ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Evaluación y propuesta de metodología aplicada a la estratificación de clientes pertenecientes al área de concesión de una Unidad de Negocio de la Corporación Nacional de Electricidad mediante la implementación de un proceso de zonificación.

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Electricidad

Presentado por:

Almeida Muñoz Alberto Sebastián

Mora López Kevin Jesús

GUAYAQUIL - ECUADOR

Año: 2020

DEDICATORIA

"A mi madre, que por mi vida daría la suya propia. Su ejemplo me ha mantenido en el sendero del trabajo y el esfuerzo, la admiro y amo más que a nada. A mi padre, quien se adelantó en el camino, nunca tendré un logro completo pues no te tengo para felicitarme, me inculcaste el estudio y heredé de ti la vocación de la enseñanza, virtudes que me han llevado a través de este camino, como estudiante, ayudante y persona. Hasta la vuelta señor".

Alberto Sebastián Almeida Muñoz

"En primer lugar dedico este proyecto a Dios por ser una guía durante mi vida. A mis padres, por darme la vida, el amor y el apoyo incondicional en mis estudios mediante sus sacrificios. A mis hermanas, que con su ejemplo y esfuerzo para cumplir sus metas me motivaron a perseverar hasta cumplir las mías. A mis maestros, amigos y enamorada, que siempre creyeron en mí y estaban para mí cuando más los necesitaba"

Kevin Jesús Mora López

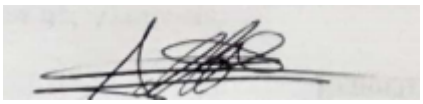
AGRADECIMIENTOS

"Agradecemos profundamente a nuestro tutor el PhD. Miguel Torres por brindarnos su asesoría y amistad a lo largo de este camino. A Valeria Zambrano, que nos ayudó con la elaboración del 5min Pitch cuyo resultado nos dejó satisfechos.

Finalmente, a todos los amigos que pudimos conocer a lo largo de estos 5 años de estudio, con quienes compartimos, clases, problemas, victorias y anhelos. Cuando llegas al final del camino el viaje ya no parece tan largo, ¿Verdad?".

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Alberto Sebastián Almeida Muñoz y Kevin Jesús Mora López* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Alberto Sebastián Almeida Muñoz

Kevin Jesús Mora López

EVALUADORES

Firmado electrónicamente por:
RENAN XAVIER ZAMBRANO ARAGUNDY

PhD. Renan Zambrano

PROFESOR DE LA MATERIA

Firmado electrónicamente por:
MIGUEL ALBERTO TORRES RODRIGUEZ

PhD. Miguel Torres

PROFESOR TUTOR

RESUMEN

El presente proyecto propone una nueva forma de estratificación de los usuarios residenciales de la empresa distribuidora *CNEL EP UN GYE*, debido a que la metodología actual se base en una definición de rangos de consumo promedio y con límites definidos por la empresa; una técnica de estratificación que no permite categorizar correctamente a los clientes según su forma de consumir.

Se utilizó la información recibida por parte del personal de la empresa distribuidora, que consistían en el consumo mensual de los usuarios a nivel residencial comprendidos para el año 2019 y para los 6 primeros meses del año 2020, se realizó un preprocesamiento de esta información y mediante el uso de *machine learning*, con un algoritmo conocido como *k-means*, se obtuvieron los clústeres.

El algoritmo proporcionó los rangos de consumo que delimitan a cada estrato y la cantidad de usuarios que los conforman, tanto para una distribución de 3 y 6 estratos. Esta información fue contrastada con la estratificación actual planteada por la empresa distribuidora.

Hay diferencias entre la estratificación inicialmente planteada por la empresa y los resultados del proyecto, ya que en la metodología propuesta los usuarios son clasificados por su patrón de consumo, es decir, no sólo cuanto consumen, sino como consumen. La clasificación por patrones de consumo permite comparar a los usuarios dentro de un mismo clúster para poder identificar a aquellos cuyo patrón de consumo sea atípico, que permite a la empresa llevar acciones a futuro ante posibles pérdidas no técnicas.

Palabras Clave: *Clúster, Machine learning, Estratificación, K-means*.

ABSTRACT

This project proposes a new form of stratification of the residential users of the distribution company CNEL EP UN GYE, because the current methodology is based on a definition of average consumption ranges, with limits defined by the company; a stratification technique that does not allow to correctly categorize customers according to their way of consuming.

The information received by the personnel of the distribution company was used, which consisted of the monthly consumption of the users at the residential level for 2019 and for the first 6 months of 2020, a pre-processing of this information was carried out and using machine learning, with an algorithm known as k-means, the clusters were obtained.

The algorithm provided the consumption ranges that delimit each stratum and the number of users that formed them, both for a distribution of 3 and 6 clusters. This information was contrasted with the current stratification proposed by the distribution company.

There are differences between the stratification initially proposed by the company and the results of the project, since in the proposed methodology, users are classified by their consumption pattern, that is, not only how much they consume, but also how they consume. Classification by consumption patterns allows users to be compared within the same cluster in order to identify those whose consumption pattern is atypical, which allows the company to take future actions against possible non-technical losses.

Keywords: *Cluster, Machine learning, Stratification, K-means.*

ÍNDICE GENERAL

ABREVIATURAS

ESPOL Escuela Superior Politécnica del Litoral

CNEL EP Corporación Nacional de Electricidad Empresa Pública

UN Unidad de Negocio

GYE Guayaquil Ecuador

MB Megabyte

SIMBOLOGÍA

kW kilovatio

GW Gigavatio

kWh Kilovatio hora

GWh Gigavatio hora

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

CAPÍTULO 1

Introducción e información general

1. INTRODUCCIÓN

1.1. Descripción del problema

De manera general, una de las mayores obligaciones que tienen las empresas de comercialización y distribución de energía eléctrica es realizar la planificación más eficiente de la distribución de energía, dentro de lo cual comprende la gestión y predicción de la demanda actual y futura. La mala ejecución de esta labor puede conllevar a pérdidas económicas para las empresas y a nivel técnico un mal dimensionamiento de los equipos [1].

Para dichas actividades las empresas eléctricas por mucho tiempo han realizado la caracterización de los usuarios en base a información fija de nivel de consumo o encuestas sociodemográficas [2], limitada por la cantidad de mediciones y la tecnología del momento [3].

Con la aparición de medidores inteligentes y la disponibilidad de mayor cantidad de mediciones, el crecimiento de la demanda, así como también el crecimiento anual del número de usuarios surge la necesidad de la revisión y la aplicación de nuevos métodos de agrupación y caracterización de los usuarios.

1.2 Justificación del problema

La estratificación de los usuarios es de mucha utilidad para las empresas de energía debido a que sirve para definir perfiles de carga de clientes, diseñar tarifas y mejorar la previsión de carga [3], por lo que el método a emplearse debe ser eficiente e ir de la mano al avance de las nuevas tecnologías de adquisición de la información.

Una de las propuestas adoptadas por algunas empresas comercializadoras de energía es la adopción de metodologías de minería de datos no supervisada para el *clustering* de los usuarios. Estas técnicas ayudan a agrupar usuarios con características similares en su consumo mensual.

Los métodos de *clustering* han sido aplicados con éxito en la estratificación de los usuarios en la Unidad de negocios Santo Domingo con una totalidad de 271809 cuentas [4] y en la Empresa Eléctrica Azogues 37029 usuarios registrados [5], por lo que para una empresa distribuidora como la Unidad de Negocios Guayaquil que registró en el año 2018 un total de 2’499.661 usuarios residenciales y comerciales [6], cobra mayor importancia un método de caracterización y agrupación eficiente para el manejo de las tarifas y previsión de la carga.

1.3 Objetivos

1.3.1 Objetivo General

Realizar el proceso técnico que permita la estratificación de los clientes de tipo residencial de la empresa distribuidora CNEL EP unidad de negocio Guayaquil.

1.3.2 Objetivos Específicos

- Identificar los estratos a los que pertenecen los clientes residenciales de la unidad de negocio Guayaquil y los límites de energía entre los que se encuentran.
- Determinar el estrato que predomina, en número de usuarios, tanto con k=3 y k=6, posterior a la aplicación del método de *clustering k-means*.
- Obtener el patrón de consumo de cada estrato que permita evaluar de manera gráfica el comportamiento de los usuarios pertenecientes al mismo.

1.4 Marco teórico

1.4.1 Minería de datos y análisis de *Clustering*

El conocimiento de los diferentes hábitos de consumo de los clientes es información importante que debe recoger y saber procesar toda empresa eléctrica con el fin de realizar planificación y manejo adecuados en la empresa, sobre todo considerando la variabilidad en cuando al sector residencial por distintos factores respecto a la forma de vida de los consumidores.

1.4.1.1 Minería de datos

La minería de datos se define como un proceso que maneja información y datos computacionales con el objetivo de crear modelos y obtener resultados que permitan descubrir patrones y tendencias de comportamiento de las variables implicadas en el estudio que se lleva a cabo [7].

Este proceso comprende cinco partes básicas:

- Clasificación de datos: En esta parte se logra definir el universo de información que se tiene y los objetivos que se pretenden alcanzar con la minería de datos y como resolver el problema que se tiene planteado.
- Preprocesamiento de datos: Esta parte reúne acciones y estrategias mediante las cuales se pretende preparar el terreno con el que se va a trabajar, actividades comunes realizadas consisten en eliminar datos innecesarios, que consistan en ruido o unificar las diferentes fuentes de información que se puedan tener en un solo universo general de información a ser procesada.
- Transformación de datos: Dependiendo del tipo de método estadístico o algoritmo computacional puede resultar necesario convertir el formato de entrada de la

información de la que se dispone de manera que pueda ser procesada facilitando el trabajo que se tiene.

- Minería de datos: Es el trabajo como tal, en esta parte se procesa la información mediante el uso de herramientas matemáticas y computacionales, el trabajo que se realiza depende más estrictamente del método que se esté utilizando y los objetivos que se quieran conseguir a través de su implementación. El realizar este proceso va devolver los resultados de este trabajo.
- Interpretación de datos: Se interpretan los resultados obtenidos en la parte anterior, dándole un sentido práctico, social, objetivo, a la salida de información obtenida. En esta parte se le da un sentido y se explican los resultados tratando de siempre encontrar una razón de por qué se obtuvo esto y si era lo que se esperaba obtener.

1.4.1.2 Análisis de *Clustering*

De acuerdo con [7] el análisis de *Clustering* es un grupo de técnicas estadísticas multi variable utilizadas para agrupar individuos en grupos homogéneos. El objetivo de aplicar este análisis es juntar grupos de individuos en base a un criterio de similitud entre ellos, misma similitud que ya está presente en la información obtenida, por lo que el *Clustering* lo que hace es interpretar esta información para formar los grupos y clasificar los datos. El criterio de homogeneidad y los diferentes patrones similares bajo los cuales se piensa reunir a estos individuos debe ser provisto con anterioridad. Estos grupos se forman mediante algoritmos los cuales tienen sus requerimientos propios en cuanto a cómo manejan la información, sin embargo, no existe un método perfecto ni generalizado para realizar *Clustering*, en términos generales cualquiera de los diferentes algoritmos conocidos puede realizar el trabajo de categorización, sin embargo, existen métodos que son preferidos sobre otros bajo diferentes

criterios como la velocidad con la que convergen, los recursos informáticos que ocupan o el tipo de principio estadístico en el que se basan.

Una de las ramas en las que se ha aplicado y es la que resulta de interés en el presente trabajo es el sector eléctrico, sobre todo orientado a la distribución y la comercialización del servicio público de energía eléctrica. Por lo general las características según las cuales se agrupa a los usuarios consumidores de energía eléctrica es por la cantidad de energía que consumen, puesto que es la mejor manera de agrupar a los usuarios y saber quiénes son los que consumen más, así también quienes son los que consumen menos, siendo este también un indicador socioeconómico pues siempre se ha relacionado la cantidad de energía que consume un cliente y la cantidad de equipos eléctricos instalados con el poder adquisitivo y el estrato social al que pertenece.

Entre los métodos que más se han empleado a lo largo de la historia en el sector eléctrico tenemos los métodos *k-means*, jerárquica, mapas auto-organizables, método modificado de seguir al líder, etc. El método en el que se enfocará este trabajo es el método *k-means*, el cual resulta útil en cuanto a que utiliza las medidas de distancias entre los datos para agrupar a los usuarios más similares posibles dentro de los mismos estratos.

1.4.2 Aplicaciones del *Clustering* en sistemas eléctricos de potencia

Con la mayor disponibilidad de información obtenida de mediciones los estudios de la carga a nivel residencial y comercial ya no solo van direccionados a mejorar los perfiles de carga [3] sino también mejorar la caracterización del usuario de la cual parte.

