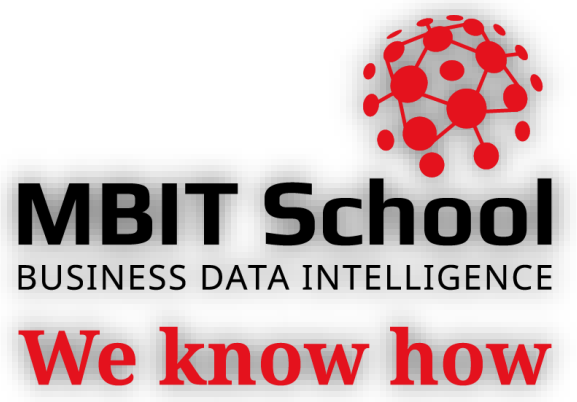


Trabajo fin de máster:

EVALUACIÓN DE DAÑOS POR TERREMOTOS

MÁSTER DATA SCIENCE PARA PROFESIONALES

Autoras:

Ana de Antonio

Pilar Campos

Silvia Saenz

Tutor:

Manuel Sánchez-Montañés

Diciembre 2020

CONTENIDO

1 INTRODUCCIÓN

Según el informe "Marco de Acción de Hyogo[1]" de la ONU, cuyo objetivo es aumentar la resiliencia de las naciones y comunidades ante los desastres:

"*En los dos últimos decenios se ha duplicado el número de desastres registrados, de aproximadamente 200 a 400 anuales. Nueve de cada diez de estos desastres están relacionados con el clima. Según las previsiones actuales con respecto al cambio climático, esta tendencia va a continuar y las situaciones de peligro relacionadas con el tiempo serán cada vez más frecuentes e imprevisibles. Los regímenes de sequía y desertificación se están intensificando y, además, muchos países son cada vez más vulnerables. Debido al incremento de la urbanización y al aumento de las concentraciones de población en asentamientos urbanos no planificados e inseguros y las zonas costeras desprotegidas, la pobreza, la prevalencia del VIH y la insuficiente atención que se presta a los cambios en los patrones de riesgo, cada vez son más las personas situadas en zonas expuestas a desastres naturales*"

Los desastres naturales son cada vez más frecuentes y destructivos. Se hace cada vez más necesario que los países sean más resilientes a los desastres naturales, pero es clave que las administraciones públicas y otras organizaciones, que se encargan de su prevención y gestión, tengan las mejores herramientas que les ayuden a la toma de decisiones de forma rápida, eficaz y temprana para minimizar los daños tanto a la población como a la economía.

En la era de la inteligencia artificial, la tecnología tiene un papel vital que desempeñar en la mejora del conocimiento del terreno por parte del personal de emergencia, para ayudar a la toma de decisiones prácticas, que salvan vidas durante la gestión de una crisis.

Existen diversas iniciativas de aplicación de la inteligencia artificial para ayudar en la respuesta a desastres naturales, algunas relacionadas con redes sociales, como AIDR [2] (Artificial Intelligence for Digital Response), que filtra y clasifica mensajes relacionados con emergencias, desastres y crisis humanitarias.

Otro ejemplo es la compañía McKinsey [3], que está utilizando algoritmos para evaluar el daño producido en edificios por terremotos usando imágenes por satélite, imágenes geoespaciales para calcular rutas, meteorológicas y otros datos.

[1] *Preparación ante los desastres para una respuesta eficaz. Conjunto de directrices e indicadores para la aplicación de la prioridad 5 del Marco de Acción de Hyogo*

[2] *http://aidr.qcri.org/*

[3] *Mckinsey*

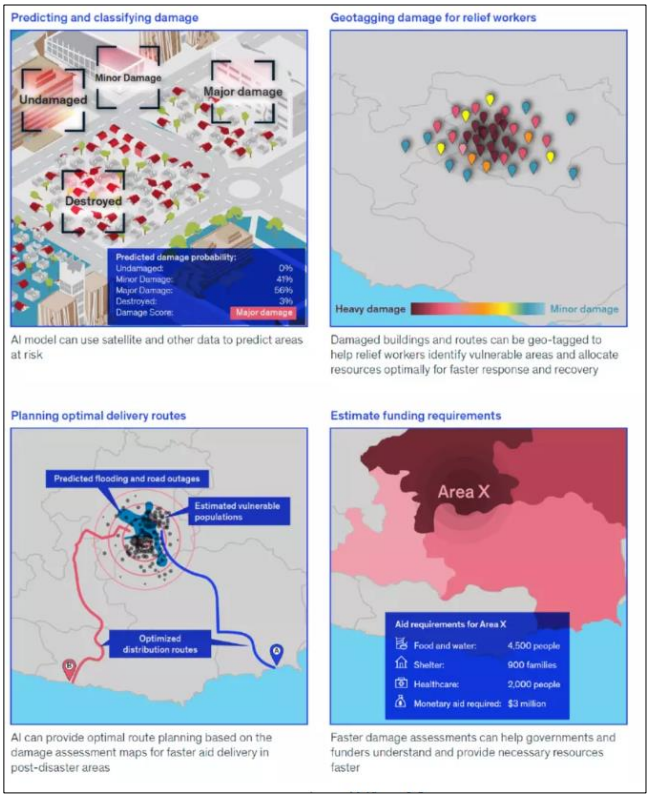


Ilustración 1 Ejemplos actuales de AI en desastres naturales

La realidad es que existen múltiples iniciativas en el sector privado para contribuir a la mejora de la gestión de crisis, aunque el impacto de estos esfuerzos aún no se materializa debido a varios desafíos[4]:

- Alcance limitado. Existen muchas iniciativas en el sector privado que involucran a algunos socios gubernamentales u ONGs y se enfocan en casos de uso concretos que no se debaten e integran en los protocolos de la amplia comunidad de gestión de crisis. Esto conduce a una fragmentación de los esfuerzos y puede dar lugar a que se proporcionen conocimientos y herramientas derivados de la inteligencia artificial a organizaciones que no pueden mantenerlos o incorporarlos de manera efectiva a sus procesos de decisión.
- Existe gran cantidad de información que podría utilizarse en la gestión de desastres (satélites, telecom, redes sociales…), el problema radica en que no siempre se puede acceder a ella cuando es necesaria. Es más, los conjuntos de datos rara vez se combinan de manera que se pueda obtener información adicional. Además, los datos sobre la visión en el terreno rara vez se capturan o analizan.
- Por último, en situaciones de desastre en las cuales vidas humanas pueden estar en peligro, es importante considerar las limitaciones de la inteligencia artificial.

[4] *Crisis management: Using Artificial Intelligence to help save lives*

2 OBJETIVO Y METODOLOGÍA

La motivación de este proyecto es aplicar ciencia de datos a un problema social. De esta forma se llega a la web Driven Data[5] "*Data science competitions to build a better world*". Indagando en las competiciones activas, se escoge "*Richter Predictor: Modeling Earthquake Damage",* cuyo objetivo es predecir el daño producido por terremotos en edificios, utilizando datos del terremoto de 7,8 Mw del 25 de abril de 2015 en Gorkha (Nepal).

El planteamiento de la competición sirvió de base para definir la idea, ya que se le debía dar forma al caso de uso. A partir de ahí, la pregunta a la hora de enfocar el proyecto consistió en "*¿Cómo se podría ayudar a la gestión de desastres humanitarios tras un terremoto usando ciencia de datos?"*

Actualmente, las tareas de clasificación de daño de los edificios se realizan por profesionales que se desplazan a la zona, tras una evaluación visual de los edificios, son etiquetados de acuerdo con su daño: edificios que no presentan peligro de colapso son marcados con una etiqueta verde y aquellos cuya reocupación es no segura, con una etiqueta roja. Existen casos en los que la ocupación no es tan obvia y se marcan con una etiqueta amarilla. Dependiendo de la intensidad del terremoto, estos trabajos pueden ser costosos en términos de recursos y de tiempo. Por ejemplo, tras el terremoto de Haití en 2010[6], se realizó una evaluación manual de 90.000 edificios en el área de Puerto Príncipe categorizando el daño de cada uno en una escala del 1 al 5, tardando semanas. En el caso del terremoto de 1994 Northridge[7], tardaron más de 2 meses en evaluar y etiquetar 100.000 edificios.

Por tanto, se pensaron posibles soluciones considerando las diferentes técnicas y fuente de datos:

- *Machine Learning:* Aplicar algoritmia supervisada para clasificar los daños de edificios. Las actividades de clasificación de este tipo de daños necesitan un gran esfuerzo en cuanto a tiempo y recursos, ya que requieren efectivos que se desplacen y evalúen el daño in situ, edificio a edificio. Con esta aproximación se usaría ciencia de datos para optimizar este proceso.

- *Deep Learning:* Usando tratamiento de imágenes, tratar de predecir el daño producidos a un edificio, también con el objetivo de optimizar el proceso de evaluación.

- *NPL Datos de Twitter.* En los días siguientes al terremoto de Nepal se produjeron más de 5M de tweets relacionados con el desastre. Parecía una aplicación útil poder evaluar el tipo de daño y dónde se había producido, aunque la imposibilidad de conseguir esos tweets (ya que la API gratuita de Twitter impone muchas limitaciones), hizo desestimar esta opción.

[5] *https://www.drivendata.org/*

[6] *https://ai.googleblog.com/2020/06/machine-learning-based-damage.html*

[7] *www.researchgate.net/publication/336014398_Classifying_Earthquake_Damage_to_Buildings_Using_Machine_Learning*

Por lo tanto, el proyecto en ciencia de datos ha consistido en la evaluación de daños en edificios causados por terremotos usando dos aproximaciones tecnológicas distintas, por un lado, Machine Learning tradicional, por otro lado, Deep Learning para el tratamiento de imágenes.

Se ha intentado mostrar cómo las técnicas actuales pueden, no sólo mejorar la actividad de etiquetado de edificios, sino que son capaces de dar información más precisa y rápida para el despliegue de efectivos, evacuación de los edificios o alerta a las personas que viven en ellos.

Durante el desarrollo, se ha seguido el ciclo de vida de un proyecto de ciencia de datos adaptado a esta casuística concreta. A su vez el presente documento se ha estructurado siguiendo esta metodología.

Tras la definición de la idea, se ha procedido a la caracterización del proyecto: problemática, tecnología a aplicar, audiencia a quien se dirige y en qué circunstancias. Como ya se ha comentado, el problema ha sido abordado utilizando las dos aproximaciones tecnológicas elegidas, por lo tanto, el desarrollo y las tareas se han realizado en paralelo.

