FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 1

[7506/9558] Organización de Datos
Curso 1
Primer cuatrimestre de 2021

Alumnos	Padrón
Elvis, Claros Castro	99879
Lerer, Joaquín	105493
Bach, Alan	91440
Reinaudo, Dante	102848

o

Índice

1. Introducción

El presente informe reúne la documentación de la solución del primer trabajo práctico de la materia Organización de Datos, el cual consiste en desarrollar un análisis exploratorio de un set de datos a partir de los conocimientos adquiridos en la materia. En particular, vamos a realizar el análisis sobre un set de datos con información acerca del impacto del terremoto Gorkha, ocurrido en Nepal durante el año 2015. Especialmente el dataset se enfoca en cómo eran las condiciones de una determinada vivienda y cuál fue su grado de daño luego del accidente.

2. Desarrollo

Para facilitar la lectura, a partir de este momento también se hará referencia al set de datos como “DataFrame”.

2.1. Preprocesamiento de los datos

Con el fin de optimizar el espacio utilizado y facilitar la futura investigación, se procedió a hacer una limpieza de los datos sobre los dos DataFrame involucrados. En principio ambos set de datos se encontraban en archivos .csv (valores separados por comas), disponibles en la web de Driven Data, y cuentan con información de encuestas realizadas por Kathmandu Living Labs y el Central Bureau of Statistics .

2.1.1. Columnas sin información

En primer lugar, se buscó si en los DataFrame existían columnas que no aporten ningún tipo de información o que tuvieran un único valor repetido a lo largo de todas sus filas, con el objetivo de descartarlas, ya que sería indiferente incluirlas e irrelevantes para la investigación. Luego se prosiguió a eliminar los valores nulos que pudieran existir en las columnas. Asombrosamente, en ninguno de los DataFrame analizados se encontraron columnas sin información, ni tampoco columnas con valores nulos.

2.1.2. Análisis del tipo de datos de las columnas

Con el objetivo de administrar el espacio utilizado, se analizó el valor de los tipos de datos de todas las columnas de nuestros dos DataFrame y se realizaron modificaciones sobre estos.

Al leer una columna del tipo entero con pandas, esta se almacena por default con un int64, este tipo de dato ocupa 64 bits para representar valores enteros [1]. Si el valor de los números a guardar es pequeño, se puede optimizar bastante el espacio utilizado empleando un tipo de dato entero que use menos bits para la representación. Se procedió a buscar el valor máximo de cada una de las columnas numéricas, para saber con cuantos bits es posible representarlas.

[1] Dado que el lector de este informe puede desconocer sobre informática y sistemas de numeración, cabe aclarar que en el sistema binario, con un número entero n de bits es posible representar valores de hasta 2^n.

Columna	Valor Máximo	Tipo de dato utilizado para la representación	Tipo de dato suficiente para la representación
building_id	1052934	Int64	Int32
geo_level_1_id	30	Int64	Int8
geo_level_2_id	1427	Int64	Int16
geo_level_3_id	12567	Int64	Int16
count_flours_pre_eq	9	Int64	Int8
age	995	Int64	Int16
area_percentage	100	Int64	Int8
height_percentage	32	Int64	Int8
count_families	9	Int64	Int8

Cuadro 1: Tipo de dato entero suficiente para almacenar la columna

Además, las columnas booleanas, inicialmente se almacenaron con un tipo de dato int64, pero fueron transformadas a un tipo de dato booleano (bool), para mejorar la perfomance de nuestro código.

Por último, las siguiente columnas fueron tratadas como tipo categórico (category):

- damage_grade
- land_surface_condition
- foundation_type
- roof_type
- ground_floor_type
- other_floor_type
- position
- plan_configuration
- legal_ownership_tatus

2.1.3. Limpieza de los datos

Luego de efectuar el análisis previo sobre los dos set de datos, se puede observar cómo se redujo considerablemente el espacio utilizado. Se pasó de trabajar con dos DataFrame, donde uno de ellos tenía un peso de 71.3 Mb, a trabajar con un único DataFrame limpio con un peso de 11.4 Mb. Queda al descubierto la importancia de realizar una limpieza de los datos.

Tras las modificaciones mencionadas, se puede observar que el DataFrame cuenta con 40 columnas y 260601 filas, sumando un total de 10424040 celdas.

