INSTITUTO POLITECNICO NACIONAL
DESARROLLO DE APLICACIONES PARA ANALISIS DE DATOS
4AM1

Limpieza de datos y clustering de disturbios eléctricos

LOPEZ SANCHEZ
IAN JOSUE

11/01/2023

CONTROL DE DOCUMENTO

DOCUMENT NAME		
Limpieza de datos y clustering de disturbios electricos		
DOCUMENT OWNER	**ISSUE DATE**	**LAST SAVED DATE**
Lopez Sanchez Ian josue	10-01-2023	15-01-2023

TABLA DE CONTENIDOS

RESUMEN EJECUTIVO

En este proyecto, se realizó un análisis de datos de disturbios eléctricos con el objetivo de clasificar los disturbios en base a su categoría, considerando el número de personas afectadas y la pérdida de energía en megavatios. Se utilizó un algoritmo de aprendizaje no supervisado para lograr este objetivo.

- Los resultados clave del análisis incluyen:
- Identificación de la relación entre la entidad de confiabilidad eléctrica (NERC) y el área geográfica.
- Determinación de las empresas con la mayor cantidad de disturbios.
- Identificación del tipo de disturbio más común.
- Análisis de la relación entre la pérdida de energía y el número de personas afectadas.
- Distribución de los disturbios entre las diferentes categorías.
- Comportamiento de los datos atípicos en el conjunto de datos.

Este análisis proporcionó una visión valiosa de los disturbios eléctricos y permitió una mejor comprensión de su impacto y distribución. El uso de técnicas de aprendizaje automático permitió una clasificación efectiva de los disturbios, lo que podría ser útil para futuras investigaciones y acciones preventivas.

INTRODUCCION

Durante el desarrollo del proyecto, se realizó un análisis exhaustivo de un conjunto de datos de disturbios eléctricos utilizando varias bibliotecas de Python, incluyendo pandas, numpy, seaborn, matplotlib, sklearn y gower.

Se propuso como meta inicial hacer una clasificación con un algoritmo de aprendizaje no supervisado y se planteo clasificar los disturbios en base a su categoría, tomando en cuenta cuantas personas fueron afectadas y cuanta perdida de energía hubo en mega watts.

El conjunto de datos se obtuvo de Kaggle llamado como "US Electric Disturbance Events". El análisis comenzó con la importación de las bibliotecas necesarias y la generación de una lista de años desde 2002 hasta 2023. Luego, se leyeron los datos de un archivo Excel para cada año y se almacenaron en un diccionario. Los datos procesados se escribieron en archivos CSV para cada año.

Se realizaron varias operaciones de limpieza de datos, incluyendo la eliminación de filas y columnas no deseadas, la conversión de tipos de datos, el reemplazo de valores y la imputación de valores faltantes. Se realizó la codificación one-hot en las columnas categóricas y se concatenaron los resultados con el DataFrame original.

Se pudo dar un recorrido histórico a través de los eventos de mayor significancia en los Estados Unidos.

Se calculó la matriz de distancia Gower para el DataFrame filtrado. Luego, se aplicó el algoritmo DBSCAN al DataFrame filtrado y se almacenaron las etiquetas de los clusters en una nueva columna. Se crearon gráficos de dispersión de los datos, coloreados por las etiquetas de los clusters.

Finalmente, se entrenó un modelo de Random Forest en los datos de entrenamiento y se evaluó el modelo en los datos de prueba. Se calculó el coeficiente de silueta para los clusters y se creó un gráfico de dispersión de las etiquetas de los clusters vs los valores de silueta.

En resumen, este chat proporcionó una visión detallada de cómo realizar un análisis completo de los datos de disturbios eléctricos, incluyendo la limpieza de datos, la visualización, el clustering y la clasificación. Todo esto se hizo utilizando varias bibliotecas de Python, demostrando la potencia y la flexibilidad de Python para el análisis de datos y el aprendizaje automático.

PROBLEMATICA

En este proyecto, se busca abordar varias problemáticas relacionadas con los disturbios eléctricos. Aquí están las problemáticas y una ampliación de cada una:

- Clasificación de los disturbios eléctricos: El objetivo principal era clasificar los disturbios eléctricos en base a su categoría. Esto implica identificar patrones en los datos que puedan ayudar a agrupar los disturbios en categorías significativas. La clasificación puede ayudar a entender mejor los tipos de disturbios que ocurren con mayor frecuencia y a identificar posibles causas y soluciones.

- Impacto de los disturbios eléctricos: Se consideró cuántas personas fueron afectadas y cuánta pérdida de energía hubo en megavatios por cada disturbio. Esto proporciona una medida del impacto de los disturbios, lo que puede ser útil para priorizar las respuestas y las estrategias de mitigación.

- Relación entre la entidad de confiabilidad eléctrica (NERC) y el área geográfica: Se identificó una relación entre la NERC y el área geográfica. Esto puede indicar que ciertas áreas son más propensas a ciertos tipos de disturbios, o que algunas NERCs son más efectivas en la gestión de disturbios que otras.

- Empresas con la mayor cantidad de disturbios: Se identificaron las empresas con la mayor cantidad de disturbios. Esto puede indicar problemas específicos en estas empresas que necesitan ser abordados.