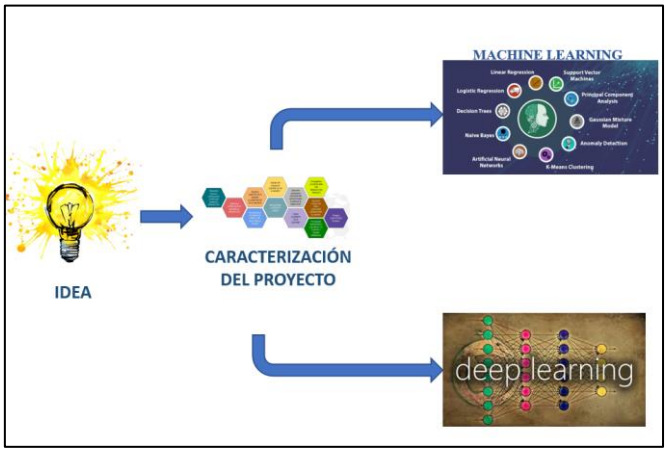


Ilustración 2 Idea y caracterización del proyecto

Las fases del proyecto en la aproximación con Machine Learning Supervisado han consistido en: una primera fase de investigación y recolección de datos, preparación del conjunto final y análisis exploratorio, división en conjunto de construcción y explotación para la modelización y prueba de concepto, aprendizaje con diferentes modelos, proceso reiterativo en base a los resultados obtenidos volviendo a modificar los datos y el planteamiento para mejorar el output.

Para finalizar, las predicciones se volcaron un dashboard con el fin de visualizar de una manera rápida y clara los edificios/zonas con una probabilidad alta de colapso tras el terremoto.

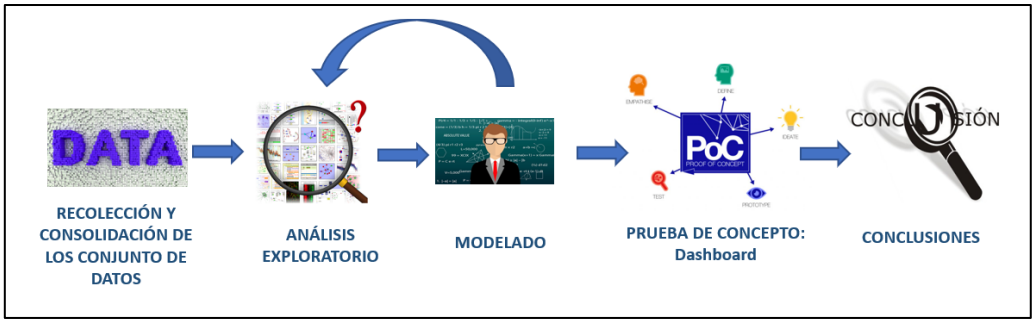


Ilustración 3 Metodología Machine learning

En cuanto a la aproximación de tratamiento de imágenes con Deep Learning, al igual que en caso anterior, un primer paso consistió en la recolección y preparación de las imágenes. A continuación, se utilizaron redes previamente entrenadas en las cuales se modificaron las últimas capas para adaptarlas a la problemática que se estaba considerando. Finalmente se realizó un proceso iterativo en el entrenamiento para mejorar los resultados.

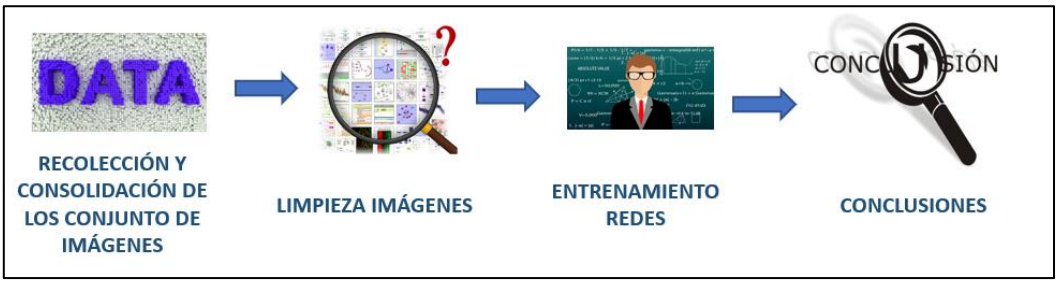


Ilustración 4 Diagrama proceso Deep Learning

La ejecución de esta prueba de concepto se realizado con datos reales del terremoto de Gorkha en 2015. Para el desarrollo, tanto del proceso de analítica como de la herramienta de visualización, se ha utilizado Python y su ecosistema, con librerías como Numpy, Pandas, Sciktlearn, Matplotlib, Dash… así como Jupyter notebooks.

3 MACHINE LEARNING: APRENDIZAJE SUPERVISADO

3.1 CONJUNTO DE DATOS

3.1.1 BÚSQUEDA Y RECOLECCIÓN

Fuente de datos 1. Encuestas a hogares afectados

La competición, en la cual se basa el proyecto, proponía un conjunto de datos, previamente filtrado con determinadas variables, que proporcionaban información de localización territorial y tipo de construcción de los edificios. De cara a enriquecer ese conjunto de datos, se decidió tomar como punto de partida el conjunto original de datos, disponible en "2015 Nepal Earthquake Open Data Portal"[8]

Tras el terremoto de 7,8 Mw en Gorkha (Nepal) el 25 de abril de 2015 y a través de *"Kathmandu Living Labs" y "Central Bureau of Statistics",* se llevó a cabo una encuesta masiva en hogares utilizando tecnología móvil, para evaluar los daños a los edificios en los distritos afectados por el terremoto. Aunque el objetivo de esta encuesta era identificar los beneficiarios a recibir ayudas del gobierno para la reconstrucción de viviendas, se recopiló información sobre el impacto del terremoto, condiciones de los hogares, así como estadísticas socioeconómicas y demográficas, convirtiéndose en uno de los conjuntos de datos más grandes recopilados hasta la fecha con las consecuencias de un desastre natural.

Los datos disponibles en el Portal [9] se dividían en tres subcategorías:

- Individuals
  - Demographics
  - Social Security
- Buildings
  - Structural Data*
  - Damage Assessment Data
  - Building Ownership and use*
- Household
  - Demographics*
  - Household Resources
  - Earthquake Impact

Analizando en detalle todo el contenido y en base al objetivo del proyecto, se usaron aquellos datos estrictamente relacionados con el tipo de edificio, así como información socioeconómica y demográfica de los hogares afectados, es decir, "*Structural data", "Building ownership and use*" y "*Household Demographics*".

[8] *https://eq2015.npc.gov.np/#/*

[9] *https://eq2015.npc.gov.np/#/download*

Fuente de Datos 2. Geolocalización municipios

Como soporte al personal de emergencia, se consideró importante disponer de una herramienta de visualización en la que poder identificar geográficamente los potenciales daños de una forma clara y actuar así de una manera rápida.

Una primera aproximación consistía en posicionar cada edificio en un mapa geográfico de Nepal a través de su geolocalización. Para obtener las coordenadas de cada uno de los edificios/hogares que participaron en la encuesta, se contactó con diferentes Instituciones en Nepal. Lamentablemente, la respuesta obtenida en todos los casos fue la misma: se trataba de información de carácter personal protegida por leyes de tratamiento de datos.

Sin embargo, se pudo obtener un mapa con las fronteras administrativas[10] en distintos formatos, siendo *geojson* el utilizado para una primera representación.

Como se intuye en el mapa inferior, la organización territorial en Nepal es un tanto compleja, ya que está dividido en 7 provincias, 111 distritos y 720 municipios.

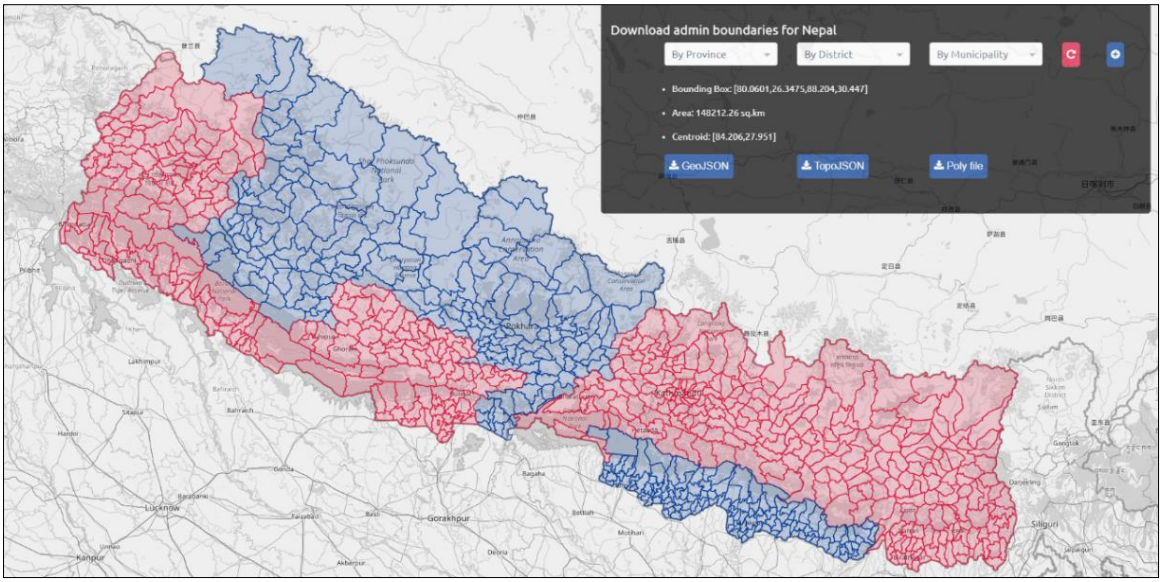


Ilustración 5 Mapa de fronteras administrativas de Nepal

De cara a consolidar toda información, era necesario cruzar cada distrito, municipio y coordenadas del geojson con los tres conjuntos de datos extraídos del Portal ("*Structural data*", *"Building ownership and use"* y *"Demographics").*

Estos tres conjuntos de datos contenían todas las variables con la que construir el modelo predictivo y, por otro lado, el geojson contenía la información necesaria para construir la herramienta de visualización. Para lograr este cruce era necesario utilizar un elemento común, en este caso el nombre del "municipio".

[10] *https://drklrd.github.io/adminboundaries-np/?fbclid=IwAR0XPuHCLS0JoDSj_KbPV7DLY8WqfzF1UgGDKC4ckO45YlE70-jNd8QUfvI*

Nepal es un país multilingüe con al menos 120 tipos de dialectos diferentes y por tanto un mismo municipio-distrito puede tomar varios nombres. Tras un proceso bastante manual y tedioso, identificando relaciones entre cada uno de los 720 municipios, se logró una estandarización en los nombres.

Se realizó una primera visualización de los distritos, seleccionando una variable input cualquiera, en este caso "*edad del cabeza de familia*", utilizando *geopandas.* Se observó que el conjunto de datos inicialmente era válido y se correspondía a los distritos más afectados por el terremoto. Además, se observa que Katmandú (epicentro del terremoto y área central color morado) no disponía de datos ya que no hubo recogida en esa zona.

