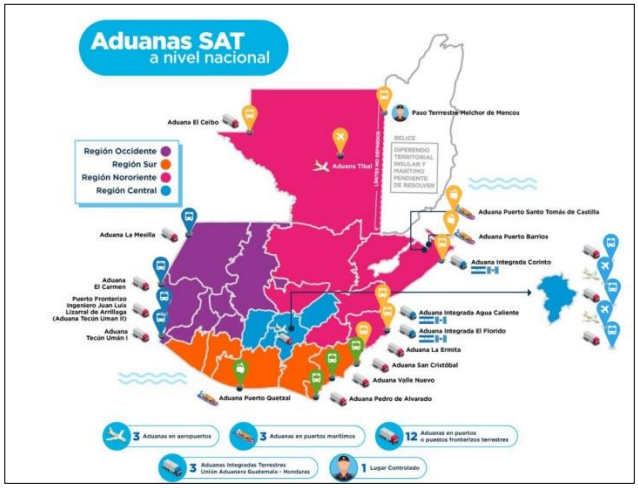
INTRODUCCIÓN

En toda entidad, industria o negocio de carácter privado o público su funcionamiento genera grandes volúmenes de datos, sin embargo, también existen problemas a resolver o procesos a mejorar, por lo que el análisis de datos es de suma importancia para la toma de decisiones. Es necesario desarrollar un proceso de minería de datos, que permita desde el punto de vista de redes neuronales recurrentes (RNNs), y utilizando la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), conocer los principales hallazgos de la información que generan sus operaciones.



Para el presente análisis se determinó la necesidad de poder resolver una de las problemáticas que tiene el sector de comercio exterior (importaciones), esto de acuerdo a las operaciones de las distintas Aduanas, (terrestres, aéreas y marítimas), ya que una de las problemáticas detectadas o áreas a mejorar es el atraso en el tiempo de despacho de mercancías.

Fuente: Superintendencia de Administración Tributaria.

El objetivo es determinar que Aduanas tienen mayor número de importaciones que generan atraso en el tiempo de salida de la importación, y así determinar la necesidad de implementar nuevas rampas de revisión para mejorar el tiempo de salida de las mercancías. El control de los datos masivos con los que se cuentan relacionados al ingreso de mercancías, se cuenta con la necesidad de determinar la medición de tiempos de despacho e identificar si la medición se puede realizar por cantidad de importaciones, por peso de las declaraciones o por valor Costo, Seguro y Flete (CIF).

PROCESAMIENTO DE DATOS

El dataset, que utilizado como base fundamental en el desarrollo del proyecto contiene una muestra de datos de transacciones de comercio exterior (por ser información confidencial fue adaptada con fines académicos), el cual cuenta con 20 variables, entre ellos:

- Correlativo: Número de fila
- Id declaración: Código de la Declaración Única Aduanera
- NIT: Número de Identificación Tributaria
- Pasaporte: Número de identificación de persona extranjera
- Consignatario: Nombre del importador
- Fecha: Fecha de aceptación de la declaración
- Días tarda Salir: Días que tarda en salir de la Aduana
- Año: Año en que realizó la importación
- Cantidad de importaciones: Comprende el total de importaciones realizadas por el consignatario.
- Aduana: Identificación de aduanas

PA	Pedro de Alvarado
EA	Express Aéreo
PQ	Puerto Quetzal
PB	Puerto Barrios
TU	Tecún Umán
ST	Santo Tomás de Castilla
VN	Valle Nuevo
SC	San Cristóbal
CG	Central de Guatemala
H8	Integrada el Florido
H6	Integrado Corinto
H7	Integrada Agua Caliente
LE	La Ermita
MM	Melchor de Mencos

- Régimen:

ID	Importación definitiva
IC	Importación Courier
DUCA-T	Transito
ZI	Importación en Zona Franca

- Peso declaración:

BAR	Barril
CBZ	Cabeza
JGO	Juego
KGS	Kilogramos
LTS	Litros
MT2	Metros cuadrados
PZA	Pieza
MTS	Metro
PAR	Par
MT3	Metros cúbicos