2.2. Análisis de las variables

Vamos a realizar un análisis exploratorio de nuestro datos con el objetivo de determinar características y variables importantes, descubrir conclusiones interesantes, y analizar la estructura de los mismos.

2.2.1. Grado de Daño (Damage Grade)

Dentro de nuestro set de datos, contamos con la información de 260601 edificios afectados por el sismo Gorkha, de 7.8 grados de magnitud en la escala Richter. Una de las variables proporcionadas, es el grado de daño que sufrió cada edificio luego de la tragedia. Basándose en esta variable de interés, se investigarán las características de cada construcción para tratar de encontrar alguna correlación entre las propiedades de los edificios y su daño sufrido a causa del terremoto.

En primer lugar, se calculó la proporción del grado de daño del total de los 26061 edificios afectados, obteniendo la siguiente tabla:

Grado de Daño	Total de edificios afectados	Porcentaje del total (%)
Low Damage	25124	9.6
Medium Damage	148259	56.9
High Damage	87218	33.5

Cuadro 2: Tipo de dato entero suficiente para almacenar la columna

Para obtener una mejor visualización de la información, se realizó el siguiente gráfico de tortas, también conocido como pie chart .

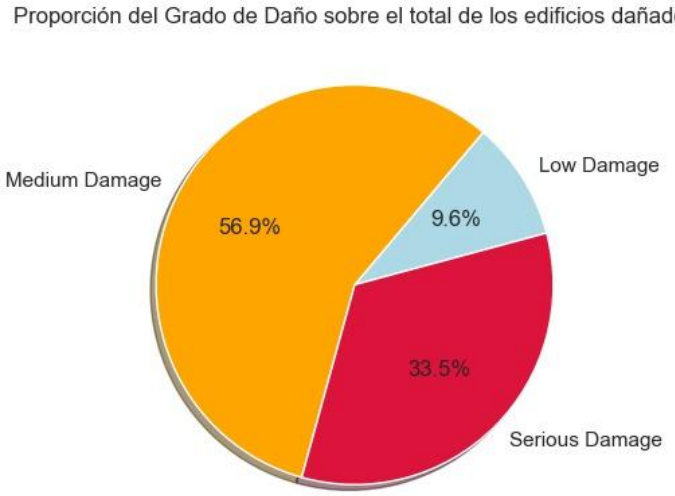

Figura 1: Proporción del Grado de Daño sobre el total de los edificios

Como puede verse en la figura anterior, la mayoría de los edificios sufrió un daño moderado, otro gran porcentaje sufrió un daño alto, mientras que tan solo una pequeña minoría sufrió un daño bajo. En las secciones posteriores vamos a investigar si existen características comunes entre los edificios que sufrieron un mismo grado de daño.

2.2.2. Hipótesis

2.2.3. Correlación de los datos

La correlación es una medida estadística que expresa hasta qué punto dos variables están relacionadas linealmente. Es una herramienta común para describir relaciones simples sin hacer afirmaciones sobre causa y efecto.

Ahora, se pasará a estudiar el grado de correlación que existe entre las distintas columnas de nuestro set de datos con respecto a la columna Damage Grade. Esto nos permitirá tener un mejor conocimiento de nuestros datos, y nos ayudará a profundizar más en nuestro análisis. Haciendo uso de las herramientas brindadas por las librerías pandas y matplotlib, se obtuvo el siguiente gráfico.

