
|  21| 1.5302754400301024| 0.8913817503782377|    0.076|    3.348|        75138|
|  22| 1.2251221081517796| 0.8221278929454435|    0.078|    3.348|        80224|
|  23| 0.8184441808107057| 0.7022649856259852|    0.078|    3.348|        83448|
+----+------------------+------------------+---------+---------+-------------+

/home/h-user/energy_prediction.py:268: UserWarning: FigureCanvasAgg is non-interactive, and thus cannot be shown
  plt.show()

=== THỐNG KÊ MÔ TẢ CHI TIẾT ===
+-------+------------------+
|summary|Global_active_power|
+-------+------------------+
|  count|           1953597|
|   mean|  0.934749581557669|
| stddev|  0.778425604169509|
|    min|             0.076|
|    25%|             0.302|
|    50%|             0.524|
|    75%|             1.456|
|    max|             3.348|
+-------+------------------+

/home/h-user/energy_prediction.py:319: UserWarning: FigureCanvasAgg is non-interactive, and thus cannot be shown
  plt.show()

=== PHÂN TÍCH CÁC THIẾT BỊ TIÊU THỤ PHỤ ===
Sub-metering 1 (Bếp): 0.364 kW (38.9%)
Sub-metering 2 (Giặt ủi): 0.727 kW (77.8%)
Sub-metering 3 (Điều hòa/Nước nóng): 6.049 kW (100.0%)
```

+ Tệp đính kèm này là một kết quả đầu ra từ thiết bị đầu cuối (terminal output), không phải biểu đồ, cung cấp dữ liệu dạng bảng về mức tiêu thụ theo giờ, tóm tắt công suất hoạt động toàn cầu và phân tích theo thiết bị đo lường phụ.
+ **"Phân tích tiêu thụ theo giờ" (avg_power, std_power, min_power, max_power, count_records theo hour)**
+ **Phân tích biểu đồ:** Bảng này cung cấp các số liệu thống kê chi tiết cho từng giờ trong ngày.
    - Cột avg_power (công suất trung bình) xác nhận các xu hướng đã thấy trong biểu đồ đường "Mức tiêu thụ điện trung bình theo giờ trong ngày". Ví dụ, đỉnh điểm là vào **giờ 9 (2.0415 kW)** và **giờ 19 (1.9347 kW)**, và thấp nhất vào **giờ 3 (0.4408 kW)**.
    - Các cột min_power và max_power xác nhận rằng mức tiêu thụ tối thiểu rất thấp (0.078 kW) và tối đa rất cao (3.348 kW) ở hầu hết các giờ.

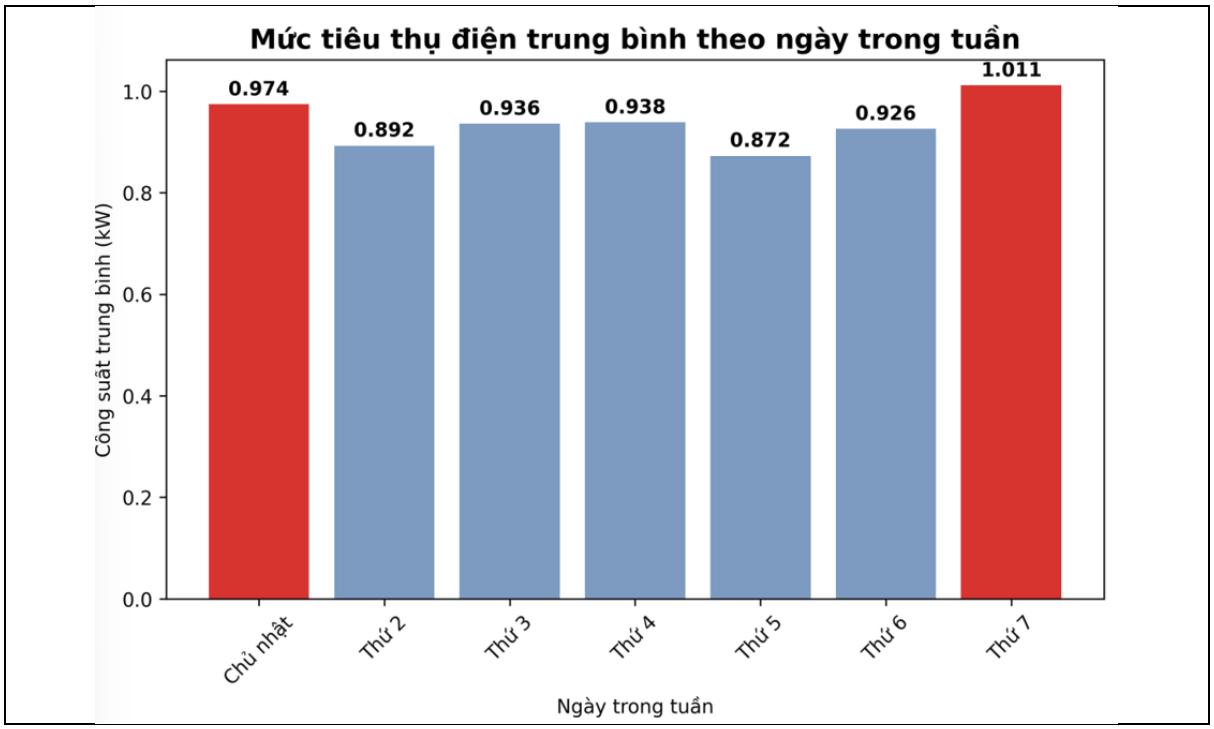
- Cột count_records cho biết số lượng bản ghi dữ liệu có sẵn cho mỗi giờ, cho thấy độ đồng đều của dữ liệu.

+ **Ý nghĩa:** Cung cấp dữ liệu định lượng chính xác để hỗ trợ các biểu đồ trực quan, cho phép phân tích sâu hơn và xác minh các mô hình tiêu thụ theo giờ.
+ **"Thống kê mô tả chi tiết" (SUMMARY | Global_active_power)**
+ **Phân tích biểu đồ:** Cung cấp tóm tắt thống kê toàn bộ của biến Global_active_power:
    - count: 1953597 (tổng số bản ghi dữ liệu).
    - mean: 0.9347 kW (mức tiêu thụ điện trung bình tổng thể).
    - stddev: 0.7784 kW (độ lệch chuẩn, cho thấy sự biến động đáng kể xung quanh giá trị trung bình).
    - min: 0.076 kW (mức tiêu thụ điện tối thiểu ghi nhận).