- Selectivo: R=Rojo, V=Verde
- Fracción arancelaria: Código numérico que se asigna a los bienes que se comercian.
- Descripción fracción: Comprende el detalle de la mercancía importada.
- País fracción: Se refiere al país de origen de las mercancías.
- Cantidad fracción: Cantidad importada según la unidad de medida.
- Unidad de medida: libras, Kilos, unidades, etc.
- Valor CIF: Valor del Impuesto determinado
- Región: Central, Sur, Occidente, Nororiente

Esta selección constituye la parte más importante del proyecto, ya que el conjunto de datos seleccionado, determinará los principales hallazgos.

Se desarrolló como paso inicial una exploración de datos, para comprender su contenido, variables, estructura, y tipos de datos (categórico, continuo o discreto), entre otros.

La calidad de los datos para el análisis de la información es de vital importancia, por lo que se inició con el proceso de limpieza de datos, mediante una serie de códigos en Phyton, que incluyo entre los más importante lo siguiente:

- Identificación de columnas con datos faltantes,
- Imputación de valores faltantes,
- Identificación de valores atípicos.

En resumen se describen los principales procesos realizados:

- Librerías utilizadas:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler
from scipy import stats
```

- Exploración de base de datos

```
df = pd.read_excel("Data_impo.xlsx")
df.head()
```

- Identificación de columnas con valores atípicos, nulos o vacíos

```
cols_con_na = [col for col in df.columns if (df[col].isnull().mean() > 0)]
cols_con_na
```

En este paso se identificaron las columnas, “Pasaporte”, Cantidad de Importaciones”

- Se verifican las columnas que son utilizables para el análisis

```
cols_rescatables = [col for col in df.columns if (df[col].isnull().mean() <= 0.05)]
cols_rescatables
```

- Se depura o eliminan las columnas que no generan valor al análisis

```
df_noNAN = df[cols_rescatables].dropna()
df.shape, df_noNAN.shape
```

- Se analizaron las columnas por tiempo de datos, discretos, continuos y categóricos

```
def getDateColTypes(df):
      categoricas = []
      continuas = []
      discretas = []

      for colName in df.columns:
            if (df_noNAN[colName].dtype == 'O'):
                  categoricas.append(colName)
            else:
                  if((df_noNAN[colName].dtype == 'int64') or (df_noNAN[colName].dtypes == 'float64')):
                        if(len(df_noNAN[colName]) <= 30):
                              discretas.append(colName)
                        else:
                              continuas.append(colName)
      return discretas, continuas, categoricas
```

- Se verificaron las columnas a utilizar, y se asignó nuevo nombre al dataset, posterior a su depuración de datos.

```
df = pd.read_excel("Data_impo.xlsx")
df.head()
```

Python

	No.	ID DECLARACIÓN	NIT	PASAPORTE	CONSIGNATARIO	FECHA	DIAS TARDA SALIR	año	Cantidad de importaciones	Aduana	Regimen	Peso Declaracion	Selectivo	Fraccion Arancelaria	Descripcion Fraccion	Pais Fraccion	Cantidad Fraccion	Unidad Medida	Valor Cif. $.	Región
0	1	295-1908743	48812894	NaN	Delgado Expo Sa	9/7/2021	16.0	2021	10.0	PA	ID	1112.00	R	8301409000	CERRADURA DE VITRINA TRADICIONALES MODELO 43535	CN	5.00	PZA	636.76	Sur
1	2	295-1908743	48812894	NaN	Delgado Expo Sa	9/7/2021	16.0	2021	10.0	PA	ID	1112.00	R	8301300000	CERRADURA PARA MUEBLES, CROMADO MODELO 43561	CN	952.00	KGS	4621.22	Sur
2	3	263-8226467	13169795	NaN	Capitolio Comercializadora	25/5/2021	12.0	2021	3.0	EA	ID	45.45	R	8301300000	CERRADURA PARA MUEBLES, ACABADO LATÓN BRILLANT...	CN	14400.00	PZA	3600.00	Central
3	4	133-2316913	82336545	NaN	Amigo Expo Sa	7/8/2021	9.0	2021	9.0	EA	ID	38.50	R	8301402000	CERRADURA PARA MUEBLES, ACABADO LATÓN BRILLANT...	CN	60.96	MTS	1892.00	Central
4	5	226-8116176	45867462	NaN	Confianza Comercializadora	9/10/2021	5.0	2021	6.0	EA	ID	0.30	R	8205599000	CUCHARA ALBAÑIL 7"", MÉRIDA, PRETUL MODELO 21057	MX	1.00	KGS	1.15	Central

- Mediante la función Summary, se analizó estadísticamente cada una de las variables.

