Dataset Information

The dataset was downloded from "https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption" and contains power consumption data of one home located in located in Sceaux (7km of Paris, France) between December 2006 and November 2010 (47 months). The variables included in the data set are as follows.

1.date: date in format dd/mm/yyyy

2.time: time in format hh:mm:ss

3.global\_active\_power: household global minute-averaged active power (in kilowatt)

4.global\_reactive\_power: household global minute-averaged reactive power (in kilowatt)

5.voltage: minute-averaged voltage (in volt)

6.global\_intensity: household global minute-averaged current intensity (in ampere)

7.sub\_metering\_1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).

8.sub\_metering\_2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.

9.sub\_metering\_3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

***Note:-The values in the sub\_metering\_\* variables when combined togather do not represent the consumption in the entire house.***

Data Preprocessing

### Time Format Correction

The time information in the dataset is represented by two columns: Date and Time. In order to make plotting data against time easier, in this section the columns "Date" and "Time" are being merged into a column named "DateTime", which is in MATLAB's datetime format.

Date\_time = datetime(year(householdpowerconsumption.Date),month(householdpowerconsumption.Date),day(householdpowerconsumption.Date),hour(householdpowerconsumption.Time),minute(householdpowerconsumption.Time),second(householdpowerconsumption.Time));

Master\_Data = addvars(householdpowerconsumption,Date\_time,'NewVariableNames',"DateTime", 'Before',"Date");

Master\_Data = removevars(Master\_Data,["Date" "Time"]);

### Handling Missing Values and Normalizing

No\_of\_Missing\_Values\_per\_Variable = sum(ismissing(Master\_Data))

plot(Master\_Data.DateTime,Master\_Data.Global\_active\_power)

hold on

Missing\_index = ismissing(Master\_Data.Global\_active\_power);

Data\_with\_filled\_entries = fillmissing(Master\_Data,"linear", "DataVariables","Global\_active\_power");

Filled\_entries = Data\_with\_filled\_entries(Missing\_index,:);

scatter(Filled\_entries.DateTime, Filled\_entries.Global\_active\_power,"red",".")

hold off

xlabel("Time")

ylabel("Active Power in KW")

title("Missing Values in Active Power Variable")

legend(["Existing Values", "Missing Values"])

xlim([datetime(2009,6,10,13,30,10) datetime(2009,6,19,7,10,56)])

ylim([0.0 12.0])

As can be seen in the output above the dataset has almost 26000 missing values. Looking at the plotted example, in some sections continuous values are missing. Efforts of filling such continuous sections will not be very effective as noticeable in the above plot. As we are planning to detect anomalies and are not doing any forecasting in the following sections, it is better to remove the missing values then to fill them by prediction. This will save the calculation from being impacted by values that were added by predictions. Normalization using Z-score is also being performed on data that will be used by the K-means function. This is being done to avoid variables with larger values disproportionately influencing the calculations.

Cleaned\_Data = rmmissing(Master\_Data);

Noramlized\_and\_Cleaned\_Data = normalize(Cleaned\_Data, "DataVariables",["Global\_active\_power","Global\_reactive\_power","Voltage","Global\_intensity","Sub\_metering\_1","Sub\_metering\_2","Sub\_metering\_3"]);

Exploratory Data Analysis (EDA)

### Exploration of the Whole Dataset

Creating the 3D histogram reveals the distribution of the active power in relation to the voltage. Voltage has been selected for processing here as we know that power consumption impacts voltage, as the resistance increases due to increased power flow. The histogram reveals that virtually all of the values are clustered in a single collection.

histogram2(Cleaned\_Data.Voltage, Cleaned\_Data.Global\_active\_power, "DisplayName","Global\_active\_power");

xlabel("Voltage in Volts")

ylabel("Active Power in KW")

title("Data of the Entire Dataset after Preprocessing")

view([-112 52])

### Exploration of the Selected portion of the Dataset

We know that the dataset contains more than 2 million entries for a period of around 4.5 years. Analyzing such a large amount of data is computationally intensive. Such analysis is also not very effective due to the variability of the data. As a result it has been decided to select entries corresponding to a particular weekday in a month for anomaly detection using K-means clustering algorithm.

