



Tecnológico de Monterrey

Maestría en Inteligencia Artificial Aplicada

Materia | Proyecto Integrador

Laura Elena Hernández Mata | A01169213

Evelyn Aylin Rendon Medina | A01748750

Samara García González | A01273001

Proyecto | Islas de Calor y Justicia Social

Avance 1 | Análisis exploratorio de datos

Asesor: Prof. Dr. Roberto Ponce López

Tutora: María de la Paz Rico Fernández

29 de septiembre de 2024

Índice

1. Análisis exploratorio.....	3
• ¿Se identifican tendencias temporales? (En caso de que el conjunto incluya una dimensión de tiempo).....	3
2. Estructura de los datos.....	4
2.1 Descripción general de la forma y los tipos de datos.....	4
• ¿Existen distribuciones sesgadas en el conjunto de datos? ¿Necesitamos aplicar alguna transformación no lineal?.....	7
• ¿Se deberían normalizar las imágenes para visualizarlas mejor?.....	7
• ¿Hay desequilibrio en las clases de la variable objetivo?.....	8
2.2 Estadísticas descriptivas para las variables del conjunto.....	8
2.3 Frecuencia de las clases en variables categóricas.....	12
• ¿Cuál es la cardinalidad de las variables categóricas?.....	13
2.4 Identificación de valores faltantes.....	13
• ¿Hay valores faltantes en el conjunto de datos? ¿Se pueden identificar patrones de ausencia?.....	13
3. Análisis univariante.....	14
• ¿Cuáles son las estadísticas resumidas del conjunto de datos?.....	14
• ¿Hay valores atípicos en el conjunto de datos?.....	14
4. Análisis bi/multivariante.....	15
• ¿Hay correlación entre las variables dependientes e independientes?.....	15
• ¿Cómo se distribuyen los datos en función de diferentes categorías?.....	16
En la siguiente imagen observamos a grandes rasgos la distribución de nuestras variables, esto por medio de histogramas. La función de esto es poder conocer cómo se comportan cada una de las variables.....	16
Cabe destacar que en algunas de las gráficas se observa un único valor, como sumatoria que nos resume en realidad el total, puesto que la segmentación por AGEB se hará en etapas posteriores.....	17
• ¿Existen patrones o agrupaciones (clusters) en los datos con características similares?.....	17
5. Procesamiento.....	19
Preparación de los Datos:.....	19
Selección de Herramientas:.....	20
Proceso de Geoprocesamiento:.....	20
6. Conclusiones.....	22
7. Referencias.....	23
8. Anexos.....	24

1. Análisis exploratorio

Entender la pregunta de negocio, es decir, resumir el objeto de estudio a la respuesta que plantea es esencial para cualquier proyecto relacionado con datos. Sin embargo, es solo el primer paso del planteamiento, pues será indispensable conocer la materia prima que se dispone para lograrlo, es decir, el estado actual de nuestros datos en calidad, cantidad y en general, qué compone nuestro set que alimentará o entrenará el modelo.

Por tanto, es de suma relevancia para este proyecto tener un entendimiento profundo de nuestro conjunto de datos, es decir, conocer nuestras variables, cómo se comportan, qué valores atípicos poseen, cómo se distribuyen y la forma en que están sesgadas las clases. Solo a partir de lo anterior, de realizar un análisis exploratorio y definición de tratamiento de acuerdo con lo hallado, será posible cimentar la construcción de un modelo que permita responder la pregunta planteada.

Si bien lo anterior se define como esencial, es importante destacar que el análisis y la preparación de los datos es lo que más tiempo puede llevar en un proyecto de esta naturaleza, pues debemos tener certeza que la selección y tratamiento son los adecuados para el siguiente proceso que se ha planteado llevar a cabo, lo que será, por tanto, parte de un proceso iterativo de mejora; hasta ser capaces de alcanzar el comportamiento adecuado de nuestro modelo.

- ¿Se identifican tendencias temporales? (En caso de que el conjunto incluya una dimensión de tiempo).

En nuestro caso de análisis particular no se analiza ninguna variable en función del tiempo, puesto que, si bien para obtener el resultado de temperatura promedio para las imágenes de las islas de calor, se analizan muestras alrededor de un año, 2022, el resultado es una imagen o fotografía estática que emplea el promedio con un intervalo de confianza, sin variación conforme el tiempo. Por otro lado, los datos provenientes del INEGI también corresponden al estudio realizado en un año particular, en este caso, 2020.

2. Estructura de los datos

En esta sección se encuentra lo relacionado con la estructura y análisis de los datos, así como la identificación de las variables faltantes. Esta parte de la entrega tiene la finalidad de contener los aspectos básicos previos al análisis un/bi/multivariable y el procesamiento de la información del proyecto.

2.1 Descripción general de la forma y los tipos de datos

A continuación explicaremos la forma en que están compuestos nuestros datos, tanto los relativos a las islas de calor, como también los datos obtenidos del INEGI.

1. Dataset de Imágenes Raster de Islas de Calor en México

- **Forma:**

Consta de imágenes en formato raster, donde cada píxel representa un valor de temperatura (en grados Celsius) correspondiente a diferentes áreas geográficas. Así mismo una segunda sección de imágenes raster pero enfocadas a zonas rurales. Las imágenes están georreferenciadas, lo que permite su ubicación precisa en un sistema de coordenadas geográficas.

- **Tipos de Datos:**

- **Valor de Temperatura:** Datos continuos que representan las temperaturas en diferentes áreas.
- **SCITEL.** Primordialmente datos numéricos, correspondientes a las variables de la población, en particular, las que consideramos
 - **Clave de entidad federativa** Código que identifica a la entidad federativa. El código 00 identifica a los registros con los totales a nivel nacional.
 - **Entidad federativa** Nombre oficial de la entidad federativa.
 - **Clave de municipio o demarcación territorial** Código que identifica al municipio o demarcación territorial al interior de una entidad federativa, conforme al Marco Geoestadístico. El código 000 identifica a los registros con los totales a nivel de entidad federativa.

- Municipio o demarcación territorial Nombre oficial del municipio o demarcación territorial en el caso de la Ciudad de México.
- Clave de localidad Código que identifica a la localidad al interior de cada municipio o demarcación territorial conforme al Marco Geoestadístico. El código 0000 identifica a los registros con los totales a nivel de municipio o demarcación territorial.
- Localidad Nombre con el que se reconoce a la localidad dado por la ley o la costumbre.
- Clave del AGEB Clave que identifica al AGEB urbana, al interior de una localidad, de acuerdo con la desagregación del Marco Geoestadístico.
- Clave de manzana Clave que identifica a la manzana, al interior de una AGEB, de acuerdo a la desagregación del Marco Geoestadístico.
- Población total Total de personas que residen habitualmente en el país, la entidad federativa, el municipio o la demarcación territorial y la localidad. Incluye la estimación del número de personas en viviendas particulares sin información de ocupantes. Incluye a la población que no especificó su edad.
- Población femenina Total de mujeres que residen habitualmente en el país, la entidad federativa, el municipio o la demarcación territorial y la localidad. Incluye la estimación del número de mujeres en viviendas particulares sin información de ocupantes. Incluye a la población que no especificó su edad.
- Población masculina Total de hombres que residen habitualmente en el país, la entidad federativa, el municipio o la demarcación territorial y la localidad. Incluye la estimación del número de hombres en viviendas particulares sin información de ocupantes. Incluye a la población que no especificó su edad.