```
dfSummary(df_noNAN)
```

```
c:\Users\50233\AppData\Local\Programs\Python\Python311\Lib\site-packages\summarytools\summarytools.py:124: RuntimeWarning: divide by zero encountered in scalar divide
  stats += f"<br>IQR (CV) : {x.quantile(0.75) - x.quantile(0.25):.1f} ({x.mean()/x.std():.1f})"
```

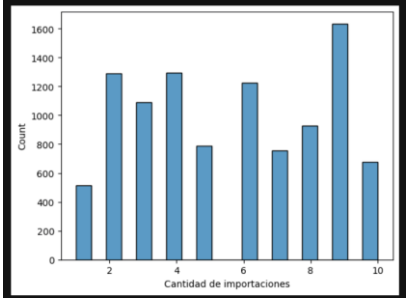
Data Frame Summary
df_noNAN
Dimensions: 10,198 x 19
Duplicates: 0

No	Variable	Stats / Values	Freqs / (% of Valid)	Graph	Missing
1	**No.** [int64]	Mean (sd) : 5099.5 (2944.1) min < med < max: 1.0 < 5099.5 < 10198.0 IQR (CV) : 5098.5 (1.7)	10,198 distinct values		0 (0.0%)
2	**ID DECLARACIÓN** [object]	1. 241-3128879 2. 221-8500628 3. 277-6783964 4. 189-8286504 5. 346-4640161 6. 312-6758013 7. 120-8000598 8. 252-8081246 9. 217-6409134 10. 255-9859719 11. other	96 (0.9%) 96 (0.9%) 61 (0.6%) 61 (0.6%) 60 (0.6%) 57 (0.6%) 57 (0.6%) 53 (0.5%) 48 (0.5%) 44 (0.4%) 9,565 (93.8%)		0 (0.0%)
3	**NIT** [int64]	Mean (sd) : 55823542.9 (26249972.4) min < med < max: 12210013.0 < 58906531.0 < 99471985.0 IQR (CV) : 42875528.0 (2.1)	86 distinct values		0 (0.0%)

- Se realizó de forma gráfica las principales variables detectadas.

```
1 sns.histplot(data= df_noNAN, x='Cantidad de importaciones')
```

```
<Axes: xlabel='Cantidad de importaciones', ylabel='Count'>
```



- Se delimitó el análisis a las nuevas columnas seleccionadas en la depuración.

```
data_clean = df_noNAN[['Peso Declaracion', 'DIAS TARDA SALIR','Cantidad de importaciones', 'Valor Cif. $.']]
data_clean
```

	Peso Declaracion	DIAS TARDA SALIR	Cantidad de importaciones	Valor Cif. $.
0	1112.00	16.0	10.0	636.76
1	1112.00	16.0	10.0	4621.22
2	45.45	12.0	3.0	3600.00
3	38.50	9.0	9.0	1892.00
4	0.30	5.0	6.0	1.15
...	...	...	...	...
10193	620.00	13.0	4.0	20.00
10194	620.00	13.0	4.0	40.00
10195	620.00	13.0	4.0	40.00
10196	620.00	13.0	4.0	40.00
10197	125400.00	23.0	1.0	54762.50

10198 rows × 4 columns

- Se desplegó la nueva estadística de los datos seleccionados