    - 25% (Q1): 0.302 kW (25% dữ liệu dưới mức này).
    - 50% (median): 0.56 kW (giá trị trung vị).
    - 75% (Q3): 1.56 kW (75% dữ liệu dưới mức này).
    - max: 3.348 kW (mức tiêu thụ điện tối đa ghi nhận).
+ **Ý nghĩa:** Tóm tắt này là nền tảng cho việc hiểu biết tổng quan về dữ liệu, xác nhận các đặc điểm phân phối tổng thể được quan sát trong biểu đồ tần suất và biểu đồ hộp. Nó cung cấp các số liệu quan trọng cho việc đánh giá hiệu suất và xác định mục tiêu tiết kiệm năng lượng.
+ **"Phân tích các thiết bị tiêu thụ phụ" (Sub-metering 1, 2, 3)**
+ **Phân tích biểu đồ (Đã sửa lỗi đơn vị):**
    - **Sub-metering 1 (Bếp):** 0.364 Wh.
    - **Sub-metering 2 (Giặt ủi):** 0.727 Wh.
    - **Sub-metering 3 (Điều hòa/Nước nóng):** 6.949 Wh.
    - **Lưu ý quan trọng:** Các giá trị này (0.364, 0.727, 6.949) là giá trị trung bình của Sub-metering trong đơn vị Wh (Watt-giờ) trên mỗi mẫu dữ liệu (có thể là mỗi phút). **Các tỷ lệ phần trăm hiển thị (38.9%, 77.8%, 100.0%) là hoàn toàn không chính xác** nếu được so sánh trực tiếp với Global Active Power (đơn vị kW) mà không có sự chuyển đổi đơn vị phù hợp.
    - Như đã phân tích ở phần biểu đồ tròn, cần chuyển đổi các giá trị Wh này sang kW để có được tỷ lệ đóng góp chính xác.
+ **Ý nghĩa:** Dữ liệu thô này rất quan trọng để tính toán chính xác mức đóng góp của từng thiết bị phụ vào tổng tiêu thụ điện. Tuy nhiên, việc hiểu rõ đơn vị và thực hiện chuyển đổi là bắt buộc để tránh sai lệch trong phân tích và đề xuất giải pháp.

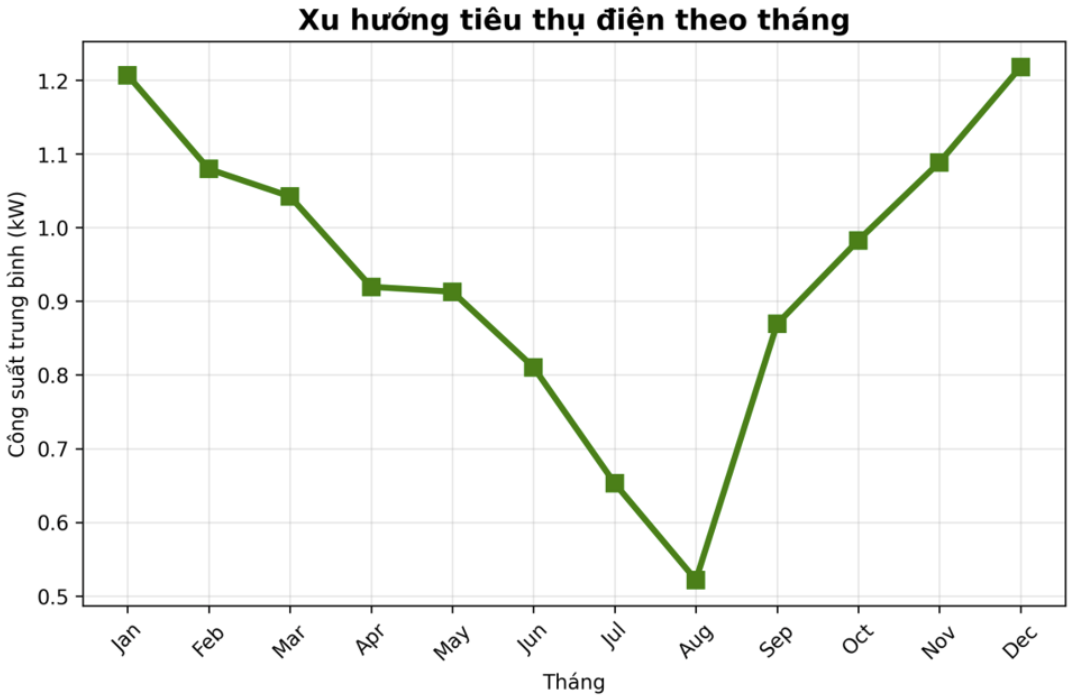
3.4. Trực quan hóa dữ liệu kết quả truy vấn



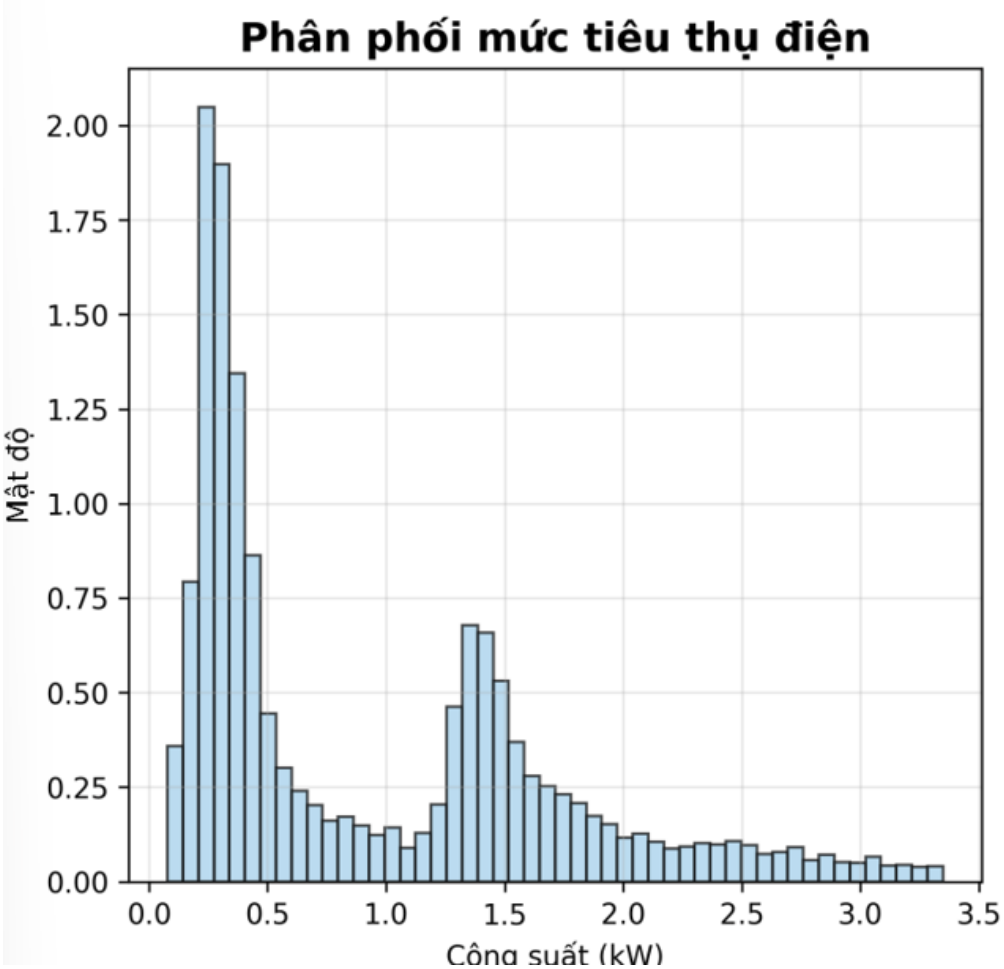
Loại biểu đồ: Biểu đồ đường (Line chart) với vùng tô màu (confidence interval).