2. Base de Datos del INEGI (Censo de Población y Vivienda 2020 - SCITEL)

A continuación mostramos un resumen de los datos en forma tabular.

Bases de datos para el proyecto			
Nombre de base de datos	SCITEL (INEGI, 2020)	Islas de calor (centígrados)	Islas de calor (categorizado)
Tipo de datos	Estructurados	Sin estructurar	Estructurados
Formato	xlsx	tif	dataframe
Fuente de información	Censo del INEGI	Información satelital provista para el proyecto	Información satelital previamente procesada para la realización del proyecto

Tabla 1. Bases de datos para el proyecto. (Elaboración propia, 2024).

- **Forma:**

Estructurada en formato Excel, con filas y columnas que organizan la información de manera tabular. Cada fila representa un municipio o una unidad geográfica (AGEB), y cada columna representa diferentes variables e indicadores.

- **Tipos de Datos:**

- **Municipio:** Identificador categórico que indica el nombre o código del municipio.
- **Estado:** Categórico, indicando a qué estado pertenece cada municipio.
- **AGEB (Área Geoestadística Básica):** Identificador categórico que clasifica áreas para fines estadísticos.
- **Población Total:** Datos numéricos que representan el número total de habitantes en cada municipio.
- **Indicadores Socioeconómicos:** Incluyen variables como el ingreso promedio, el nivel de educación, la vivienda, el acceso a servicios básicos, entre otros.

- ¿Existen distribuciones sesgadas en el conjunto de datos? ¿Necesitamos aplicar alguna transformación no lineal?

De primera instancia, sabemos que se tiene sesgo en la distribución de los datos relativos a las temperaturas encontradas en el análisis. Lo anterior puede jugar en contra al generar clasificaciones, dependiendo de la sensibilidad del modelo seleccionado, el tratamiento específico que se le brinden a los datos y los hiperparámetros.

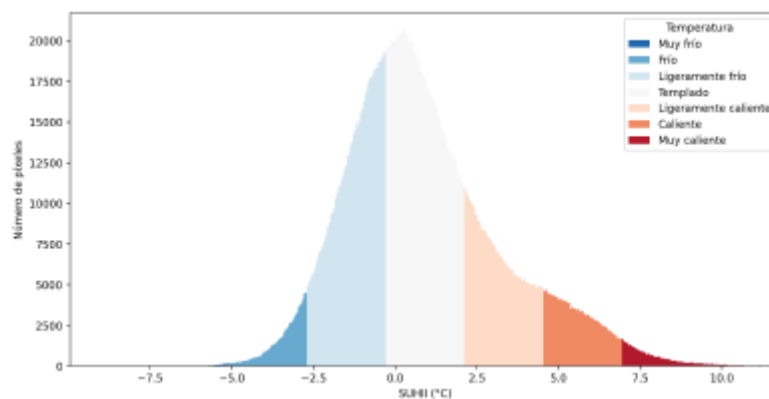


Imagen 1. Distribución de temperatura basadas en las estadísticas de la imagen. (Torre, et. al. 2023)

Sin embargo, es importante entender y asociar este desbalanceo de clases, porque es precisamente la búsqueda de las zonas intensas de calor lo que más nos interesa, es decir, la vulnerabilidad contra el calor y, como podemos observar, las zonas de mayor calor son las áreas con menor cantidad de observaciones.

Así pues, en este primer análisis exploratorio se observa dicho comportamiento, pero aún es necesario evaluar cómo afectará al modelo de clasificación, pues el tratamiento es también un proceso iterativo y adaptativo.

- ¿Se deberían normalizar las imágenes para visualizarlas mejor?

Sí, se planea realizar una normalización de las imágenes del dataset, el método considerado para esto es la normalización por percentiles, ya que nos permite escalar los valores de temperatura usando los percentiles 1 y 99 y disminuir los valores extremos o atípicos. Además es importante recalcar que se deben redimensionar las imágenes a un solo tamaño, ya que actualmente vienen en diferentes tamaños y esto dificulta su análisis y comparación visual.

- ¿Hay desequilibrio en las clases de la variable objetivo?

Sí, como se observa en la gráfica superior (Torre, et. al. 2023), el enfoque en las zonas de calor intenso revela que estas áreas, que representan mayor vulnerabilidad, tienen menos observaciones. Este desbalance puede afectar el rendimiento del modelo por lo que se debe evaluar su impacto y considerar técnicas de ajuste.

2.2 Estadísticas descriptivas para las variables del conjunto

Realizamos un primer abordaje para conocer cómo se realiza la lectura de nuestras variables, describiendo cada columna.

```
In [ ]: # Observamos la estructura
df_concat.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1640722 entries, 0 to 1640721
Data columns (total 22 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   AGE8             1640720 non-null object
1   POBTOT           1640720 non-null float64
2   POBFEM           1640720 non-null object
3   POBMAS           1640720 non-null object
4   P_0A2            1640720 non-null object
5   P_3A5            1640720 non-null object
6   P_60YMAS         1640720 non-null object
7   POB65_MAS        1640720 non-null object
8   P3HLINHE         1640720 non-null object
9   PHOG_IND         1640720 non-null object
10  PCON_DISC        1640720 non-null object
11  PCON_LIMI        1640720 non-null object
12  PSIND_LIM        1640720 non-null object
13  P15YM_AN         1640720 non-null object
14  P15YM_SE         1640720 non-null object
15  GRAPROES         1640720 non-null object
16  PEA              1640720 non-null object
17  PSINDER          1640720 non-null object
18  PDER_SS          1640720 non-null object
19  TOTHOOG          1640720 non-null object
20  POBHOG           1640720 non-null object
21  VIVTOT           1640720 non-null float64
dtypes: float64(2), object(20)
memory usage: 275.4+ MB
```

Imagen 2. Estructura de datos. (Elaboración propia, 2023)

Tal como se aprecia en la imagen anterior, la mayoría de los datos no están catalogados como numéricos, esto a pesar que al abrir las bases de datos sí son números. Esto puede pasar por diversas circunstancias, entre las que destaca, por ejemplo, el uso de caracteres especiales.