```
data_clean.describe()
```
✓ 0.0s

	Peso Declaracion	DIAS TARDA SALIR	Cantidad de importaciones	Valor Cif. $.
count	1.019800e+04	10198.000000	10198.000000	1.019800e+04
mean	5.464611e+04	52.881742	5.594332	2.745698e+04
std	5.154167e+05	29.164612	2.758509	1.847308e+05
min	1.500000e-01	1.000000	1.000000	5.000000e-02
25%	1.399360e+03	28.000000	3.000000	2.115850e+02
50%	8.870000e+03	54.000000	6.000000	1.752525e+03
75%	2.040880e+04	78.000000	8.000000	9.918778e+03
max	2.600000e+07	100.000000	10.000000	7.800000e+06

- Importante analizar la correlación de las variables detectadas entre el peso de declaración, cantidad de importaciones y valor CIF

```
sns.heatmap( data_clean[['Peso Declaracion', 'DIAS TARDA SALIR','Cantidad de importaciones', 'Valor Cif. $.']].corr(),annot=True)
```

- Se identificaron los valores atípicos

```
temp = pd.DataFrame()

for col in data_clean.columns:
    column = data_clean[col]

    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)

    IQR = Q3 - Q1

    umbral_inferior = Q1 - 1.5 * IQR
    umbral_superior = Q3 + 1.5 * IQR

    data_clean[col + '_outliers'] = (column < umbral_inferior) | (column > umbral_superior)

    #temp[col + '_outliers'] = ((column < umbral_inferior) | (column > umbral_superior))

    print(f'{Q1 =} {Q3=} {IQR =} {umbral_inferior = } {umbral_superior =}')
```

- Identificación de los outlier

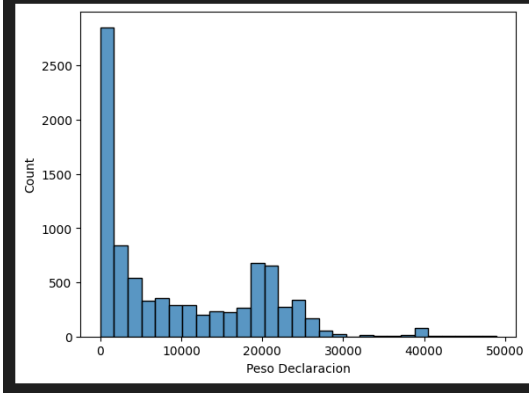
```
data_clean.groupby(by='Peso Declaracion_outliers').count()
```

Peso Declaracion_outliers	Peso Declaracion	DIAS TARDA SALIR	Cantidad de importaciones	Valor Cif. $.	DIAS TARDA SALIR_outliers	Cantidad de importaciones_outliers	Valor Cif. $._outliers
False	9531	9531	9531	9531	9531	9531	9531
True	667	667	667	667	667	667	667

1. Eliminación de filas con outlier identificados, para evitar sesgo en la información del dataset

```
sns.histplot(data=data_noout, x='Peso Declaracion')
```

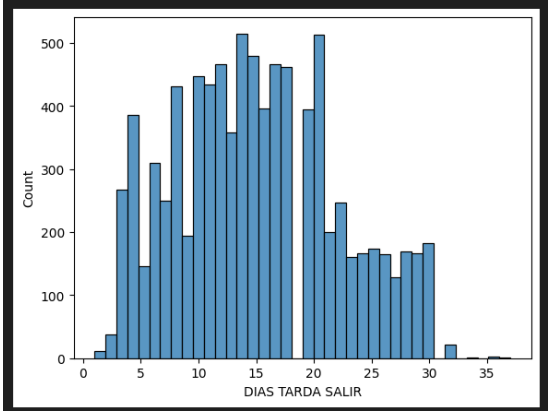
```
<Axes: xlabel='Peso Declaracion', ylabel='Count'>
```



2. Generación de nuevo histograma con la aplicación de limpieza de datos, aplicada a cada una de las variables identificadas.