La característica de la carga residencial y comercial es que representa en términos de número de usuarios (o cuentas) la mayor cantidad del total de consumidores finales. Adicionalmente la carga residencial se diferencia de la carga comercial e industrial que presenta una mayor

heterogeneidad entre los usuarios, que no se puede agrupar por similitudes socioeconómicas o localización geográfica, ya que los patrones de consumo cambian de una residencia a otra. Esta variabilidad de patrones de consumo hace interesante la aplicación de métodos de agrupamiento como el *clustering*, que conlleva a varias aplicaciones

1.4.2.1 Diseño de tarifa

El diseño de las tarifas se realiza normalmente bajo consideraciones económicas, y en el caso de necesitarse, aplicando subsidios a ciertos grupos de consumidores en base a características sociodemográficas [3]. En este sentido, las técnicas de *clustering* pueden ser utilizadas para maximizar beneficios anuales en base a la agrupación de consumidores, proponiendo nuevas alternativas como la creación de tarifas múltiples para los grupos formados, es decir, que se busca asignar la tarifa más adecuada a cada grupo de clientes con características similares. Para este punto cabe mencionar que el número de grupos o clústeres puede ser establecido por la empresa comercializadora de energía o hallado el número óptimo en base a los algoritmos de *clustering*.

1.4.2.2 Pronóstico de carga

Las aplicaciones del *clustering* en la previsión de carga se basan en una técnica llamada *Clúster-based Aggregate Forecasting (CBAF),* que consiste en realizar grupos o clúster de los usuarios residenciales, realizar el pronóstico de la carga para cada clúster en intervalos de 1-24 horas y posteriormente calcular el total de la carga pronosticada [8]. Esta metodología depende del número de clústeres utilizados y la cantidad total de usuarios residenciales.

1.4.2.3 Respuesta de demanda

Similar al diseño tarifario, los métodos de *clustering* pueden ayudar a identificar en base a los patrones de consumo a aquellos grupos a los que se puede direccionar incentivos de reducción de consumo para disminuir la demanda en horas pico y mejorar la estabilidad del

sistema. Para este enfoque algunas metodologías o técnicas de minería de datos se apoya de las curvas de carga, mientras que en otros estudios se ha realizado una segmentación demográfica [3]

1.4.2.4 Clasificación

Una de las más recientes aplicaciones en estudio es la aplicación de *clustering* a partir de los datos obtenidos de medidores inteligentes para poder agrupar a aquellos usuarios de los que aún no se posee información proveniente de medidores inteligentes. La metodología consiste en que a partir de los clústeres creados se determina un perfil de carga característico de cada clúster que sea posteriormente comparado con el perfil de carga de los usuarios no agrupados o de los que se posea poca información por las limitaciones de la medición. [3]

1.4.3 Método *k-means*

Este método es conocido y ampliamente utilizado por el nivel de adaptabilidad que puede tener en diferentes ramas y campos de investigación, siendo fácil de implementar, eficiente y no consume mucha memoria. Pertenece a las técnicas de agrupamiento basadas en particiones, de tal manera que se busca obtener un alto grado de similitud intergrupal y muy bajo en similitud intragrupal, es decir, los individuos que forman parte de los clústeres van a tener un grado de homogeneidad alto y sus características serán muy similares, a pesar de que los grupos no tengan mayor similitud entre ellos, siendo un clúster muy diferente de otro y esto no implicaría ningún problema. Esto se logra resolviendo un problema de optimización que se basa en minimizar la suma de las distancias cuadráticas de cada dato con respecto al centroide de su grupo, tal como lo modela la ecuación (1)

Ecuación 1. Expresión para el algoritmo k-means.

$$min\mathrm{E}(\mu_i) = min \sum_{i=1}^{k} \sum_{x_j \epsilon S_i} \|x_j - \mu_i\|^2$$

Para esto S representa un conglomerado de datos o valores que son denotados como x y pertenecen a un grupo con un centroide μ, siendo un total de k grupos.

Para poder determinar qué tan similares son los datos entre sí se calcula la distancia o proximidad con alguno de los diferentes métodos, como la distancia euclidiana, euclidiana cuadrada, entre otras. Existen múltiples variantes para poder realizar el método, entre los más conocidos destacan:

- Algoritmo Forgy/Lloyd

  Ambos son básicamente el mismo algoritmo, la diferencia es que el primero utiliza una distribución continua mientras que el segundo una discreta, siendo diferente la función objetivo que piensan minimizar, pero en esencia es la misma ya presentada, cualquiera de los 2 métodos es indiferentes de cual función de distancia se utilice.

  Lo que el algoritmo hace se puede separar en 3 pasos:

  a) Escoger centroides iniciales, exactamente k.
  b) Asignar los datos a un clúster según la distancia calculada entre cada dato y cada centroide inicial.
  c) Se calcula la media de los valores asignados al clúster de modo que con este valor se recalculan los centroides.

  Luego se sigue repitiendo b y c hasta que los centroides varíe por debajo de un cierto valor de tolerancia o que dejen de variar completamente.

- Algoritmo MacQueen

  Es de tipo iterativo, se considera más eficiente que los anteriores pues actualiza los centroides con mayor frecuencia, esto se debe que no sólo se actualizan los centriolos al asignar los datos a los clúster sino también cuando cambia el subespacio. De modo que, si uno de los centroides deja de ser el más cercano y ahora resulta ser otro, se

mueven los casos al nuevo centroide más cercano, pero además se recalculan los 2 centroides involucrados y se procede a partir de los nuevos valores obtenidos.

- Algoritmo de Hartigan & Wong

  Se caracteriza por basarse en la suma de los cuadrados de los errores de los datos del clúster de modo que podría darse el caso de que un dato se mueva de un clúster a otro, aunque el centroide del nuevo clúster tenga una distancia mayor que la del primero, pues el criterio va a ser la disminución del error entre los datos de los grupos. Se repiten los pasos a, b y c de los métodos de Forgy y Lloyd con la adición de 2 pasos:

  Calcular la suma de los cuadrados de los errores dentro de cada clúster

  Comparar los errores entre los clúster y si uno es menor a otro lo cambia de clúster.

1.4.3.1 Limitaciones del método

Cualquiera de los algoritmos de tipo *k-means* siempre va a poder converger, el problema con esto puede ser que no se obtenga una respuesta global sino local y de tal manera el resultado no es el óptimo. Para los métodos de Forgy y Lloyd es también posible que se muevan todos los datos de un clúster a otro en una sola iteración por lo que podrían resultar en la existencia de grupos sin elementos. Por su parte los otros dos métodos tienen una notable susceptibilidad al orden en que se tienen agrupados los datos. Este método es un buscador local por lo que es sensible también a los centroides iniciales escogidos.

Dado que todos estos algoritmos consideran la media existe un problema al momento de trabajar con los valores *outliers*, es decir aquellos que son datos aberrantes que no siguen el comportamiento del resto de los elementos del clúster, estos suelen ser eliminados del sistema, pero en la realidad no necesariamente son valores errados sino simplemente son casualidades. Así mismo el método tiene la tendencia de crear clúster con el mismo tamaño, cosa que no necesariamente va a tener sentido práctico, pues desde luego no necesariamente

van a haber el mismo número de usuarios en la categoría de mayor consumo energético que en el primero.

CAPÍTULO 2

Metodología

2. METODOLOGÍA

2.1.Preprocesamiento de datos

Para empezar el trabajo de *clustering* es importante recibir los datos sobre los cuales se va a trabajar y adecuarlos a las necesidades que implica la estratificación de datos, esto es, obtener la información realmente importante que va a permitir categorizar a los elementos del clúster. Este trabajo pretende estratificar a los usuarios de la CNEL EP UN Guayaquil en base a su consumo histórico de energía por lo que es importante obtener un distintivo de cada usuario y la información del consumo energético en kWh para cada uno de los meses que se han dispuesto en el análisis.

2.1.1. Datos obtenidos de CNEL EP UN Guayaquil

La CNEL por su lado proporcionó la información de sus usuarios para 18 meses que constituyen todo el año 2019 y la primera mitad del 2020, en archivos con el modelo que se presenta en la ilustración 1.

Ilustración 1. Información de CNEL EP UN Guayaquil para el mes de enero 2019.

En la ilustración 1 se visualiza el archivo *".csv"* proporcionado por empresa para el primer mes del 2019, este archivo separado por comas involucra una gran cantidad de información innecesaria para cada usuario, desde el punto de vista de la estratificación, como es la dirección del usuario, número de cédula, teléfono, entre otros por lo que esta información debe ser depurada de modo que se pueda procesar, también cabe resaltar que el peso aproximado de cada uno de estos archivos es de cerca de 220 MB por toda la información innecesaria que ya se explicó anteriormente.

2.1.2. Criterios de procesamiento de datos y metodología aplicada

De estos archivos se va a extraer el distintivo de cada usuario que para este caso será el código de cuenta contrato proporcionado por la empresa para cada usuario. La información del consumo de energía para cada mes registrado mediante lecturas debidamente tomadas por parte del personal de la empresa eléctrica es la información relevante para el *clustering* por lo cual es la información que debe extraerse de cada uno de los archivos proporcionados.

Para extraer esta información se emplea el código en Matlab presentado en la ilustración 2.

```
%% Close the text file.
fclose(fileID);
frmto='%*s %[^\n\r]';
consumo=[];
codigo=[];
for i= 1:length(dataArray{1,2})
    linea=dataArray{1,2}{i};
    if length(strfind(linea,'@'))>0
    linea=string(textscan(linea,frmto,'Delimiter', '@'));
    end
    filtr1=textscan(linea,frmto,'Delimiter', '->');
    filtr2=textscan(string(filtr1),frmto,'Delimiter','0');
    filtr3=textscan(string(filtr2),frmto,'Delimiter','1');
    filtr4=textscan(string(filtr3),'%*s %q %*[^\n\r]','Delimiter',',');
    cons=textscan(string(filtr4),'%f %*[^\n\r]','Delimiter','"');
    cod=textscan(string(dataArray{1,1}{i}),'%f %*[^\n\r]','Delimiter','"');

    codigo(i,1)=cell2mat(cod);
    consumo(i,1)=cell2mat(cons);
end
result=[codigo consumo];

clearvars filename delimiter formatSpec fileID linea filtr1 filtr2 filtr3 filtr4 cons ans frmto cod i;
dlmwrite('C:\Users\kevin\Downloads\DATOS_CNEL\DatosSemiproces\pre202006.csv',result,'precision','%.2f')
```

Ilustración 2. Código de datos semiprocesados.

La ilustración 2 muestra el código en Matlab utilizado para extraer el código y consumo de cada usuario, el mismo tiene por finalidad la creación de un archivo con 2 columnas, una que es el código y otra que es el consumo mensual, de modo que por cada archivo de cada mes se crea uno nuevo que sólo presenta esta información, dando 18 nuevos archivos que para efectos de este documento serán llamados *"Datos semiprocesados"*.

Hecho esto es necesario unir estos 18 archivos en uno sólo, por lo cual se utiliza un nuevo código en Matlab, se presenta este código en las ilustraciones 3 y 4.

Posteriormente sus resultados en la ilustración 5.

```
clc
clear all

ruta= 'C:\Users\alber\Downloads\DatosSemiproces\DatosSemiproces\pre';
anl9=(201901:01:201912);
an20=(202001:01:202006);
meses=[anl9 an20];
```

Ilustración 3. Código para carga de Datos semiprocesados.

La ilustración 3 muestra el código que carga los archivos que contienen los datos semiprocesados creados anteriormente, con la finalidad de unificarlos.

```
fileID = fopen(strcat(ruta,num2str(meses(1)),'.csv'),'r');
datacell = textscan(fileID, '%f%f%*[^\n\r]', 'Delimiter', ',');
fclose(fileID);
A=cell2mat(datacell);
Enel9=table(A(:,1),A(:,2),'VariableNames',{'Codigo' 'Cons_201901'});
clearvars fileID datacell A
GranM=Enel9;
columnas={'Cons_201901'};
for i =2:length(meses)
    fileID = fopen(strcat(ruta,num2str(meses(i)),'.csv'),'r');
    datacell = textscan(fileID, '%f%f%*[^\n\r]', 'Delimiter', ',');
    fclose(fileID);
    col_name=strcat('Cons_',num2str(meses(i)));
    columnas{i}=col_name;
    B=cell2mat(datacell);
    mes=table(B(:,1),B(:,2),'VariableNames',{'Codigo' col_name});
    clearvars fileID datacell B

    GranM = outerjoin(GranM,mes,'MergeKeys',true);
end
clearvars i Enel9  anl9 an20 colname
idx_miss= ismissing(GranM(:,columnas));
GranM{:,columnas}(idx_miss) = 0;
```

Ilustración 4. Código para unificar los Datos semiprocesados.