Una vez que se analizaron los datos, encontramos el caracter “*” como parte de la información, por esto es que no aparecían los datos como numéricos, tal como se explicó anteriormente. Por tanto, decidimos optar por distintas formas de tratamiento de los datos no congruentes, como eran en este caso los caracteres “*”.

```
In [ ]: # Eliminamos los caracteres de tipo *
df_concat.replace('*', 0, inplace=True)
df_concat.head()
```

```
Out[ ]:  AGEB  POSTOT  POBFEM  POBMAS  P_0A2  P_3A5  P_60YMAS  POB65_MAS  P3HLINHE  PHOG_IND  ...  PSIH
```

0	0000	2368467.0	1211647	1156820	108229	119277	240222	159493	594	62207	...	2
1	0000	66841.0	34606	32235	4151	4297	6767	4902	452	21072	...	
2	0000	7953.0	4224	3729	364	361	996	705	0	156	...	
3	0042	84.0	43	41	4	3	6	5	0	0	...	
4	0042	13.0	0	0	0	0	0	0	0	0	...	

5 rows x 22 columns

Imagen 3. Tratamiento de caracteres tipo “*”. (Elaboración propia, 2023)

Decidimos realizar distintas estrategias para conocer cuál sería mejor en el tratamiento de la información. Cada una de éstas se caracteriza por la copia del DataFrame original y la realización de distintos cambios en dichas copias. Las estrategias abordadas fueron:

1. Eliminación de renglones/columnas faltantes

En esta parte mostraremos el paso a paso y cuáles son los resultados de cada una de las modificaciones. Iniciamos con la eliminación de las columnas con valores nulos y mantenemos un seguimiento de los renglones. En este primer paso, notamos que el número de renglones es de 1,640,722; esto se aprecia en la siguiente imagen.

```
[71] ndfs1 = df_concat.copy()
      ndfs1.dropna(axis = 1, inplace = True) # axis 1 son columnas / axis 0 son renglones.
      print("Número de renglones: ", len(ndfs1))
      ndfs1.head()
```

Número de renglones: 1640722

0
1
2
3
4

Imagen 4. Eliminación de columnas con valores nulos. (Elaboración propia, 2023)

Posteriormente, realizamos una eliminación de todos los renglones donde haya valores nulos. Podemos observar que en el resultado tenemos aún el total de 1,640,720 renglones. Es importante mencionar que nos aseguramos de mantener los DataFrames originales intactos y sólo trabajar en nuevos.

```
[72] ndfs2 = df_concat.copy()
      ndfs2.dropna(how='all', inplace = True)
      print("Número de renglones: ", len(ndfs2))
      ndfs2.head()
```

Número de renglones: 1640720

	AGEB	POBTOT	POBFEM	POBMAS	P_OA2	P_3A5	P_60YMAS	POB65_MAS	P3HLINHE	PHOG_IND	...	PSIND_LIM	P15YH_LAN	P15YH_SE	GRAPROES	PEA	PSINDER	POER_SS	TOTHOG	POBHOG	VIVTOT
0	0000	2368467.0	1211647	1156820	108229	119277	240222	159493	594	62207	...	2004940	61734	75526	10.48	1233080	486467	1873160	668487	2362208	826353.0
1	0000	66841.0	34606	32235	4151	4297	6767	4902	452	21072	...	56134	5099	5509	7.24	33329	15269	51533	17122	66793	24475.0
2	0000	7953.0	4224	3729	364	361	996	705	0	156	...	6797	173	259	9.9	4103	2191	5753	2142	7941	2824.0
3	0042	84.0	43	41	4	3	6	5	0	0	...	66	3	4	11.17	42	39	45	18	84	25.0
4	0042	13.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0

5 rows x 22 columns

Imagen 5. Eliminación de renglones con valores nulos. (Elaboración propia, 2024)

Aplicamos otro método de eliminación para nulos. Observamos que la cantidad de filas se mantiene intacta con un total de 1,640,722.

```
[74] ndfs4 = df_concat.copy()
      ndfs4.dropna(thresh = 5, axis = 1, inplace = True)
      print("Número de renglones: ", len(ndfs4))
      ndfs4.head()
```

Número de renglones: 1640722

	AGEB	POBTOT	POBFEM	POBMAS	P_OA2	P_3A5	P_60YMAS	POB65_MAS	P3HLINHE	PHOG_IND	...	PSIND_LIM	P15YH_LAN	P15YH_SE	GRAPROES	PEA	PSINDER	POER_SS	TOTHOG	POBHOG	VIVTOT
0	0000	2368467.0	1211647	1156820	108229	119277	240222	159493	594	62207	...	2004940	61734	75526	10.48	1233080	486467	1873160	668487	2362208	826353.0
1	0000	66841.0	34606	32235	4151	4297	6767	4902	452	21072	...	56134	5099	5509	7.24	33329	15269	51533	17122	66793	24475.0
2	0000	7953.0	4224	3729	364	361	996	705	0	156	...	6797	173	259	9.9	4103	2191	5753	2142	7941	2824.0
3	0042	84.0	43	41	4	3	6	5	0	0	...	66	3	4	11.17	42	39	45	18	84	25.0
4	0042	13.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0

5 rows x 22 columns

Imagen 6. Eliminación de renglones con valores nulos. (Elaboración propia, 2024)

2. Estrategia 2: Imputación de datos faltantes

Decidimos también realizar la imputación de datos faltantes a través de diversas técnicas tales como la moda, la media y la mediana. Si bien esto lo hicimos para experimentar, cabe mencionar que no elegimos ninguna de estas técnicas, ya que podrían ensuciar los datos poblacionales. Igualmente, al final lo que decidimos es tratar los nulos como datos faltantes y, posteriormente, esos fueron eliminados de la base para los análisis subsecuentes.

2.1.- Como primer paso realizamos una copia del dataset original para no perder los cambios realizados hasta ahora, posteriormente obtenemos la media del campo a imputar

```
[76] nfp10 = df_concat.copy()
      TutuChiclanas = nfp10.VIVTOT.mean()
      print(TutuChiclanas, "\n", TutuChiclanas)

Total de elementos: 126.8883736288524

[77] # Una vez que tenemos la media, imputamos el valor en la columna
      nfp10['VIVTOT'] = nfp10.VIVTOT.fillna(TutuChiclanas, inplace = True)
      nfp10.head()
```

	AJER	POBTOT	POBTEN	POBTEN	P_2002	P_2003	P_2004	P_2005	POBTEN_2004	POBTEN_2005	POBTEN_2006	POBTEN_2007	POBTEN_2008	POBTEN_2009	POBTEN_2010	POBTEN_2011	POBTEN_2012	POBTEN_2013	POBTEN_2014	POBTEN_2015	POBTEN_2016	POBTEN_2017
0	0000	208407.0	121947	108628	108228	116277	246222	108463	394	62207	---	208407	61734	70526	10.40	1233000	408407	1071450	888407	2362200	826321.8	
1	0000	58841.0	34806	32238	4191	4287	6707	4892	452	21072	---	58134	5888	5888	7.24	32238	15288	81533	17122	98783	24476.8	
2	0000	17851.0	4224	3728	384	381	585	785	0	156	---	6707	172	258	9.9	4191	2191	5753	2142	7841	2824.8	
3	0042	84.0	43	41	4	3	0	0	0	0	---	88	3	4	11.17	43	38	43	18	84	25.8	
4	0042	11.0	7401	7401	7401	7401	7401	7401	7401	7401	---	7401	7401	7401	7401	7401	7401	7401	7401	7401	7401	