- Tipo de disturbio más común: Se identificó el tipo de disturbio más común. Esto puede ayudar a las empresas y a las NERCs a prepararse mejor para este tipo de disturbio.

- Distribución de los disturbios entre las diferentes categorías: Se analizó la distribución de los disturbios entre las diferentes categorías. Esto proporciona una visión general de la frecuencia de los diferentes tipos de disturbios.

En resumen, este proyecto busca entender mejor los disturbios eléctricos y su impacto a través de un análisis detallado de los datos disponibles.

METODOLOGIA

En este proyecto, se utilizó una metodología basada en el epiciclo del análisis de datos, que es un enfoque iterativo para el análisis de datos que incluye las etapas de problematización, abstracción, solución y reflexión.

Problematización: El primer paso en el epiciclo del análisis de datos fue definir claramente el problema. En este caso, el objetivo era clasificar los disturbios eléctricos

en base a su categoría, considerando cuántas personas fueron afectadas y cuánta pérdida de energía hubo en megavatios.

Una vez hecha la problematización se inicio con el desarrollo del proyecto, iniciamos dividiendo el desarrollo en 4 fases, construcción del dataset, limpieza, preprocesamiento y modelado.

Construcción del dataset: la "base de datos" o mas bien la fuente de datos estaba dada en un formato "xlsx" con una hoja de Excel por cada año desde enero de 2002 a julio de 2023, por lo que se creo un método de que se encargara de leer los datos de un archivo Excel llamado 'DOE_Electric_Disturbance_Events.xlsx' para cada año en una estructura de datos de tipo lista con contenido de 2002 a 2023. Cada hoja en el archivo Excel correspondía a un año específico. Los datos de cada año se almacenaron en un DataFrame de Pandas y luego se añadieron a un diccionario llamado year_data, donde la clave es el año y el valor es el DataFrame correspondiente. Nos quedamos con year_data el cual es un diccionario que contiene los datos de todos los años desde 2002 hasta 2023. Cada entrada en el diccionario es un DataFrame de Pandas que contiene los datos para un año específico.

Posterior se observó que los formatos de cada archivo csv, variaban desde su estructura hasta el formato de sus datos entre cada año por lo que se prosiguió a la creación de cuatro funciones de Python que fueron diseñadas para procesar y limpiar conjuntos de datos específicos. Cada función tomaba un DataFrame de pandas (df) y un nombre de archivo (name) como entrada.
A continuacion un breve resumen de las funciones.

fstWR(df, name): Esta función renombra las columnas del DataFrame a una lista predefinida de nombres de columnas. Luego, elimina las filas donde el tipo de la columna 'Date' es un número flotante o una cadena. Finalmente, guarda el DataFrame procesado en un archivo CSV con el nombre proporcionado.

sndWR(df, name): Similar a fstWR, esta función también renombra las columnas y elimina las filas basadas en el tipo de la columna 'Date'. Sin embargo, los nombres de las columnas y el orden son diferentes en comparación con fstWR.

trdWR(df, name): Esta función renombra las columnas, elimina las columnas 'Month' y 'Alert', y descarta la primera fila. Luego, elimina las filas donde el tipo de la columna 'Date' es un número flotante. Finalmente, guarda el DataFrame procesado en un archivo CSV.

fthWR(df, name): Similar a trdWR, pero elimina las columnas ‘Year’, ‘Month’ y ‘Alert’ en lugar de solo ‘Month’ y ‘Alert’.

Estas funciones son hechas a medida por lo que su replicación en otro proyecto de análisis o de índole similar no es recomendable.

Posterior una vez que se hizo ese wrangling inicial, procedimos a ver como quedaron los datasets individuales.

Ahí fue donde se dio la realización de que había ciertos formatos como los de “Restoration Date” los cuales variaban mucho entre conjuntos por lo que se pasó a la siguiente fase del desarrollo, la de limpieza.

Abstracción: En esta etapa, se identificaron las variables clave para el análisis, incluyendo el número de personas afectadas y la pérdida de energía. Se decidió utilizar un algoritmo de aprendizaje no supervisado para la clasificación, ya que no se contaba con etiquetas predefinidas para los datos. Y se procedió a limpiar el dataset a fin de poder aplicar el modelo que queremos emplear, el cual de inicio se planteo como K-Means o Elastic Net Regression.

Limpieza de datos: Se realizaron varias operaciones de limpieza de datos, incluyendo la eliminación de filas y columnas no deseadas, la conversión de tipos de datos, el reemplazo de valores y la imputación de valores faltantes. Estos pasos fueron necesarios para asegurar que los datos estuvieran en un formato adecuado para el análisis y para mejorar la calidad de los resultados.

Para iniciar la limpieza de cada dataset individual primero volvimos a hacer uso de la lista “years” que teníamos previamente hecha en pasos anteriores y se procedió a la creación de funciones hechas a la medida para limpiar cada dataset

A continuación, se da un resumen de lo hecho

fstCleaning(df, name):
Recibe un DataFrame (df) y un nombre (name).
Itera sobre las filas del DataFrame.
Convierte la columna 'Restoration Date' al formato de fecha si comienza con un dígito, de lo contrario, usa la columna 'Date'.
Guarda el DataFrame resultante en un archivo CSV con el nombre proporcionado (name).
sndCleaning(df, year):

Recibe un DataFrame (df) y un año (year).