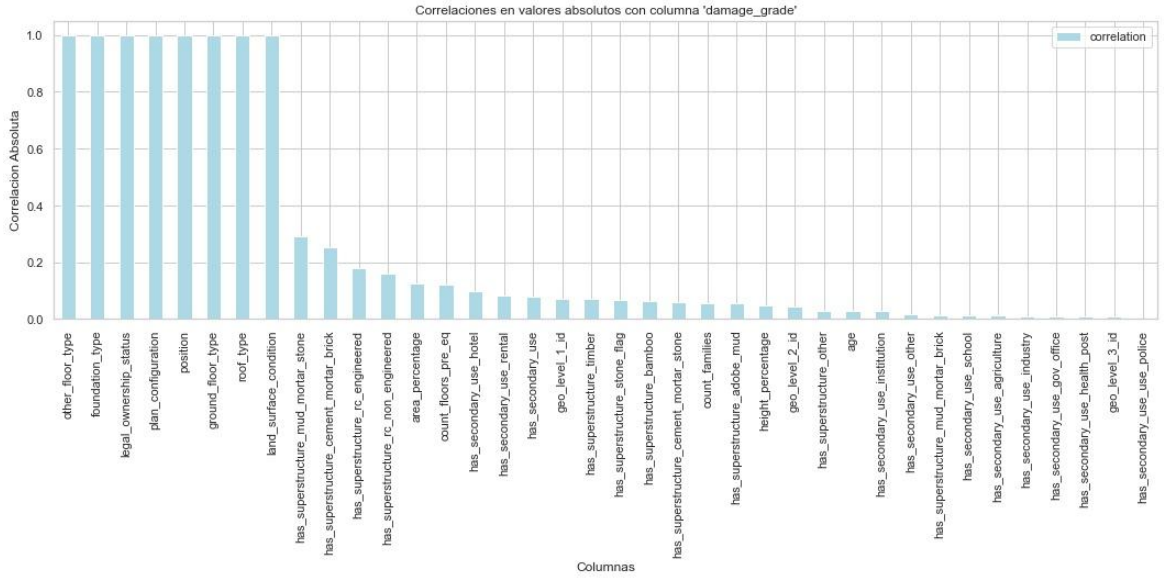

Figura 2: Correlación en valores absolutos de las columnas con Damage Grade

Al observar la gráfica superior, queda al descubierto cuales son las columnas que guardan más relación lineal con el grado de daño, esto quiere decir……

2.2.5. Daño por ubicación geográfica

Dentro de nuestro Data Frame, contamos con información sobre la localización geográfica de cada edificio afectado. Esta información está separada en tres columnas diferentes: “geo_level_1_id”, “geo_level_2_id” y “geo_level_3_id”, que van desde la ubicación más general a la más específica respectivamente. En primera instancia, se buscó graficar en un mapa la distribución de las zonas más afectadas por el sismo, pero desgraciadamente no fue posible realizar un mapeo de esta información con ubicación reales de Nepal.

Para llevar a cabo el análisis, se dividieron a los edificios afectados en 30 regiones, definidas por su código “geo_level_1_id”, y se dejó de lado las otras dos columnas. Se tomaron las 10 regiones con mas cantidad de edificios dañados, y se calculo la proporcion de grado de daño de cada una de estas. Se obtuvo la siguiente tabla.

Región	Cantidad de edificios con Low Damage	Cantidad de edificios con Medium Damage	Cantidad de edificios con High Damage	Cantidad total de edificios dañados
4	521	11164	2883	14568
6	2108	16222	6051	24381
7	1033	11273	6688	18994
8	654	8513	9913	19080
10	1211	12107	8761	22079
17	285	3913	17615	21813
20	3311	11860	2045	17216
21	322	5857	8710	14889
26	8028	12645	1942	22615
27	465	6007	6060	12532

Cuadro 3: Top 10 edificios mas dañados

Mediante la grafica de un stacked bar plot vamos a ver cuales fueron las 10 regiones con mayor cantidad de edificios dañados y que proporcion de grado de daño tuvo cada una.

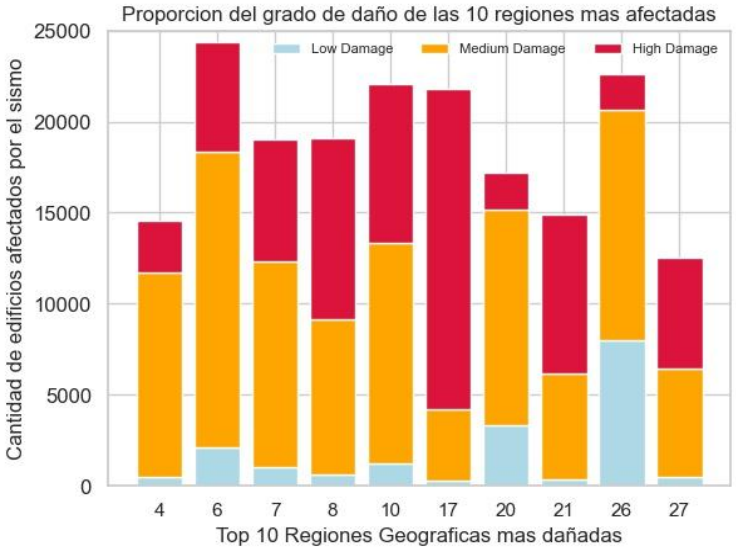

Figura 3: Top 10 regiones mas afectadas por el sismo

Del grafico podemos destacar 3 regiones:

- ➢ Region 6: es la region con mayor cantidad de edificios dañados, y a su vez la que mayo cantidad de edificios con daño moderado.
- ➢ Region 17: es la region con mayor cantidad de edificios don daño critico y la que menor cantidad con daño bajo presenta.
- ➢ Region 26: si bien es la segunda region con mayor cantidad de edificios dañados, presenta una considerable cantidad de edificios con daño bajo y ,tambien, un bajo grado de edificios con daño critico .

2.2.6. Daño por antigüedad del edificio

En nuestro set de datos, tambien contamos con informacion acerca de la edad de los edificios, por lo tanto vamos a analizar como afecta la antigüedad del edificio al grado de daño que sufrio. Naturalmente, uno tenderia a pensar que en cuanto mas antiguo es un edificio, mayor probabilidad hay de que este se derrumbe, debido al deterioro ocasionado por el paso del tiempo. Procederemos a investigar la informacion disponible, para determinar si la afirmacion anterior es valida. En primer lugar, realizaremos un grafico de densidad, tambien llamado Density Plot, de la columna age de nuestro set de datos, esta visualizacion es muy util para analizar la distribucion de una variable.

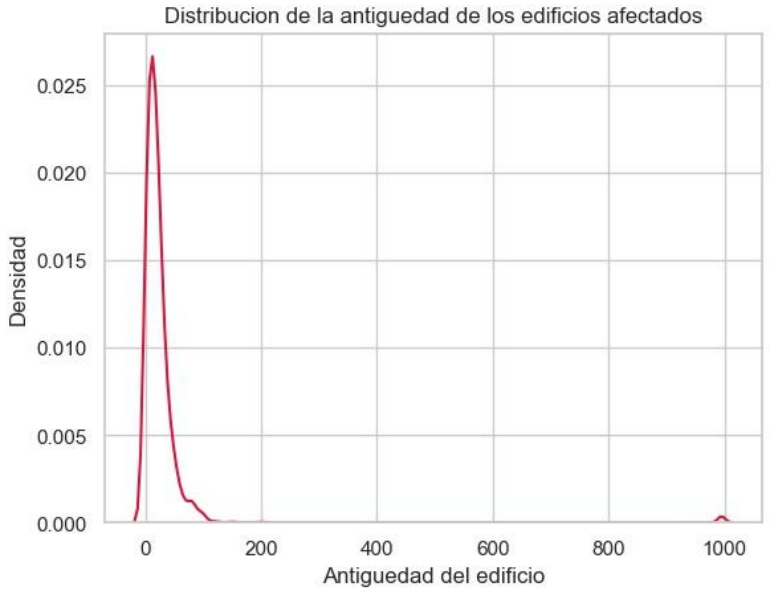

Figura 4: Distribución de la antiguedad de los edificios afectados

Hechandole un vistazo a la grafica superior, podemos destacar dos resultados:

- ➢ a. La gran mayoria de los edificios dañados tiene una edad entre los 0 y 100 años aproximadamente.
- ➢ b. La cantidad de edificios afectados con edades entre los 200 y 900 años es practicamente despreciable o nula.