La ilustración 4 presenta el código que leyendo cada uno de los 18 archivos devuelve una matriz completa con 19 columnas, la primera que contiene los códigos de cada usuario ya

existente desde enero 2019 y los nuevos usuarios agregados a partir de los demás meses, las columnas 2 a 19 tienen el consumo de cada usuario para cada uno de los 18 meses, siendo natural que existan usuarios con valores no tomados en todos los meses puesto que son cuentas que se agregaron después de la fecha de inicio del análisis.

Debido a que estos valores extraños son contraproducentes para el trabajo de *clustering* es necesario aplicar un filtrado a la información como se muestra en la ilustración 5.

```
%% Filtrado de Datos
idx_inc_rows=sum(idx_miss,2)>0;% Indices de filas (usuarios) que le faltan datos (NaN)
Users_dat_incomp=sum(idx_inc_rows);% Numero de usuarios con datos incompletos(NaN)
Users_dat_compl=height(GranM)-Users_dat_incomp;% Numero de usuarios con datos completos, pero con 0
M_no_miss=GranM;
M_no_miss(idx_inc_rows,:) = []; %Tabla sin los usuarios que tienen NaN
idx_nonzero=table2array(GranM(:,columnas))>0;
num_med=sum(idx_nonzero,2);% Numero de mediciones por usuario
idx_compl_rows=num_med<18;% Indices de filas(usuarios) que tienen 18 mediciones diferente de 0
M_no_zero=GranM;
M_no_zero(idx_compl_rows,:)=[]%Tabla sin los usuarios que tienen algún 0 o NaN
```

Ilustración 5. Código para filtrado de datos.

El código mostrado en la ilustración 5 toma la matriz previamente obtenida y elimina los usuarios con datos faltantes y con mediciones de cero, dejando únicamente a los que tienen sus 18 mediciones completas y con valores reales de medición.

Estos usuarios son los que se utilizarán para el *clustering* siendo un total de 498991, para mostrar el número de usuarios con mediciones completas se elaboró un código que se presenta en la ilustración 6.

```
%% Resumen de mediciones
Med=histogram(num_med,'BinLimits',[0,max(num_med)])
Dist_med=histcounts(num_med,'BinLimits',[0,max(num_med)])';
T_med=table((0:1:max(num_med))',Dist_med,'VariableNames',{'Total_Mediciones' 'Total_Usuarios'});
```

Ilustración 6. Código para la creación del histograma.

El código presentado en la ilustración 6 muestra el código que cuenta el número de usuarios que tienen mediciones completas, que les falta una medición, que les falta 2 mediciones, etc.

El código presenta esta información en forma de histograma el mismo que se muestra en la ilustración 7.

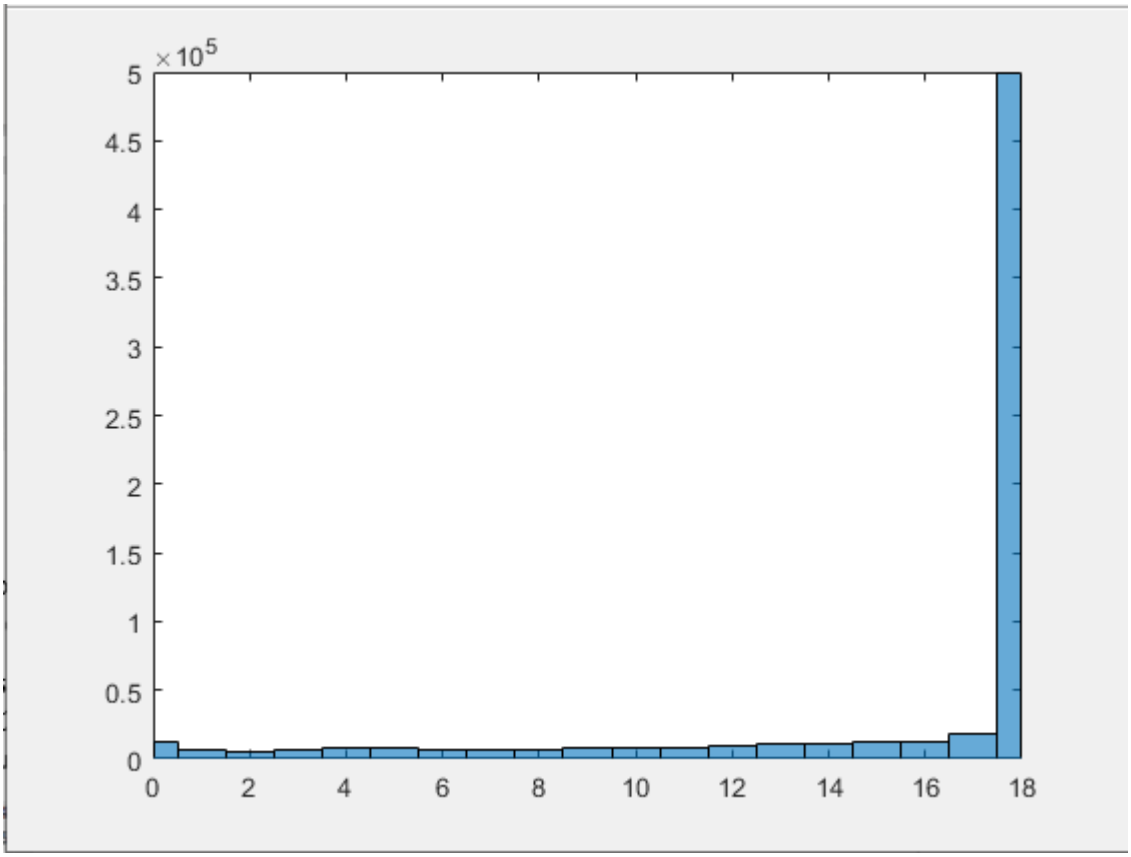


Ilustración 7. Histograma de usuarios en función de número de mediciones.

En el histograma presentado en la ilustración 7 se muestra la comparativa entre los usuarios con mediciones faltantes, donde vemos que la mayoría tienen las 18 mediciones completas, de modo que los usuarios eliminados en el *clustering* representan un número reducido en comparación al global de datos iniciales, esto confirma que los usuarios presentes en la matriz resultante que da el filtrado de datos son una muestra significativa para la estratificación de usuarios.

A continuación, nos interesa eliminar a los usuarios que tengan todas sus mediciones repetidas, muestra el código empleado para eliminar a estos usuarios de la matriz en la ilustración 8.

```
48      %% Eliminacion de datos repetidos
49 -    varianzas=var(table2array(M_no_zero(:,columnas)),0,2);
50 -    idx_var_rp=varianzas<=0.1;
51 -    M_no_repeat=M_no_zero;
52 -    M_no_repeat(idx_var_rp,:)=[];
53 -    M_rep=M_no_zero(idx_var_rp,:);
```

Ilustración 8. Código para la eliminación de datos repetidos.

Para poder eliminar estos valores utilizamos el criterio de la varianza, como se muestra en el código de la ilustración 8, se calcula la varianza para las filas de la matriz, es decir, para cada uno de los usuarios, luego con este valor para cada usuario se toman aquellos que tienen una varianza menor o igual a 0.1. Se declara la matriz M_no_repeat, la cual toma la matriz sin datos iguales a cero y compara esta matriz con la varianza de los usuarios y elimina a aquellos que están fuera del parámetro ingresado, esta varianza podría reajustarse para aceptar valores mayores o menores.

La importancia de eliminar a estos usuarios con datos repetidos recae en que, si bien puede parecer normal, a primera vista, que un usuario consuma lo mismo todos los meses, esto no corresponde a la realidad, pues ignora que existe una época de mayor calor en la ciudad de Guayaquil, donde la experiencia nos indica que el consumo energético crece por el uso de ventiladores y aires acondicionados. Tampoco tiene sentido que en 18 meses un usuario consuma el mismo valor de energía, pues los usuarios tienden a adquirir nuevos equipos como electrodomésticos, televisores, consolas, etc.

El código que se utilizó para la eliminación de *outliers* se presenta en la ilustración 9.

```
%% Filtrado
Info=table2array(M_no_repeat(:,2:19));
codigo=table2array(M_no_repeat(:,1));
[idx_outlier,lower,upper,center]=isoutlier(Info,'median',2,'ThresholdFactor',5);
num_haveoutl=sum(idx_outlier,2);
Depurado=Info;
Info=[Info,num_haveoutl];
Depurado(idx_outlier)=NaN;
F=fillmissing(Depurado,'movmean',7,2);
idx_Oneoutl=num_haveoutl<=1;
M_final=array2table([codigo(idx_Oneoutl),F(idx_Oneoutl,:)],'VariableNames',['Codigo' columnas]);
Data_Out=[codigo(~idx_Oneoutl),Info(~idx_Oneoutl,:)];
```

Ilustración 9. Código para el filtrado de ouliers.

Los *outliers* son datos que están muy alejados de lo normal, siendo que presentan valores muy elevados o comportamientos erráticos como un mes cuyo valor no tiene ningún sentido en comparación a meses posteriores y siguientes, por lo que se utiliza la función *isoutlier* para poder identificarlos y eliminarlos de la matriz, con el fin de poder tener un mejor desempeño en los resultados.

2.2. Normalizando datos

Una vez eliminando valores de cero, repetidos y *outliers* podemos hacer un reescalamiento de los datos, esto suele llamarse también normalización y se presenta el código correspondiente en la ilustración 10. En este proyecto se realizó el *clustering* para los datos normalizados y reales.

```
%% Reescalamiento
M_rescale=M_no_repeat;
for k = 1:length(columnas)
    col_M=table2array(M_rescale(:,k+1));
    max_value=max(col_M);
    min_value=min(col_M);
    col_resc=(col_M- min_value)/ (max_value - min_value);
    M_rescale.(k+1)=col_resc;
end
clearvars k col_M col_resc min_value max_value
```

Ilustración 10. Código para el reescalamiento de los datos en Matlab.

Para esto se definió una matriz M_rescale, la cual toma como base la matriz sin datos *outliers* previamente obtenida con el código de la ilustración 9. En la ilustración 10 se muestra que se

toma para cada columna, es decir, para cada mes, el valor máximo y mínimo del consumo eléctrico, a continuación, se toma cada valor y lo reescala al restar el valor mínimo y dividir esta diferencia para la diferencia entre el valor de consumo máximo y el mínimo del mes. Esto se itera hasta obtener una matriz de datos reescalados, una vez realizado este proceso se procede con el algoritmo *k-means.*

La expresión utilizada para el reescalamiento se presenta en la ecuación 2.

Ecuación 2. Expresión para el reescalamiento de datos.

$$Data_{Resc} = \frac{Data_{Real} - Min}{Max - Min}$$

2.3. Clustering

El trabajo de *clustering* se va a realiza para los datos reales y los datos normalizados, en ambos casos se utiliza el algoritmo *k-means* con k=3 y k=6.

Con el preprocesamiento de datos realizado ya nos es posible trabajar con el algoritmo propuesto que es *k-means*. Este tiene como objetivo crear el número de clústeres deseados en base al consumo eléctrico de los usuarios, pero tomando en cuenta, más que sólo el rango de valores en el que se encuentra el usuario sino el patrón de comportamiento de estos.

2.3.1. Detalles del algoritmo empleado

Para emplear el algoritmo definimos en primera instancia los datos que vamos a utilizar, estas líneas de código se muestran en la ilustración 11.

```
79      %% Datos para Kmeans
80 -    Data=table2array(M_final(:,2:19));
81 -    Data_norm=table2array(M_rescale(:,2:19));
82 -    max_data=max(Data,[],1);
83 -    min_data=min(Data,[],1);
84 -    std_data=std(Data_norm,0,2);
85 -    cons_data=mean(Data,2);
```

Ilustración 11. Código para definir los datos a usar en el algoritmo k-means.

Como se muestra en la ilustración 11, se definen Data y Data_norm, matrices que no contienen el número de identificación de los usuarios sino sólo sus consumos para los 18 meses, adicionalmente se obtiene los datos mínimo y máximo y se define para trabajar en el algoritmo con el promedio.

2.3.2. Ejemplo del algoritmo empleado en MATLAB

A continuación, se mostrará un ejemplo de cómo funciona el algoritmo empleado, en posta de que el lector conozca de manera práctica el funcionamiento de *k-means*, que posteriormente será aplicado directamente como una función de Matlab para la estratificación de usuarios de CNEL EP UN Guayaquil.