Itera sobre las filas del DataFrame.
Convierte la columna 'Restoration Date' al formato de fecha si comienza con un dígito, de lo contrario, usa la columna 'Date'.
Guarda el DataFrame resultante en un archivo CSV con el nombre del año proporcionado (year).
is_format1(date_str):

Verifica si una cadena de fecha está en el formato 'mm/dd/yyyy'.
convert_format1(date_str):

Convierte una cadena de fecha en el formato 'mm/dd/yyyy' al formato 'yyyy-mm-dd 00:00:00'.
detect_and_convert(date_str):

Detecta el formato de la cadena de fecha y la convierte según sea necesario. Devuelve la fecha convertida y un indicador booleano que indica si se detectó el formato.
trdCleaning(df, year):

Itera sobre las filas del DataFrame.
Utiliza detect_and_convert para convertir la columna 'Restoration Date' al formato deseado.
Guarda el DataFrame resultante en un archivo CSV con el nombre del año proporcionado (year).
trdexCleaning(df, year):

Itera sobre las filas del DataFrame.
Utiliza detect_and_convert para convertir la columna 'Date' al formato deseado.
Guarda el DataFrame resultante en un archivo CSV con el nombre del año proporcionado (year).

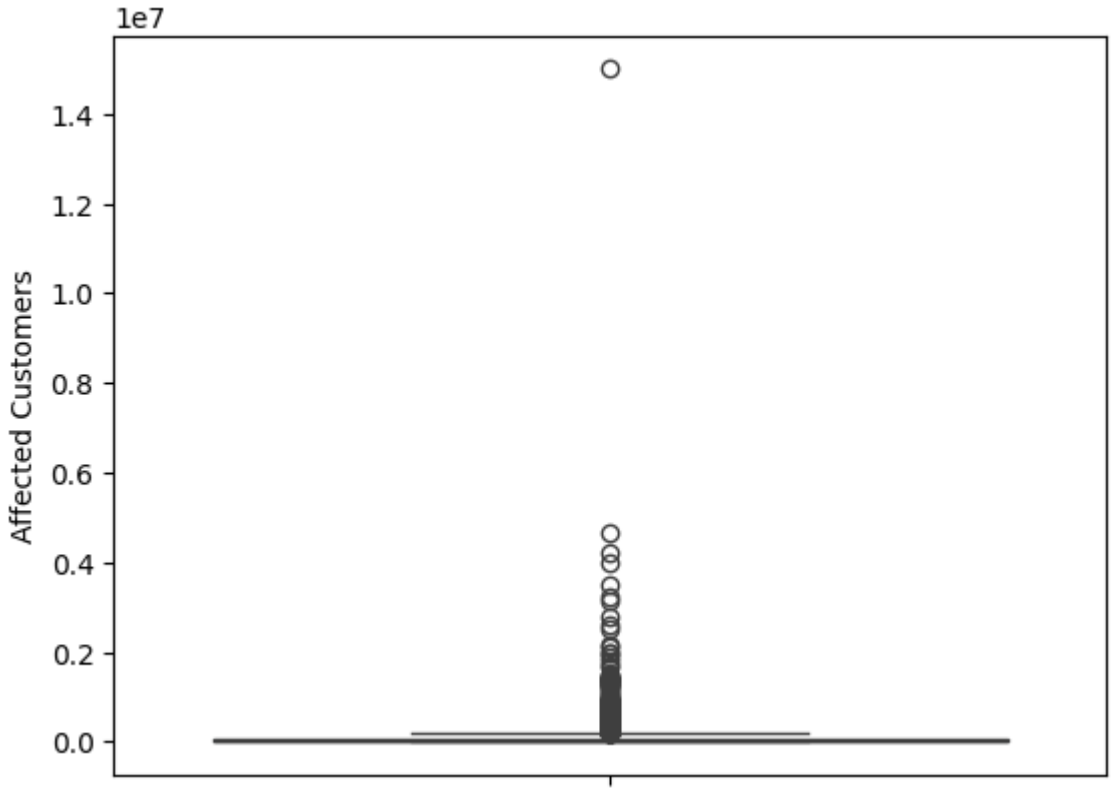
Luego en el ciclo principal de esta parte de la limpieza:
Se leen los archivos CSV para cada año y almacena los DataFrames resultantes en un diccionario (year_data).
Itera sobre los años y aplica las funciones de limpieza y transformación basadas en condiciones específicas.
Elimina columnas no deseadas ('Unnamed' y 'index') de los DataFrames.
Guarda los DataFrames resultantes en archivos CSV.

Una vez hecho esto procedemos con la otra vuelta para limpiar los datos. A continuación se describen los métodos y formas que se uso para limpiar esta parte.

Combinación de datos: Combina todos los DataFrames en year_data en un solo DataFrame combined_df y lo guarda en un archivo CSV ‘distrubances.csv’. Luego, carga este archivo CSV en un nuevo DataFrame df.

Limpieza inicial: Elimina varias columnas no deseadas de df, incluyendo ‘Restoration Time’, ‘Event’, y varias columnas ‘Unnamed’.