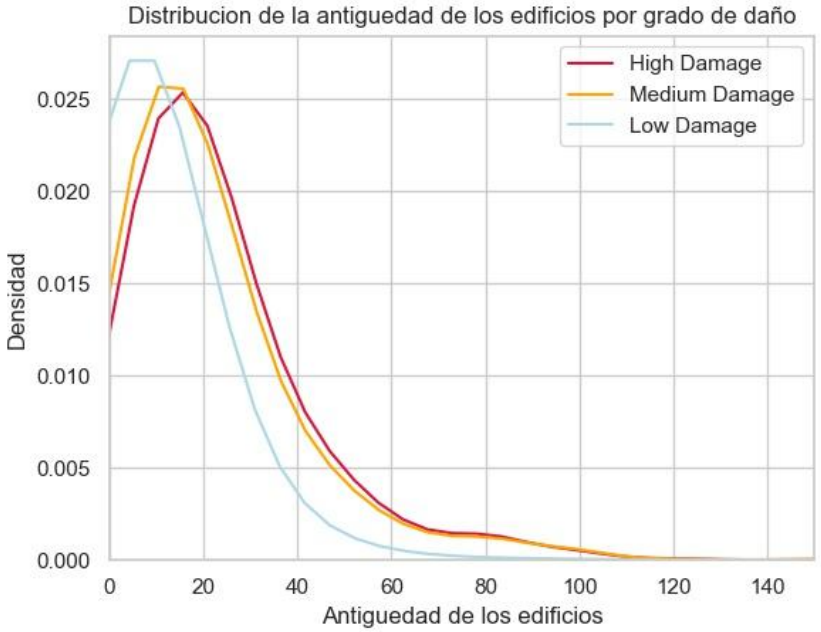

Figura 5: Distribución de la antiguedad de los edificios afectados por grado de daño