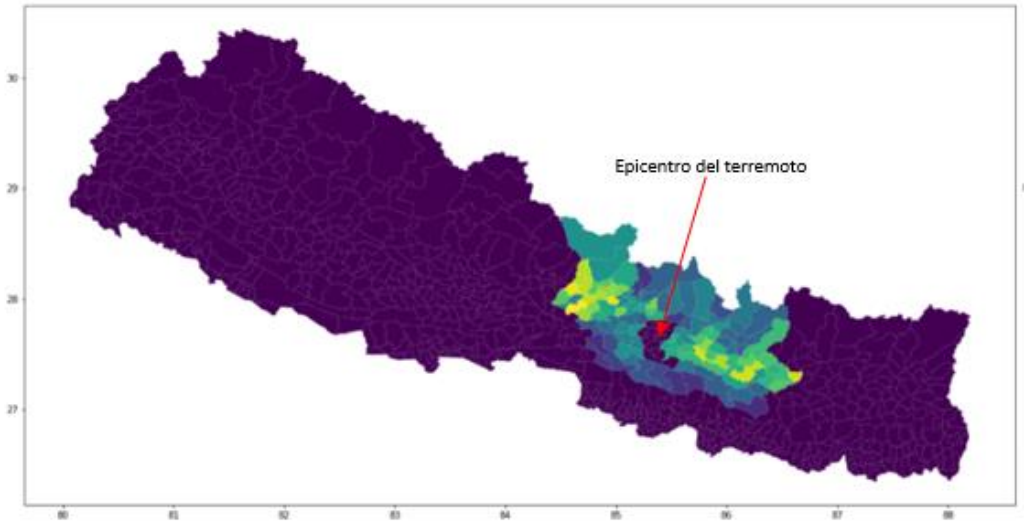


Ilustración 6 Ejemplo visualización de datos en Nepal

3.1.2 CONSTRUCCIÓN DEL CONJUNTO DE DATOS FINAL

En el dibujo inferior se muestra el esquema de trabajo seguido para llegar a un solo conjunto de datos con el que trabajar de cara al modelo predictivo (*"Conjunto de datos final")*

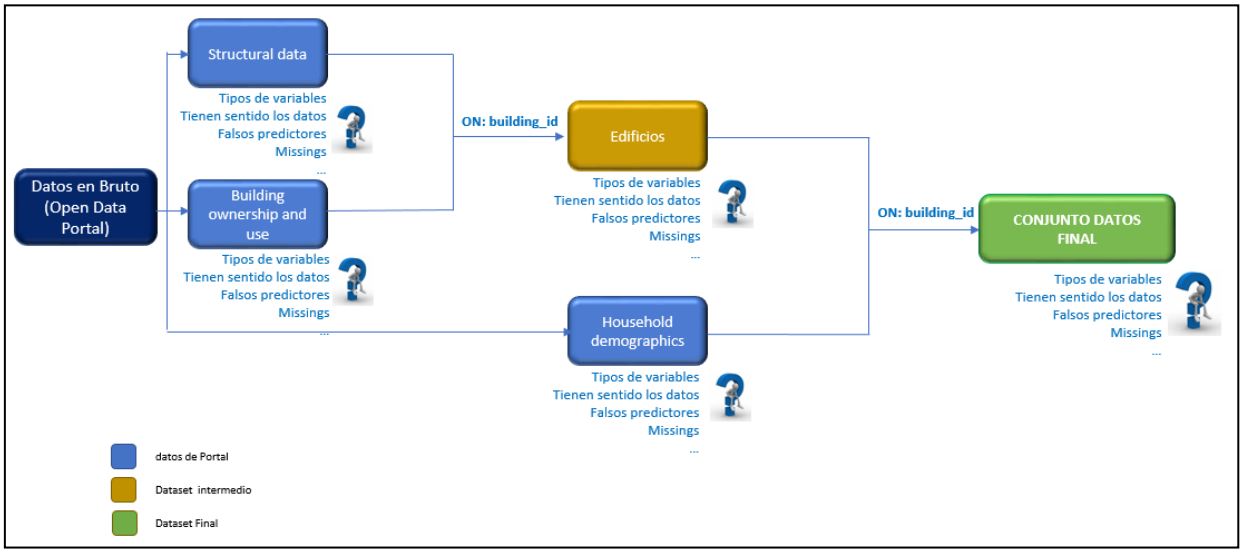


Ilustración 7 Diagrama de consolidación de los datos

Paso 1. Creación de "Edificios"

CSV 1: "Building Structure.csv", información sobre el edificio en cuanto a estructura, construcción y tipos de materiales. Contiene 31 variables y 762.106 registros. Estudiando el sentido de cada variable en particular, se identificaron 3 variables predictoras que debían ser eliminadas (*"count floors post eq", "condition post eq" y "technical solution proposed").*

CSV 2: "Building ownership and use.csv", información sobre la propiedad del edificio, número de familias o su uso, caracterizado por 17 variables y 762.106 registros.

Existía una variable única y común, *"building id"* con la que el cruce de ambos documentos fue una tarea sencilla, por tanto, se obtuvo un nuevo conjunto de datos llamado *"Edificios",* donde el número de valores ausentes era despreciable por lo que se eliminaron:

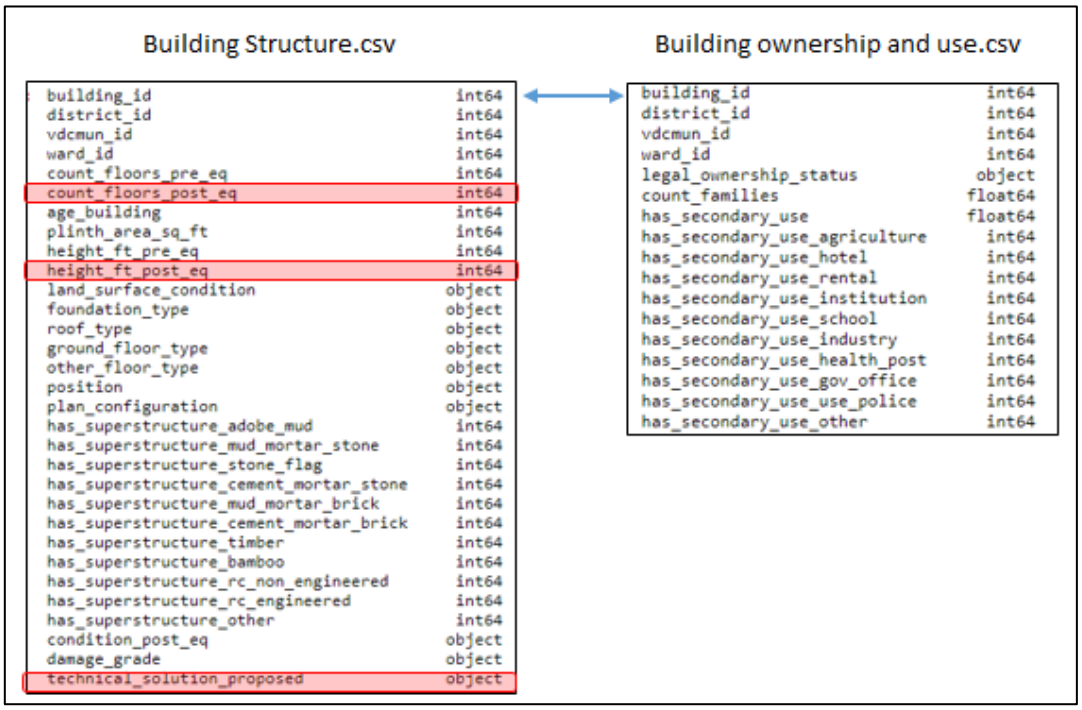


Ilustración 8 Tipos de variables

Your selected dataframe has 39 columns.
There are 5 columns that have missing values.

	Missing Values	% of Total Values
damage_grade	12	0.0
has_secondary_use	10	0.0
count_families	2	0.0
position	1	0.0
plan_configuration	1	0.0

Ilustración 9 Missing values

De cara a entender más en profundidad los datos obtenidos tras este cruce (39 variables y 762.093 líneas), los registros se agruparon en base a la variable "*count families*".

Se observaron 71.576 edificios con 0 familias, concluyendo que se correspondían a edificios públicos y/o privados no destinados a vivienda (iglesias, hospitales, colegios etc..). Estos edificios se llamarán de aquí en adelante *"edificios sin familias".*

count_families	building_id	district_id	vdcmun_id	ward_id	count_floors_pre_eq	age_building	plinth_area_sq_ft	height_ft_pre_eq	land_surface_condition	f
0.0	71576	71576	71576	71576	71576	71576	71576	71576	71576	
1.0	643409	643409	643409	643409	643409	643409	643409	643409	643409	
2.0	39751	39751	39751	39751	39751	39751	39751	39751	39751	
3.0	5685	5685	5685	5685	5685	5685	5685	5685	5685	
4.0	1215	1215	1215	1215	1215	1215	1215	1215	1215	
5.0	302	302	302	302	302	302	302	302	302	
6.0	104	104	104	104	104	104	104	104	104	
7.0	27	27	27	27	27	27	27	27	27	
8.0	15	15	15	15	15	15	15	15	15	
9.0	8	8	8	8	8	8	8	8	8	
11.0	1	1	1	1	1	1	1	1	1	

Ilustración 10 Variables agrupadas "count_families"

Paso 2. Cruce de "Edificios" con "Household Demographics"

CSV 3 "Household_demographics.csv**",** con información específica sobre los hogares afectados tales como ingresos, tamaño del hogar, edad del cabeza de familia, nivel educativo o etnia. Contiene 11 variables y 747.365 registros.

Hay que destacar que en este caso no existe una variable llamada "*building id"* sino *"Household id*", por lo que fue necesario buscar una solución de cara a lograr ese cruce con "*Edificios":*

household_id	int64
district_id	int64
vdcmun_id	int64
ward_id	int64
gender_household_head	object
age_household_head	float64
caste_household	object
education_level_household_head	object
income_level_household	object
size_household	float64
is_bank_account_present_in_household	float64

Ilustración 11 Variable Household_id

Tras un estudio de las cuatro variables id, se llegó a la siguiente conclusión:

- "*Building id*": código de 12 dígitos, por ejemplo, 120101000011

- "*Household id*": código de 14 dígitos, donde los dos últimos dígitos hacen referencia al número de familias viviendo en el mismo edificio, siendo los 12 primeros dígitos el edificio en el que residen, es decir, el building id que se busca.

    - 12010100001101 - 1 familia en el edificio 120101000011
    - 12010100001102 - 2 familias en el edificio 120101000011

Al igual que en el caso anterior, antes de consolidar la información, se trataron los valores ausentes. En este caso, el número también se consideró despreciable frente a su contenido total, por lo que esos registros fueron eliminados:

Your selected dataframe has 11 columns.
There are 7 columns that have missing values.

	Missing Values	% of Total Values
caste_household	228	0.0
education_level_household_head	228	0.0
income_level_household	228	0.0
is_bank_account_present_in_household	228	0.0
gender_household_head	2	0.0
age_household_head	2	0.0
size_household	2	0.0

Ilustración 12 Missing values

Paso 3: Creación del conjunto de datos final

Tras el cruce de "*Edificios*" y "*household demographics.csv*", se obtiene el conjunto de datos final para la construcción del modelo predictivo.