Limpieza de ‘Affected Customers’: Realiza una serie de operaciones de limpieza en la columna ‘Affected Customers’. Esto incluye reemplazar ciertos valores de cadena por 0, extraer números de cadenas que contienen ciertos caracteres y multiplicarlos por un factor apropiado, y finalmente convertir todos los valores a tipo float.



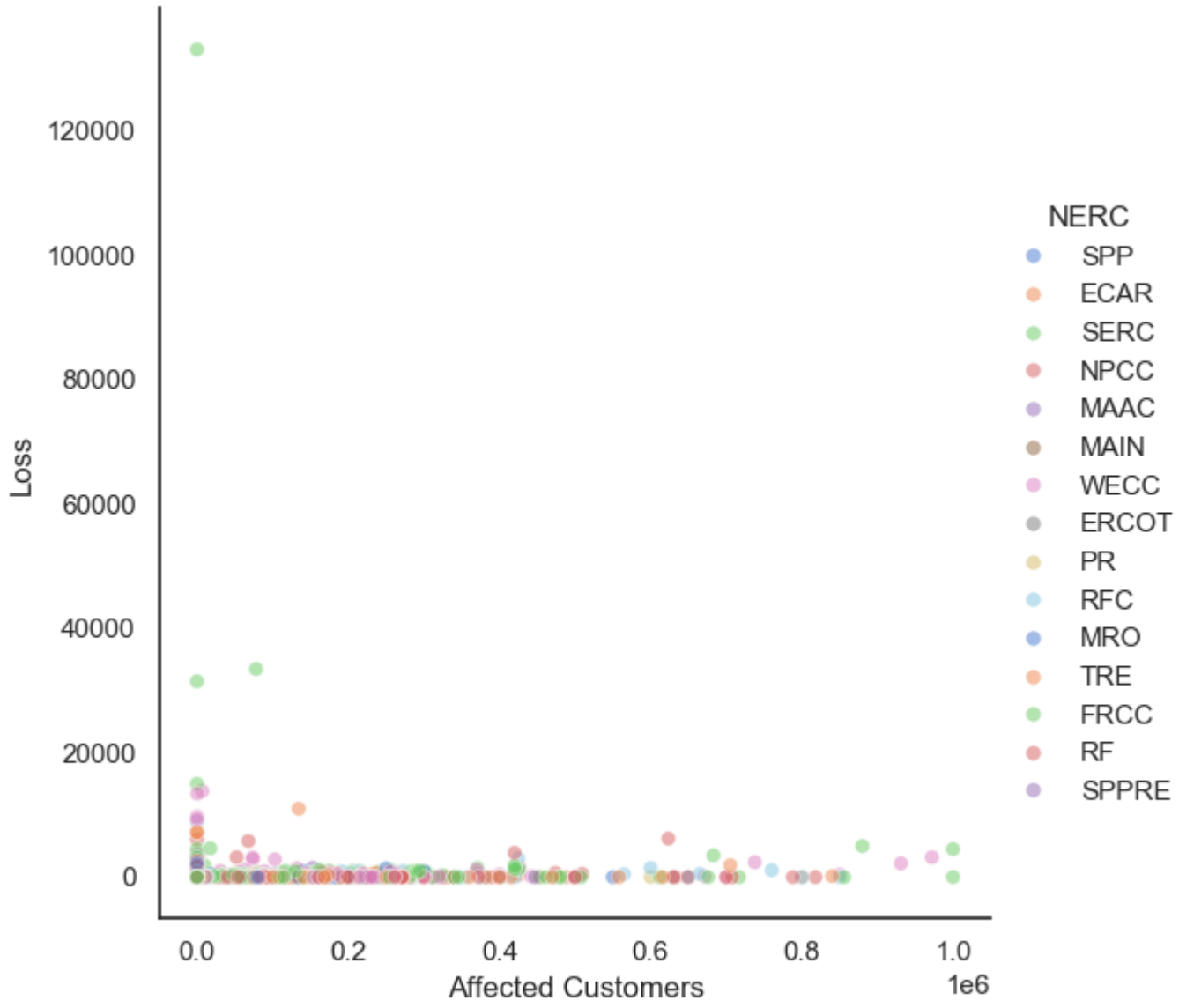
Limpieza de ‘Loss’: Realiza una serie de operaciones de limpieza en la columna ‘Loss’. Esto incluye reemplazar ciertos valores de cadena por 0, extraer números de cadenas que contienen ciertos caracteres y multiplicarlos por un factor apropiado, y finalmente convertir todos los valores a tipo numérico.

Limpieza de ‘NERC’: Reemplaza los valores no string en la columna ‘NERC’ por el valor más común (moda) en esa columna. Luego, elimina los espacios en blanco de los valores de ‘NERC’. Finalmente, filtra el DataFrame para mantener sólo las filas donde ‘NERC’ está en la lista de valores que ocurren más de 14 veces.

Limpieza de ‘Area’: Reemplaza los valores en la columna ‘Area’ por el nombre del estado si el nombre del estado está presente en el valor original. Luego, filtra el DataFrame para mantener sólo las filas donde ‘Area’ está en la lista de valores que ocurren al menos 10 veces.