El algoritmo que utiliza *k-means* para la realización de los clústeres es el siguiente:

2.3.2.1. Inicialización

En esta primera etapa se seleccionan los *k* centroides iniciales para los clústeres de manera aleatoria. En el algoritmo tradicional se escogen aleatoriamente los *k* centroides al mismo tiempo a partir de *k* observaciones presentes en el conjunto de observaciones *X*.

Matlab, hace uso de la inicialización *k-means* , cuando se repite varias veces el algoritmo para hallar la solución óptima. Esta variante para la inicialización consiste en lo siguiente:

I. Seleccionar el primer centroide C_1 de manera aleatoria de entre el conjunto de observaciones *X*, asumiendo que cada observación tiene la misma probabilidad de ser escogida que el resto.
II. Calcular las distancias $\boldsymbol{d\ (x,\ C_1)}$ de cada una de las observaciones $x \in X$ al centroide C_1.
III. Seleccionar el siguiente centroide C_2 de manera aleatoria de entre el conjunto de observaciones *X* asumiendo ahora que la probabilidad que tiene de ser escogida una observación $x \in X$, viene dada por la ecuación 3.

Ecuación 3. Expresión de probabilidad de que una muestra x sea escogida como centroide C2.

$$P(x_m) = \frac{d^2(x_m,\ C_1)}{\sum_{i=1}^{n} d^2(x_i, C_1)}$$

IV. Calcular las distancias de cada una de las observaciones *x* ϵ *X* al nuevo centroide recién seleccionado.

V. Seleccionar el siguiente centroide C_j de manera aleatoria de entre el conjunto de observaciones *X* asumiendo ahora que la probabilidad que tiene de ser escogida una observación *x* es proporcional a la ecuación 4.

Ecuación 4. Función de probabilidad de que una muestra sea escogida para ser el siguiente centroide.

$$P(x_m) = \frac{d^2(x_m,\ C_p)}{\sum_{i=1}^{n} d^2(x_i, C_p)}$$

Donde el centroide C_p que se use para cada x_m o x_i será el que le produce una menor distancia de entre los centroides ya calculados.

VI. Se itera sobre el paso IV y V hasta hallar los *k* Centroides.

2.3.2.2. Clasificación

Para cada observación *x* ϵ *X* se calcula la distancia a los centroides y se asigna la observación al centroide con el que tenga una menor distancia, y por ende asignándolo al clúster que el centroide representa.

2.3.2.3. Recálculo de los centroides

Se recalculan los *k* nuevos centroides tomando el promedio de las observaciones de cada uno de los *k* grupos.

2.3.2.4. Iteración o convergencia

Se repiten los pasos 2 y 3 hasta que la asignación de las observaciones a sus clústeres no cambie.

Un ejemplo sencillo de la aplicación de clúster sería el siguiente, realizado con los datos de mostrados en la tabla 1.

Dado el siguiente conjunto de observaciones de partículas aleatorias en un espacio de $\mathbb{R}^3$ *se desea catalogarlas en 2 clústeres:*

Tabla 1. Muestras para ejemplo del funcionamiento de k-means.

Observación	**X**	**Y**	**Z**
A	5,2	3,4	1,4
B	4,8	3,1	1,6
C	7,1	3,3	4,9
D	6,4	3,2	4,6
E	6,4	3,3	5,7
F	5,8	3,7	5,2

1. Se escogen aleatoriamente 2 centroides para los clústeres provenientes de las observaciones, en este caso se escogerá la observación 1 y 2 como centroides, presentados en la tabla 2.

Tabla 2. Centroides del ejemplo.

	X	**Y**	**Z**
Centroide 1	5,2	3,4	1,4
Centroide 2	4,8	3,1	1,6

2. Se calculan las distancias de las observaciones a cada uno de los centroides, para el presente caso se usará la Distancia Euclidiana Cuadrática, que se presenta en la ecuación 5.

Ecuación 5. Expresión para el cálculo de la Distancia Euclidiana Cuadrática.

$$d(P_1, P_2) = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2$$

Calculando con la expresión anterior las distancias y asignando las observaciones según la menor distancia calculada se tienen los resultados presentados en la tabla 3.

Tabla 3. Distancia entre muestras al centroide.

Observación	**Distancia a C1**	**Distancia a C2**	**Clúster**
A	0	0,29	1
B	0,29	0	2
C	15,87	16,22	1
D	11,72	11,57	2
E	19,94	19,41	2
F	14,89	14,32	2

3. Se recalculan los centroides para cada clúster obteniendo el promedio de las observaciones que resultaron clasificadas dentro de cada grupo, siendo los nuevos centroides los mostrados en la tabla 4.

Tabla 4. Centroides recalculados para el ejemplo.

	X	**Y**	**Z**
Centroide 1	6,15	3,35	3,15
Centroide 2	5,85	3,325	4,275

4. Se repiten los pasos 2 y 3 en lo que correspondería a la segunda iteración y se observa si es que existe algún cambio en la clasificación de las observaciones a cada clúster, conforme a los datos presentados en la tabla 5.

Tabla 5. Resumen de la segunda iteración para el ejemplo.

Observación	**X**	**Y**	**Z**	**Distancia a C1**	**Distancia a C2**	**Clúster**
A	5,2	3,4	1,4	3,9675	8,69375	1
B	4,8	3,1	1,6	4,2875	8,30875	1
C	7,1	3,3	4,9	3,9675	1,95375	2
D	6,4	3,2	4,6	2,1875	0,42375	2
E	6,4	3,3	5,7	6,5675	2,33375	2
F	5,8	3,7	5,2	4,4475	0,99875	2
Centroide 1	6,15	3,35	3,15			
Centroide 2	5,85	3,36	4,28			

5. Como se observa que existe un cambio de grupo para la observación B y C se calculan nuevos centroides y realiza una tercera iteración, que se presenta en la tabla 6.

Tabla 6. Resumen de la tercera iteración para el ejemplo.

Observación	**X**	**Y**	**Z**	**Distancia a C1**	**Distancia a C2**	**Clúster**
A	5,2	3,4	1,4	0,0725	15,19125	1
B	4,8	3,1	1,6	0,0725	14,96625	1
C	7,1	3,3	4,9	15,9725	0,50125	2
D	6,4	3,2	4,6	11,5725	0,28125	2
E	6,4	3,3	5,7	19,6025	0,36625	2
F	5,8	3,7	5,2	14,5325	0,50625	2
Centroide 1	5	3,25	1,5			
Centroide 2	6,43	3,38	5,1			

Como no existe algún cambio en la clasificación de las observaciones, se da por concluido el algoritmo.

2.4. Estratificación de usuarios

Con los datos obtenidos se procede a realiza el algoritmo *k-means*, primero definiendo k igual a 3, de modo que se obtengan 3 clústeres en función de lo solicitado por la empresa distribuidora CNEL EP UN Guayaquil. En la ilustración 12 se muestra el código empleado para la creación de los clústeres.

2.4.1. Estratos de CNEL EP UN Guayaquil

2.4.1.1.Estratificación de usuarios con datos normalizados

La estratificación se realizó mediante el uso de la función *k-means* de Matlab, misma que devuelve como resultado un vector con el número de usuarios e índices, 1, 2 y 3, según cual clúster está ubicado cada usuario, de modo que tenemos cada clúster y su respectivo centroide. A continuación, tomamos los centroides, los cuales se calcularon como datos normalizados, y se lo regresa a valores reales, mediante el despeje de la ecuación 2. Se plotea la gráfica que muestre los 3 clústeres y sus respectivos centroides en la ilustración 12.

```
%% Gráfica K=3
[idx_clus3,Cntroid3]=kmeans(Data_norm,3);
Cent_real3=[];
for n=1:length(Cntroid3)
    Cent_real3(:,n)=Cntroid3(:,n)*(max_data(n)-min_data(n))+min_data(n);
end
Cent_avg3=mean(Cent_real3,2);
Cent_std3=std(Cntroid3,0,2);
figure()
plot(cons_data(idx_clus3==1),std_data(idx_clus3==1),'r.','MarkerSize',12)
hold on
plot(cons_data(idx_clus3==2),std_data(idx_clus3==2),'b.','MarkerSize',12)
plot(cons_data(idx_clus3==3),std_data(idx_clus3==3),'g.','MarkerSize',12)
plot(Cent_avg3,Cent_std3,'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Centroids')
title 'Cluster Assignments and Centroids'
hold off
clearvars n
```

Ilustración 12. Código para la creación de Clústeres con k=3.

En la ilustración 13 se muestra el código para el resumen del proceso de estratificación con k=3.

```
%% Resumen k=3
array_res=[];
for i=1:3
    tot=length(cons_data(idx_clus3==i));
    max_rang=round(max(cons_data(idx_clus3==i)),2);
    min_rang=round(min(cons_data(idx_clus3==i)),2);
    line=[i min_rang max_rang tot];
    array_res=[array_res;line];
end
T_res3=array2table(array_res,'VariableNames',{'Cluster' 'Cons_min' 'Cons_max' 'Total_usuarios'});
clearvars i tot max_rang min_rang line array_res
```

Ilustración 13. Resumen de la estratificación para k=3.

Este resumen procesa los resultados del algoritmo *k-means* para obtener el total de usuarios en cada clúster, así como sus valores máximo y mínimo y almacena esta información en una matriz de resultados.

También se procede a realizar la estratificación con k igual a 6, de modo que el algoritmo encuentre 6 clústeres en base al patrón de consumo, el código utilizado para la obtención de los clústeres y sus respectivos centroides se presenta en la ilustración 14.

```
%% Grafica k=6
[idx_clus6,Cntroid6]=kmeans(Data_norm,6,'MaxIter',1000);
Cent_real6=[];
for n=1:length(Cntroid6)
    Cent_real6(:,n)=Cntroid6(:,n)*(max_data(n)-min_data(n))+min_data(n);
end
Cent_avg6=mean(Cent_real6,2);
Cent_std6=std(Cntroid6,0,2);
figure()
plot(cons_data(idx_clus6==1),std_data(idx_clus6==1),'r.','MarkerSize',12)
hold on
plot(cons_data(idx_clus6==2),std_data(idx_clus6==2),'b.','MarkerSize',12)
plot(cons_data(idx_clus6==3),std_data(idx_clus6==3),'g.','MarkerSize',12)
plot(cons_data(idx_clus6==4),std_data(idx_clus6==4),'m.','MarkerSize',12)
plot(cons_data(idx_clus6==5),std_data(idx_clus6==5),'c.','MarkerSize',12)
plot(cons_data(idx_clus6==6),std_data(idx_clus6==6),'y.','MarkerSize',12)
plot(Cent_avg6,Cent_std6,'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Centroids')
title 'Cluster Assignments and Centroids'
hold off
clearvars n
```

Ilustración 14. Código para la estratificación con k=6.

De forma similar a la realizada para k=3, procesamos los datos para obtener los centroides y obtener los 6 clústeres, estos se van a graficar en Matlab y se pueden observar en la ilustración 15.

```
%% Resumen k=6
array_res=[];
for i=1:6
    tot=length(cons_data(idx_clus6==i));
    max_rang=round(max(cons_data(idx_clus6==i)),2);
    min_rang=round(min(cons_data(idx_clus6==i)),2);
    line=[i min_rang max_rang tot];
    array_res=[array_res;line];
end
T_res6=array2table(array_res,'VariableNames',{'Cluster' 'Cons_min' 'Cons_max' 'Total_usuarios'});
clearvars i tot max_rang min_rang line array_res
```

Ilustración 15. Código para el resumen de estratificación k=6.

Este resumen tiene como salida una matriz que resume la cantidad de usuarios, valores máximos y mínimos para cada uno de los clústeres.

2.4.1.2.Estratificación de usuarios con datos reales

Habiendo creado los clústeres con datos normalizados producto del reescalamiento mostrado en la sección 2.2., se procedió a omitir este reescalamiento en función de obtener mejores resultados, al trabajar con los datos reales objeto del estudio realizado, estos datos reales son los que utilizarán a partir de este punto y constituyen el resultado final de este trabajo.