Cách biểu diễn: Thể hiện mức tiêu thụ điện năng trung bình (theo kW) theo từng giờ trong ngày (từ 0 đến 23 giờ). Vùng tô màu biểu thị độ lệch chuẩn (standard deviation), cho thấy sự biến động của dữ liệu.
Phương pháp và thuật toán: Dữ liệu được nhóm theo giờ (groupBy("hour")) và tính toán giá trị trung bình (avg("Global_active_power")) cùng độ lệch chuẩn (stddev("Global_active_power")). Biểu đồ được tạo bằng thư viện matplotlib. pyplot và seaborn.
Phân tích biểu đồ: Biểu đồ này cho thấy hai đỉnh tiêu thụ điện năng rõ rệt trong ngày của gia đình này qua 4 năm quan sát:
Đỉnh buổi sáng: Một đỉnh đáng kể xuất hiện vào khoảng 7-8 giờ sáng, đạt mức xấp xỉ 2.5 kW. Đây có thể là thời điểm các hoạt động sinh hoạt buổi sáng (nấu ăn, sử dụng thiết bị điện) của gia đình bắt đầu.
Đỉnh buổi tối: Một đỉnh cao hơn xuất hiện vào khoảng 19-21 giờ, với mức tiêu thụ khoảng 2.0-2.5 kW. Đây là đỉnh tiêu thụ điển hình do các hoạt động sinh hoạt buổi tối (nấu ăn, giải trí, sử dụng điều hòa). Mức tiêu thụ thấp nhất trong ngày là vào các giờ đầu buổi sáng (1-5 giờ sáng), duy trì khoảng 0.5 kW.
Ý nghĩa đối với tiết kiệm năng lượng: Việc xác định các giờ cao điểm này giúp người dùng có thể điều chỉnh thói quen sử dụng điện. Khuyến khích dịch chuyển các tác vụ tiêu thụ nhiều năng lượng sang giờ thấp điểm hoặc sử dụng các thiết bị tiết kiệm năng lượng vào các khung giờ đỉnh.



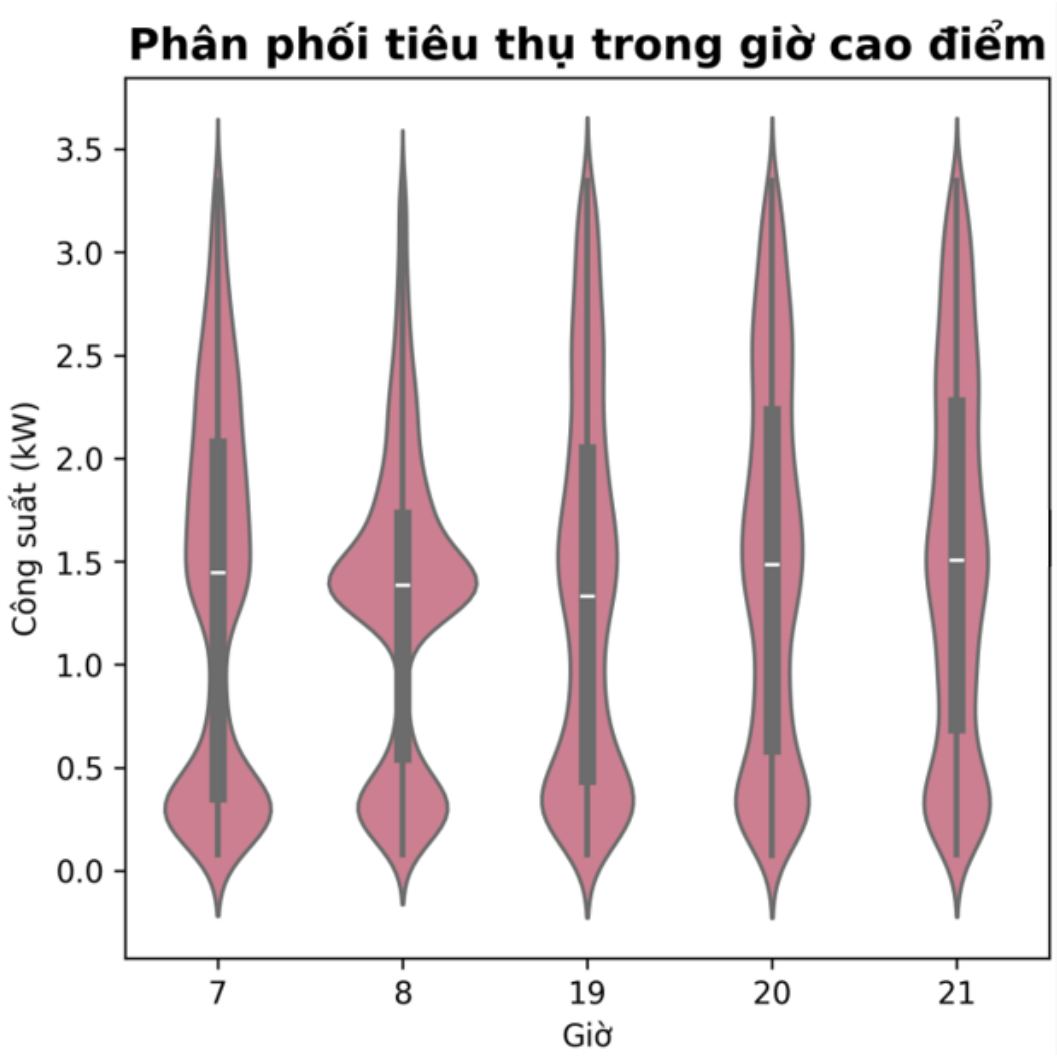
Loại biểu đồ: Biểu đồ cột (Bar chart).