Limpieza de ‘Disturbance’: Filtra el DataFrame para mantener sólo las filas donde ‘Disturbance’ está en la lista de valores que ocurren al menos 10 veces. Luego, reemplaza los valores en la columna ‘Disturbance’ por uno de los valores predefinidos en la lista distrubances si ese valor está presente en el valor original.

Visualización de datos: Crea un gráfico de dispersión de ‘Affected Customers’ vs ‘Loss’, coloreado por ‘NERC’. Los datos se filtran para incluir sólo las filas donde ‘Affected Customers’ y ‘Loss’ son menores o iguales a 1,000,000.



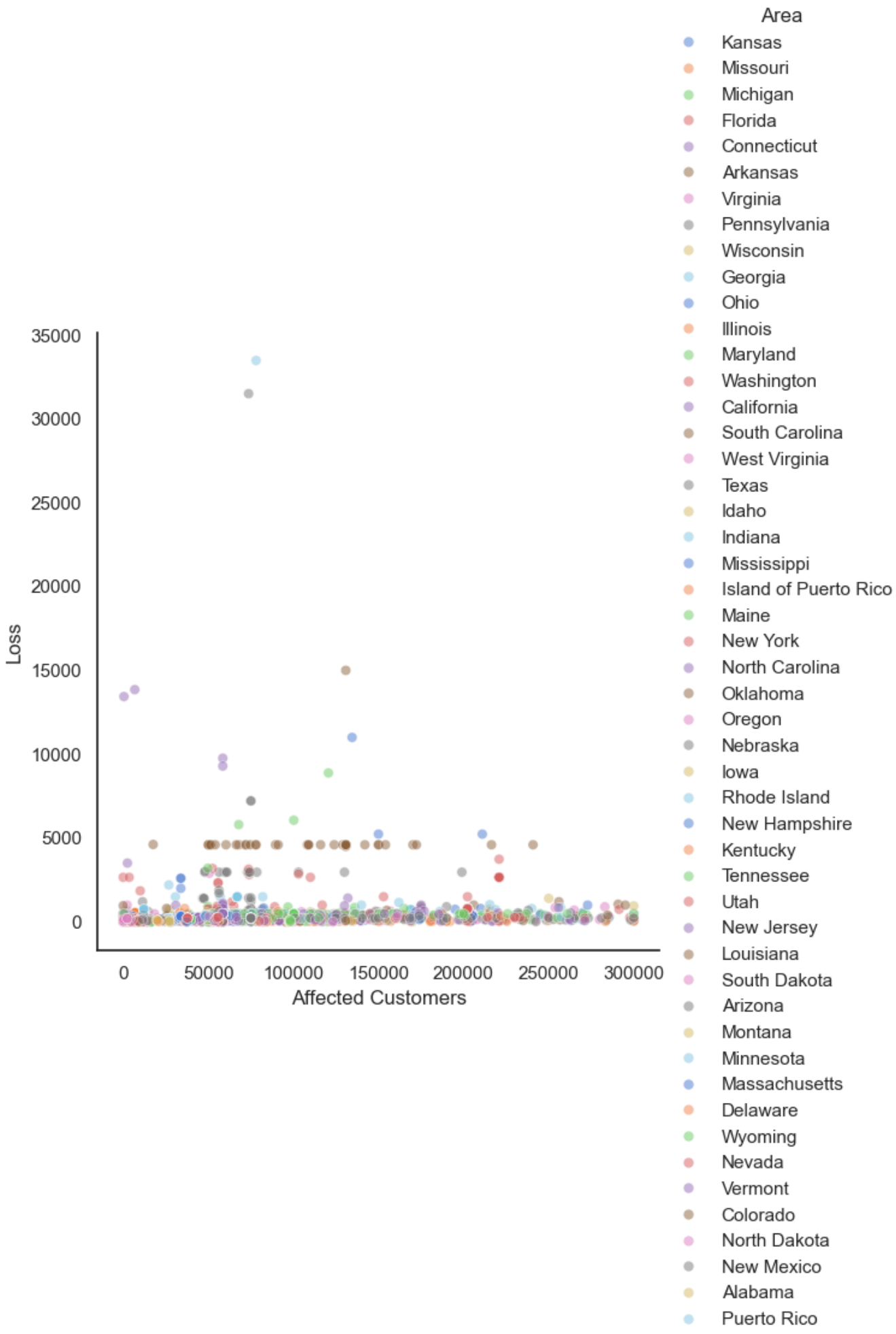
Limpieza de ‘Affected Customers’: Si ‘Affected Customers’ es menor que 1, reemplaza el valor por la media de ‘Affected Customers’ para la misma ‘NERC’ y ‘Area’. Si ‘Affected Customers’ sigue siendo menor que 1, reemplaza el valor por la media de ‘Affected Customers’ para la misma ‘NERC’.

Cálculo de la puntuación: Calcula una puntuación como la relación entre ‘Loss’ y ‘Affected Customers’ para cada fila donde ‘Loss’ es mayor que 0. Imprime la puntuación para cada fila, así como la media de todas las puntuaciones.

Limpieza de ‘Loss’: Si ‘Loss’ es menor que 1, reemplaza el valor por la media de ‘Loss’ para la misma ‘NERC’ y ‘Area’, multiplicada por un factor de 1.8690183434213. Si la media de grupo es menor que 1, utiliza la media de ‘Loss’ para la misma ‘NERC’.