5 rows x 22 columns

[78] # Verificamos que ya no tenemos valores nulos en la columna Inputada

```
print(nfp10['VIVTOT'].isnull().values.any())
```

False

2.2.- Como primer paso realizamos una copia del dataset original para no perder los cambios realizados hasta ahora, posteriormente imputamos con la mediana

```
[79] nfp2 = df_concat.copy()
      nfp2['VIVTOT'] = nfp2.VIVTOT.fillna(nfp2.VIVTOT.median(), inplace = True)
      nfp2.head()
```

	AJER	POBTOT	POBTEN	POBTEN	P_2002	P_2003	P_2004	P_2005	POBTEN_2004	POBTEN_2005	POBTEN_2006	POBTEN_2007	POBTEN_2008	POBTEN_2009	POBTEN_2010	POBTEN_2011	POBTEN_2012	POBTEN_2013	POBTEN_2014	POBTEN_2015	POBTEN_2016	POBTEN_2017
0	0000	208407.0	121947	108628	108228	116277	246222	108463	394	62207	---	208407	61734	70526	10.40	1233000	408407	1071450	888407	2362200	826321.8	
1	0000	58841.0	34806	32238	4191	4287	6707	4892	452	21072	---	58134	5888	5888	7.24	32238	15288	81533	17122	98783	24476.8	
2	0000	17851.0	4224	3728	384	381	585	785	0	156	---	6707	172	258	9.9	4191	2191	5753	2142	7841	2824.8	
3	0042	84.0	43	41	4	3	0	0	0	0	---	88	3	4	11.17	43	38	43	18	84	25.8	
4	0042	11.0	7401	7401	7401	7401	7401	7401	7401	7401	---	7401	7401	7401	7401	7401	7401	7401	7401	7401	7401	

5 rows x 22 columns

[80] # Verificamos que ya no tenemos valores nulos en la columna Inputada

```
print(nfp2['VIVTOT'].isnull().values.any())
```

False

2.3.- Como primer paso realizamos una copia del dataset original para no perder los cambios realizados hasta ahora, posteriormente imputamos con la moda

```
[81] nfp3 = df_concat.copy()
      nfp3['VIVTOT'] = nfp3.VIVTOT.fillna(nfp3.VIVTOT.mode()[0], inplace = True)
      nfp3.head()
```

	AJER	POBTOT	POBTEN	POBTEN	P_2002	P_2003	P_2004	P_2005	POBTEN_2004	POBTEN_2005	POBTEN_2006	POBTEN_2007	POBTEN_2008	POBTEN_2009	POBTEN_2010	POBTEN_2011	POBTEN_2012	POBTEN_2013	POBTEN_2014	POBTEN_2015	POBTEN_2016	POBTEN_2017
0	0000	208407.0	121947	108628	108228	116277	246222	108463	394	62207	---	208407	61734	70526	10.40	1233000	408407	1071450	888407	2362200	826321.8	
1	0000	58841.0	34806	32238	4191	4287	6707	4892	452	21072	---	58134	5888	5888	7.24	32238	15288	81533	17122	98783	24476.8	
2	0000	17851.0	4224	3728	384	381	585	785	0	156	---	6707	172	258	9.9	4191	2191	5753	2142	7841	2824.8	
3	0042	84.0	43	41	4	3	0	0	0	0	---	88	3	4	11.17	43	38	43	18	84	25.8	
4	0042	11.0	7401	7401	7401	7401	7401	7401	7401	7401	---	7401	7401	7401	7401	7401	7401	7401	7401	7401	7401	

5 rows x 22 columns

[82] # Verificamos que ya no tenemos valores nulos en la columna Inputada

```
print(nfp3['VIVTOT'].isnull().values.any())
```

False

2.4.- Como primer paso realizamos una copia del dataset original para no perder los cambios realizados hasta ahora, posteriormente eliminamos los rengiones especificos donde hay nulos

```
[84] nfp5b = df_concat.copy()
      nfp5b.dropna(subset=['VIVTOT', 'POBTOT'], inplace = True)
      len(nfp5b)
```

1640720

2.5.- Como primer paso realizamos una copia del dataset original para no perder los cambios realizados hasta ahora, imputamos dos columnas con dos estrategias diferentes

```
[86] nfp510 = df_concat.copy()
      nfp510.VIVTOT.fillna(nfp510.VIVTOT.mode()[0], inplace=True)
      nfp510.POBTOT.fillna(nfp510.POBTOT.mean(), inplace=True)
      nfp510.head()
      print(nfp510['VIVTOT'].isnull().values.any())
      print(nfp510['POBTOT'].isnull().values.any())
```

False

False

Imagen 7. Estrategia de imputación de datos faltantes. *(Elaboración propia, 2024)*

De forma que, una vez determinada la estrategia final, se lograron tratar los caracteres “*” y poder convertir las columnas de tipo object a int, puesto que correspondía que el tipo de datos que contienen

```
In [ ]: # Hacemos el cambio para que todas las variables sean numéricas
dfs= df_concat.apply(pd.to_numeric, errors='coerce').fillna(0).astype(int)
dfs.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1495346 entries, 0 to 1495345
Data columns (total 22 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   AGEB        1495346 non-null  int64
1   POBTOT      1495346 non-null  int64
2   POBFEM      1495346 non-null  int64
3   POBMAS      1495346 non-null  int64
4   P_OA2       1495346 non-null  int64
5   P_3A5       1495346 non-null  int64
6   P_60YMAS    1495346 non-null  int64
7   POB65_MAS   1495346 non-null  int64
8   P3HLINHE    1495346 non-null  int64
9   PHOG_IND    1495346 non-null  int64
10  PCON_DISC   1495346 non-null  int64
11  PCON_LIMI   1495346 non-null  int64
12  PSIND_LIH   1495346 non-null  int64
13  P1SYM_AN    1495346 non-null  int64
14  P1SYM_SE    1495346 non-null  int64
15  GRAPROES    1495346 non-null  int64
16  PEA         1495346 non-null  int64
17  PSINDER     1495346 non-null  int64
18  PDER_SS     1495346 non-null  int64
19  TOTHOG      1495346 non-null  int64
20  POBHOG      1495346 non-null  int64
21  VIVTOT      1495346 non-null  int64
dtypes: int64(22)
memory usage: 251.0 MB
```

Imagen 8. Conversión de datos. *(Elaboración propia, 2024)*

2.3 Frecuencia de las clases en variables categóricas

Respecto a los sets provenientes del INEGI, nuestra primordial categoría corresponde a los Estados, teniendo por tanto al nivel más superficial, datos para cada uno de los 32 Estados de la República. Sin embargo, ya que se determinó que el estudio sería realizado por AGEB, es dicho nivel de detalles relativo a las categorías que podríamos considerar como el primario.