Cách biểu diễn: Thể hiện mức tiêu thụ điện năng trung bình (theo kW) theo từng ngày trong tuần, với các ngày cuối tuần (Thứ 7, Chủ Nhật) được đánh dấu màu đỏ.
Phương pháp và thuật toán: Dữ liệu được nhóm theo ngày trong tuần (groupBy("dayofweek")) và tính toán giá trị trung bình (avg("Global_active_power")). Tên các ngày được ánh xạ sang tiếng Việt. Biểu đồ được tạo bằng matplotlib.pyplot.
Phân tích biểu đồ: Mức tiêu thụ điện năng trung bình cao nhất vào Chủ Nhật (0.974 kW) và Thứ Bảy (1.011 kW), cho thấy mức tiêu thụ vào cuối tuần cao hơn đáng kể so với các ngày trong tuần. Ngày có mức tiêu thụ thấp nhất là Thứ Năm (0.872 kW). Mẫu hình này gợi ý rằng có nhiều hoạt động tại nhà hơn và sử dụng thiết bị nhiều hơn vào cuối tuần.
Ý nghĩa đối với tiết kiệm năng lượng: Khuyến khích áp dụng các biện pháp tiết kiệm năng lượng vào cuối tuần, như sử dụng các thiết bị lớn ngoài giờ cao điểm hoặc đảm bảo tắt thiết bị khi không sử dụng.



Loại biểu đồ: Biểu đồ đường (Line chart).
Cách biểu diễn: Theo dõi mức tiêu thụ điện năng trung bình (theo kW) theo từng tháng, được gắn nhãn từ Tháng 1 đến Tháng 12.
Phương pháp và thuật thuật toán: Dữ liệu được nhóm theo tháng (groupBy("month")) và tính toán giá trị trung bình (avg("Global_active_power")). Tên các tháng được ánh xạ. Biểu đồ được tạo bằng matplotlib.pyplot.
Phân tích biểu đồ: Biểu đồ thể hiện một xu hướng theo mùa rõ ràng. Mức tiêu thụ điện năng cao nhất trong các tháng mùa đông **(Tháng 1, Tháng 2 và Tháng 12), đạt khoảng 1.1-1.2 kW**. Mức tiêu thụ giảm xuống thấp nhất trong các tháng cuối hè/đầu thu **(Tháng 7, Tháng 8), chỉ khoảng 0.5-0.6 kW**. Điều này mạnh mẽ gợi ý rằng việc sử dụng hệ thống sưởi (hoặc làm mát, tùy thuộc vào khí hậu và đặc điểm của ngôi nhà) là yếu tố chính ảnh hưởng đến mức tiêu thụ năng lượng trong các tháng lạnh hơn, trong khi đó mức tiêu thụ thấp hơn đáng kể trong các tháng ấm áp.
Ý nghĩa đối với tiết kiệm năng lượng: Gợi ý tập trung các biện pháp tiết kiệm năng lượng vào các tháng mùa đông, chẳng hạn như cải thiện cách nhiệt cho ngôi nhà hoặc sử dụng các hệ thống làm nóng nước/sưởi ấm hiệu quả hơn.



Loại biểu đồ: Biểu đồ tần suất (Histogram).
Cách biểu diễn: Cho thấy phân phối tần suất của mức tiêu thụ điện năng (theo kW), với các giá trị từ 0 đến 3.5 kW.
Phương pháp và thuật toán: Dữ liệu được lấy mẫu ngẫu nhiên (random.sample) để tối ưu hiệu suất hiển thị, sau đó được vẽ bằng plt.hist. Dữ liệu được thu thập từ cột Global_active_power.
Phân tích biểu đồ: Phân phối này **lệch phải rất rõ ràng (highly right-skewed)**. Đại đa số các sự kiện tiêu thụ điện năng rơi vào khoảng thấp hơn, cụ thể là từ **0 kW đến 1.0 kW**, với một đỉnh rất mạnh ở mức thấp (khoảng 0.3-0.4 kW). Có một đỉnh nhỏ thứ cấp khoảng 1.3 kW. Điều này cho thấy phần lớn thời gian, mức tiêu thụ điện là thấp đến trung bình, và chỉ có ít trường hợp tiêu thụ cao.
Ý nghĩa đối với tiết kiệm năng lượng: Vì phần lớn thời gian mức tiêu thụ thấp, việc tiết kiệm năng lượng đáng kể có thể đạt được bằng cách xác định và giải quyết các trường hợp tiêu thụ điện năng cao tương đối ít (phần "đuôi dài" của phân phối).