Al final de este proceso, df queda como un DataFrame limpio y preprocesado que contiene los datos de disturbios eléctricos para todos los años desde 2002 hasta 2023.

Modelado: Como se había planteado desde un inicio el objetivo del proyecto es realizar una clasificación por categorías para poder tener una idea de como se comportan estos tipos de disturbios y categorizar su impacto en la vida diaria de las personas.

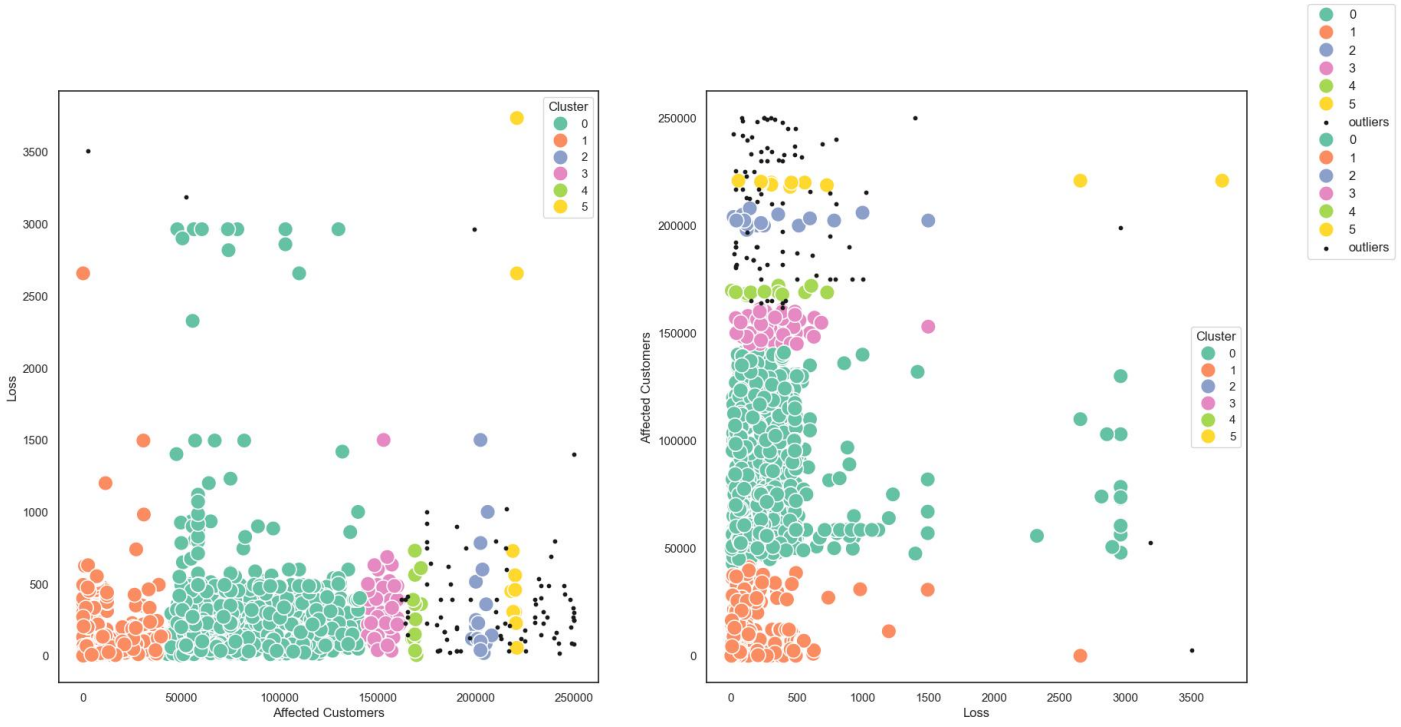

Area
Kansas
Missouri
Michigan
Florida
Connecticut
Arkansas
Virginia
Pennsylvania
Wisconsin
Georgia
Ohio
Illinois
Maryland
Washington
California
South Carolina
West Virginia
Texas
Idaho
Indiana
Mississippi
Island of Puerto Rico
Maine
New York
North Carolina
Oklahoma
Oregon
Nebraska
Iowa
Rhode Island
New Hampshire
Kentucky
Tennessee
Utah
New Jersey
Louisiana
South Dakota
Arizona
Montana
Minnesota
Massachusetts
Delaware
Wyoming
Nevada
Vermont
Colorado
North Dakota
New Mexico
Alabama
Puerto Rico
Loss
35000
30000
25000
20000
15000
10000
5000
0
0
50000
100000
150000
200000
250000
300000
Affected Customers

Por lo que tras preparar los datos para poder aplicar un modelo, se noto que primero que nada se tiene que filtrar y modificar algunos datos para no ser susceptibles a datos anómalos.

Después se realizo una codificación de tipo one-hot con las variables categóricas 'Disturbance', 'NERC' y 'Area' fueron transformadas utilizando la codificación one-hot. Este es un proceso que convierte variables categóricas en un formato que puede ser proporcionado a los algoritmos de aprendizaje automático para mejorar su rendimiento.