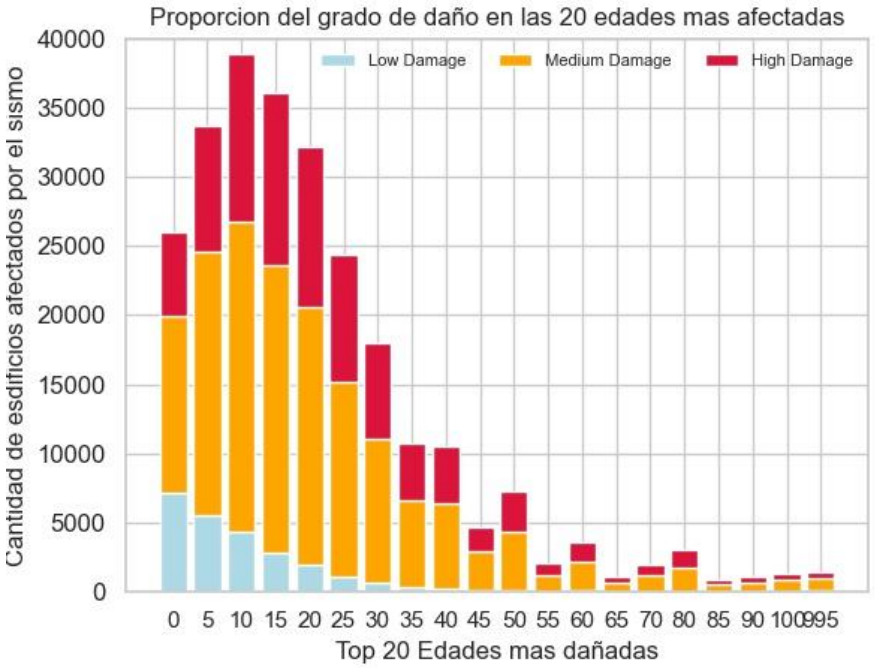

Figura 6: Las 20 edades de edificios mas afectadas

2.2.7. Daño por cantidad de pisos del edificio

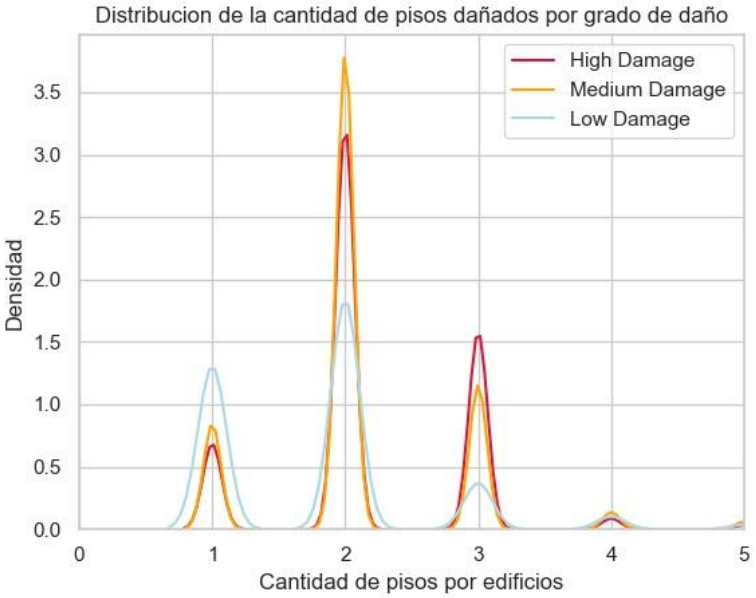

Figura 7: Distribucion de la cantidad de pisos dañados por grado de daño

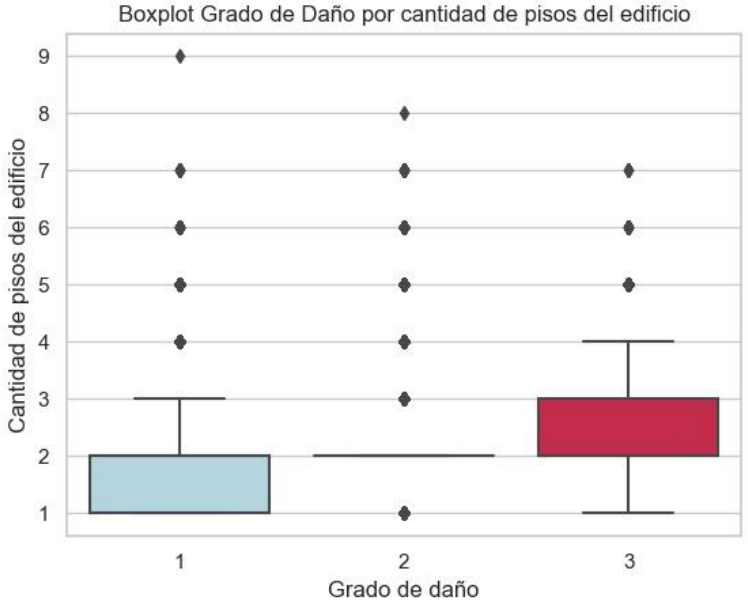

Figura 8: Box plot de la cantidad de pisos de los edificios afectados

2.2.8. Daño por altura del edificio

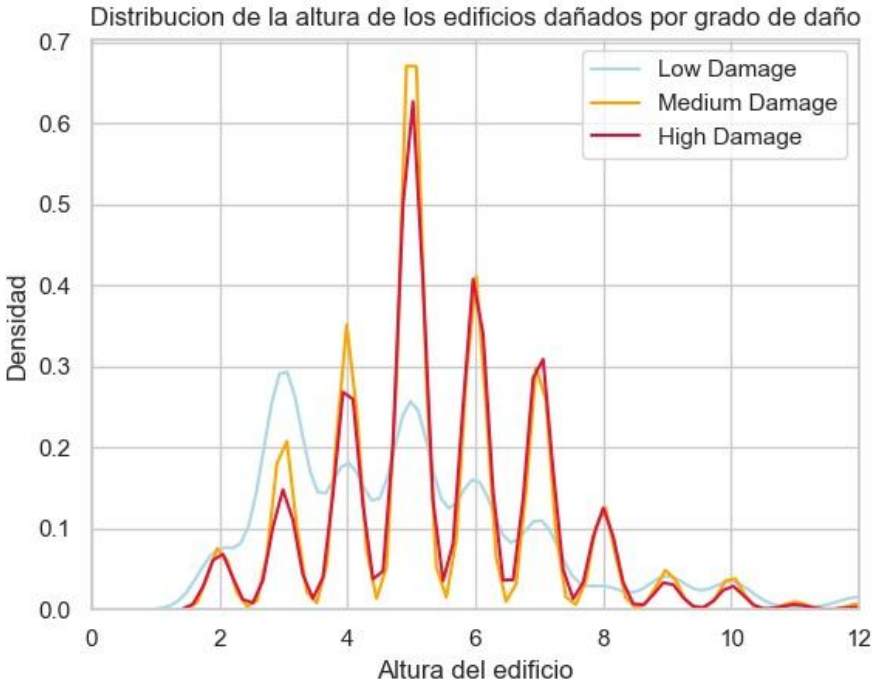

Figura 9: Distribucion de la altura de los edificios por grado de daño

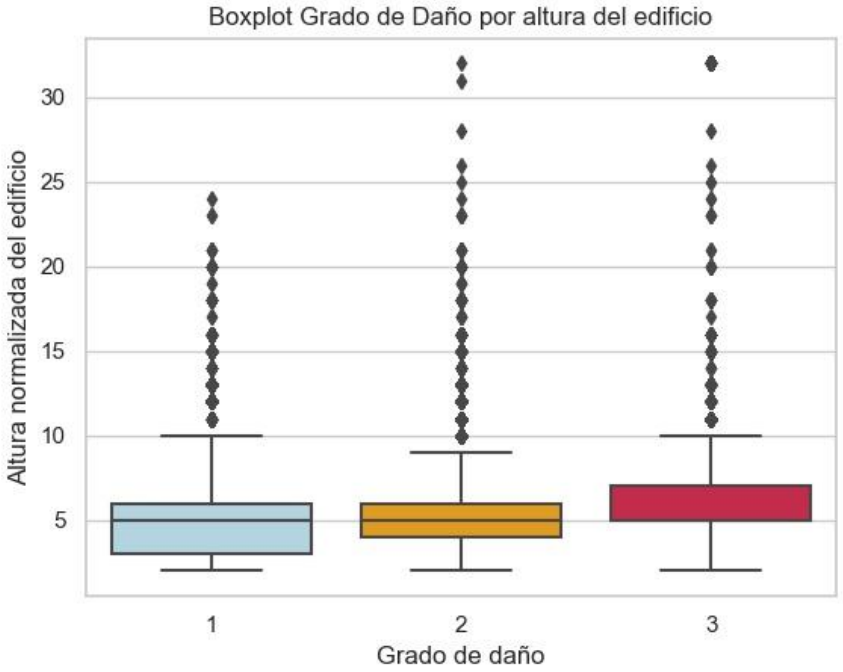

Figura 10: Box plot de la altura de los edificios afectados

2.2.9. Daño por area de pisos del edificio

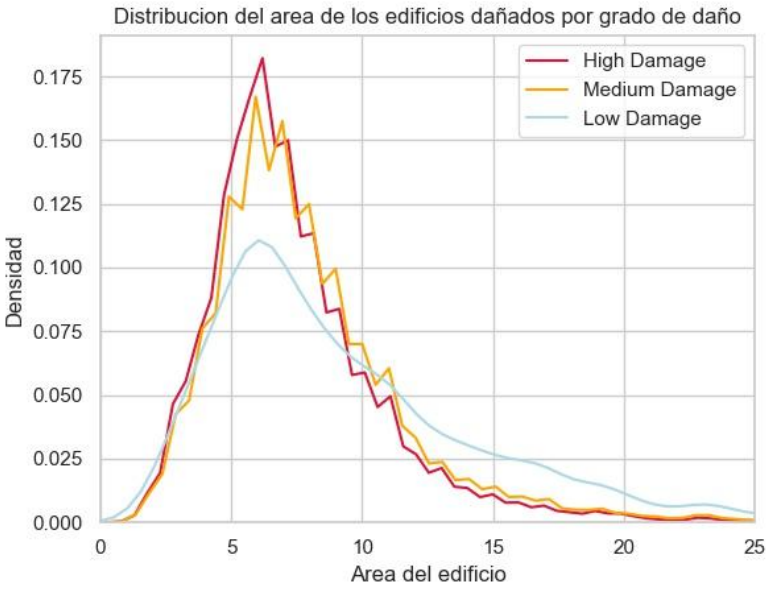

Figura 11: Distribucion del area de los edificios afectados

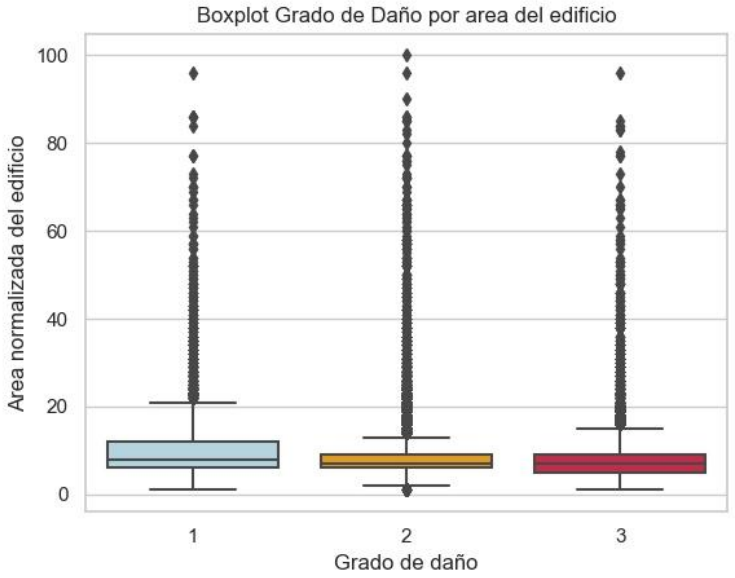

Figura 12: Box plot del área de los edificios afectados

3. Conclusiones

En conclusión, se cree que con el presente informe se ha logrado desarrollar interesantes aspectos de la base de datos con la que se trabajó. Tras analizar su estructura, relaciones, composición y propiedades, fue posible familiarizarse con la información con la que estábamos teniendo contacto, permitiéndonos adentrarnos en ella.