Se realizaó el *clustering* con k=6 utilizando los datos reales, estos se obtienen a partir de la matriz M_final, la cual se obtiene después de eliminar los datos *outliers*, es decir, no se utilizan los datos normalizados. Se utiliza la función *k-means* y se grafican los clústeres con sus respectivos centroides. Este código se muestra en la ilustración 16.

```
%% Grafica k=6
Data=table2array(M_final(:,2:19));
codigo=table2array(M_final(:,1));
max_data=max(Data,[],1);
min_data=min(Data,[],1);
std_data=std(Data,0,2);
std_normal=(std_data-min(std_data))/(max(std_data)-min(std_data));
cons_data=mean(Data,2);
[idx_clus6,Cntroid6,suma6,Dist6]=kmeans(Data,6,'MaxIter',1000,'Replicates',10);
Cent_avg6=mean(Cntroid6,2);
Cent_std6=(std(Cntroid6,0,2)-min(std_data))/(max(std_data)-min(std_data));
figure()
plot(cons_data(idx_clus6==1),std_normal(idx_clus6==1),'r.','MarkerSize',12)
hold on
plot(cons_data(idx_clus6==2),std_normal(idx_clus6==2),'b.','MarkerSize',12)
plot(cons_data(idx_clus6==3),std_normal(idx_clus6==3),'g.','MarkerSize',12)
plot(cons_data(idx_clus6==4),std_normal(idx_clus6==4),'m.','MarkerSize',12)
plot(cons_data(idx_clus6==5),std_normal(idx_clus6==5),'c.','MarkerSize',12)
plot(cons_data(idx_clus6==6),std_normal(idx_clus6==6),'y.','MarkerSize',12)
plot(Cent_avg6,Cent_std6,'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Centroids')
title 'Cluster Assignments and Centroids'
hold off
clearvars n
```

Ilustración 16. Código para la estratificación con datos reales para k=6.

El código mostrado en la ilustración 17 permite obtener información sobre los clústeres creados, estos son el número de usuarios que pertenece a cada clúster, valor mayor y menor, mismos que serán presentados en forma de tabla.

```
%% Resumen k=6 promedio normal
cons_data=mean(Data,2);
array_res=[];
for i=1:6
    tot=length(cons_data(idx_clus6==i));
    max_rang=max(cons_data(idx_clus6==i));
    min_rang=min(cons_data(idx_clus6==i));
    line=[i min_rang max_rang tot];
    array_res=[array_res;line];
end
T_res6_simple=array2table(array_res,'VariableNames',{'Cluster' 'Cons_min' 'Cons_max' 'Total_usuarios'});
T_res6_simple=sortrows(T_res6_simple,'Cons_min');
clearvars i tot max_rang min_rang line array_res

```

Ilustración 17. Código para el resumen de los datos reales para k=6.

En la ilustración 18 se visualiza el código utilizado para realizar la estratificación de 3 clústeres con los datos reales de los usuarios de la empresa distribuidora, estos se grafican de forma similar a la realizada para el caso con k=6.

```
%% Gráfica K=3
[idx_clus3,Cntroid3,suma3,Dist3]=kmeans(Data,3,'Replicates',10);
Cent_avg3=mean(Cntroid3,2);
Cent_std3=(std(Cntroid3,0,2)-min(std_data))/(max(std_data)-min(std_data));
figure()
plot(cons_data(idx_clus3==1),std_normal(idx_clus3==1),'r.','MarkerSize',12)
hold on
plot(cons_data(idx_clus3==2),std_normal(idx_clus3==2),'b.','MarkerSize',12)
plot(cons_data(idx_clus3==3),std_normal(idx_clus3==3),'g.','MarkerSize',12)
plot(Cent_avg3,Cent_std3,'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Centroids')
title 'Cluster Assignments and Centroids'
hold off
clearvars n
```

Ilustración 18. Código para la estratificación con datos reales para k=3.

La ilustración 19 presenta el código empleado para obtener la información más importante para cada clúster, siendo esto valor máximo, mínimo y número de usuarios por cada clúster. Esta información será presentada en forma de tabla en el capítulo 3.

```
%% Resumen k=3 Promedio normal
array_res=[];
cons_data=mean(Data,2);
for i=1:3
    tot=length(cons_data(idx_clus3==i));
    max_rang=round(max(cons_data(idx_clus3==i)),2);
    min_rang=round(min(cons_data(idx_clus3==i)),2);
    line=[i min_rang max_rang tot];
    array_res=[array_res;line];
end
T_res3_simple=array2table(array_res,'VariableNames',{'Cluster' 'Cons_min' 'Cons_max' 'Total_usuarios'});
clearvars i tot max_rang min_rang line array_res
```

Ilustración 19. Código para el resumen de la estratificación con k=3.

Adicionalmente se consideró una estratificación para 3 y 6 grupos que fue propuesta por CNEL EP UN Guayaquil, la cual comprende rangos de valores y cuyos códigos se muestran en las ilustraciones 20 y 21.

```
%% Claificacion actual
% 6 estratos
rangos=['0-200';'200-250';'250-350';'350-600';'600-800';'800-1200';'>1200'];
clasif=[sum(cons_data<=200);...
    sum((cons_data>200)&(cons_data<=250));...
    sum((cons_data>250)&(cons_data<=350));...
    sum((cons_data>350)&(cons_data<=600));...
    sum((cons_data>600)&(cons_data<=800));...
    sum((cons_data>800)&(cons_data<=1200));...
    sum(cons_data>1200)];
Clas6estr=table(rangos,clasif,'VariableNames',{'Rangos' 'Total_Usuarios'});
clearvars rangos clasif
```

Ilustración 20. Clasificación actual para 6 estratos.

```
%% Clasificacion actual
% 3 rangos
rangos=['0-150';'150-500';'>500'];
clasif=[sum(cons_data<=150);...
    sum((cons_data>150)&(cons_data<=500));...
    sum(cons_data>500)];
Clas3estr=table(rangos,clasif,'VariableNames',{'Rangos' 'Total_Usuarios'});
clearvars rangos clasif

```

Ilustración 21. Clasificación actual para 3 estratos.

Esta clasificación se va a presentar en forma de tabla, para poder contrastarse con la estratificación realizada como resultado del presente trabajo. Cabe destacar que esta clasificación no es *clustering*, simplemente es segmentar por grupos a los usuarios en base a su consumo y no a su comportamiento, los mismos límites que se han definido son impuestos por la empresa distribuidora de forma arbitraria y pueden o no responder a la realidad de sus usuarios.

CAPÍTULO 3

Resultados y análisis

3. RESULTADOS Y ANÁLISIS

3.1.Resultados con datos normalizados

Para el proceso con datos normalizados se presentan los gráficos de los clústeres con sus respectivos centroides realizados en Matlab.

La ilustración 22 muestra los 3 clústeres producto de aplicar el algoritmo, las cruces negras son los centroides de cada uno, como se puede observar los estratos están divididos por colores, donde cada punto es un usuario de la empresa distribuidora y el algoritmo los agrupa en la cercanía de cada centroide.

3.1.1. Resultados para la estratificación con k=3

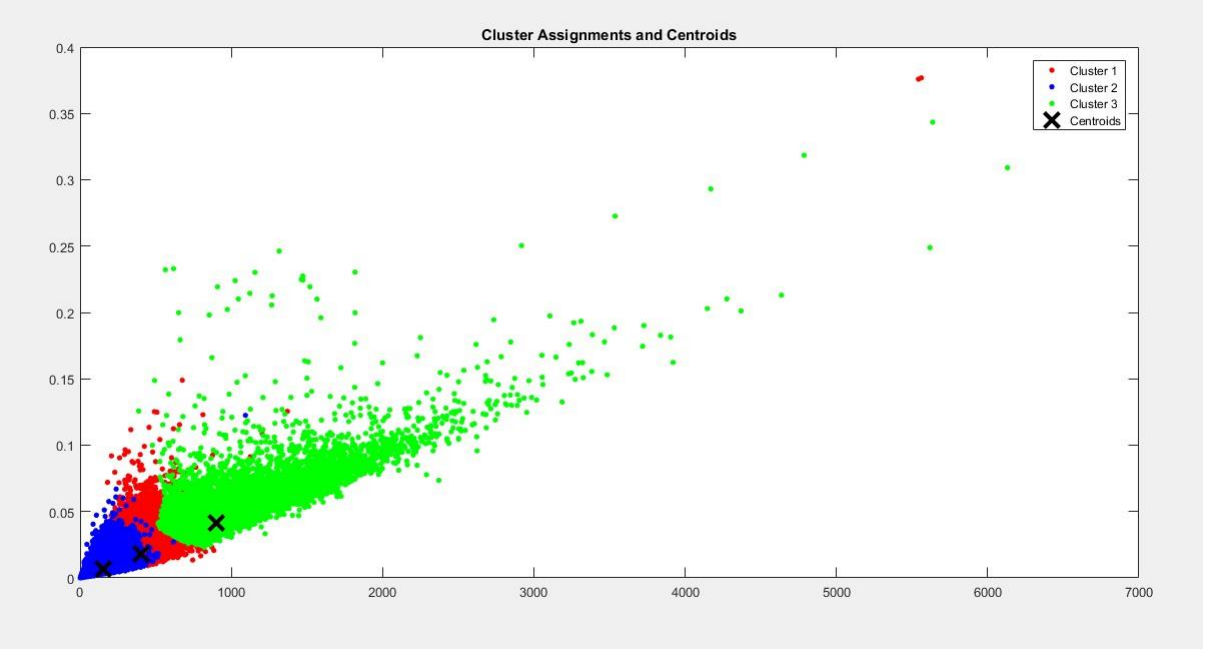


Ilustración 22. Gráfico de la estratificación para k=3.

Así mismo es evidente que están repartidos en función de sus consumos, sin embargo, existen datos que pertenecen a un clúster y que se encuentran alejados de los demás valores de ese

clúster. Estos datos pertenecen al clúster porque el algoritmo identifica que su patrón de consumo está relacionado con los otros usuarios de este clúster, aunque su consumo pueda ser notablemente mayor.

Exceptuando este tipo de datos, la mayoría de los usuarios agrupados en un clúster tienen no sólo patrones de consumo similares sino también consumos mensuales parecidos, lo mismo que puede ser relacionado con estar dentro del mismo estrato social.

El resumen de los resultados obtenidos para esta estratificación se presenta en la tabla 7.

Tabla 7. Resumen de estratificación de usuarios para k=3 con datos normalizados.

Clúster	***Consumo Mínimo (KWh)***	***Consumo máximo (KWh)***	***Total de usuarios***
1	175,83	5561	132927
2	1,11	1093,22	341901
3	386,83	6131,11	24006

En la tabla 7 se tiene la información recopilada de los clústeres donde se evidencia que el clúster con mayor número de usuarios es el segundo, de color rojo en la ilustración 22. Así mismo se evidencia que el valor de consumo más pequeño pertenece justamente a este estrato, mientras que el valor máximo es mayor al valor mínimo del clúster 3 que es el que le sigue.

El clúster 3 tiene el menor número de usuarios, lo que corresponde a la realidad, donde los usuarios con los valores más altos de consumo se relacionan con la clase social más alta, los mismos que son minoría en la sociedad. Dicho eso se tiene que resaltar que el valor más bajo de este clúster es menor que el máximo de los otros 2 clústeres, esto debido a aquellos valores que, si bien no encajan dentro del criterio de consumo, son parte del clúster por su patrón de consumo.

3.1.2. Resultados para la estratificación con k=6

La ilustración 23 muestra el gráfico de los 6 clústeres con datos normalizados, estos clústeres y sus centroides fueron hallados por el algoritmo en base a sus patrones de consumo, los mismos a simple vista respetan el criterio de consumos similares, sin embargo, es notable que existen valores que se traslapan entre un clúster y otro.

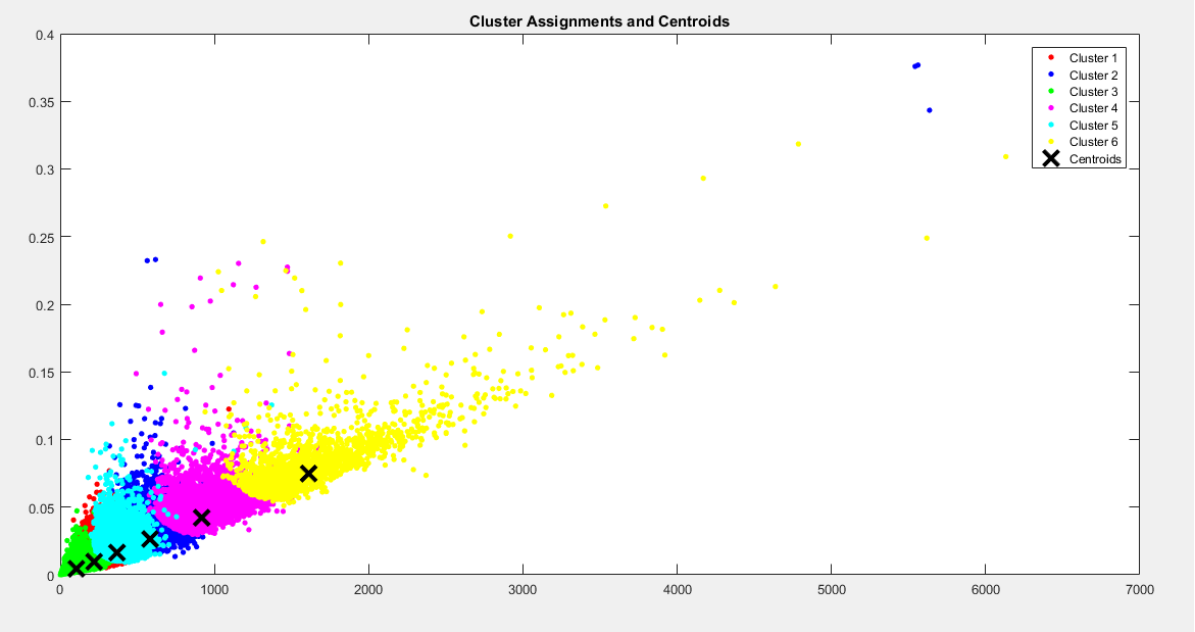


Ilustración 23. Gráfica de la estratificación para k=6 con datos normalizados.