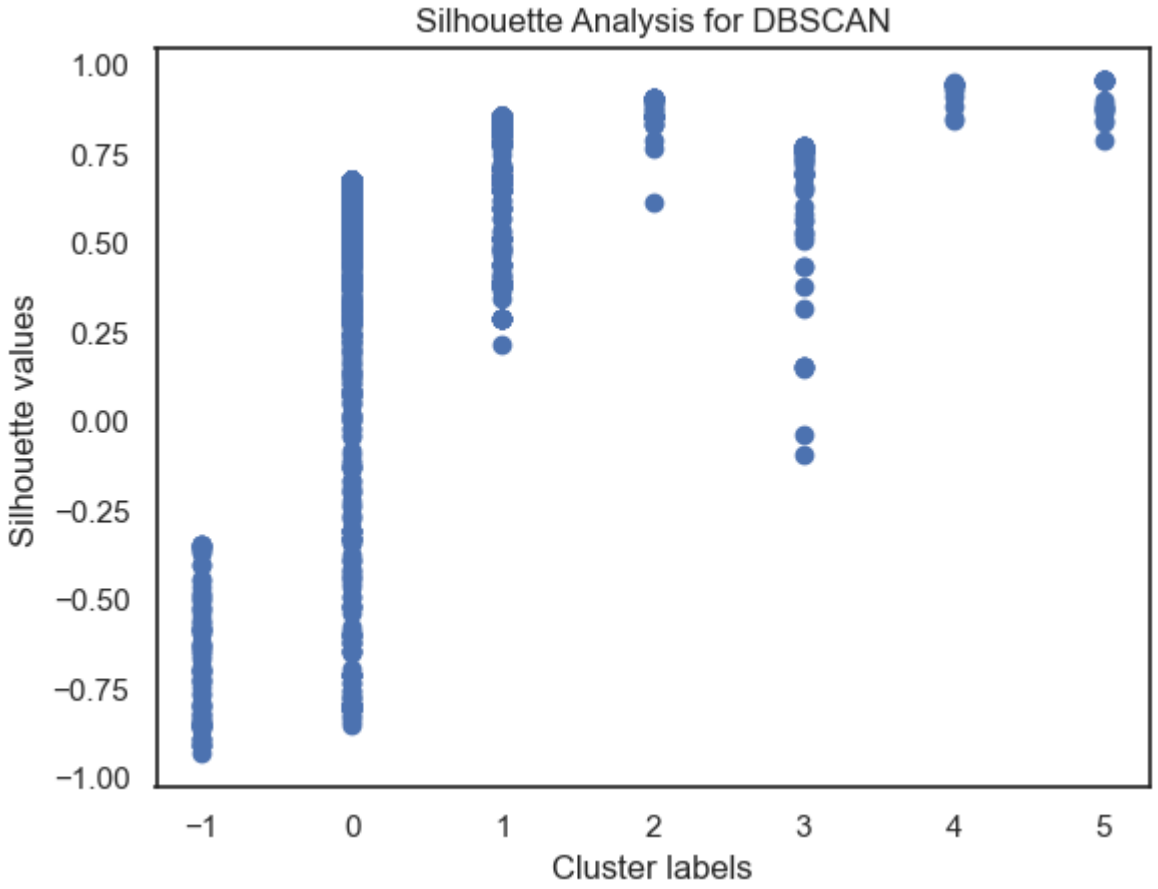
Cálculo de la matriz de distancia: Se calculó la matriz de distancia Gower para el DataFrame filtrado. Esta matriz de distancia es una representación de la similitud o disimilitud entre los diferentes puntos de datos en el conjunto de datos.

Clustering DBSCAN: Se aplicó el algoritmo DBSCAN al DataFrame filtrado. DBSCAN es un algoritmo de clustering que divide los puntos de datos en varios clusters basándose en la densidad de los puntos.



Clasificación con Random Forest: Se entrenó un modelo de Random Forest en los datos de entrenamiento y se evaluó el modelo en los datos de prueba. Random Forest es un algoritmo de aprendizaje supervisado que puede ser utilizado para tareas de clasificación.

Análisis de silueta: Calcula el coeficiente de silueta para los clusters y lo imprime. También crea un gráfico de dispersión de las etiquetas de los clusters vs los valores de silueta.



Reflexión: En la última etapa del epiciclo del análisis de datos, se reflexionó sobre los resultados del análisis. Se identificaron hallazgos clave, como la relación entre la entidad de confiabilidad eléctrica (NERC) y el área geográfica, las empresas con la mayor cantidad de disturbios, el tipo de disturbio más común, la relación entre la pérdida de energía y el número de personas afectadas, la distribución de los disturbios entre las diferentes categorías y el comportamiento de los datos atípicos. Estos hallazgos proporcionaron una visión valiosa de los disturbios eléctricos y permitieron una mejor comprensión de su impacto y distribución.