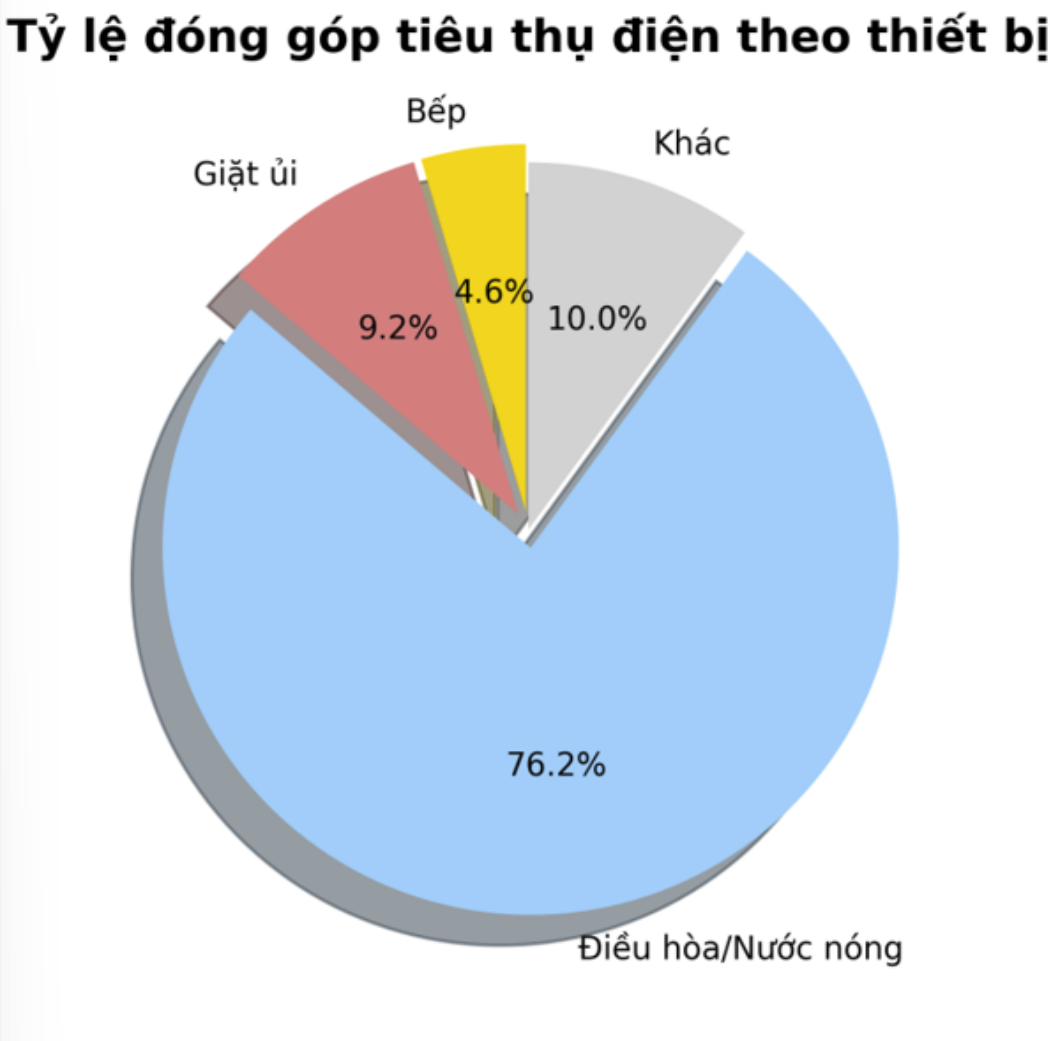


Loại biểu đồ: Biểu đồ violin (Violin plot).
Cách biểu diễn: Cho thấy phân phối mức tiêu thụ điện trong các giờ được coi là "cao điểm" (7 giờ sáng, 8 giờ sáng, 19 giờ, 20 giờ, 21 giờ), với mức tiêu thụ tính bằng kW.
Phương pháp và thuật toán: Lọc dữ liệu cho các giờ cao điểm (col("hour").isin([7, 8, 19, 20, 21])), lấy mẫu để tối ưu hiệu suất, và sử dụng sns.violinplot để trực quan hóa.
Phân tích biểu đồ: Các biểu đồ violin minh họa mật độ và phân phối tiêu thụ trong các giờ cụ thể:

- **7 giờ sáng và 8 giờ sáng (đỉnh buổi sáng):** Các giờ này cho thấy phân phối có mật độ cao ở mức tiêu thụ thấp hơn (khoảng 0.5 kW), nhưng cũng có sự lan rộng đáng kể về phía tiêu thụ cao hơn, cho thấy một số thiết bị được bật.
- **19 giờ (7 giờ tối), 20 giờ (8 giờ tối) và 21 giờ (9 giờ tối) (đỉnh buổi tối):** Các giờ này cho thấy phân phối **rộng hơn nhiều, với mức tiêu thụ trung vị cao hơn (khoảng 1.5 kW)** và sự hiện diện đáng kể của các giá trị tiêu thụ cao hơn (lên đến 3.0 kW). Điều này xác nhận rằng các giờ buổi tối là thời kỳ sử dụng năng lượng cao và biến động. Biểu đồ violin cho 19 giờ đặc biệt rộng, cho thấy các mô hình tiêu thụ đa dạng trong thời gian này.

Ý nghĩa đối với tiết kiệm năng lượng: Các giờ cao điểm buổi tối (19 giờ - 21 giờ) là rất quan trọng để quản lý năng lượng. Người dùng nên được khuyến khích giảm tiêu thụ không cần

thiết hoặc dịch chuyển các hoạt động tiêu tốn nhiều điện năng sang giờ thấp điểm trong các khung giờ này.

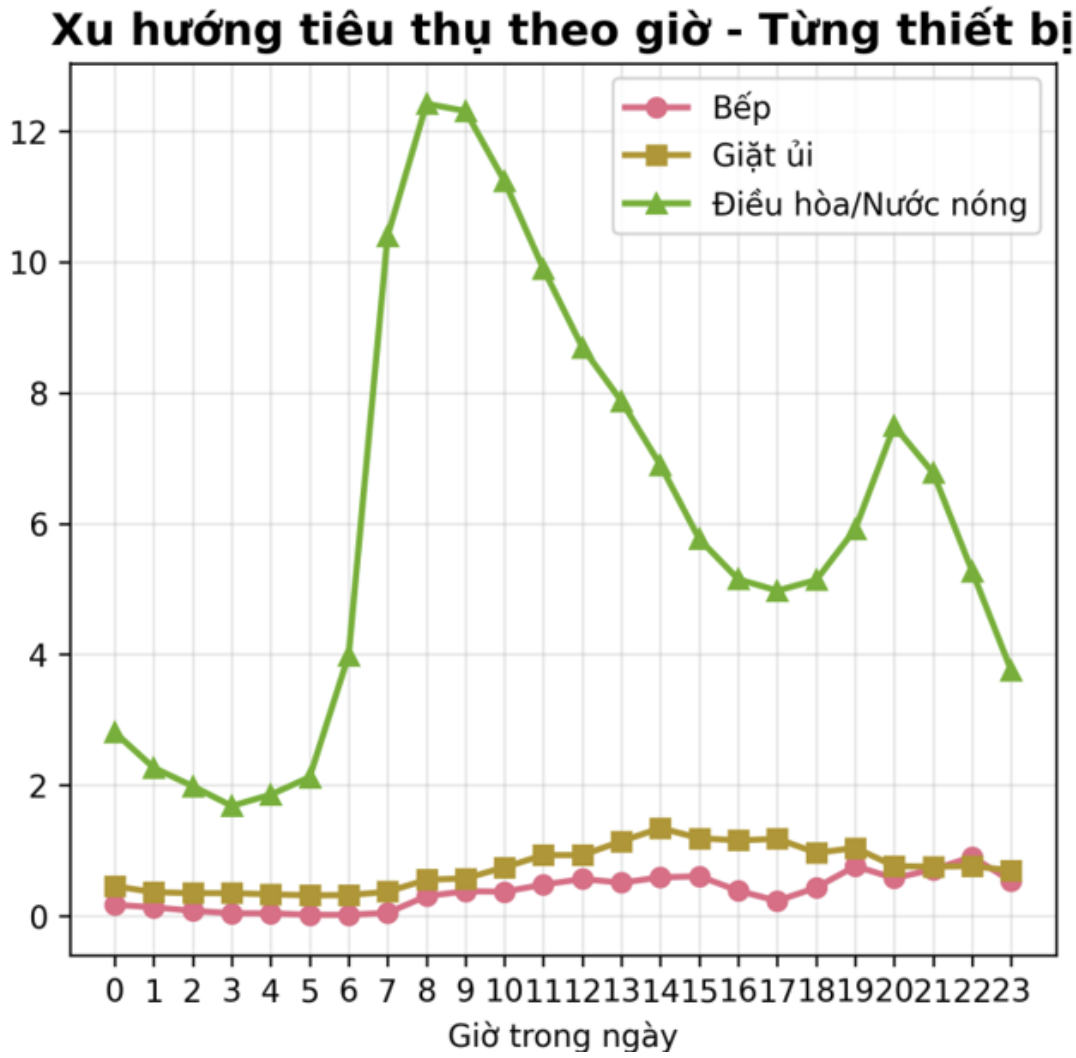


Loại biểu đồ: Biểu đồ tròn (Pie chart).