El resumen de esta estratificación se muestra en la tabla 8.

Tabla 8. Resumen de la estratificación para k=6 con datos normalizados.

Clúster	***Consumo Mínimo (KWh)***	***Consumo máximo (KWh)***	***Total de usuarios***
1	86,39	1093,22	164830
2	319,28	5635,94	40163
3	1,11	447,67	190440
4	492,5	1673,06	13298
5	181,78	1370,89	87968
6	939,28	6131,11	2135

De la tabla 8 se puede obtener la información del número de usuarios, donde el clúster con mayor cantidad de clientes es el 3, mientras que el que tiene el menor número de usuarios es el clúster 6, mismo que representa el consumo más alto entre todos los demás, por lo que son el estrato social más alto en esta distribución si nos guiamos por el consumo energético.

Existe el traslape de un clúster a otro, debido a los datos que se alejan más del centroide a pesar de tener un valor mayor de consumo.

3.2. Resultados sin normalizar

Por lo presentado en la sección 3.1, la estratificación con datos normalizados arrojó resultados que no satisfacen al presente trabajo. Se va a realizar la estratificación con datos reales, es decir, saltando la parte de la normalización, pero si se elimina de las muestras originares ceros, repetidos y *outliers*.

3.2.1. Resultados para la estratificación con k=3

En la ilustración 24 se muestra la estratificación en 3 clústeres, esta se ha realizado con los datos reales, sin normalizar. Claramente se aprecia una mejor separación de los datos, de modo que se observan todos los datos correctamente divididos en los 3 grupos y sus respectivos centroides.

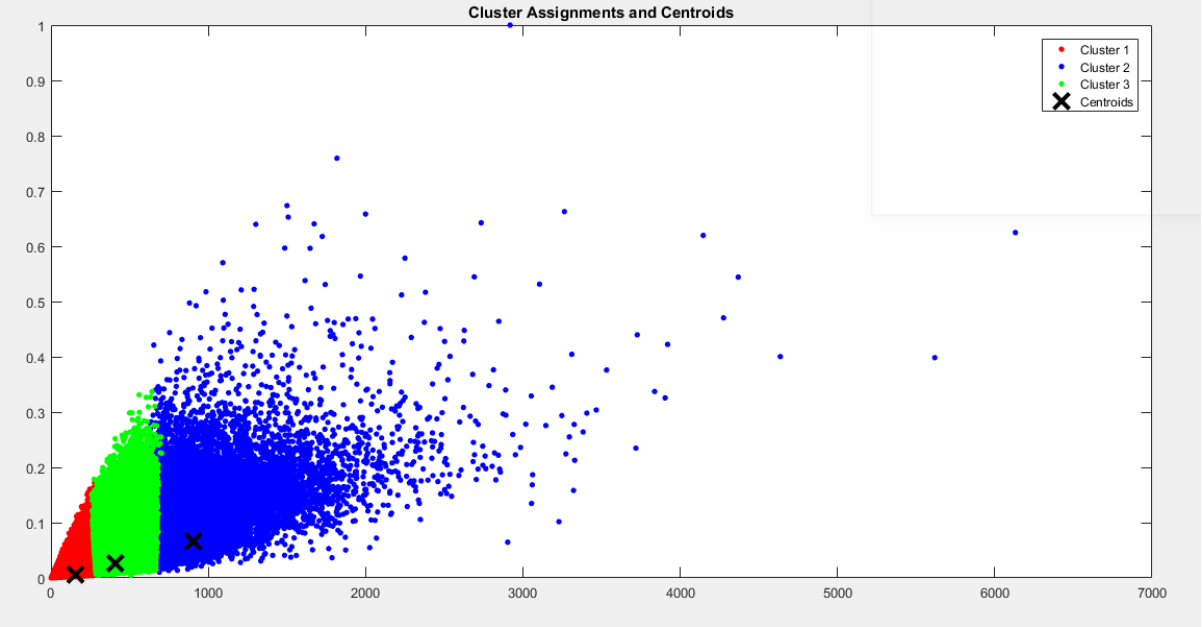


Ilustración 24. Estratificación de usuarios para k=3 con datos reales.

Observando con más atención se puede ver que aún existen datos que se traslapan de un clúster a otro, aunque este problema se presenta en mucha menor medida que en el caso de los datos normalizados. Esta estratificación arrojó un resultado mucho más satisfactorio que el anterior, lo que plantea que es mejor trabajar con los datos reales.

La ilustración 25 presenta la estratificación realiza con datos reales para k=6. De modo que se obtuvo un resultado a simple vista es más favorable que el obtenido con datos normalizados. Estos 6 clústeres aún presentan traslape entre grupos, sin embargo, esto se da en mucho menor medida, los centroides y los datos de cada clúster no exceden distancias mayores como en el caso de datos normalizados donde los datos de un clúster podían aparecer junto a los datos de 2 clústeres siguientes o posteriores.

3.2.2. Resultados para la estratificación con k=6

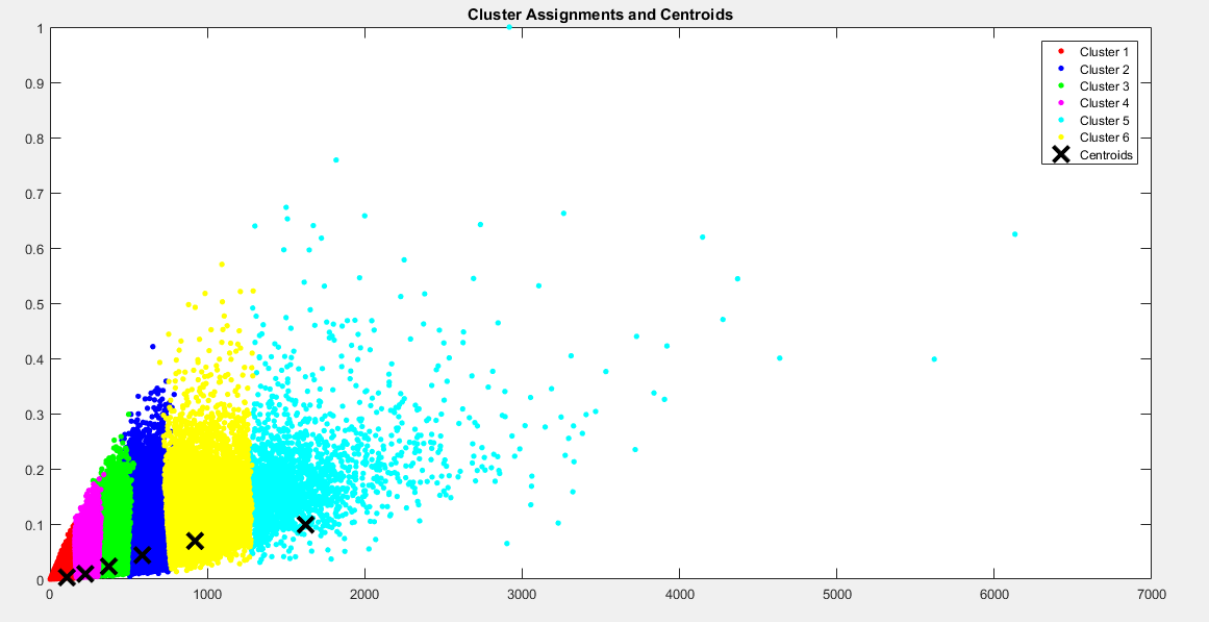


Ilustración 25. Estratificación de usuarios para k=6 con datos reales.

Esto comprobó que es más factible trabajar con los datos reales, dado que los datos normalizados, si bien tienen un sentido dentro de la dinámica de la minería de datos y el procesamiento de información, no aporta un impacto positivo real en la estratificación presentada en este trabajo.

A partir de este punto se va a trabajar únicamente con los datos reales y los ajustes que se realicen tendrán como base estos resultados preliminares obtenidos en la sección 3.2.

Como se vio en las ilustraciones 24 y 25, si bien los clústeres se ven bien definidos a primera vista, existe aún traslape de un clúster a otro. Esto se va a tratar a continuación.

3.3. Redistribución de datos.

Si bien el algoritmo agrupa los datos por patrón de consumo, dentro de la práctica lo que se espera es tener clústeres de tal manera que los usuarios estén estratificados de forma secuencial, si, por su patrón de consumo, pero también por el valor de ese consumo, es decir, con rangos bien definidos de tal manera que los clústeres sean secuenciales.

Para efectos prácticos es conveniente que donde termine un clúster comience el siguiente, considerando que así funcionaban los rangos propuestos por la empresa distribuidora. Ambos criterios, tanto el patrón de consumo como la cantidad del consumo son relevantes, por lo que se realizó un ajuste a los resultados previamente obtenidos para estratificación con datos reales.

3.3.1. Redistribución de datos con k=3.

El código presentado en la ilustración 26 tiene por finalidad obtener los rangos en los que están los clústeres y obtener los percentiles para los mismos.

```
93      %% Redistribución con Datos simples
94 -    cons_data=mean(Data,2);
95 -    array_res=[];
96 -    for i=1:3
97 -        miembros=cons_data(idx_clus3==i);
98 -        Y=prctile(miembros,[4 99]);
99 -        array_res=[array_res;[i Y(1) Y(2)]];
100 -   end
101 -   New3_rang=array2table(array_res,'VariableNames',{'Cluster' 'Min' 'Max'});
102 -   New3_rang=sortrows(New3_rang,'Min');
103 -   info=table2array(New3_rang);
104 -   New3_rang(1,2)={min(cons_data(idx_clus3==info(1,1)))};
105
106
107 -   clearvars i array_res Y miembros info
```

Ilustración 26. Redistribución de datos simples con k=3.

Con los cuales se procede a hacer la redistribución de las muestras como se muestra en la ilustración 27.

```
108      %% Armado de tablas
109 -    array_res=[];
110 -    data_no_out=[];
111 -    desglose=[];
112 -    Data_clas=table2array(New3_rang);
113
114 -  for i=1:2...
130 -    clust=Data_clas(3,1);
131 -    low=Data_clas(3,2);
132 -    up=Data_clas(3,3);
133 -    idx_cumple=(cons_data>=low)&(cons_data<up)&(idx_clus3==clust);
134 -    std_fin=std_normal(idx_cumple);
135 -    cod_fin=codigo(idx_cumple);
136 -    Data_fin=Data(idx_cumple,:);
137 -    cons_fin=mean(Data_fin,2);
138 -    avg=mean(cons_fin);
139 -    n=sum(idx_cumple);
140 -    tot_clus=sum(idx_clus3==clust);
141 -    desglose=[desglose;clust*ones(length(Data_fin),1) cod_fin Data_fin];
142 -    data_no_out=[data_no_out; clust*ones(length(cons_fin),1) cons_fin std_fin];
143 -    array_res=[array_res;[clust low avg up n (n/tot_clus)*100]];
144 -    New3_clust=array2table(array_res,'VariableNames',{'Cluster' 'Min' 'Promedio' 'Max' 'Total_usuarios' 'Percent_Cluste
145 -    clearvars clust low up idx_cumple cons_fin cod_fin std_fin Data_fin avg  tot_clus i array_res n
```

Ilustración 27. Armado de tablas para k=3.

En la ilustración 27 se muestra en código para el armado de tablas, estas tablas son los mismos clústeres que ya están armados con datos reales, lo que se hizo con estos datos es identificar los valores mínimo y máximo de cada clúster de manera que el máximo del clúster 1 sea el nuevo mínimo del clúster 2. Así mismo, el mínimo del clúster 3 será el máximo del clúster 2, de modo que ya no existan datos que se traslapen entre clústeres.