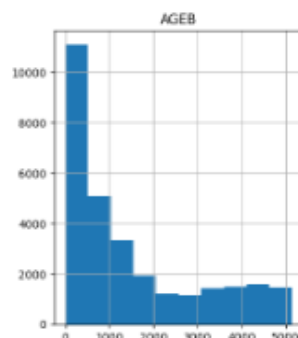


Imagen 9. Distribución de registros por AGEB. *(Fuente propia 2024)*

- ¿Cuál es la cardinalidad de las variables categóricas?

Una de las principales categorías en nuestra base de datos de islas de calor está dada por los rangos de temperatura, que están determinados de acuerdo con la información compartida. Tal como se muestra en la siguiente imagen, donde se consideran los rangos de temperatura y se clasifican con nomenclatura nominal.

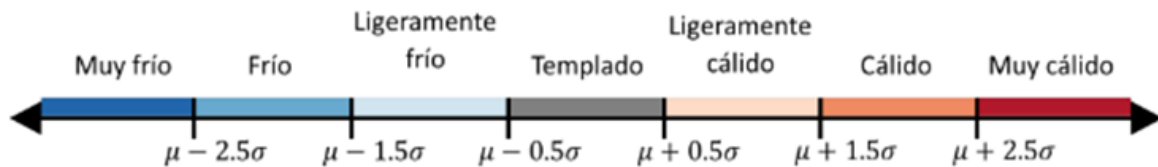


Imagen 10. Categorías de temperatura basadas en las estadísticas de la imagen. (Torre, et. al. 2023)

2.4 Identificación de valores faltantes

Los valores faltantes son aquellos datos que no están presentes en un conjunto de datos, ya sea porque no se registraron, se perdieron o no se recopilaron. En el análisis de datos, los valores faltantes pueden aparecer en cualquier variable y pueden afectar la calidad y precisión del análisis, causando sesgos o errores en los resultados.

- ¿Hay valores faltantes en el conjunto de datos? ¿Se pueden identificar patrones de ausencia?

En las bases de datos del INEGI y en el dataset de imágenes raster con las islas de calor por estado, no hay valores faltantes per se. Sin embargo, nos encontramos con una encrucijada, ya que el dataset del INEGI no contaba con las coordenadas de las AGEb. Esto representa un desafío, ya que la falta de información espacial puede limitar el análisis y la integración con los datos de temperatura. Sobre todo tomando en cuenta que los datos de INEGI están en formato tabular, y se necesitará crear una geometría como en este caso centroides de los AGEb para poder relacionarlos espacialmente.

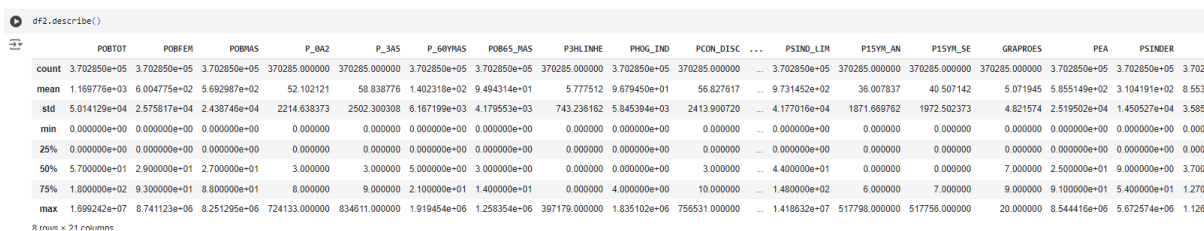
Por esta razón fue necesario recurrir al Marco Geoestadístico. Censo de Población y Vivienda 2020 (INEGI, s.f.) y descargar los archivos .shp por estado.

3. Análisis univariante

El análisis univariante es aquel que permite estudiar, como su nombre lo indica, una variable a través de distintos instrumentos; el cual se abordará en esta sección. Cabe mencionar que se distingue del análisis multivariante ya que en este punto lo que se busca es describir, y, a pesar de que se estudien distintas variables, no se determinará cómo están asociadas, sino más bien conocer sus características. (Mas, 2019)

- ¿Cuáles son las estadísticas resumidas del conjunto de datos?

Al ser un conjunto de datos a total país, se puede apreciar que las estadísticas de la base de datos no muestran una tendencia como tal. Más bien, esto se apreciará de mejor manera cuando se trabajen las mallas por AGEB, donde también se utilizará la información de temperatura.



```
df2.describe()
```

	POBTOT	POBFER	POBMAS	P_0A2	P_3A5	P_60YMAS	POB65_MAS	P3HLINHE	PHOG_IND	PCON_DISC	...	PSIND_LIR	P15YILAN	P15YIL_SE	GRAPROES	PEA	PSINDER
count	3.702850e+05	3.702850e+05	3.702850e+05	370285.000000	370285.000000	3.702850e+05	3.702850e+05	370285.000000	3.702850e+05	370285.000000	...	3.702850e+05	370285.000000	370285.000000	370285.000000	3.702850e+05	3.702850e+05
mean	1.169776e+03	6.004775e+02	5.692987e+02	52.102121	58.838776	1.402318e+02	9.494314e+01	5.777512	9.679450e+01	56.827617	...	9.731452e+02	36.007837	40.507142	5.071945	5.855149e+02	3.104191e+02
std	5.014129e+04	2.575817e+04	2.438746e+04	2214.638373	2502.300308	6.167199e+03	4.179553e+03	743.236162	5.845394e+03	2413.900720	...	4.177016e+04	1871.669762	1972.502373	4.821574	2.519502e+04	1.450527e+04
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000	...	0.000000e+00	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000	...	0.000000e+00	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00
50%	5.700000e+01	2.900000e+01	2.700000e+01	3.000000	3.000000	5.000000e+00	3.000000e+00	0.000000	0.000000e+00	3.000000	...	4.400000e+01	0.000000	0.000000	7.000000	2.500000e+01	9.000000e+00
75%	1.800000e+02	9.300000e+01	8.800000e+01	8.000000	9.000000	2.100000e+01	1.400000e+01	0.000000	4.000000e+00	10.000000	...	1.480000e+02	6.000000	7.000000	9.000000	9.100000e+01	5.400000e+01
max	1.699242e+07	8.741123e+06	8.251295e+06	724133.000000	834611.000000	1.919454e+06	1.258354e+06	397179.000000	1.835102e+06	756531.000000	...	1.418632e+07	517798.000000	517756.000000	20.000000	8.544416e+06	5.672574e+06

R console > 21 columns

Imagen 11. Estadísticas resumidas del conjunto de datos. (Elaboración propia, 2024)

- ¿Hay valores atípicos en el conjunto de datos?