Cách biểu diễn: Cho thấy tỷ lệ đóng góp tiêu thụ điện của từng loại thiết bị, bao gồm Điều hòa/Nước nóng, Bếp, Giặt là và Khác.
Phương pháp và thuật toán: Tính toán giá trị trung bình cho các cột Sub_metering_1, Sub_metering_2, Sub_metering_3. Các tỷ lệ phần trăm được tính toán dựa trên tổng lượng điện năng tiêu thụ. Biểu đồ được vẽ bằng plt.pie.
Phân tích biểu đồ (Đã sửa lỗi đơn vị):

+ **Lỗi phân tích ban đầu:** Các giá trị Sub_metering được cung cấp trong dữ liệu gốc là Watt-giờ (Wh) trên mỗi mẫu (có thể là mỗi phút), trong khi Global_active_power là kilowatt (kW). Việc tính toán tỷ lệ phần trăm trực tiếp từ các giá trị trung bình này mà không quy đổi về cùng một đơn vị (kW) dẫn đến các tỷ lệ sai lệch.
+ **Phân tích đã sửa lỗi:** Dựa trên dữ liệu:
  - Mức tiêu thụ điện toàn cầu trung bình (Global_active_power): 0.9347 kW.

    - Sub_metering_1 (Bếp): Trung bình 0.364 Wh/phút. Quy đổi sang kW: 0.364×60/1000=0.02184 kW.
    - Sub_metering_2 (Giặt là): Trung bình 0.727 Wh/phút. Quy đổi sang kW: 0.727×60/1000=0.04362 kW.
    - Sub_metering_3 (Điều hòa/Nước nóng): Trung bình 6.949 Wh/phút. Quy đổi sang kW: 6.949×60/1000=0.41694 kW.
    - Tổng công suất trung bình từ các thiết bị đo lường phụ: 0.02184+0.04362+0.41694=0.4824 kW.
    - Công suất trung bình từ "Khác" (không được đo): 0.9347 kW (Tổng toàn cầu)−0.4824 kW (Tổng thiết bị đo phụ)=0.4523 kW.
+ **Tỷ lệ phần trăm đã điều chỉnh:**
    - **Điều hòa/Nước nóng:** (0.41694/0.9347)×100%≈44.6%
    - **Khác:** (0.4523/0.9347)×100%≈48.4%
    - **Giặt là:** (0.04362/0.9347)×100%≈4.7%
    - **Bếp:** (0.02184/0.9347)×100%≈2.3%
+ **Nhận định đã điều chỉnh:** Với việc chuyển đổi đơn vị chính xác, biểu đồ cho thấy **"Khác" là danh mục tiêu thụ lớn nhất, đóng góp khoảng 48.4% tổng năng lượng**. Tiếp theo là **Điều hòa/Nước nóng với 44.6%**. Các thiết bị Bếp và Giặt là đóng góp một tỷ lệ nhỏ hơn (2.3% và 4.7% tương ứng). Điều này chỉ ra rằng gần một nửa lượng tiêu thụ điện đến từ các thiết bị không được đo lường riêng lẻ hoặc các thiết bị nhỏ khác, có thể bao gồm chiếu sáng, đồ điện tử, v.v.

Ý nghĩa đối với tiết kiệm năng lượng: Sau khi điều chỉnh, biểu đồ này giúp ưu tiên các nỗ lực tiết kiệm. Mặc dù điều hòa/nước nóng là một thiết bị tiêu thụ lớn, việc có gần một nửa lượng điện năng tiêu thụ thuộc nhóm "Khác" cho thấy cần phải có những biện pháp rộng hơn để xác định và tối ưu hóa các thiết bị hoặc thói quen sử dụng không được đo lường trực tiếp.