Aunque no se han analizado todas las columnas, se considera que se hizo un buen trabajo seleccionando aquellas que eran más relevantes a partir de su relevancia supuesta para las oportu- nidades y lo ofrecido por la empresa. Como continuación del trabajo realizado, se podría terminar de estudiar todas las columnas, pero más importante aún, se podría encontrar aún más vínculos como los hallados en múltiples secciones del informe. Además, dado el interés que se tiene sobre el éxito de las oportunidades, creemos que dado el extenso análisis realizado será posible estudiar la probabilidad de éxito de una oportunidad gracias a todo lo mencion

1. Código Fuente

El código fuente se encuentra disponible en: https://github.com/DatosOrga2021/orga_datos_1c_2021

2. Librerías Externas

2.1. Pandas

Pandas es una librería para análisis y manipulación de datos. Fue la principal herramienta utilizada para este proyecto, siendo la base de todos los análisis aquí presentados.

2.2. Numpy

Numpy es una librería para análisis numérico muy conocida. Dado su practicalidad estuvo involucrada de fondo en muchas operaciones que realizamos, así como también forma parte de otras de las librerías usadas.

2.3. Matplotlib

Matplotlib es una librería para creación de visualizaciones estáticas y animadas. Fue la principal herramienta utilizada para crear gráficos, partiendo de nuestro DataFrame que era modificado para llegar a información que le cargaríamos a los gráficos.

2.4. SeaBorn