Sí, encontramos que hay valores atípicos en la información de SCITEL (INEGI, 2020). Tal como se aprecia en la siguiente imagen; cabe mencionar que para el procesamiento de datos, decidimos normalizarlos.

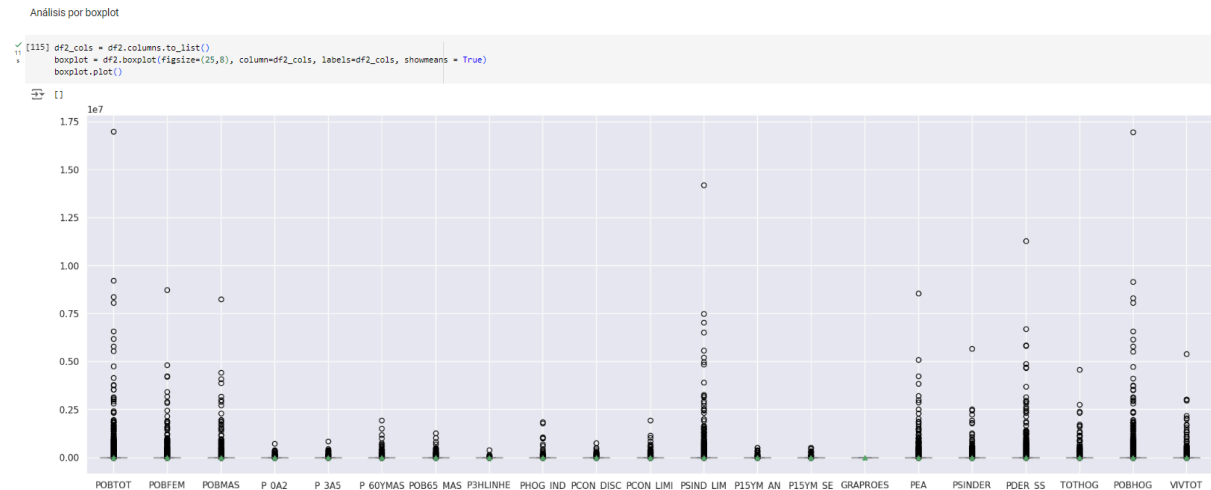


Imagen 12. Gráfica de boxplot para análisis de outliers. (Elaboración propia, 2024)

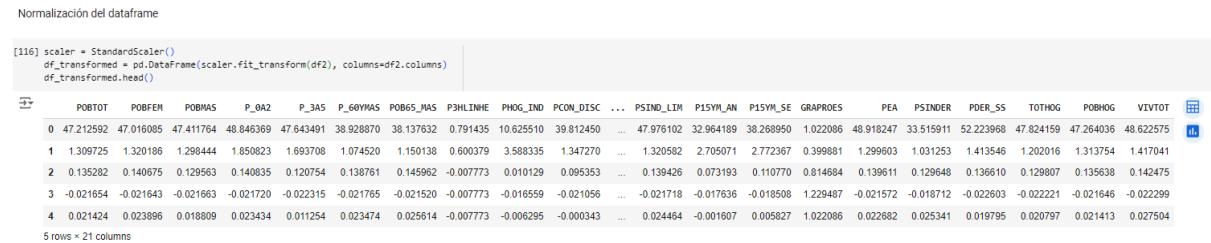


Imagen 13. Normalización del dataframe. (Elaboración propia, 2024)

4. Análisis bi/multivariante

El análisis bivariable nos sirve para conocer la relación entre dos variables, y con ello determinar si el comportamiento de una afecta a la otra. Es decir, si el proceder de una variable está en función de otra. Naturalmente, el análisis multivariante nos sirve para determinar esas relaciones entre más de dos variables. (Bech, 2019)

- ¿Hay correlación entre las variables dependientes e independientes?

Sí, observamos correlación en algunas de las variables, tal como se muestra en la siguiente imagen. Recordemos que aquellas variables que tienen una correlación perfecta son las que tienen un 1; que normalmente esto ocurre con la misma variable, que es lo que se espera. Sin embargo, se puede apreciar que hay, por

ejemplo, una alta correlación cuando hablamos de personas de 60 años o más con alguna discapacidad.