Aquellos datos que pertenezcan al clúster 2, pero que estén por debajo del máximo del clúster 1 ya no van a pertenecer al clúster 2. Si bien su patrón de consumo fue identificado por el algoritmo como parte del clúster 2, estos datos son los que producen un traslape, por lo que se optó por cambiar a estas muestras de clúster, es decir, ya no pertenecerán al clúster 2 sino al clúster 1.

Este movimiento no afecta el procedimiento realizado con *k-means* ni contradice los resultados obtenidos, puesto que son pocas las muestras que se desplazarán de un clúster a otro. En efecto, la mayoría de las muestras contemplan ambos criterios, por lo que el procedimiento de redistribución sólo contempla un pequeño reajuste con criterios de practicidad.

El código presentado en la ilustración 28 realiza la gráfica de los clústeres redistribuidos para la estratificación con k=3.

```
%% Grafica 3 cluster Redistribuidos
cons=data_no_out(:,2);
std=data_no_out(:,3);
idx=data_no_out(:,1);
figure()
plot(cons(idx==1),std(idx==1),'r.','MarkerSize',12)
hold on
plot(cons(idx==2),std(idx==2),'b.','MarkerSize',12)
plot(cons(idx==3),std(idx==3),'g.','MarkerSize',12)
plot(Cent_avg3,Cent_std3,'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Centroids')
title 'Cluster Assignments and Centroids'
hold off
clearvars idx cons std
```

Ilustración 28. Código para graficar los clústeres redistribuidos con k=3.

La ilustración 29 permite visualizar los clústeres resultantes de la estratificación con k=3 pero con la redistribución de datos. Se pudo observar de manera clara la segmentación entre clústeres.

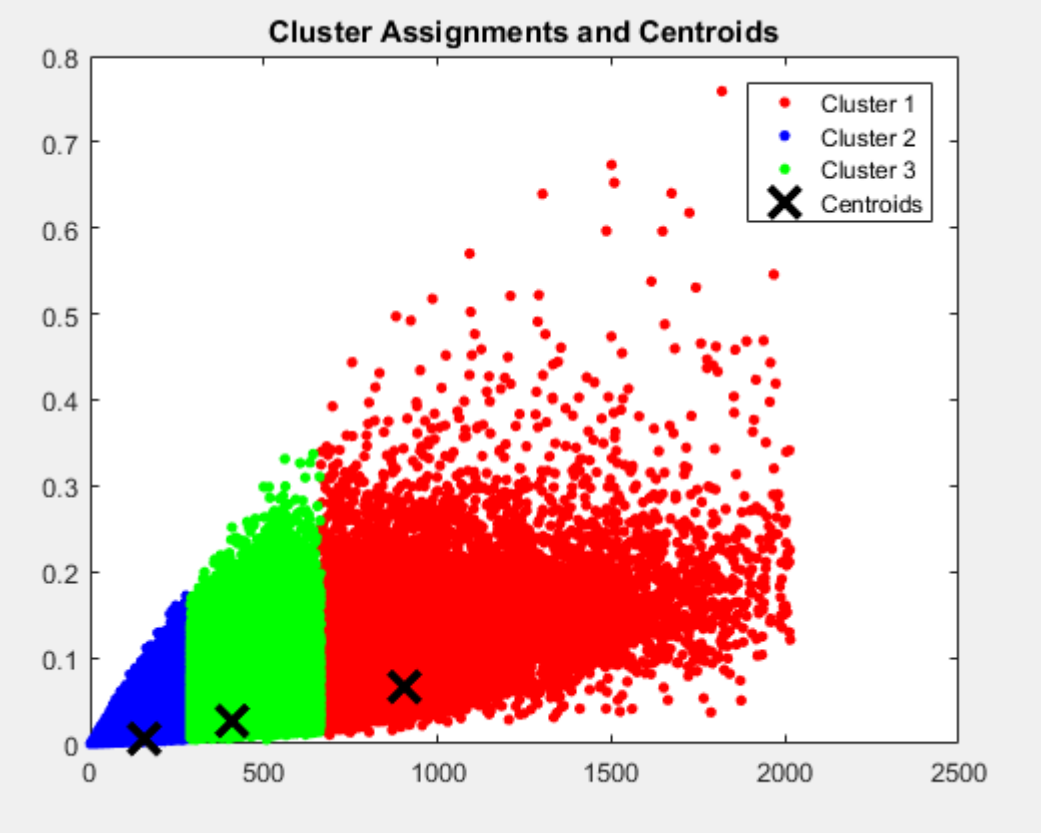


Ilustración 29. Clústeres redistribuidos con k=3.

La información relevante los 3 clústeres redistribuidos se presenta en la tabla 9.

Tabla 9. Clústeres para la estratificación con k=3 y datos redistribuidos.

Clúster	***Consumo mínimo (KWh)***	***Consumo máximo (KWh)***	***Consumo Promedio (KWh)***	***Número de usuarios***	***Porcentaje de usuarios incluidos***
1	1	288,56	155,71	311610	99,92
2	288,56	663,96	413,58	117478	95,91
3	663,96	2015,16	898,10	22057	94,99

Como se observa en la información presentada en la tabla 9, el máximo de un clúster se convierte en el mínimo del clúster siguiente por lo que ya no existe traslape entre estos, así mismo se puede extraer el promedio de consumo para ese clúster y el número de usuarios en cada uno de estos estratos.

El comportamiento de los clústeres muestra una mayor cantidad de usuarios en el estrato de menor consumo, el cual asciende hasta los 288.56 kWh promedio mensuales, el cuál es un consumo regular para hogares de clase media en la demografía ecuatoriana. Por número de usuarios el que menor número de clientes contiene es el clúster 3, correspondiente a los usuarios con mayor consumo, donde su promedio es cercano a los 900 kWh, un consumo bastante elevado y que a su vez corresponde a un gasto considerable en la planilla eléctrica, por lo que cabe decir que este estrato representa a hogares pudientes de la demografía guayaquileña.

3.3.2. Redistribución de datos con k=6

El proceso para la redistribución con 6 estratos se hace de manera similar a como se hizo con 3, los códigos utilizados para conseguir que el mínimo de un clúster sea el máximo del anterior, de modo que los usuarios que quedan fuera de esa condición se reagrupen en el clúster que les corresponde, se presentan en las ilustraciones 30 y 31.

```
112      %% Redistribución con Datos simples
113 -    cons_data=mean(Data,2);
114 -    array_res=[];
115 -    for i=1:6
116 -        miembros=cons_data(idx_clus6==i);
117 -        Y=prctile(miembros,[4 99]);
118 -        array_res=[array_res;[i Y(1) Y(2)]];
119 -    end
120 -    New6_rang=array2table(array_res,'VariableNames',{'Cluster' 'Min' 'Max'});
121 -    New6_rang=sortrows(New6_rang,'Min');
122 -    info=table2array(New6_rang);
123 -    New6_rang(1,2)={min(cons_data(idx_clus6==info(1,1)))};
124
125
126 -    clearvars i array_res Y miembros info
```

Ilustración 30. Código para la redistribución de datos para k=6.

```
%% Armado de tablas
array_res=[];
data_no_out=[];
desglose=[];
Data_clas=table2array(New6_rang);
for i=1:5...
clust=Data_clas(6,1);
low=Data_clas(6,2);
up=Data_clas(6,3);
idx_cumple=(cons_data>=low)&(cons_data<up)&(idx_clus6==clust);
std_fin=std_normal(idx_cumple);
cod_fin=codigo(idx_cumple);
Data_fin=Data(idx_cumple,:);
cons_fin=mean(Data_fin,2);
avg=mean(cons_fin);
n=sum(idx_cumple);
tot_clus=sum(idx_clus6==clust);
desglose=[desglose;clust*ones(length(Data_fin),1) cod_fin Data_fin];
data_no_out=[data_no_out; clust*ones(length(cons_fin),1) cons_fin std_fin];
array_res=[array_res;[clust low avg up n (n/tot_clus)*100]];
New6_clust=array2table(array_res,'VariableNames',{'Cluster' 'Min' 'Promedio' 'Max' 'Total_usuarios' 'Percent_Cluste
clearvars clust low up idx_cumple cons_fin cod_fin std_fin Data_fin avg  tot_clus i array_res n
```

Ilustración 31. Código para el armado de tablas con k=6.

Con estos códigos se redistribuyen los clústeres y con el código de la ilustración 32 se obtiene la gráfica de los clústeres redistribuidos que se presentan en la ilustración 33.

```
%% Grafica 6 cluster Redistribuidos
cons=data_no_out(:,2);
std=data_no_out(:,3);
idx=data_no_out(:,1);
figure()
plot(cons(idx==1),std(idx==1),'r.','MarkerSize',12)
hold on
plot(cons(idx==2),std(idx==2),'b.','MarkerSize',12)
plot(cons(idx==3),std(idx==3),'g.','MarkerSize',12)
plot(cons(idx==4),std(idx==4),'m.','MarkerSize',12)
plot(cons(idx==5),std(idx==5),'c.','MarkerSize',12)
plot(cons(idx==6),std(idx==6),'y.','MarkerSize',12)
plot(Cent_avg6,Cent_std6,'kx','MarkerSize',15,'LineWidth',3)
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Centroids')
title 'Cluster Assignments and Centroids'
hold off
clearvars idx cons std
```

Ilustración 32. Código para la obtención de la gráfica de los clústeres redistribuidos con k=6.

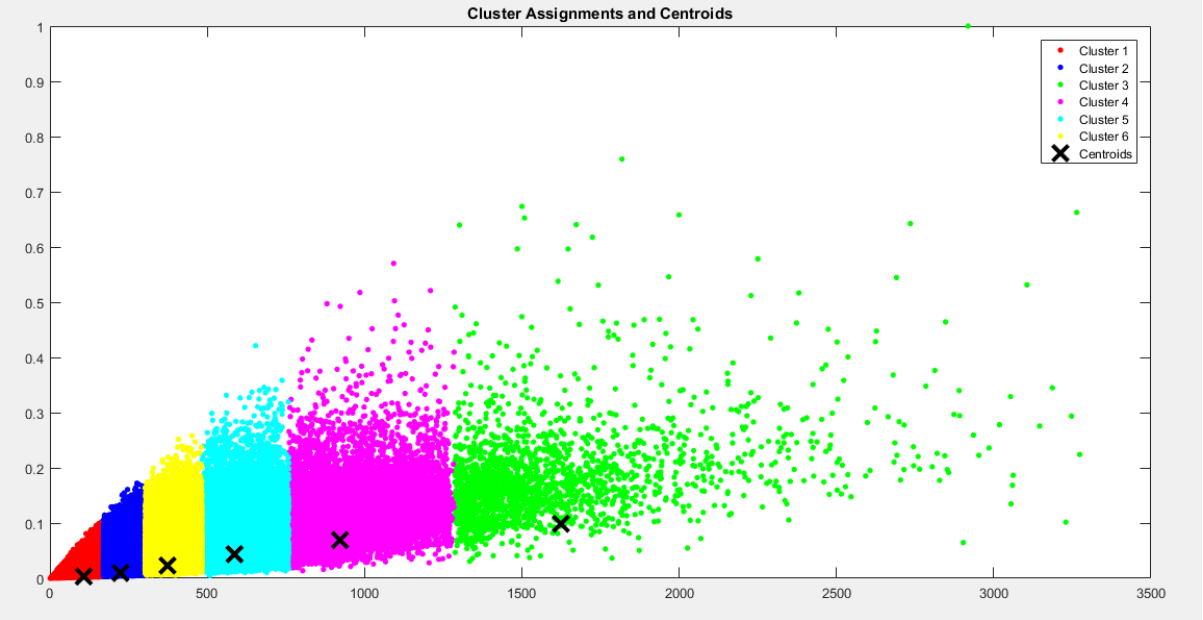


Ilustración 33. Gráfica de los clústeres redistribuidos con k=6.

En el gráfico de la ilustración 33 se tiene la gráfica con los clústeres resultantes después de la redistribución, claramente se ve que los clústeres no se traslapan entre sí gracias al código de la ilustración 31. La información relevante de esta redistribución se presenta en la tabla 10.

Tabla 10. Información relevante de los clústeres redistribuidos con k=6.

Clúster	***Consumo mínimo (KWh)***	***Consumo máximo (KWh)***	***Consumo Promedio (KWh)***	***Número de usuarios***	***Porcentaje de usuarios incluidos***
1	1	170,05	107,49	175768	99,95
2	170,05	302,83	226,49	142135	95,59
3	302,83	484,11	376,01	76568	95,37
4	484,11	759,61	590,65	36093	95,70
5	759,61	1286,04	928,87	12479	95,99
6	1286,04	3282,18	1614,80	1910	95,03

Mientras que la estratificación propuesta por la empresa distribuidora, como se mencionó anteriormente, no corresponde a un trabajo de *clustering*, sino a agrupar a los usuarios en base a su consumo de manera arbitraria, estos rangos no están definidos por un algoritmo, ni

los usuarios que pertenecen a cada estrato están agrupados por patrón de consumo. El resumen de esta estratificación se presenta en la tabla 11.