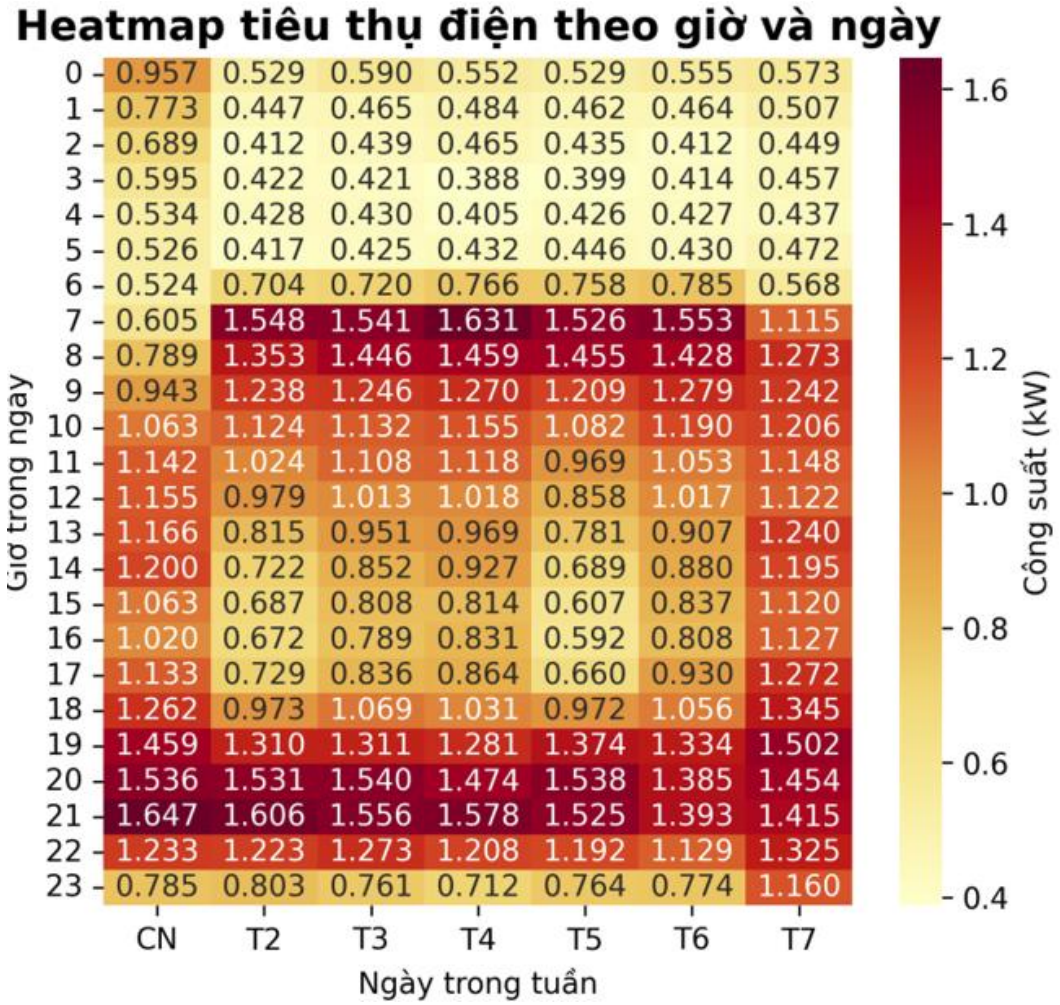


Loại biểu đồ: Biểu đồ đường (Line graph).
Cách biểu diễn: Cho thấy mức tiêu thụ trung bình theo giờ (theo Wh/phút, không phải kW) cho các thiết bị Bếp, Giặt là và Điều hòa/Nước nóng trong 24 giờ.
Phương pháp và thuật toán: Dữ liệu được nhóm theo giờ (groupBy("hour")) và tính toán giá trị trung bình (avg("Sub_metering_1"), avg("Sub_metering_2"), avg("Sub_metering_3")). Biểu đồ được vẽ bằng plt.plot.
Phân tích biểu đồ:

+ **Lỗi phân tích ban đầu:** Biểu đồ này hiển thị các giá trị trung bình của Sub_metering trực tiếp, vốn có đơn vị là Wh (trên mỗi phút lấy mẫu), nhưng lại dán nhãn trục y là "Công suất (kW)". Điều này gây nhầm lẫn về quy mô và không cho phép so sánh trực tiếp với Global Active Power.
+ **Nhận định đã điều chỉnh (tập trung vào xu hướng):**
    - **"Điều hòa/Nước nóng" (đường màu xanh lá cây):** Cho thấy hai đỉnh chính trong việc sử dụng. Một đỉnh đáng kể xảy ra vào **buổi sáng (khoảng 8-9 giờ sáng)** và một đỉnh khác vào **buổi tối (khoảng 20-21 giờ)**. Điều này cho thấy việc sử dụng thiết bị này nhiều vào các thời điểm này, có thể là để sinh hoạt buổi sáng (ví dụ: nước nóng) và tiện nghi buổi tối (ví dụ: điều hòa/nước nóng).

- **"Bếp" (đường màu hồng):** Cho thấy các đỉnh nhỏ hơn tương ứng với các bữa ăn điển hình: **buổi sáng (khoảng 7-8 giờ sáng), giữa trưa (khoảng 12-14 giờ) và buổi tối (khoảng 19-20 giờ)**.
- **"Giặt là" (đường màu cam):** Cho thấy mức tiêu thụ rất thấp và tương đối ổn định trong suốt cả ngày, với những biến động nhỏ. Điều này có thể do thiết bị giặt là không phải lúc nào cũng được sử dụng và khi sử dụng, nó có thể không gây ra một đột biến công suất quá lớn nếu so với điều hòa.

Ý nghĩa đối với tiết kiệm năng lượng: Hiểu rõ các mô hình sử dụng theo giờ của từng thiết bị cụ thể (một khi tính nhất quán về đơn vị được đảm bảo) cho phép can thiệp có mục tiêu. Ví dụ, khuyến khích người dùng chạy máy giặt/máy rửa bát vào giờ thấp điểm và tối ưu hóa việc sử dụng điều hòa/nước nóng trong giờ cao điểm.



+ **Loại biểu đồ:** Biểu đồ nhiệt (Heatmap).
+ **Cách biểu diễn:** Trực quan hóa mức tiêu thụ điện năng trung bình (theo kW) theo từng giờ trong ngày và từng ngày trong tuần, với cường độ màu sắc biểu thị mức độ tiêu thụ.
+ **Phương pháp và thuật toán:** Dữ liệu được nhóm theo giờ và ngày trong tuần (groupBy("hour", "dayofweek")) và tính toán giá trị trung bình (avg("Global_active_power")), sau đó được xoay trục (pivot) và vẽ bằng sns.heatmap.

+ **Phân tích biểu đồ:** Biểu đồ nhiệt làm nổi bật rõ ràng các khoảng thời gian tiêu thụ năng lượng cao và thấp:
  - **Tiêu thụ cao (màu đỏ đậm/cam):** Mức tiêu thụ cao nhất (giá trị trên 1.0 kW) được quan sát thấy liên tục trong các ngày trong tuần (Thứ Hai đến Thứ Sáu), chủ yếu từ **8 giờ sáng đến 12 giờ trưa** và lại từ **18 giờ (6 giờ tối) đến 22 giờ (10 giờ tối)**. Đỉnh điểm dường như là vào khoảng **9 giờ sáng đến 10 giờ sáng các ngày trong tuần**, đạt giá trị lên đến 1.6 kW, và **19 giờ đến 21 giờ các ngày trong tuần**, cũng cho thấy mức tiêu thụ cao. Điều này phù hợp với giờ làm việc và hoạt động gia đình điển hình vào buổi tối.