En resumen, este proyecto demostró cómo se pueden utilizar varias técnicas de análisis de datos y aprendizaje automático para obtener información valiosa a partir de un conjunto de datos de disturbios eléctricos. A través de un proceso iterativo de problematización, abstracción, solución y reflexión, se logró el objetivo de clasificar los disturbios eléctricos y entender mejor su impacto

RESULTADOS

A continuación se presenta un breve resumen de los resultados encontrados, la primera imagen corresponde al conteo de valores de los diferentes grupos nuevos que se crearon con DBScan. Donde -1 son valores atípicos y todos los demás grupos son como se distribuyen los datos entre categorías

Cluster	count
0	1157
1	537
-1	86
3	59
2	42
5	35
4	16

La siguiente metrica presentada es una de K-fold cross validation, se tomo en cuenta 10 folds para realizar la prueba

Scores [0.8709677419354839]

Scores [0.8709677419354839, 0.8580645161290322]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258, 0.8774193548387097]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258, 0.8774193548387097, 0.8766233766233766]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258, 0.8774193548387097, 0.8766233766233766, 0.8441558441558441]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258, 0.8774193548387097, 0.8766233766233766, 0.8441558441558441, 0.8376623376623377]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258, 0.8774193548387097, 0.8766233766233766, 0.8441558441558441, 0.8376623376623377, 0.8636363636363636]

Scores [0.8709677419354839, 0.8580645161290322, 0.8516129032258064, 0.864516129032258, 0.8774193548387097, 0.8766233766233766, 0.8441558441558441, 0.8376623376623377, 0.8636363636363636, 0.8246753246753247]

Cross-Validation accuracy: 0.857 +/- 0.016

Al final nos deja con un accuracy score de 85% en la prueba de K-folds con un margen de error de 1.6%

Se uso también silhouette score para determinar la precisión del agrupamiento

Silhouette score: 0.48357090617049153

Donde el score va de -1 a 1, siendo la mas cercana a 1 un mejor agrupamiento de datos.

También se presentan los resultados de recall, f1-score, precisión y accuracy siendo este de 94%

Accuracy: 0.9483204134366925

	precision	recall	f1-score	support
-1	0.62	0.67	0.65	15
0	0.97	1.00	0.99	236
1	1.00	1.00	1.00	103
2	0.70	0.54	0.61	13
3	0.75	0.82	0.78	11
4	0.00	0.00	0.00	4
5	1.00	0.40	0.57	5
accuracy			0.95	387
macro avg	0.72	0.63	0.66	387
weighted avg	0.94	0.95	0.94	387

CONCLUSIONES

En conclusión, este proyecto de análisis de disturbios eléctricos ha proporcionado una visión profunda y valiosa de la frecuencia, distribución y clasificación de eventos perturbadores en el suministro eléctrico. A través de un enfoque iterativo del epiciclo del análisis de datos, se logró abordar diversas problemáticas, desde la clasificación de disturbios hasta la identificación de relaciones clave entre variables.

El uso de técnicas avanzadas como DBScan para la creación de grupos, K-fold cross-validation para evaluar la precisión del modelo, y silhouette score para medir la calidad del agrupamiento, ha demostrado la robustez del análisis. El modelo resultante exhibe un sólido rendimiento con un accuracy score del 85%, respaldado por la evaluación detallada de métricas de clasificación como recall, f1-score, precisión y accuracy.

La visualización de los datos, la limpieza y procesamiento cuidadoso de los conjuntos de datos, y la aplicación de algoritmos de aprendizaje automático han contribuido a la

identificación de patrones significativos y la comprensión profunda de la dinámica de los disturbios eléctricos.

Estos hallazgos no solo tienen aplicaciones inmediatas en la gestión y mitigación de disturbios eléctricos, sino que también sientan las bases para investigaciones futuras y la implementación de estrategias preventivas. En última instancia, este proyecto destaca la utilidad y la potencia de las técnicas de análisis de datos avanzadas para abordar problemas complejos. Y sirvió como una buena practica para poder explorar mas sobre como se ven los datos en un ámbito mas profesional.

REFERENCIAS

Microsoft, (2020). ¿Qué es el Proceso de ciencia de datos en equipo

(TDSP)?. Recuperado de:https://www.ibm.com/downloads/cas/6RZMKDN8

IBM, (2020). Metodología fundamental para la ciencia de datos.

Recuperado de: https://www.ibm.com/downloads/cas/6RZMKDN8

IBM, (2022). Descripción de los datos - Documentación de IBM.

Recuperado de:
https://www.ibm.com/docs/es/spssmodeler/18.4.0?topic=understanding-describing-data

• AMPLN. (2019). CICLing: International Conference on

Computational Linguistics and Intelligent Text Processing. Obtenido

de AMPLN Asociación Mexicana para el Procesamiento del

Lenguaje: https://www.cicling.org/ampln/

• Justicia de la T., M. d. (2017). Nuevas Técnicas de Minería de Textos:

Aplicaciones. Universidad de Granada. Tesis Doctorales. Obtenido

de http://hdl.handle.net/10481/46975

• Gomez-Adorno, H., Bel-Enguix, G., Sierra, G., Sánchez, O., &

Quezada, D. (2018). A machine learning approach for detecting

aggressive tweets in spanish. In Proceedings of the Third Workshop

on Evaluation of Human Language Technologies for Iberian

Languages (IberEval 2018), CEUR WS Proceeding.

• Sidorov, G., Markov, I., Kolesnikova, O., & Chanona-Hernández, L. (2019). Human interaction with shopping assistant robot in natural language. Journal of Intelligent & Fuzzy Systems, 36(5), 4889-4899.