Recordemos que los valores ausentes ya fueron tratados anteriormente, sin embargo, en el dataset final se identificaron 8 variables demográficas con 71.647 ausentes y 35 variables estructurales con 13 valores ausentes.

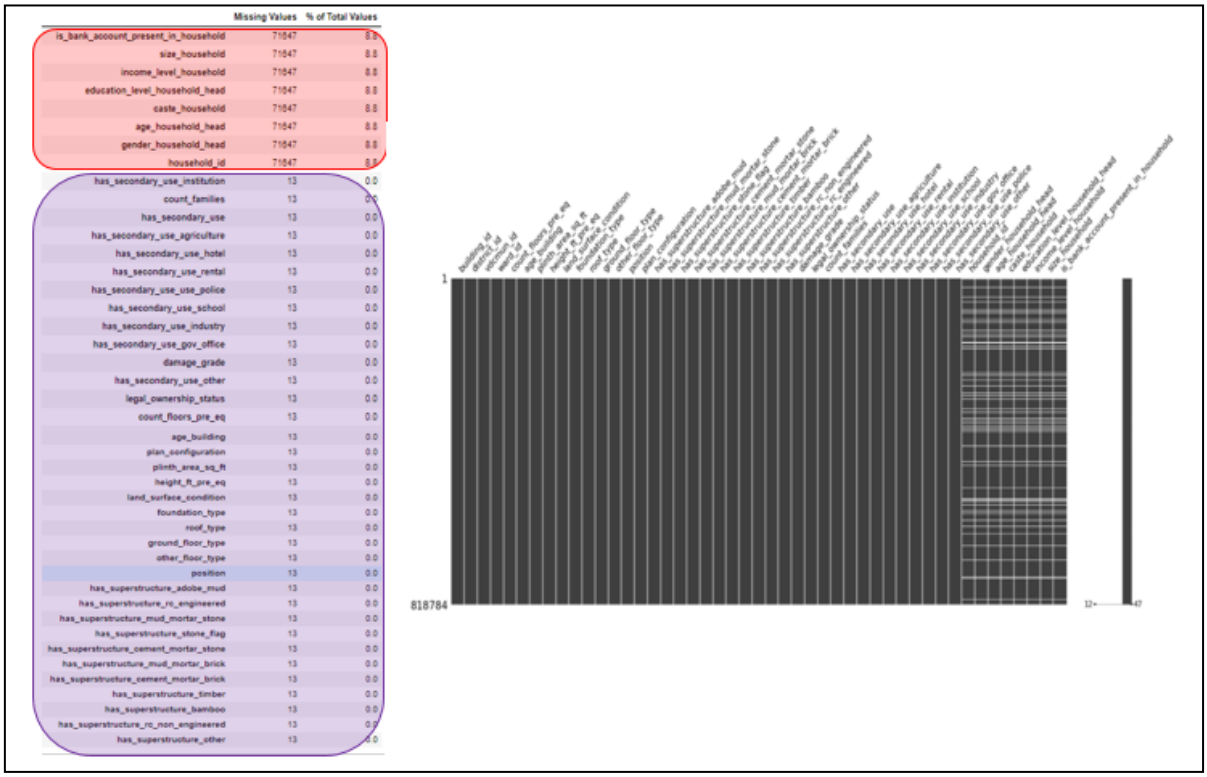


Ilustración 13 Missing values

En este punto, se debía profundizar y entender la razón de esos valores ausentes distribuidos además de la misma manera.

Era obvio que los 71.647 missing provenía de edificios no destinados a la vivienda y que por tanto carecen de información demográfica, pero ¿por qué no se corresponden en número con los "*edificios sin familias*" (71.576) identificados anteriormente? ¿De dónde procedía esa diferencia de 71 registros?

Tras estudiarlo en profundidad, se concluyó de la existencia de un error en la recogida de los datos y de 71 edificios con familias no recogidos en el conjunto de datos socioeconómicos y demográficos.

Tras la eliminación de esos valores, se obtienen, como era de esperar, 71.576 valores ausentes que se corresponden a los "*edificios sin familias*".

Sin embargo, hay que apuntar que estos 71.576 registros con valores ausentes, en realidad no los son y no hay que tratarlos como tal. Por tanto, en las variables numéricas se sustituye el valor ausente por el valor 0 y en las categóricas, se trata como un nivel más llamado "*no familias*".

	Missing Values	Tipo variable	Tratamiento
is_bank_account_present_in_household	71.647	int64	0
size_household	71.647	int64	0
income_level_household	71.647	object	no_familias
education_level_household_head	71.647	object	no_familias
caste_household	71.647	object	no_familias
age_household_head	71.647	int64	0
gender_household_head	71.647	object	no_familias

Ilustración 14 Tratamiento de missing values

Otro punto importante que considerar radica en la visualización de estas variables, recordando que surgirán valores atípicos en valor 0 que se corresponderán con este tipo de casuística.

3.2 ANÁLISIS EXPLORATORIO INICIAL

3.2.1 DESCRIPCIÓN DEL CONJUNTO DE DATOS

Como punto de partida se cuenta con un conjunto de datos formado por 818.700 registros y 48 variables de diferente naturaleza:

- 5 variables id referente a distrito, municipio, barrio, edificio y hogar *(apuntar que el id no es un número aleatorio, sino que está constituido por una consecución de números lógica dando una cierta información)*
- 7 variables numéricas
- 12 variables categóricas
- 22 variables booleanas

Numéricas (Ids)	
building_id	int64
district_id	int64
vdcmun_id	int64
ward_id	int64
household_id	float64

Numéricas	
count_floors_pre_eq	int64
age_building	int64
plinth_area_sq_ft	int64
height_ft_pre_eq	int64
count_families	int64
age_household_head	int64
size_household	int64

Categóricas	
land_surface_condition	object
foundation_type	object
roof_type	object
ground_floor_type	object
other_floor_type	object
position	object
plan_configuration	object
legal_ownership_status	object
gender_household_head	object
caste_household	object
education_level_household_head	object
income_level_household	object
damage_grade	object

Booleanas	
is_bank_account_present_in_household	int64
has_superstructure_adobe_mud	int64
has_superstructure_mud_mortar_stone	int64
has_superstructure_stone_flag	int64
has_superstructure_cement_mortar_stone	int64
has_superstructure_mud_mortar_brick	int64
has_superstructure_cement_mortar_brick	int64
has_superstructure_timber	int64
has_superstructure_bamboo	int64
has_superstructure_rc_non_engineered	int64
has_superstructure_rc_engineered	int64
has_superstructure_other	int64
has_secondary_use	int64
has_secondary_use_agriculture	int64
has_secondary_use_hotel	int64
has_secondary_use_rental	int64
has_secondary_use_institution	int64
has_secondary_use_school	int64
has_secondary_use_industry	int64
has_secondary_use_gov_office	int64
has_secondary_use_use_police	int64
has_secondary_use_other	int64

Ilustración 15 Tipos de variables

A continuación, un breve descriptivo de cada una de ellas, recordando que la información disponible hace referencia a la propia construcción y naturaleza de los edificios, así como a datos socioeconómicos de las familias que conviven en los edificios.

La variable a predecir (resaltada en la tabla) será el nivel de daño de cada edificio, constando de cinco niveles: “Grade 1”, “Grade 2”, “Grade 3”, “Grade 4” y “Grade 5”, siendo 5 el más dañado.

Variable	Tipo información	Descripción
building_id	Structural data	A unique ID that identifies a unique building from the survey
district_id	Structural data	District where the building is located
vdcmun_id	Structural data	Municipality where the building is located
ward_id	Structural data	Ward Number in which the building is located
household_id	Structural data	A unique ID that identifies a unique household by building from the survey
count_floors_pre_eq	Structural data	Number of floors that the building had before the earthquake
age_building	Structural data	Age of the building (in years)
plinth_area_sq_ft	Structural data	Plinth area of the building (in square feet)
height_ft_pre_eq	Structural data	Height of the building before the earthquake (in feet)
land_surface_condition	Structural data	Surface condition of the land in which the building is built
foundation_type	Structural data	Type of foundation used in the building
roof_type	Structural data	Type of roof used in the building
ground_floor_type	Structural data	Ground floor type
other_floor_type	Structural data	Type of construction used in other floors (except ground floor and roof)
position	Structural data	Position of the building
plan_configuration	Structural data	Building plan configuration
has_superstructure_adobe_mud	Structural data	Flag variable that indicates if the superstructure of the building is made of Adobe/Mud
has_superstructure_mud_mortar_stone	Structural data	Flag variable that indicates if the superstructure of the building is made of Mud Mortar - Stone
has_superstructure_stone_flag	Structural data	Flag variable that indicates if the superstructure of the building is made of Stone
has_superstructure_cement_mortar_stone	Structural data	Flag variable that indicates if the superstructure of the building is made of Stone
has_superstructure_mud_mortar_brick	Structural data	Flag variable that indicates if the superstructure of the building is made of Cement Mortar - Stone
has_superstructure_cement_mortar_brick	Structural data	Flag variable that indicates if the superstructure of the building is made of Mud Mortar - Brick
has_superstructure_timber	Structural data	Flag variable that indicates if the superstructure of the building is made of Timber
has_superstructure_bamboo	Structural data	Flag variable that indicates if the superstructure of the building is made of Bamboo
has_superstructure_rc_non_engineered	Structural data	Flag variable that indicates if the superstructure of the building is made of RC (Non Engineered)
has_superstructure_rc_engineered	Structural data	Flag variable that indicates if the superstructure of the building is made of RC (Engineered)
has_superstructure_other	Structural data	Flag variable that indicates if the superstructure of the building is made of any other material
damage_grade (TARGET)	Structural data	Damage grade assigned to the building by the surveyor after assessment
technical_solution_proposed	Structural data	Technical solution proposed by the surveyor after assessment
legal_ownership_status	Owership	Legal ownership status of the land in which the building was built
count_families	Owership	Number of families in the building
age_household_head	Owership	Age of the head member of each household
size_household	Owership	Number of people belonging to the same family
has_secondary_use	Owership	Flag variable that indicates if the building is used for any secondary purpose
has_secondary_use_agriculture	Owership	Flag variable that indicates if the building is secondarily used for agricultural purpose
has_secondary_use_hotel	Owership	Flag variable that indicates if the building is secondarily used as hotel
has_secondary_use_rental	Owership	Flag variable that indicates if the building is secondarily used for rental purpose
has_secondary_use_institution	Owership	Flag variable that indicates if the building is secondarily used for institutional purpose
has_secondary_use_school	Owership	Flag variable that indicates if the building is secondarily used as school
has_secondary_use_industry	Owership	Flag variable that indicates if the building is secondarily used for industrial purpose
has_secondary_use_health_post	Owership	Flag variable that indicates if the building is secondarily used as health post
has_secondary_use_gov_office	Owership	Flag variable that indicates if the building is secondarily used as government office
has_secondary_use_use_police	Owership	Flag variable that indicates if the building is secondarily used as police station
has_secondary_use_other	Owership	Flag variable that indicates if the building is secondarily used for other purpose

Ilustración 16 Descriptivo de las variables

Analizando más en profundidad las **variables categóricas**, se observa que algunas de ellas tiene gran número de niveles : *"caste household"* con 98, *"education level household head"* con 20, o *"plan configuration"* con 10.

	land_surface_condition	foundation_type	roof_type	ground_floor_type	other_floor_type	position	plan_configuration	damage_grade
count	818700	818700	818700	818700	818700	818700	818700	818700
unique	3	5	3	5	4	4	10	5
top	Flat	Mud mortar-Stone/Brick	Bamboo/Timber-Light roof	Mud	TImber/Bamboo-Mud	Not attached	Rectangular	Grade 5
freq	680062	674181	534966	660723	523438	647377	785131	297707

	legal_ownership_status	gender_household_head	caste_household	education_level_household_head	income_level_household
count	818700	818700	818700	818700	818700
unique	4	3	97	20	6
top	Private	Male	Tamang	Illiterate	Rs. 10 thousand
freq	785995	507739	209066	263154	409100

Ilustración 17 Categorías de variables

En consecuencia, ante una transformación a matriz numérica, el tamaño del conjunto de datos se hará inmanejable, por tanto, una selección de variables es de vital importancia.