Tabla 11. Estratificación propuesta por CNEL EP, 6 estratos.

Consumo mínimo (KWh)	***Consumo máximo (KWh)***	***Número de Usuarios***
0	200	224395
200	250	58171
250	350	74056
350	600	70964
600	800	17700
800	1200	9671
>1200		2590

En la tabla 12 se presentan los resultados para la estratificación propuesta por la distribuidora con 3 estratos, obtenida de manera similar.

Tabla 12. Estratificación propuesta por CNEL EP, 3 estratos.

Consumo mínimo (KWh)	***Consumo máximo (KWh)***	***Número de Usuarios***
0	150	150261
150	500	259379
>500		47907

Al comparar las tablas 9 y 10 con las tablas 11 y 12 se evidencia la diferencia existente entre definir rangos de consumo de forma arbitraria y utilizar un método de *machine learning* como lo es el algoritmo *k-means*, donde este define los clústeres que no sólo están agrupados por cantidad de consumo sino también por patrón de consumo, es decir, no consideran únicamente el cuanto sino también el cómo están consumiendo los usuarios el suministro de energía eléctrica.

3.4. Potencialidad del procedimiento.

Dentro de lo mencionado en la sección 1.1.2. sobre la eliminación de *outliers*, el algoritmo elaborado para la solución del *clustering* permite la parte de detección de valores aberrantes mediante el análisis y detección visual de los resultados por clúster.

Los valores aberrantes pueden ser indicativos de un error en la medición del consumo mensual de un cliente, ya sea este un error humano o un error causado por el medidor. A continuación, se muestra un ejemplo de los valores atípicos que constan en la base de datos de usuarios del consumo mensual en la ilustración 34.

	6 201905	7 Cons_201906	8 Cons_201907	9 Cons_201908	10 Cons_201909	11 Cons_201910	12 Cons_201911	13 Cons_201912	14 Cons_202001	15 Cons_202002	16 Cons_202003	17 Cons_202004	18 Cons_202005	19 Cons_202006
1	1	4	3	6	1	10	10	3	4	8	4	34064	31866	34065
2	5	4	5	4	2	6	6	5	2	2	2	33596	33597	32513
3	180	128	104	131	51	30	25	50	23	11	24998	24192	24192	26612
4	375	388	375	414	413	363	208	202	183	256	244	261	236	19105
5	320	444	198	276	328	309	339	298	319	241	150	236	237	16621
6	28	26	20	25	20	21	2867	3	21	27	29	25	26	16436
7	508	507	514	551	579	510	674	599	312	6878	24514	16691	16265	15216

Ilustración 34. Datos atípicos en base de datos para meses puntuales

Estos valores pueden ser identificados fácilmente de manera gráfica como se muestra en la ilustración 35.

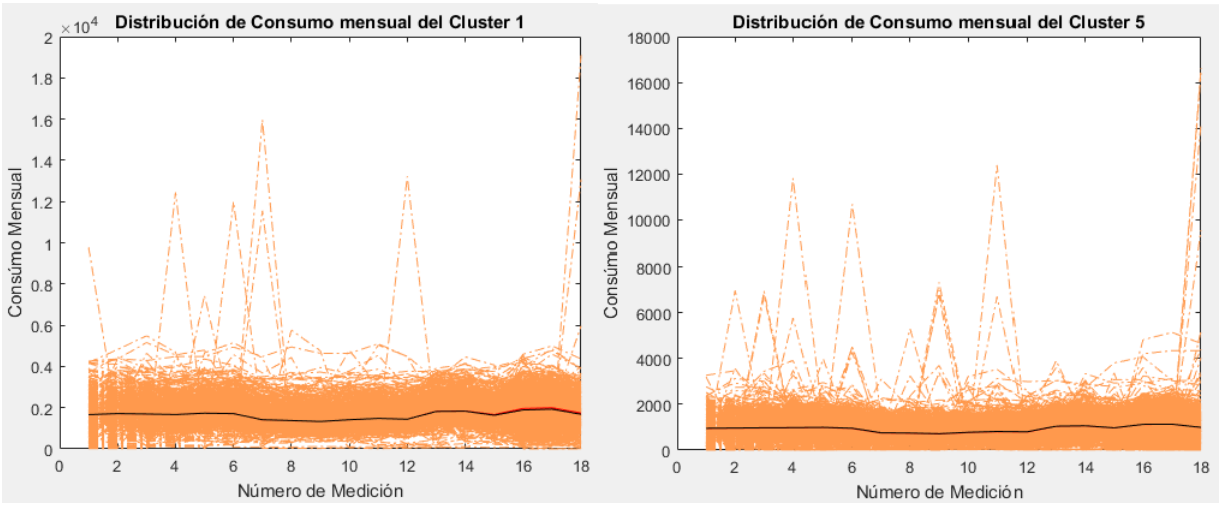


Ilustración 35. Consumo por clustering con datos atípicos

Ya con la eliminación de los datos atípicos se puede observar una mejor redistribución de los usuarios con respecto a su centroide o comportamiento típico del clúster, como se evidencia en la ilustración 36.

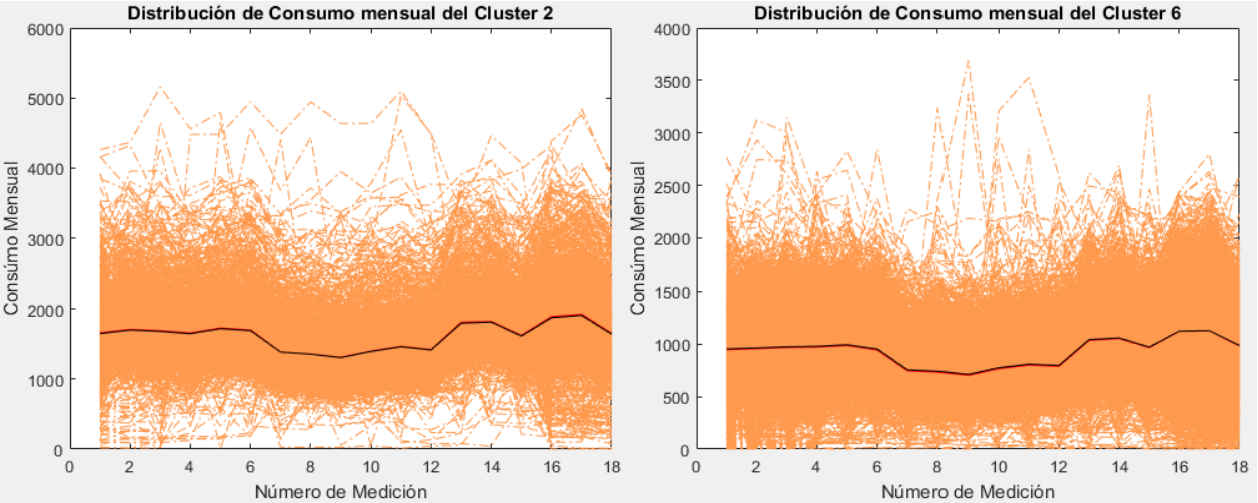


Ilustración 36. Consumo por clustering sin datos atípicos

CAPÍTULO 4

Conclusiones y Recomendaciones

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Se realizó la estratificación de usuarios a nivel residencial de la empresa eléctrica distribuidora CNEL EP Unidad de Negocios Guayaquil, para lo cual se realizó un filtrado de datos que elimine a los usuarios con valores de cero, valores iguales todos los meses y usuarios con patrones atípicos de consumo, a fin de obtener dos estratificaciones, una dividida en tres grupos y otra en seis, conforme a lo acordado con la empresa distribuidora.
- El proceso de estratificación permitió no sólo clasificar a los usuarios por grupos sino también el definir los límites de consumo que acotan a cada estrato, misma estratificación que, para la distribuidora, antes consistía en separar a los usuarios en grupos según su rango de consumo promedio, en rangos definidos por la misma empresa. Esta nueva estratificación realizada mediante *machine learning* revela rangos de consumo como parte del mismo algoritmo, es decir, no son impuestos de forma arbitraria bajo ningún sesgo, sino que son un resultado del mismo trabajo.
- Habiéndose aplicado el algoritmo *k-means* y después de redistribuir a los usuarios para definir los límites de consumo de cada estrato, se obtuvo la información resumida de cada estrato, donde también se encontraba el número de usuarios pertenecientes a cada uno, donde para k=3 se evidencia que el estrato predominante es el número 1. Mientras que para k=6 se repite el primer clúster como predominante, seguido de cerca por el estrato 2. Estos resultados muestran que la mayor cantidad de usuarios de la empresa tienen consumos promedio de entre 100 a 150 kWh,

aproximadamente. El consumo de energía eléctrica se ha considerado como una forma de interpretar la clase social de los usuarios, de modo que un consumo menor de energía corresponde a la clase baja y un consumo mayor a la clase alta, considerando la capacidad adquisitiva de esta última para obtener equipos eléctricos y electrónicos. La estratificación realizada en este trabajo denota que la mayor cantidad de usuarios de la empresa distribuidora pertenecen a la clase baja y media-baja de la ciudad de Guayaquil.

- Se obtuvo el patrón de consumo de cada uno de los estratos para ambas estratificaciones, este permite visualizar el comportamiento de los usuarios que pertenecen al estrato, donde los usuarios con comportamientos atípicos destacan por presentar picos en sus curvas, si bien estos usuarios son clasificados en el estrato por el algoritmo, estos usuarios son potenciales fuentes de pérdidas no técnicas, es decir, son usuarios que pueden tener errores en lecturas, medidores dañados o con alteraciones intencionales, por lo que esto da la potencialidad a la realización de nuevos estudios que permitan identificar a estos usuarios con el fin de que la distribuidora tome acciones respecto a estos.

4.2. Recomendaciones

- Es recomendable mantener actualizada esta estratificación, si bien fue realizada en un periodo de estudio de 18 meses de consumo de la empresa distribuidora, este tiempo puede variar en función de la disponibilidad de la información, por lo que replicar este estudio para meses posteriores permitiría comparar los resultados y ver como evolucionan los estratos y el comportamiento de los usuarios residenciales en Guayaquil.
- Se recomienda así mismo, explotar la potencialidad del proyecto, con datos como los usuarios con patrones atípicos, mediante acciones por parte de la empresa que les

permitan revelar la razón de este comportamiento aberrante por parte de estos usuarios y tomar medidas correctivas al respecto, medidas que puede ahorrar miles de dólares mensuales a la empresa de ser bien aplicadas.

Referencias

[1] J. E. Parra y F. L. &. A. H. N. Quilumba, «Customers' demand clustering analysis — A case study using smart meter data,» IEEE, Morelia, 2016.

[2] A. Rajabi, L. Li, J. Zhang, J. Zhu y S. &. G. M. J. Ghavidel, «A review on clustering of residential electricity customers and its applications,» de *2017 20th International Conference on Electrical Machines and Systems* , Sidney, 2017.

[3] A. Rajabi, M. Eskandari, M. J. Ghadi, L. Li y J. &. S. P. Zhang, «A comparative study of clustering techniques for electrical load pattern segmentation,» *Renewable and Sustainable Energy Reviews*, vol. 120, nº 109628, 2020.

[4] M. Hajiaghapour-Moghimi, K. Azimi-Hosseini y E. &. V. M. Hajipour, «Residential Load Clustering Contribution to Accurate Distribution Transformer Sizing,» de *2019 International Power System Conference (PSC), IEEE,*, Teherán, 2019.

[5] T. K. Wijaya, M. Vasirani y S. &. A. K. Humeau, «Cluster-based aggregate forecasting for residential electricity demand using smart meter data,» de *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, 2015.

[6] J. C. C. Parra, Zonificación del mercado de consumo eléctrico de la empresa eléctrica Azogues, Cuenca: Universidad de Cuenca, 2018.

[7] CNEL EP, «ACTUALIZACIÓN DEL ANÁLISIS DE ESTRATIFICACIÓN DE CLIENTES DE LA UN SANTO DOMINGO.,» CNEL EP, Santo Domingo, 2020.

[8] CNEL EP, «Informe de Rendición de cuentas 2018,» 2018. [En línea]. Available: https://www.cnelep.gob.ec/wp-content/uploads/2019/02/resumen-ejecutivo-RC_2018_vf.pdf. [Último acceso: Noviembre 2020].