Data = Cleaned\_Data;

Noramlized\_Data = Noramlized\_and\_Cleaned\_Data;

Selected\_Year = 2008;

Selected\_Month = 3;

% Selected\_Day = 18; % Rather than selecting WeekDay a particular Date can be selected

Selected\_Weekday = 5;

Selected\_entries = year(Data.DateTime)==Selected\_Year & month(Data.DateTime)==Selected\_Month & weekday(Data.DateTime)==Selected\_Weekday; %& day(Data.DateTime)==Selected\_Day;

Data = Data(Selected\_entries,:);

Noramlized\_Data = Noramlized\_Data(Selected\_entries,:);

Month\_name = cell2mat(month(Data.DateTime(1,:),'name'));

[~, Weekday\_name] =weekday(Data.DateTime(1,:),'long');

Title\_text = sprintf('Data of All the %ss of %s %d', Weekday\_name, Month\_name, Selected\_Year);

X = Data.Voltage;

Y = Data.Global\_active\_power;

X\_Normalized = Noramlized\_Data.Voltage;

Y\_Normalized = Noramlized\_Data.Global\_active\_power;

scatter(X, Y)

xlabel("Voltage in Volts")

ylabel("Active Power in KW")

title(Title\_text)

% Create histogram2 of selected data

histogram2(X, Y,"DisplayName","Global\_active\_power");

xlabel("Voltage in Volts")

ylabel("Active Power in KW")

title(Title\_text)

view([-113 68])

Anomaly Detection using K-means Clustering

The Task asked for anomaly detection in power consumption. As large deviations in power consumption were not present it has been decided to shift to anomaly detection in voltage while also taking into account the power consumption at that point in time.

### Feature Selection

As mentioned, consumption can have a significant impact on voltage. The anomalies that we are planning to detect are issues in voltage values when considered with contemporary power consumption. As a result the 'Voltage" variable along with "Global\_active\_power" has been selected to be passed to the Kmeans function.

### **Selection of K - Number of Clusters**

The selection of the right number of clusters is essential for detecting the anomalies. We know the values are not making multiple clusters which are clearly set apart, but EDA has revealed that values are concentrated in some areas. If the values for a weekday in the selected month are more dispersed than increasing the number clusters will be beneficial, as the anomaly detection will be done at cluster level without getting impacted by large concentrations of points in other clusters that are at some distance from other points.

No\_of\_clusters = 5;

[groupNum,C] = kmeans([X\_Normalized Y\_Normalized],No\_of\_clusters);

legend\_names = [];

figure

hold on

for i = 1:No\_of\_clusters

scatter(X\_Normalized(groupNum==i), Y\_Normalized(groupNum==i))

legend\_names = [legend\_names, sprintf("Cluster %d",i)];

end

scatter(C(:,1), C(:,2),'filled',"black")

xlabel("Voltage (Normalized)")

ylabel("Active Power (Normalized)")

title(sprintf("Created Clusters in the %s",Title\_text))

legend([legend\_names,"Centroids"])

hold off

### Detecting Anomalies using Euclidean Distance Between Cluster Values

After the clusters have been formed, anomaly detection is to be done based on the total euclidean distance each point has with other points in the same cluster. A portion of points that is farthest from the other values in the cluster will be classified as anomalies. The size of this portion can be selected using the "Portion\_of\_Anomalies" variable. The portion size is the same for all the clusters, and is to be adjusted as the number of clusters are changed.

Portion\_of\_Anomalies = 0.01; % 0.1 means 10 anomalies in 100 values

clf

plot\_X = [];

plot\_Y = [];

plot\_anomalies\_X = [];

plot\_anomalies\_Y = [];

for i = 1:No\_of\_clusters

X\_of\_cluster = X(groupNum==i);

Y\_of\_cluster = Y(groupNum==i);

plot\_X = [plot\_X ; X\_of\_cluster];

plot\_Y = [plot\_Y ; Y\_of\_cluster];

distances = pdist([X\_of\_cluster, Y\_of\_cluster]);

[~,I] = sort(sum(squareform(distances)));

cluster\_size = length(I);

anomaly\_index\_of\_cluster = I((end-floor(cluster\_size\*Portion\_of\_Anomalies)):end);

plot\_anomalies\_X = [plot\_anomalies\_X ; X\_of\_cluster(anomaly\_index\_of\_cluster)];

plot\_anomalies\_Y = [plot\_anomalies\_Y ; Y\_of\_cluster(anomaly\_index\_of\_cluster)];

end

scatter(plot\_X, plot\_Y, "black")

hold on

scatter(plot\_anomalies\_X, plot\_anomalies\_Y,"red")

legend("Normal Values","Detected Anomalies")

xlabel("Voltage in Volts")

ylabel("Active Power in KW")

title(sprintf("Anomalies in the %s",Title\_text))

hold off

Comments on Results

The anomalies being detected are combinations of voltage and active power that do not frequently occur on the selected weekday, as a result they fall on the border of the portion of the plot that contains the main concentration of points. If the ratio of the height and width of the dense part of the plotted data is very different from 1 then using more number of clusters is better as the anomaly detection will be localized in clusters spread over the larger dimension of the concentration.