En cuanto a las variables **numéricas**, se observan outliers y/o errores de medición que deberán tratarse:

- *"age household head"*, con un máximo 122 años.
- "*size household"* con un máximo de 40 miembros en 1 familia

Sin embargo, resaltar "*age building"* con un máximo de 999 años de antigüedad y 0 años de mínimo. Podría deberse a un error o simplemente edificios antiguos (templos, iglesias y edificios recién construidos), por lo que no se aplicará ninguna técnica de tratamiento de outliers en este caso concreto.

	count_floors_pre_eq	age_building	plinth_area_sq_ft	height_ft_pre_eq	count_families	age_household_head	size_household
count	818700.000000	818700.000000	818700.000000	818700.000000	818700.000000	818700.000000	818700.000000
mean	2.107502	24.598400	416.747600	16.193325	1.083211	42.553381	4.491334
std	0.662849	65.619261	239.295952	5.569587	0.620012	19.513617	2.741697
min	1.000000	0.000000	70.000000	6.000000	0.000000	0.000000	0.000000
25%	2.000000	9.000000	280.000000	13.000000	1.000000	31.000000	3.000000
50%	2.000000	16.000000	360.000000	16.000000	1.000000	44.000000	4.000000
75%	2.000000	27.000000	480.000000	18.000000	1.000000	56.000000	6.000000
max	9.000000	999.000000	5000.000000	99.000000	11.000000	122.000000	40.000000

Ilustración 18 Outliers

3.2.2 DESBALANCEO DE CLASES

Visualizando en un primer momento la variable a predecir, *"damage grade",* se observa un desbalanceo sobre todo en los extremos.

La mayor frecuencia recae en G5 y G4, llegando a un 60% de los edificios y reduciéndose en G2 y G1 (poco dañados). Estos resultados arrojan sentido ya que los datos están recogidos en los distritos cercanos al epicentro del terremoto.

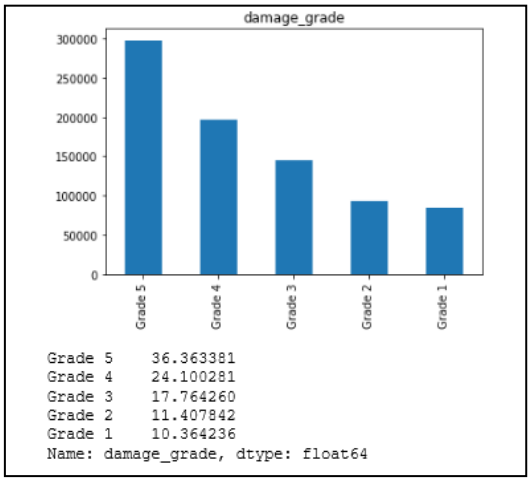


Ilustración 19 Variable damage grade inicial

La aproximación inicial consistía en afrontar el problema con las 5 clases originales. Sin embargo, se observó una capacidad predictiva llamativamente pequeña y muy cercano a un modelo dummy.

En este problema es importante recordar el sistema de recogida de datos. La encuesta fue enviada a cada una de las familias afectadas con el objetivo de identificar a los beneficiarios de ayudas, un año y medio después del terremoto, por tanto, es un ejercicio bastante declarativo y personal sin intervención de personal técnico para valorar cada una de las variables y sobre todo la asignación de clases a cada edificio afectado.

Ello podría implicar un riesgo en el manejo de unas etiquetas un tanto difusas. Es decir, las diferencias entre G5 y G1 deberían ser claras y contar con poco error muestral, sin embargo ¿es así para el resto de los grados?

Asumiendo esa potencial desviación, las clases fueron agrupadas, pasando de los 5 originales a 3 ("G1+G2", "G3+G4", "G5"), consiguiendo además un mejor balanceo. El resultado de los modelos mejoró, pero no como se esperaba, siendo solo ligeramente superior a un modelo dummy.

En base a esto, se testó una tercera situación, agrupando en tan solo dos clases:

- G4+G5 (*Severe/High damage*, para edificios destruidos o muy dañados)
- G3+G2+G1 (*Medium/Low*, para edificios medio o poco dañados).

De esta manera se consigue, por un lado, una mayor claridad o separación de las clases, pero sobre todo, en un caso real, identificar rápidamente los edificios más afectados y que por tanto, necesitarían una intervención más inmediata, optimizando así los recursos humanos.

Por tanto, el nuevo punto de partida sería un problema con 2 clases repartidas en 60%-40%

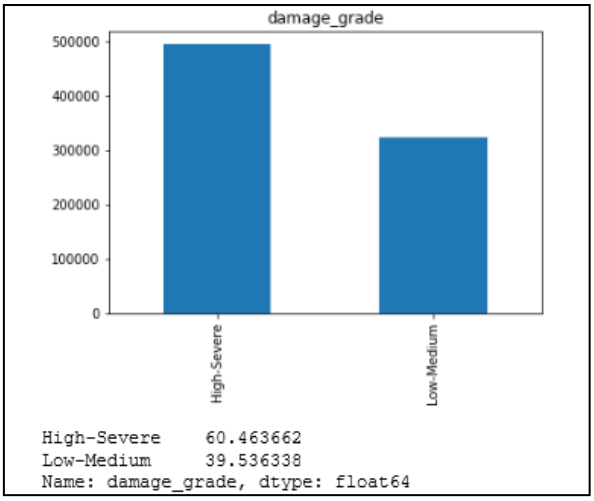


Ilustración 20 Variable damage grade agrupada

3.2.3 PRIMERAS RELACIONES CON EL TARGET Y TRANSFORMACIÓN DE VARIABLES

El objetivo de esta sección es tomar una primera sensibilización de las variables input y cómo se comportan con la clase. Pero además se aprovechará este punto para:

1. Valorar la posibilidad hacer transformaciones logarítmicas en variables numéricas.
2. Tratamiento de outliers donde sea necesario.
3. "Limpiar" las variables categóricas, agrupando niveles con poca frecuencia o en aquellas que tenga un sentido lógico. De esta manera se reducirá el número de variables en la matriz numérica.

VARIABLES NUMÉRICAS

Recordemos que contamos con 7 variables numéricas (excluyendo los IDs). A continuación, se detalla cada una de ellas.

- **Age Building**

La distribución original está muy centrada en torno a 0-100, por lo que expandirla a través del logaritmo es una buena práctica, obteniendo un histograma totalmente abierto y con simetría gaussiana.

En cuanto a los outliers, como ya se indicó, no se aplicará ninguna técnica dado que pueden corresponderse a edificios muy antiguos (templos, iglesias o edificios recién construidos)

En el boxplot contra la variable objetivo, se observa la caja ligeramente desplazada hacia arriba en "*high-severe*", por lo que parece que los edificios más antiguos sufren más daños altos.

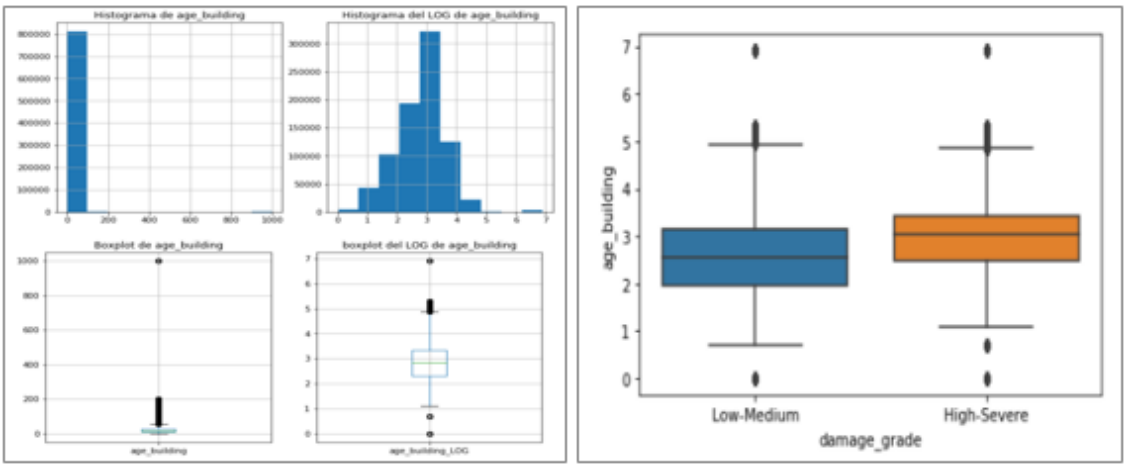


Illustration 21 Age building vs Damage Grade

❖ Count Floors pre eq

Esta variable no se transformará ya que como se observa es su histograma, tiene una buena distribución. La media de los edificios tiene dos plantas, sin embargo, los outliers en este caso tampoco se consideran atípicos, estando su máximo en nueve. En cuanto a la relación con el nivel de daño, se observa más impacto en edificios con más plantas.

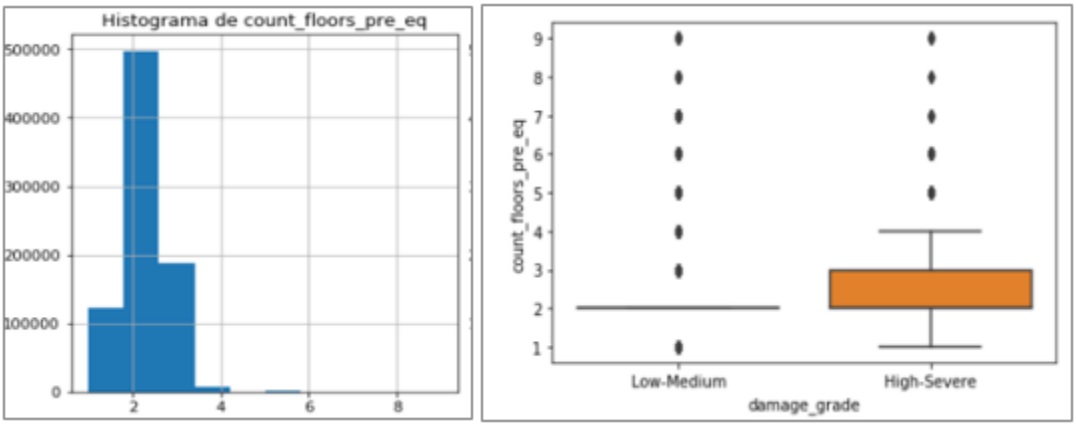


Illustration 22 Count Floors pre eq vs Damage Grade

❖ Height ft pre eq

Esta variable también es perfecta para una transformación logarítmica, ya que se obtiene una distribución mucho más abierta.