```
sns.histplot(data=data_noout, x='DIAS TARDA SALIR')
```

```
<Axes: xlabel='DIAS TARDA SALIR', ylabel='Count'>
```



PARA EL DESARROLLO DE REDES NEURONALES RECURRENTES, SE DETERMINÓ UNA SEMILLA ALEATORIA CON VALOR DE 10, UTILIZANDO LA FUNCIÓN SEED (10)

```
import numpy as np
np.random.seed(10)
```

3. Se muestran los valores Lamda

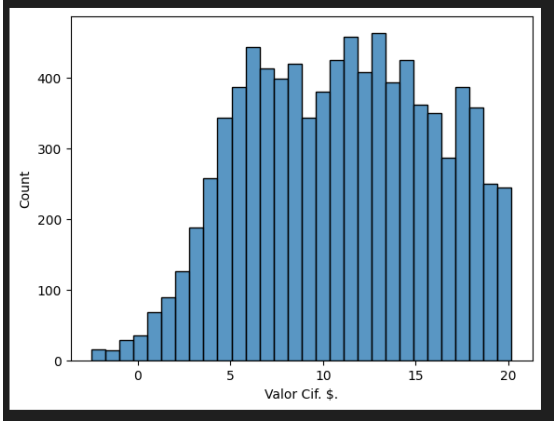
```
df_lambdas = {
    'nombre' : nombre,
    'lambda' : lambdas
}

df_lambdas = pd.DataFrame(df_lambdas)
df_lambdas
```

	nombre	lambda
0	Peso Declaracion	0.275843
1	DIAS TARDA SALIR	0.681006
2	Cantidad de importaciones	0.708408
3	Valor Cif. $.	0.124225

4. Se genera un nuevo dato estadístico para analizar los nuevos valores

```
sns.histplot(data_ab,x= 'Valor Cif. $.')
```

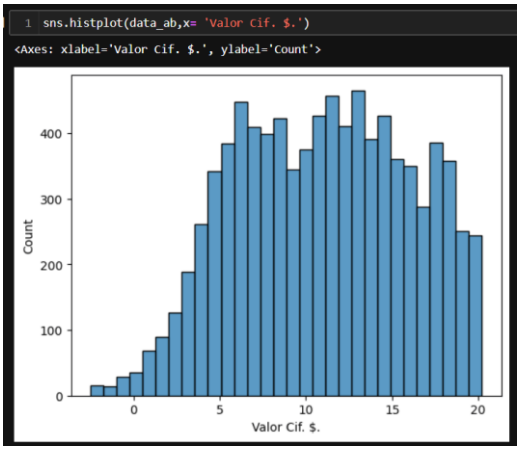
```
<Axes: xlabel='Valor Cif. $.', ylabel='Count'>
```



5. Mediante representación gráfica se presenta el conjunto de datos con el ajuste realizado.

```
1 sns.histplot(data_ab,x= 'Valor Cif. $.')
```

```
<Axes: xlabel='Valor Cif. $.', ylabel='Count'>
```



6. Se generó y ajusto, mediante modelo KMeans con diferentes valores de k y guardando la inercia de cada modelo en una lista, así como los valores de k correspondientes en otra lista.

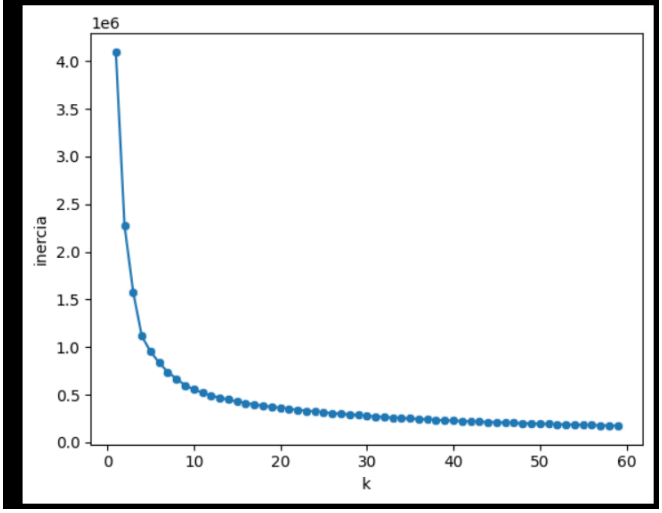
```
inercia = []
ks = []

for k in range(1,60):
    modelo_iterado = KMeans(n_clusters = k, random_state = 5)
    modelo_iterado.fit(data_ab)
    inercia.append(modelo_iterado.inertia_)
    ks.append(k)
```

7. Una vez aplicado y creado el data frame “codo”, se toma la variable “inercia” y con ello se desarrolla un nuevo gráfico de dispersión que muestra los puntos que conecta el gráfico 2D.