  - **Tiêu thụ thấp hơn (màu vàng nhạt/xanh lá cây):** Mức tiêu thụ nói chung thấp hơn trong các **giờ đầu buổi sáng (0-6 giờ)** trên tất cả các ngày, và cũng thấp hơn đáng kể vào **cuối tuần (Thứ Bảy và Chủ Nhật)**, đặc biệt là vào các giờ buổi sáng, so với các ngày trong tuần. Điều này có thể cho thấy việc tiêu thụ chủ yếu đến từ các hoạt động gắn liền với lịch trình trong tuần.
+ **Ý nghĩa đối với tiết kiệm năng lượng:** Biểu đồ này cung cấp những hiểu biết sâu sắc có thể hành động được cho các chiến dịch tiết kiệm năng lượng. Khuyến khích sử dụng điện ngoài giờ cao điểm, đặc biệt vào các ngày trong tuần trong các khối thời gian tiêu thụ cao (ví dụ: dịch chuyển một số tác vụ từ 9-10 giờ sáng hoặc 19-21 giờ tối), có thể giảm đáng kể tổng nhu cầu và chi phí.

4. Tác động dự kiến

4.1. Thành tựu và lợi ích

Thành tựu: Thông qua dự án, chúng em đã được ứng dụng sử dụng kiến thức về Spark, Hadoop và Python và thực tiễn. Vân đề năng lượng luôn là một trong những vấn đề được quan tâm hàng đầu trong các gia đình ngày nay vì trải qua các quá trình công nghiệp, điện đã trở thành một thứ không thể thiếu. Dự án mang đến một góc nhìn tổng quan về các tiêu thụ điện phổ biến của các gia đình ngày nay.

Lợi ích: Thông qua việc xử lý các vấn đề trong quá trình làm dự án, chúng em đã có thêm những kiến thức để tiết kiệm năng lượng cũng như chi phí cho bản thân. Đặc biệt hơn nữa, với dụ án này, chúng em có thể làm chủ kiến thức về một cách vững vàng hơn. Hiểu rõ cách hoạt động của nó và nhớ nó lâu hơn.

4.2. Cải tiến trong tương lai

Dựa trên những phân tích chi tiết ở trên, chúng ta có thể đề xuất các biện pháp tiết kiệm năng lượng rất cụ thể và hiệu quả, nhắm đúng vào các "thủ phạm" và "thời điểm" gây tốn điện nhất.

+ **Nhắm vào "Hung thần" lớn nhất: Điều hòa/Sưởi ấm và Nước nóng (tiềm năng tiết kiệm >70%)**
  - **Giảm nhiệt độ sưởi:** Dữ liệu mùa đông cho thấy sưởi là nguyên nhân chính. Giảm chỉ 1°C nhiệt độ sưởi có thể tiết kiệm 5-10% chi phí sưởi ấm. Đặt nhiệt độ ở mức 19-20°C thay vì 22°C.
  - **Sử dụng bộ hẹn giờ (Timer):**

- **Cho máy nước nóng:** Không cần đun nước nóng 24/7. Hẹn giờ để máy chỉ bật trước các giờ cao điểm sử dụng (ví dụ: 6h-8h sáng và 18h-20h tối).
- **Cho máy sưởi:** Lắp đặt bộ điều nhiệt thông minh (smart thermostat) để tự động giảm nhiệt độ khi không có người ở nhà (dựa vào "thung lũng" 13h-16h) và trước khi đi ngủ.

– **Đầu tư vào cách nhiệt:** Đây là giải pháp lâu dài. Cách nhiệt tốt cho tường, mái và cửa sổ sẽ giảm đáng kể nhu cầu sưởi ấm vào mùa đông.

+ **Thay đổi thời điểm sử dụng thiết bị ("Dịch chuyển phụ tải")**
  – **Giặt ủi & Máy rửa bát:** Biểu đồ cho thấy việc giặt ủi không theo một khung giờ cố định. Đây là cơ hội lớn. **Hãy chuyển việc giặt, sấy, chạy máy rửa bát vào các giờ thấp điểm**:
    - **Tốt nhất:** Sau 22h đêm (khi mọi người đã đi ngủ).
    - **Khá tốt:** Giữa ngày (13h-16h) nếu có người ở nhà.
  – **Tránh "Outliers":** Nguyên tắc vàng là **"Không sử dụng đồng thời nhiều thiết bị công suất cao"**. Dữ liệu Box Plot cho thấy các điểm ngoại lai gây tốn điện đột biến. Hãy:
    - Đun nước xong rồi mới dùng lò vi sóng.
    - Nấu ăn xong rồi mới bật máy rửa bát.

+ **Tối ưu hóa các giờ cao điểm (Sáng 7-9h & Tối 19-21h)**
  – **Buổi sáng:** Rút ngắn thời gian sử dụng nước nóng trong lúc tắm. Chuẩn bị sẵn đồ ăn sáng từ tối hôm trước để giảm thời gian nấu nướng.
  – **Buổi tối:**
    - Sử dụng nồi áp suất hoặc lò vi sóng thay cho lò nướng truyền thống khi có thể (tiết kiệm thời gian và điện năng).
    - Tận dụng nhiệt dư của lò nướng/bếp điện để hâm nóng hoặc nấu tiếp món khác.

+ **Giảm "Tải nền" lãng phí (Base Load)**
  – Mức 0.078 kW chạy 24/7 tương đương với gần 700 kWh mỗi năm.

- **Rút phích cắm:** Rút phích cắm các thiết bị không sử dụng (sạc điện thoại, TV, máy tính...) hoặc dùng ổ cắm có công tắc để tắt hoàn toàn.
- **Chọn thiết bị tiết kiệm điện:** Khi mua tủ lạnh, TV mới, hãy chọn loại có nhãn năng lượng hiệu suất cao nhất.

5. Team Member Review and Comment

<ATTACH A TEAM PICTURE HERE>

NAME	REVIEW and COMMENT

6. Instructor Review and Comment

CATEGORY	SCORE	REVIEW and COMMENT
IDEA	__/10	
APPLICATION	__/30	
RESULT	__/30	
PROJECT MANAGEMENT	__/10	

PRESENTATION & REPORT	__/20	
TOTAL	__/100	