En su boxplot se observan muchos puntos fuera de los límites, sin embargo, no se considera una buena práctica tratarlos por la propia información que arroja la variable, siendo la altura mínima 6 y la máxima 99.

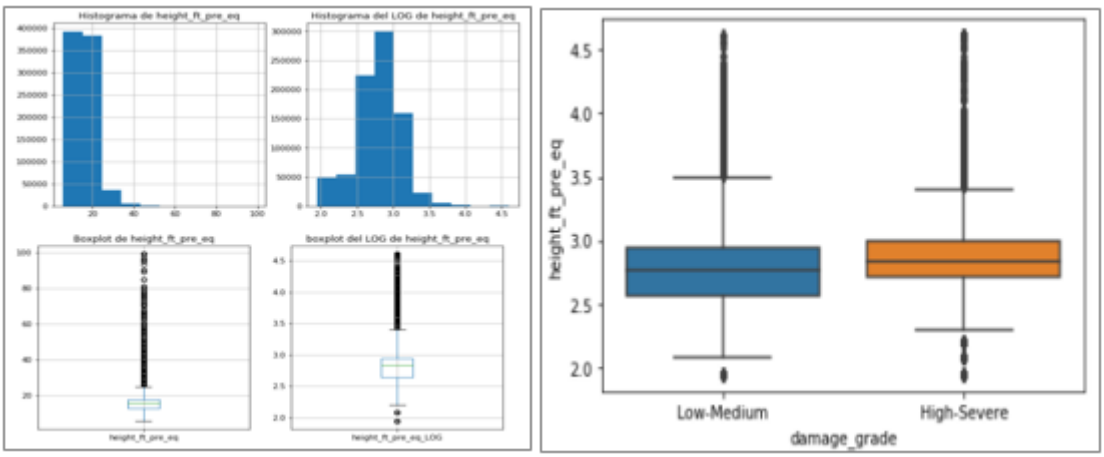


Illustration 23 Height ft pre eq vs Damage Grade

❖ Plinth área sq ft

Esta variable también tratará de forma logarítmica. En cuanto a su relación con el nivel de daño, quizás los edificios con números más bajos son más impactados por daños altos, sin embargo, no es una relación muy marcada.

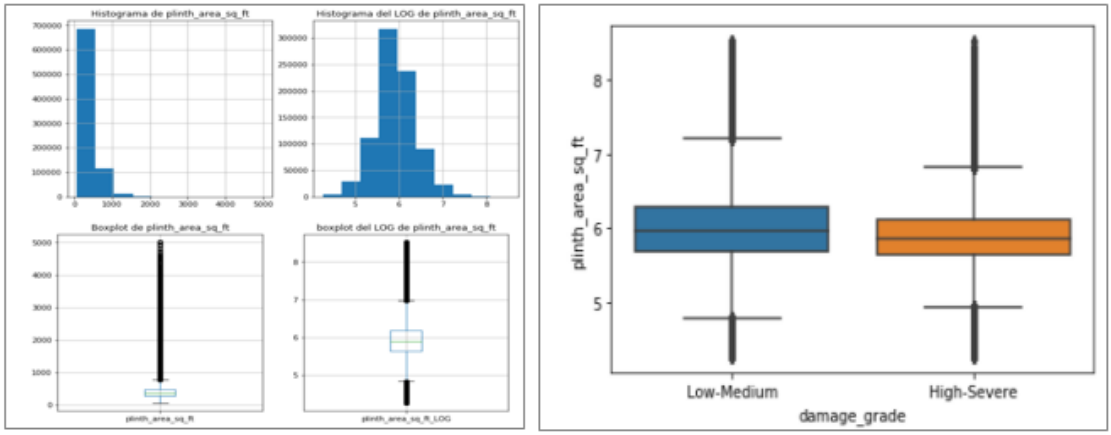


Illustration 24 Plinth area sq ft vs Damage Grade

❖ Age household head

Como ya se observó anteriormente, en esta variable existen outliers que se deberían tratar ya que, como primera conclusión, no tendría sentido encontrar edificios cuyo cabeza de familia sea 0 o mayor a 100. Sin embargo, recordemos que el valor 0 de esta variable hace referencia (o en gran parte) a esos *"edificios sin familias"*. Por tanto, se tratarán los outliers, pero tan solo los que sobrepasen el límite superior, poniendo el corte en 100 años.

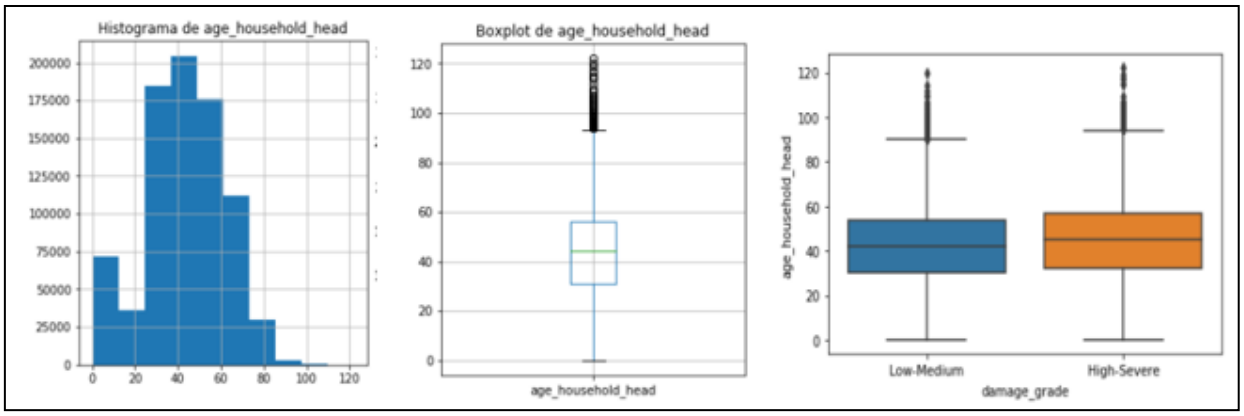


Illustration 25 Age household head vs Damage Grade

❖ Size household

Esta variable se comporta como la anterior, es decir, los atípicos en punto 0 no son outliers reales, sino relativos a los edificios sin familias. En cuanto a su distribución, conviene hacer una transformación logarítmica

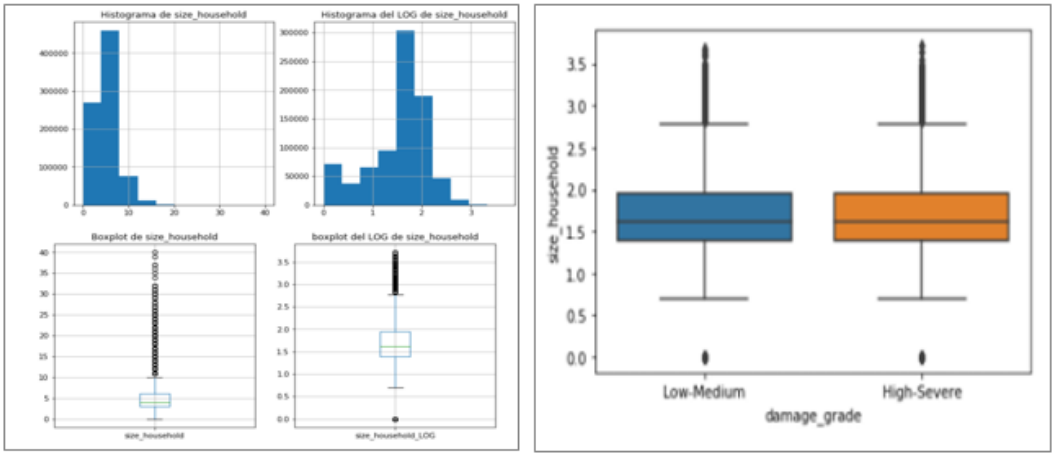


Illustration 26 Size household vs Damage Grade

A modo resumen, la siguiente tabla muestra este primer análisis en las variables numéricas y en código semafórico las potenciales relaciones observadas en los gráficos:

Variables numéricas	¿Tratamiento Outliers?	¿Transf. logaritmica?	¿Relacion con el nivel de daño?
Age building	No	Si	Daños altos en edificios más antiguos
Count Floors	No	No	Daños altos en edificios con más plantas
Height	No	Si	Daños altos en edificios con mas altura
Plinth area	No	Si	Ligeramente en daños altos
Age household head	Si	No	No se aprecia una relación clara
Count families	No	No	No se aprecia una relación clara
Size household	No	Si	No se aprecia una relación clara

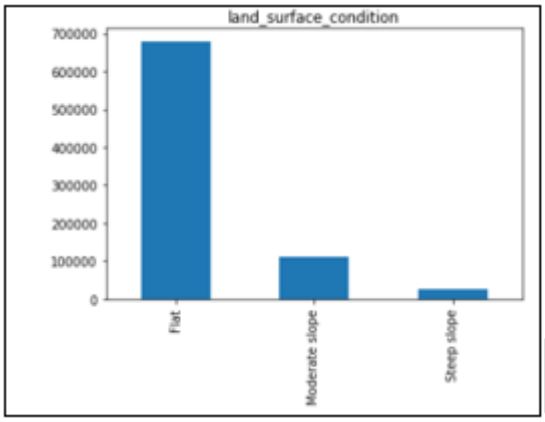
Ilustración 27 Análisis variables numéricas

VARIABLES CATEGÓRICAS

Pasemos a mostrar el racional seguido en una muestra de variables, extensibles a la totalidad de las 12 variables categóricas.

❖ Land Surface Condition

Se observan tres categorías, *"flat", "moderate slope"* y *"steep slope",* donde el nivel mayoritario es *"flat"* con un 83% de frecuencia.