```
sns.scatterplot(data=codo,x='k',y='inercia',markers=True)
sns.lineplot(data=codo, x='k', y='inercia',)
```

✓ 0.1s

```
<Axes: xlabel='k', ylabel='inercia'>
```



8. El siguiente paso fue detectar los clúster, mediante el algoritmo de clustering, con el objetivo de agrupar los datos utilizando el parámetro “randomstate=10.

```
modelo = KMeans(n_clusters=5,random_state = 10)
modelo.fit(data_ab)
```

9. Se muestra la agrupación obtenida de los clúster

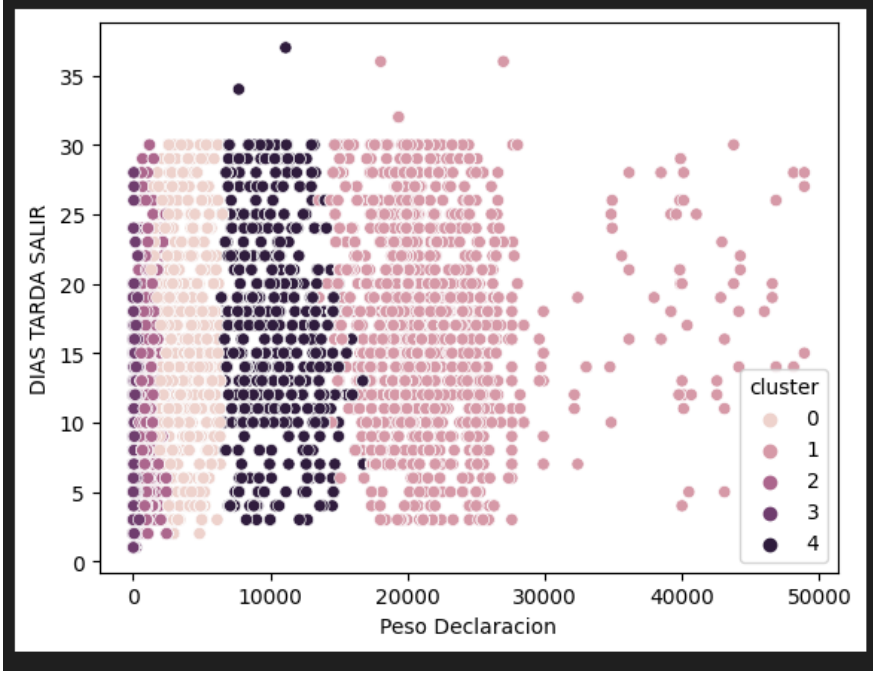
```
data_noout.groupby(by='cluster').agg(['min','mean','max'])
```

	Peso Declaracion			DIAS TARDA SALIR			Cantidad de importaciones			Valor Cif. $.		
	min	mean	max	min	mean	max	min	mean	max	min	mean	max
cluster												
0	1326.00	4013.403902	6577.51	2.0	15.556981	30.0	1.0	5.619623	10.0	2.00	4316.264702	24150.00
1	13417.00	21909.351930	48916.00	3.0	17.561397	36.0	1.0	5.604446	10.0	0.77	6200.360388	24332.00
2	400.24	1269.095697	3020.71	2.0	12.648251	30.0	1.0	5.758236	10.0	0.65	901.470071	24149.48
3	0.15	159.790740	462.16	1.0	10.751220	28.0	1.0	5.464228	10.0	0.05	1262.596618	22156.96
4	6426.12	10472.526664	16828.56	3.0	16.516175	37.0	1.0	5.486700	10.0	0.50	3996.460726	24113.28

10. Una vez identificados los clúster y agrupado los valores, se presenta en forma gráfica el resultado obtenido de la detección realizada.

