Patterns Recognition

Objective:

To detect at least 3 household's patterns in time series data, label them with name (like appliance1,2,3 etc), Visualize which appliance is used in 1 hour and 1 day interval.

Introduction:

Pattern recognition is the automated recognition of patterns and regularities in data. It has applications in statistical data analysis, signal processing, image analysis, information retrievals etc. Here we are using 10 days smart energy meter data set to find specific pattern of appliances based on r phase current values.

Methodology:

I used Pearson Correlation to find the similarity b/w patterns, Correlation provides quantitive information regarding the relationship between continuous variables. This measures the strength and direction of a linear relationship between two variables. **Values** always **range** between -1 (strong negative relationship) and +1 (strong positive relationship). Here I took 200 size windows as a generalize pattern and compare this pattern with another patterns.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Here γ = correlation value

xi = values of the x variable in the sample x

$\bar{x}$ = sample mean value

yi = values of the y variable in sample y

$\bar{y}$ = sample mean

Code steps:

1) Import important libraries (Pandas, Matplotlib, NumPy)

2) Read csv file using pandas.

3) Make a data frame with only three features including r_phase_current, total_active_power, timestamp.

4) plot line plot for r_phase_current values in different window for visualization of its patterns.

5) calculate in how many list we can separate the whole r_phase_current data in 200 windows frame

6) Perform window framing (window size = 200) and create 144 list of r_phase_current values with 200 elements.

7) Define a function name pearson (s1, s2) for calculating pearson correlation b/w patterns. Where s1is pattern that we want to match and s2 is pattern through which we match our patterns.

8) Check is there any null values in r_phase_current values before calling pearson(s1,s2) function

9) Call pearson(s1,s2) function and store all correlation values into a list. Also handle error.

10) Create a list name label, store all the labels (appliance name) selected from correlation values conditions.

- Conditions are if abs(corr) >= 0.70 represents Appliance 1
- If abs(corr) <0.70 and abs(corr)>=0.30 represents Appliance 2
- If abs(corr)>=0.10 and abs(corr)<0.30 represents Appliance 3
- If abs(corr) < 0.10 then it represents Appliance 4

Here abs() is called for absolute value of corr (correlation value)

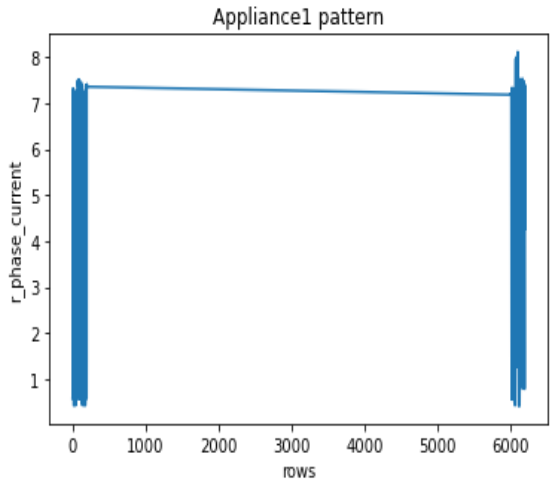
11) Create a data frame of target (Appliances) using list capture from step 10

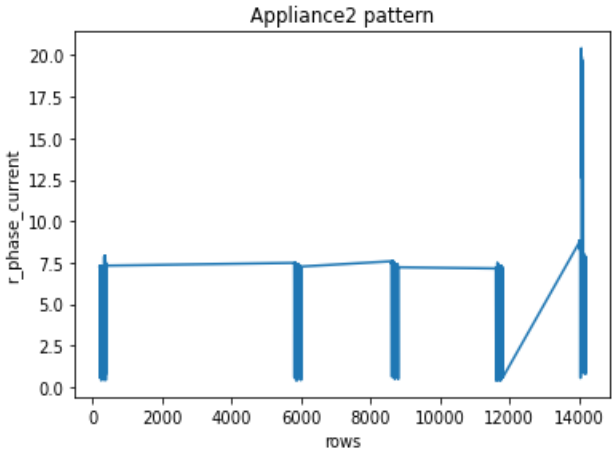
12) join this label data frame (df2) with features data frame (df1) and make a new data frame (df3) including independent features and dependent features.

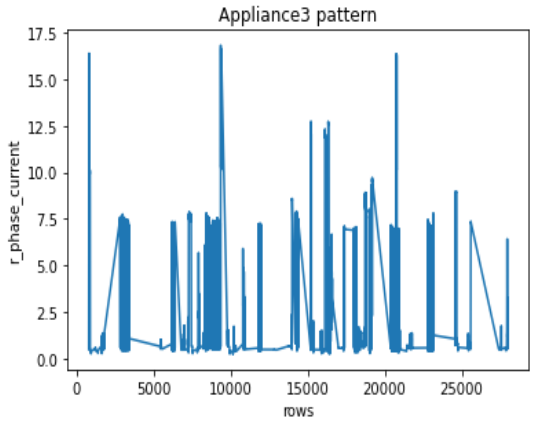
13) Observe the length of frame of one hour and one day

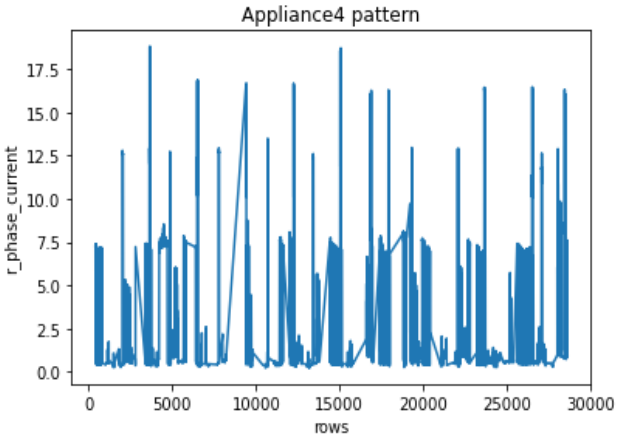
14) Visualize how many appliances are used in first one hour, first 2,3,4 hours, also visualize this for day one.

15) Visualize all four appliances patterns.









Tools & Technology:

Anaconda, jupyter notebook, python, matplotlib, pandas, NumPy.

Code link:

https://github.com/ShubhamJha21/Pattern-Recognition/blob/main/Pattern-Recognition-1.ipynb