Flat	83.066080
Moderate slope	13.712349
Steep slope	3.221571

Ilustración 28 Land Surface condition count

Analizando por nivel de daño, no se intuye a priori ninguna relación con el target, ya que los porcentajes se mantienen prácticamente iguales en "*High-Severe*" y "*Low-Medium*".

damage_grade	High-Severe	Low-Medium
land_surface_condition		
Flat	82.575311	83.816624
Moderate slope	13.974498	13.311440
Steep slope	3.450192	2.871937

Ilustración 29 Land Surface condition vs Damage Grade

Por tanto, parece razonable hacer una agrupación de estas categorías en tan solo dos: "Flat" y "No Flat", dando como resultado:

damage_grade	High-Severe	Low-Medium
land_surface_condition		
Flat	82.575311	83.816624
No Flat	17.424689	16.183376

Ilustración 30 Agrupación Land Surface condition

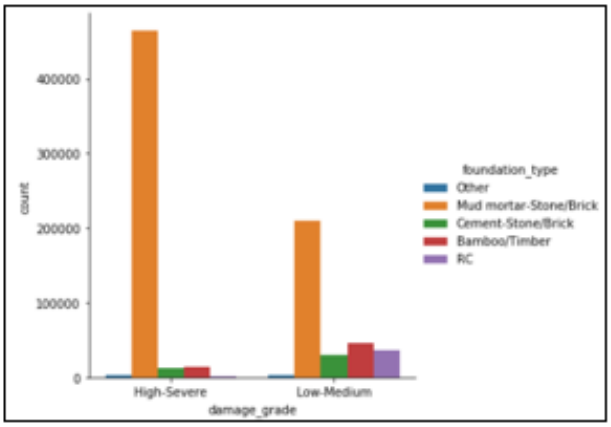
❖ Foundation Type

Existen 5 niveles donde claramente "*mud mortar – stone/brick"* es la mayoritaria:

Mud mortar-Stone/Brick	82.347746
Bamboo/Timber	7.269696
Cement-Stone/Brick	5.203371
RC	4.592158
Other	0.587028

Ilustración 31 Foundation type count

En este caso si se puede intuir una relación entre el tipo de estructura y el nivel de daño, donde edificios con otros tipos de materiales como *"RC", "Bamboo/Timber" y "Cement-Stone/Brick*" recaen proporcionalmente en edificios menos dañados.



damage_grade	High-Severe	Low-Medium
foundation_type		
Bamboo/Timber	2.848595	14.030968
Cement-Stone/Brick	2.521131	9.305372
Mud mortar-Stone/Brick	93.866259	64.732270
Other	0.494731	0.728179
RC	0.269284	11.203211

Ilustración 32 Foundation type vs Damage Grade

Por tanto, esta variable también permite una agrupación de categorías en aquellas en las que se observa un comportamiento similar, creando un material llamado *"others"* que supone la agrupación de los materiales comentados anteriormente:

damage_grade	High-Severe	Low-Medium
foundation_type		
Mud mortar-Stone/Brick	93.866259	64.73227
others	6.133741	35.26773

Ilustración 33 Agrupación Foundation type

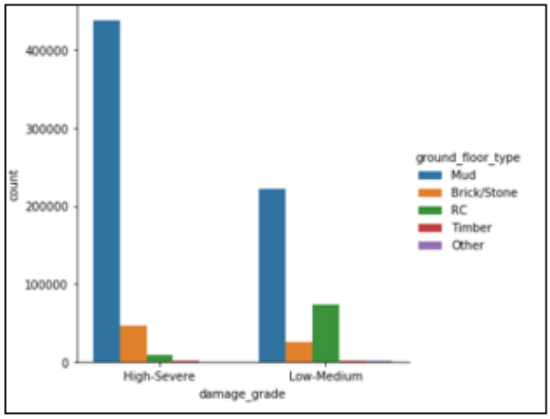
❖ Ground Floor Type

El material usado para la base de los edificios es *"mud"* es un 80% de los casos, siendo *"RC"* el siguiente material más utilizado pero su frecuencia baja a un 10% de los registros.

Mud	80.703921
RC	10.028460
Brick/Stone	8.679980
Timber	0.451692
Other	0.135947

Ilustración 34 Ground floor type count

Además, se observa que este material, *"RC"* está presente solo en edificios con bajo daño, mientras que el resto de los materiales (a excepción de *"mud")* se distribuye de manera similar.



damage_grade	High-Severe	Low-Medium
ground_floor_type		
Brick/Stone	9.288387	7.749533
Mud	88.497948	68.784370
Other	0.068483	0.239122
RC	1.788629	22.629787
Timber	0.356554	0.597187

Ilustración 35 Ground floor type vs Damage Grade

Siguiendo el racional expuesto, se agrupan categorías con mismos comportamientos:

damage_grade	High-Severe	Low-Medium
ground_floor_type		
Mud	88.497948	68.784370
RC	1.788629	22.629787
others	9.713423	8.585843

Ilustración 36 Ground floor type vs Damage Grade

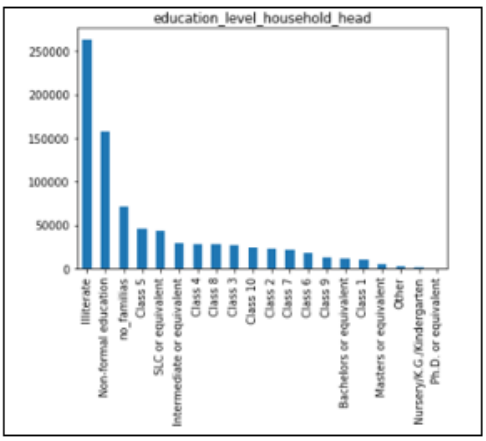
Además, cruzando variables input entre sí, en este caso con *"legal ownership*", se confirma que este material es más común en edificios públicos ("No Private"), lo cual puede tener cierta lógica asumiendo que los edificios gubernamentales se construyen con materiales de calidad:

legal_ownership_status	No Private	Private
ground_floor_type		
Mud	77.367375	80.842753
RC	15.658156	9.794210
others	6.974469	9.363037

Ilustración 37 Agrupación Ground floor type

❖ Education Level

Originalmente esta variable cuenta con 20 categorías, siendo "*Illiterate*" la mayoritaria seguida de "*non formal education*".



Illiterate	32.142909
Non-formal education	19.264932
no_familias	8.742641
Class 5	5.528643
SLC or equivalent	5.250641
Intermediate or equivalent	3.564309
Class 4	3.399902
Class 8	3.361427
Class 3	3.202150
Class 10	2.943325
Class 2	2.727617
Class 7	2.621229
Class 6	2.088433
Class 9	1.488213
Bachelors or equivalent	1.334433
Class 1	1.220960
Masters or equivalent	0.550507
Other	0.346647
Nursery/K.G./Kindergarten	0.212654
Ph.D. or equivalent	0.008428

El sistema educativo de Nepal [11]está organizado en 12 niveles y a su vez en tipos de educación (primaria y secundaria):

Primary education: class 1-5
Lower secondary: class 6-8
Secondary: class 9-10
Higher secondary: class 10-12

Por tanto, en base a esta información, una manera óptima de trabajar con esta variable podría consistir en tener 5 niveles: *"Illiterate", "low", "medium", "high", "No_familias"* (esta última corresponde con edificios sin familias).

Low	35.344204
Illiterate	32.142909
Medium	21.876878
no_familias	8.742641
High	1.893368

damage_grade	High-Severe	Low-Medium
education_level_household_head		
High	1.679744	2.220066
Illiterate	33.930822	29.408621
Low	35.782076	34.674559
Medium	21.074672	23.103706
no_familias	7.532686	10.593048

Ilustración 38 Education Level vs Damage Grade

A priori, no se ve una relación directa con el nivel de daño, es decir dentro de cada nivel de educación, el porcentaje de distribución entre "*high-severe*" y "*low-medium*" permanece prácticamente estable.

❖ Income Level

Se observa que el 50% de los edificios tienen una renta baja, en torno a *"Rs. 10 thousand"*

Rs. 10 thousand	49.969464
Rs. 10-20 thousand	28.298400
Rs. 20-30 thousand	9.298033
no_familias	8.742641
Rs. 30-50 thousand	2.680225
Rs. 50 thousand or more	1.011237

Ilustración 39 Income Level count

El sueldo medio [12]ronda los Rs. 20-23,5 thousand, por tanto, esta variable se podría agrupar en *"low", "medium", "high" y "no familias",* tomando ese umbral como referencia:

- Rs. 10 + Rs. 10-20: *"low"*
- Rs. 20-30: *"medium"*
- Rs. 30-50 + Rs >50: *"high"*
- *"no Familias"*

[11] *www.nuffic.nl/sites/default/files/2020-08/education-system-nepal.pdf*

[12] *datosmacro.expansion.com/paises/nepal#:~:text=Su%20capital%20es%20Katmand%C3%BA%20y,habitante312%24%20dolares%20por%20habitante*

Sorprendentemente, no hay una gran correlación a priori entre nivel de daño y salario.

damage_grade	High-Severe	Low-Medium
income_level_household		
High	3.295449	4.297092
Low	80.538003	74.796097
Medium	8.633862	10.313763
no_familias	7.532686	10.593048

Ilustración 40 Income Level vs Damage Grade

Con esta primera aproximación, se puede estudiar una potencial relación con el target, que se terminará de confirmar con otras técnicas más adelante.

A continuación, una tabla resumen de este análisis, así como las categorías resultantes:

Variable	Categoría mayoritaria		Categorías "pre"	Categorías "post"	¿Relacion con el nivel de daño?
Land surface condition	Flat	83%	3	2	damage_grade / land_surface_condition: Flat 82.575311 / 83.816624; No Flat 17.424689 / 16.183376 — no se obseva una relacion clara
Foundation type	Mud mortar-Stone/Brick	82%	5	2	damage_grade / foundation_type: Mud mortar-Stone/Brick 93.866259 / 64.73227; others 6.133741 / 35.26773 — **Daño bajo/medio:** "foundation_type_others"
Roof type	Bamboo/Timber-Light roof	65%	3	3	damage_grade / roof_type: Bamboo/Timber-Heavy roof 30.213771 / 25.413366; Bamboo/Timber-Light roof 69.205036 / 59.437600; RCC/RB/RBC 0.581193 / 15.149034 — **Daño bajo/medio:** "roof_type_RCC/RB/RBC"
Ground Floor type	Mud	80%	5	3	damage_grade / ground_floor_type: Mud 88.497948 / 68.784370; RC 1.788629 / 22.629787; others 9.713423 / 8.585843 — **Daño bajo/medio:** "ground_floor_type_RC"
Other Floor type	TImber/Bamboo-Mud	80%	4	3	damage_grade / other_floor_type: Not applicable 10.219064 / 22.461104; RCC/RB/RBC 0.612707 / 11.271178; Timber 89.168229 / 66.267718 — **Daño bajo/medio:** "other_floor_type_RCC/RB/RBC"
Position	Not attached	79%	4	3	damage_grade / position: Attached >1 side 2.948996 / 5.250491; Attached-1 side 18.367689 / 15.078595; Not attached 78.683315 / 79.670914 — Si el edificio está anclado por un lado o más, potencialmente hay menos riesgo de collapso
Plan configuration	Rectangular	96%	10	3	damage_grade / plan_configuration: L-shape 0.693311 / 2.447758; Rectangular 96.804548 / 94.515948; other 2.502141 / 3.036295 — Si el edificio tiene una base en forma de L, potencialmente hay menos riesgo de daño alto
Legal Ownership	Private	96%	4	2	damage_grade / legal_ownership_status: No Private 2.860514 / 5.729353; Private 97.139486 / 94.270647 — Si el edificio es de carácter no privado, puede potencialmente hay menos riesgo de daños alltos
Gender household head	Male	62%	2	2	damage_grade / gender_household_head: Female 26.590252 / 33.291420; Male 65.877063 / 56.115532; no_familias 7.532686 / 10.593048 — **Daño bajo/medio:** "other_floor_type_RCC/RB/RBC"
Education Level	Low	35%	20	5	damage_grade / education_level_household_head: High 1.679744 / 2.220066; Illiterate 33.930822 / 29.408621; Low 35.782076 / 34.674559; Medium 21.074672 / 23.103706; no_familias 7.532686 / 10.593048 — A priori, no se observa una relación clara
Income level	Low	78%	6	4	damage_grade / income_level_household: High 3.295449 / 4.297092; Low 80.538003 / 74.796097; Medium 8.633862 / 10.313763; no_familias 7.532686 / 10.593048 — A priori, no se observa una relación clara
Caste_household	Tamag	26%	97	13	damage_grade / caste_household: Brahman-Hill 13.423607 / 13.369830; Chhetree 16.805921 / 13.963928; Damai/Dholi 2.052459 / 1.629058; Gurung 3.696648 / 2.209871; Kami 3.330195 / 2.846912; Magar 4.456826 / 7.185403; Newar 7.910653 / 9.094673; Other 9.535449 / 9.190136; Rai 0.876739 / 2.080115; Sarki 2.822939 / 2.339319; Sherpa 1.486013 / 0.777301; Tamang 26.069864 / 24.720406; no_familias 7.532686 / 10.593048 — No se observa una relación clara en las castas mayoritarias.