```
sns.scatterplot(data=data_noout,x='Peso Declaracion', y='DIAS TARDA SALIR', hue='cluster')
```

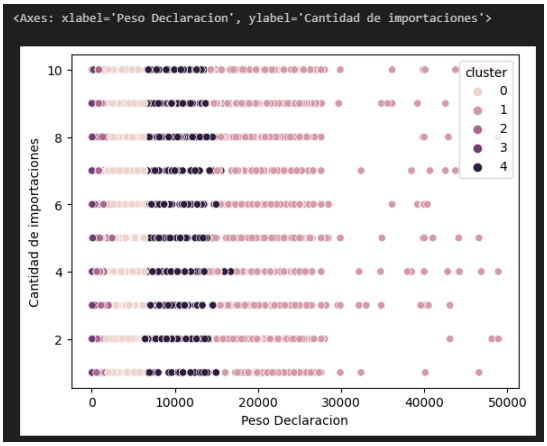
```
<Axes: xlabel='Peso Declaracion', ylabel='DIAS TARDA SALIR'>
```



En la siguiente representación gráfica se identifica los cuatro clúster y la forma en que se agrupan, utilizando la función scatterplot.

```
sns.scatterplot(data=data_noout,x='Peso Declaracion', y='Cantidad de importaciones', hue='cluster')
```

```
<Axes: xlabel='Peso Declaracion', ylabel='Cantidad de importaciones'>
```



CONCLUSIÓN

En este proyecto se abordó la problemática que tienen algunas aduanas en el país, en cuanto al atraso de días durante el despacho de mercancías, por lo que se aplicó técnicas mediante análisis exploratorio de Dataset de una muestra de transacciones de importación, además técnicas de ingeniería de características, que su objetivo fundamental fue aplicar limpieza de datos en la información para posteriormente realizar análisis mediante Redes Recurrentes.

El problema detectada fueron los días que tardan en salir las declaraciones en las diferentes Aduanas del territorio guatemalteco; se establecen riesgos en el ingreso de mercancías, en ocasiones existen verificaciones que conllevan una mayor cantidad de tiempo en su revisión física y documental.

El objetivo de la revisión en aduanas es el cumplimiento de los pagos arancelarios de ello deriva la importancia de las verificaciones física y documental de las mercancías que ingresan al país, sin embargo, se ha detectado atrasos en el despacho de estas, por lo que fue necesario realizar un proceso de análisis en las transacciones para la detectar hallazgos y posibles soluciones.

En el análisis se establecieron las siguientes características de los clúster:

Clúster	Característica
0	Medio
1	Alto
2	Medio
3	Bajo
4	Alto

Cluster	Peso Declaracion			Dias tarda salir			Cantidad de Importaciones			Valor Cif. $.		
	min	mean	max	min	mean	max	min	mean	max	min	mean	max
0	1326	4013.4039	6577.51	2	15.556981	30	1	5.619623	10	2	4316.2647	24150
1	13417	21909.3519	48916	3	17.561397	36	1	5.604446	10	0.77	6200.36039	24332
2	400.24	1269.0957	3020.71	2	12.648251	30	1	5.758236	10	0.65	901.470071	24149.48
3	0.15	159.79074	462.16	1	10.75122	28	1	5.464228	10	0.05	1262.59662	22156.96
4	6426.12	10472.5267	16828.56	3	16.516175	37	1	5.4867	10	0.5	3996.46073	24113.28

De acuerdo a la información obtenida en el procesamiento de datos y detección de principales hallazgos, se recomienda para mejorar los tiempos de despacho de mercancías lo siguiente:

Invertir en sistemas de gestión aduanera y tecnología que permita automatizar los procesos de despacho y con ello acelerar el flujo en las aduanas.
Se sugiere programar la distribución del personal previamente, para atender de manera efectiva el despacho de mercancías, así mismo realizar estudio de carga laboral con la finalidad de establecer si se cuenta con la necesidad de incrementar el personal que realiza las verificaciones de mercancías o se deben redistribuir las atribuciones con el fin de mejorar los tiempos de atención y salida de mercancías.

Anexos o Enlace

https://github.com/ecristal80/Proyecto_final_Statistial.git