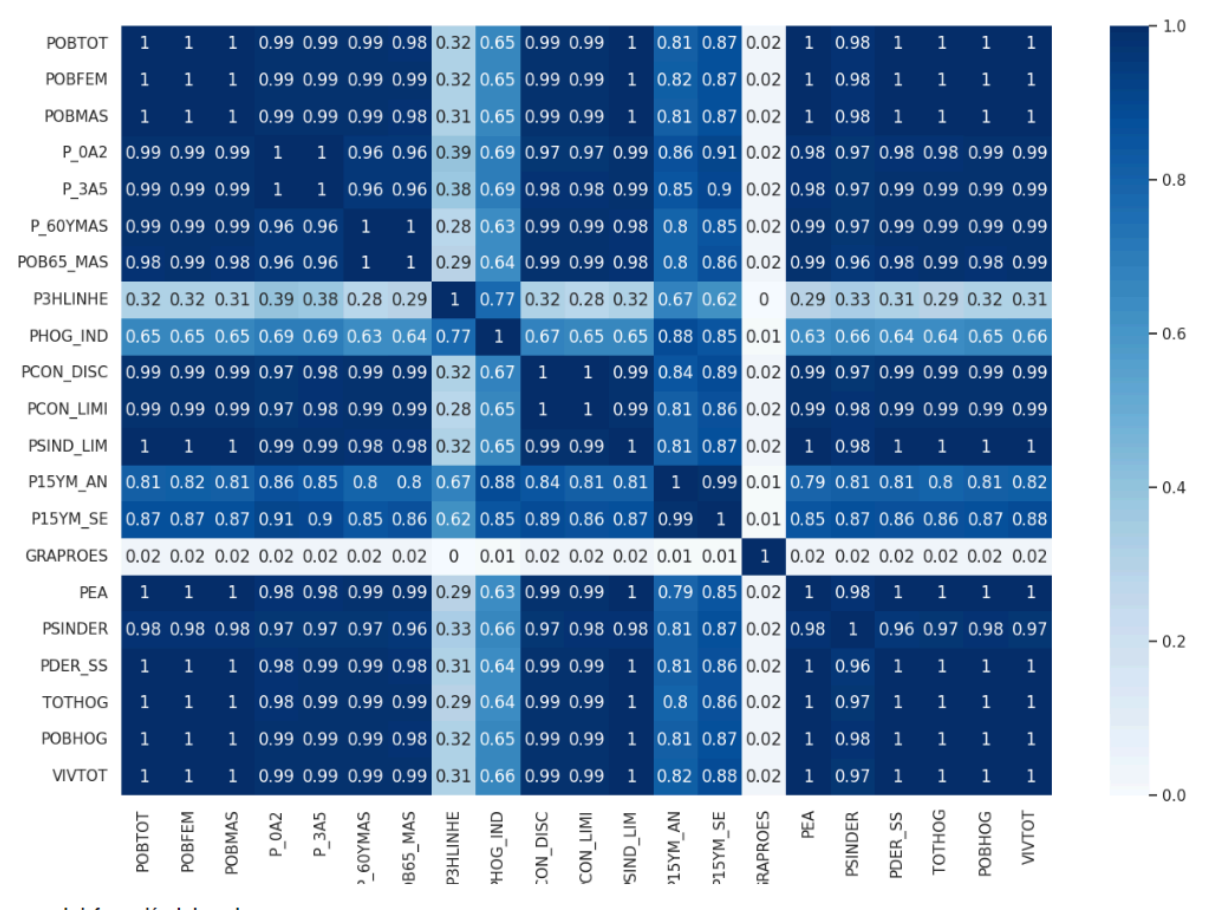


Imagen 14. Mapa de correlación . (Elaboración propia, 2024)

- ¿Cómo se distribuyen los datos en función de diferentes categorías?

En la siguiente imagen observamos a grandes rasgos la distribución de nuestras variables, esto por medio de histogramas. La función de esto es poder conocer cómo se comportan cada una de las variables.



Imagen 15. Distribución de los datos . *(Elaboración propia, 2024)*

Cabe destacar que en algunas de las gráficas se observa un único valor, como sumatoria que nos resume en realidad el total, puesto que la segmentación por AGEB se hará en etapas posteriores.

- ¿Existen patrones o agrupaciones (clusters) en los datos con características similares?

Inicialmente, realizamos un análisis PCA para poder conocer cómo se comportan las variables entre sí y, de esta manera, conocer los escenarios a los cuales nos podremos enfrentar más adelante cuando trabajemos con clústeres más avanzados. En la siguiente imagen se aprecia el resumen del análisis PCA, así como la proporción acumulada de cada variable en la explicación del fenómeno.

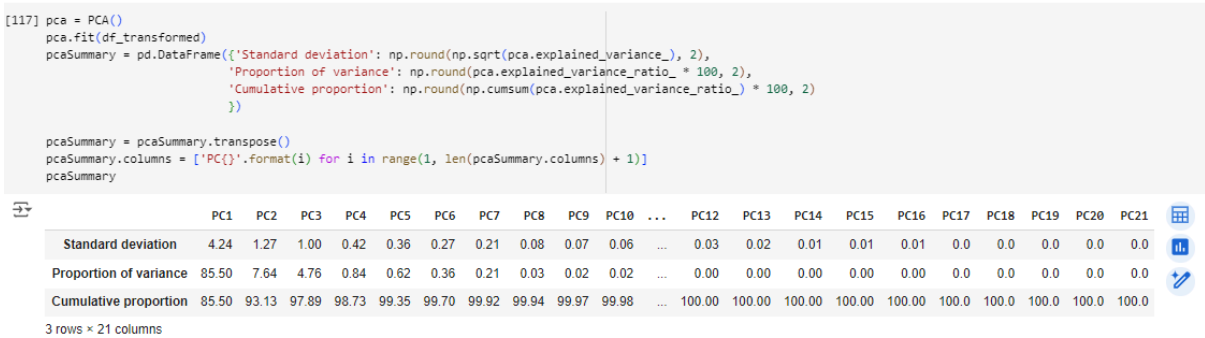


Imagen 16. Resumen análisis PCA. (Elaboración propia, 2024)

Para comprender mejor la varianza explicada, decidimos graficar los resultados. En la siguiente imagen se puede observar que a través de los primeros tres componentes se tendría la totalidad de la varianza explicada.

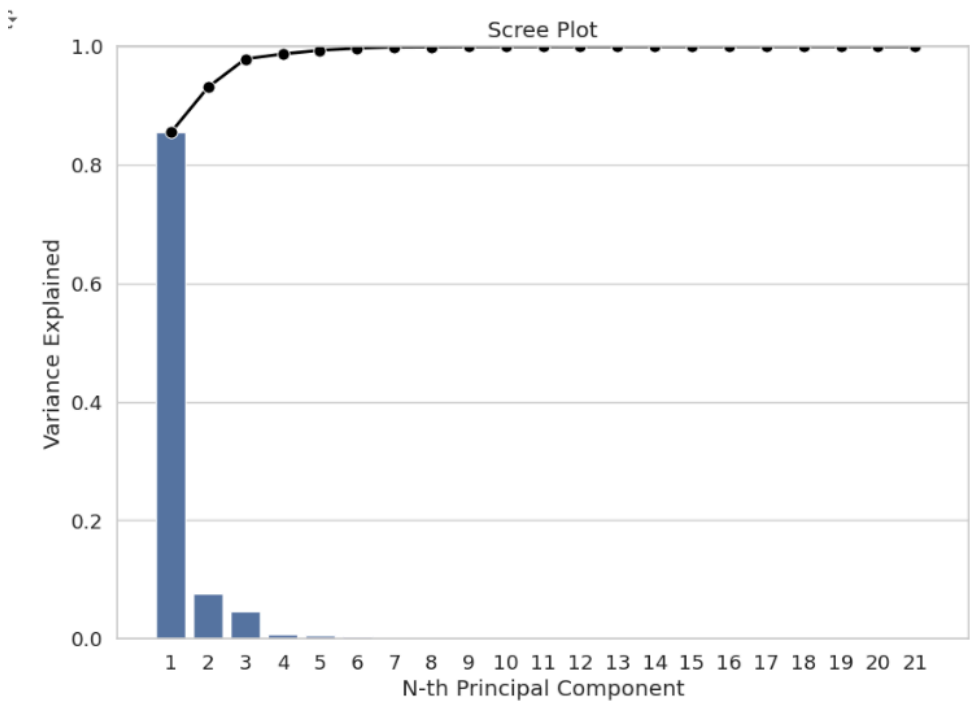


Imagen 17. Gráfica análisis PCA. (Elaboración propia, 2024)

5. Procesamiento

A continuación se describe el proceso de procesamiento inicial de los datos tabulares obtenidos en el portal del INEGI y su comparación con las imágenes raster que se nos han proporcionado:

Preparación de los Datos:

Datos INEGI: Asegurarnos de que los datos estén en un formato geográfico, como están en formato tabular, necesitamos crear una geometría (por ejemplo, centroides de los AGEB) para poder relacionarlos espacialmente. Verificar que el sistema de coordenadas de los datos INEGI coincida con el de las imágenes satelitales.

```
# Cargar el shapefile en este primer approach usamos el estado de Querétaro
agebs = gpd.read_file('/content/sample_data/conjunto_de_datos/22a.shp')

# Calcular los centroides y agregarlos al GeoDataFrame
agebs['centroid'] = agebs['geometry'].centroid

# Crear un nuevo GeoDataFrame solo con los centroides
centroides = gpd.GeoDataFrame(agebs['centroid'], geometry='centroid')

# Guardar el nuevo GeoDataFrame
centroides.to_file("centroides_agebs.shp")
```