Seaborn es una librería para visualización de datos estadísticos basada en Matplotlib. Fue utilizada para la creación de algunas visualizaciones más complejas, como los HeatMaps.

2.5. GeoPandas

GeoPandas es una librería para trabajar con datos geoespaciales. Fue utilizada para la creación de las visualizaciones de mapas. Para esto fue requerido modificar varios datos para adaptarse a las especificaciones de GeoPandas.

Referencias

[1] Benford's law. En.wikipedia.org. (2020). Retrieved from https://en.wikipedia.org/wiki/Benford

[2] GeoPandas 0.8.0. Geopandas.org. (2013). Retrieved from https://geopandas.org/.

[3] Hunter, J. (2007). Matplotlib: A 2D graphics environment. Matplotlib: Python plotting. Re- trieved from https://app.flourish.studio/visualisation/4347256/edit.

[4] Michael Waskom and the seaborn development team. (2020). mwaskom/seaborn. seaborn: statistical data visualization. Retrieved from https://seaborn.pydata.org/.

[5] NumPy. Numpy.org. (2019). Retrieved from https://numpy.org/.

[6] Regla de Sturges. Es.wikipedia.org. (2020). Retrieved from https://es.wikipedia.org/wiki/Regla_de_Sturges.

[7] The pandas development team. (2020). pandas - Python Data Analysis Library. Pan- das.pydata.org. Retrieved from https://pandas.pydata.org/.

[8] Mlxtend. http://rasbt.github.io/mlxtend/. En especial se utilizo http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/.

[9] Florecer https://app.flourish.studio/visualisation/4347256/ed it?