Imagen 18. Obtención de centroides de las AGEB por estado. *(Elaboración propia, 2024)*

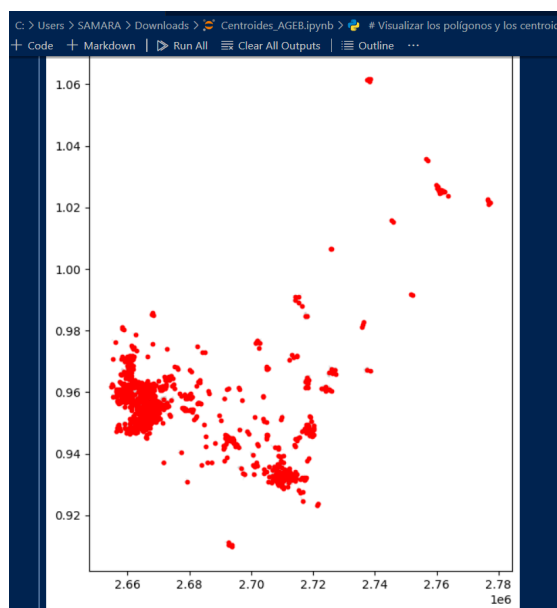


Imagen 19. Plot de los centroides de las AGEB del estado de Querétaro. *(Elaboración propia, 2024)*

Imágenes Satelitales: Registrar las imágenes satelitales para que estén georreferenciadas correctamente.

Selección de Herramientas:

Python: Es una excelente opción para este tipo de análisis, además de ser el lenguaje con el que hemos estado trabajando a lo largo de la maestría.

Librerías:

- Geopandas: Para trabajar con datos geográficos en Python.
- Rasterio: Para leer y manipular imágenes raster (como las tif).
- Shapely: Para realizar operaciones geométricas.
- Matplotlib/ Seaborn: Para crear visualizaciones.
- Raster Stats: es una herramienta muy útil para este tipo de análisis geoespacial.

```
#Importando librerías
import geopandas as gpd
import matplotlib.pyplot as plt
import zipfile
import os
```

Imagen 20. Librerías utilizadas hasta el momento . (Elaboración propia, 2024)

Proceso de Geoprocesamiento:

1.-Cargar los Datos: Utilizamos Geopandas para cargar los datos INEGI y Rasterio para cargar las imágenes satelitales.

2.-Realizar un Join Espacial: Utiliza la función overlay de Geopandas para realizar un join espacial entre los AGEb y las imágenes satelitales. Esto te permitirá asignar los valores de las imágenes (temperatura) a cada AGEb.

3.- Agrupar y Resumir: Agrupa los datos por AGEb y calcula estadísticas (por ejemplo, temperatura promedio) para cada uno.

4.- Visualización: Utilizar Matplotlib o Seaborn para crear mapas que muestren la distribución espacial de las islas de calor y su relación con las características socioeconómicas de los AGEB.

6. Conclusiones

Al terminar este análisis exploratorio de los datos EDA , corroboramos que es una etapa crucial que permite tener una comprensión más profunda de la estructura y características de nuestros datos , a través de técnicas y visualizaciones como el análisis univariante y bi/multivariante, el EDA ayuda a identificar patrones, tendencias y relaciones significativas que pueden influir en los resultados del modelado.

Además, este proceso nos resultó de utilidad para detectar y justificar operaciones de preprocesamiento, como el manejo de valores faltantes y atípicos, así como la reducción de la alta cardinalidad, lo que contribuye a mejorar la calidad de los datos y la eficacia de nuestro modelo y aplicación.

Cabe destacar que el análisis exploratorio es una de las fases fundamentales y que más tiempo lleva a cualquier analista de datos, ya sea en proyectos de IA, Machine Learning, Aprendizaje Profundo, etc. En nuestro caso particular, pudimos observar que de inicio, las bases de datos parecían limpias.

Sin embargo, a través de este proceso, al observar las medidas de distribución central y la descripción de tipos de datos, pudimos percibir que eran “objetos”, lo cual implicó ahondar en las razones por las cuales se obtenía esta lectura.

Así pues, identificamos caracteres no coincidentes con el rango de los datos y exploramos diferentes formas de tratarlo, apegándonos al caso de estudio. En este sentido, al trabajar con datos poblacionales, no fue posible realizar la imputación de datos. Ya que buscamos apegarnos a la información que realmente se pudo obtener de las bases de datos, esto al ser de carácter poblacional.

Finalmente, el análisis exploratorio nos permitirá clusterizar la información poblacional. Para luego comprender cómo estos datos tienen un impacto en las Islas de Calor y así determinar el nivel de criticidad, esto considerando los factores que nos permitirán realizar planes de acción sustentados en datos fehacientes.

7. Referencias

Bech, J. (2019). Análisis Multivariado. Universidad Autónoma de Aguascalientes. ISBN 978-607-8652-68-6.

https://editorial.uaa.mx/docs/analisis_multivariado.pdf

INEGI. (2020). Sistema de consulta de integración territorial (SCITEL). Principales resultados por AGEB y manzana urbana. INEGI.

<https://www.inegi.org.mx/app/scitel/Default?ev=10>

INEGI. (s. f.). *Publicaciones y mapas*.

<https://www.inegi.org.mx/app/biblioteca/ficha.html?upc=889463807469>

Kumar Mukhiya, S., y Ahmed, U. (2020). Hands-On Exploratory Data Analysis with Python. Packt Publishing.

<https://learning.oreilly.com/library/view/hands-on-exploratory-data/9781789537253/0957090f-fa4d-4145-95dd-6d3782e5c04d.xhtml>

Mas, J. (2019). Análisis univariante. Universitat Oberta de Catalunya. PID_00268326.

<https://openaccess.uoc.edu/bitstream/10609/148455/3/AnalisisUnivariante.pdf>

Torre, J., *et. al.* (2023). Metodología para identificar y cuantificar islas de calor en entornos urbanos con imágenes satelitales. Centro para el Futuro de las Ciudades, Tecnológico de Monterrey.

https://drive.google.com/drive/folders/1p-hPh6o_heBx-HAEKY1CsAioUi1XuRcS?hl=es

Studer, S., *et. al.* (2021). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. Preprints 2021, 1, 0.

<https://doi.org/10.48550/arXiv.2003.05155>

Visengeriyeva, L., Kammer, A., Bär, I., Kniesz, A., y Plöd, M. (2023). CRISP-ML(Q). The ML Lifecycle Process. MLOps. INNOQ.

<https://ml-ops.org/content/crisp-ml>

8. Anexos

Anexo - [Repositorio en GitHub](#)