Ilustración 41 Análisis variables categóricas

```
Categorías de land_surface_condition : ['Flat' 'No Flat']

Categorías de foundation_type : ['others' 'Mud mortar-Stone/Brick']

Categorías de roof_type : ['Bamboo/Timber-Light roof' 'Bamboo/Timber-Heavy roof' 'RCC/RB/RBC']

Categorías de ground_floor_type : ['Mud' 'others' 'RC']

Categorías de other_floor_type : ['Not applicable' 'Timber' 'RCC/RB/RBC']

Categorías de position : ['Not attached' 'Attached-1 side' 'Attached >1 side']

Categorías de plan_configuration : ['Rectangular' 'L-shape' 'other']

Categorías de damage_grade : ['Low-Medium' 'High-Severe']

Categorías de legal_ownership_status : ['Private' 'No Private']

Categorías de gender_household_head : ['Male' 'Female' 'no_familias']

Categorías de caste_household : ['Rai' 'Other' 'Brahman-Hill' 'no_familias' 'Chhetree' 'Tamang' 'Kami'
 'Newar' 'Gurung' 'Magar' 'Sarki' 'Damai/Dholi' 'Sherpa']

Categorías de education_level_household_head : ['Illiterate' 'Low' 'Medium' 'no_familias' 'High']

Categorías de income_level_household : ['Low' 'Medium' 'no_familias' 'High']
```

Ilustración 42 Categorías resultantes

VARIABLES BOOLEANAS

Recordemos que el problema cuenta con 22 variables booleanas, que se dividen en 2 grupos de información:

- Variables referentes a "*superestructura*"
- Variables referentes a "*uso secundario*" del edificio

Booleanas	
is_bank_account_present_in_househol	int64
has_superstructure_adobe_mud	int64
has_superstructure_mud_mortar_stone	int64
has_superstructure_stone_flag	int64
has_superstructure_cement_mortar_st	int64
has_superstructure_mud_mortar_brick	int64
has_superstructure_cement_mortar_br	int64
has_superstructure_timber	int64
has_superstructure_bamboo	int64
has_superstructure_rc_non_engineere	int64
has_superstructure_rc_engineered	int64
has_superstructure_other	int64
has_secondary_use	int64
has_secondary_use_agriculture	int64
has_secondary_use_hotel	int64
has_secondary_use_rental	int64
has_secondary_use_institution	int64
has_secondary_use_school	int64
has_secondary_use_industry	int64
has_secondary_use_gov_office	int64
has_secondary_use_use_police	int64
has_secondary_use_other	int64

Ilustración 43 Variables booleanas

El segundo grupo merece un análisis en detalle y sobre todo entender la relación de *"has secondary use"* con el resto de *"has secondary use_X"* específicos.

En un primer paso, se observa que en torno al 85% de edificios no tienen un uso secundario y tan solo un 12% se destina a otro uso además de vivienda.

Visualizándolo por tipo de daño, se observa que efectivamente cuando hay un uso adicional, un 16% recae en daños bajos/medios. Por tanto, en general cuando el edificio tiene un uso secundario, parece que tienen un daño bajo/medio.

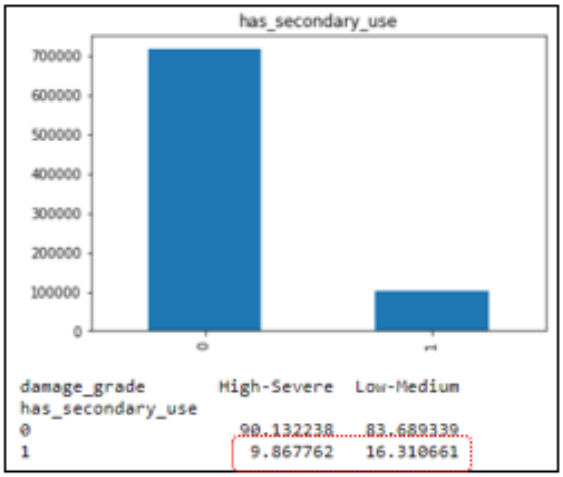


Ilustración 44 Has_secondary_use vs Damage Grade

Analizando los 9 usos secundarios (agricultura, hotel, school, institution…), se concluye que tener el detalle de uso secundario no aporta información adicional, ya que de alguna manera toda está contenida en "*has secondary use*".

Además, sus frecuencias son muy bajas, por tanto, podemos eliminar estas 9 variables:

Agricultura	Severe/high	Medium /Low
0	92,5	93,4
1	7,4	6,5

Institution	Severe/high	Medium /Low
0	99,9	99,7
1	0,0	0,2

Gov Office	Severe/high	Medium /Low
0	99,9	99,6
1	0,1	0,0

Hotel	Severe/high	Medium /Low
0	98,0	93,9
1	1,9	6,3

School	Severe/high	Medium /Low
0	99,8	99,9
1	0,0	0,1

Police	Severe/high	Medium /Low
0	99,9	99,9
1	0,1	0,0

Rental	Severe/high	Medium /Low
0	99,7	97,5
1	0,2	2,5

Industry	Severe/high	Medium /Low
0	99,9	99,9
1	0,1	0,2

Other	Severe/high	Medium /Low
0	99,6	99,3
1	0,3	0,6

Ilustración 45 Análisis "usos secundarios"

En cuando al resto de booleanas, la metodología seguida es la misma que con las variables categóricas, analizando la relación de cada una de ellas con el objetivo, tratando de intuir en base a sus porcentajes cómo pueden afectar o no.

A continuación, se muestra una tabla resumen con una primera conclusión:

has superstructure_	¿Relacion con el nivel de daño?	
adobe mud	damage_grade / has_superstructure_adobe_mud: 0 → High-Severe 95.774884, Low-Medium 95.610225; 1 → High-Severe 4.225116, Low-Medium 4.389775	no se obseva una relacion clara
Mud mortar stone	damage_grade / has_superstructure_mud_mortar_stone: 0 → High-Severe 7.259563, Low-Medium 40.217311; 1 → High-Severe 92.740437, Low-Medium 59.782689	**Daño bajo/medio:** "has superstrucuture_Mud mortar stone" = 0
stone flag	damage_grade / has_superstructure_stone_flag: 0 → High-Severe 95.706401, Low-Medium 97.713202; 1 → High-Severe 4.293599, Low-Medium 2.286798	no se obseva una relacion clara
cement mortar stone	damage_grade / has_superstructure_cement_mortar_stone: 0 → High-Severe 99.03094, Low-Medium 97.40395; 1 → High-Severe 0.96906, Low-Medium 2.59605	Si el edificio tiene "superestructura de cement mortar stone" potencialmente puede ayudar a que los daños sean bajos
mud mortar brick	damage_grade / has_superstructure_mud_mortar_brick: 0 → High-Severe 98.099657, Low-Medium 96.930031; 1 → High-Severe 1.900343, Low-Medium 3.069969	Si el edificio tiene "superestructura de mud mortar brick" potencialmente puede ayudar a que los daños sean bajos
cement mortar brick	damage_grade / has_superstructure_cement_mortar_brick: 0 → High-Severe 98.87539, Low-Medium 83.122119; 1 → High-Severe 1.12461, Low-Medium 16.877881	**Daño bajo/medio:** "has superstrucuture_cement mortark brick" = 1
timber	damage_grade / has_superstructure_timber: 0 → High-Severe 77.953642, Low-Medium 69.329964; 1 → High-Severe 22.046358, Low-Medium 30.670036	**Daño bajo/medio:** "has superstrucuture_timber" = 1
bamboo	damage_grade / has_superstructure_bamboo: 0 → High-Severe 94.320184, Low-Medium 88.794009; 1 → High-Severe 5.679816, Low-Medium 11.205991	**Daño bajo/medio:** "has superstrucuture_bamboo" = 1
rc_non_engineered	damage_grade / has_superstructure_rc_non_engineered: 0 → High-Severe 98.687315, Low-Medium 91.278222; 1 → High-Severe 1.312685, Low-Medium 8.721778	**Daño bajo/medio:** "has superstrucuture_rc_non_enginiered" = 1
rc_engineered	damage_grade / has_superstructure_rc_engineered: 0 → High-Severe 99.943638, Low-Medium 95.517851; 1 → High-Severe 0.056362, Low-Medium 4.482149	**Daño bajo/medio:** "has superstrucuture_rc_enginiered" = 1
other	damage_grade / has_superstructure_other: 0 → High-Severe 99.126291, Low-Medium 98.295251; 1 → High-Severe 0.873709, Low-Medium 1.704749	no se obseva una relacion clara

Ilustración 46 Análisis variables booleanas

3.3 ESTRATEGIAS DE SELECCIÓN DE VARIABLES

De cara a hacer una selección de variables, el proceso seguido se basó en:

1. Agrupaciones de categorías con poca frecuencia o con sentido de negocio *(apartado anterior)*
2. Filtrado inicial:
   2.1 Contraste chi2 para variables categóricas
   2.2 Análisis de correlaciones
3. Filtrado adicional: métodos wrapping (RFECV)

3.3.1 CONTRASTE CHI2

El objetivo consiste en entender si existe una relación de significación entre el target (categórica) y las variables propiamente categóricas y booleanas.

Por tanto, a un nivel de significancia del 95%:

Ho: no hay una relación significativa (variables independientes)

H1: existe una relación

Debido al gran número de registros, el contraste estadístico en todos los casos sugería no rechazar la Ho, es decir, siempre existe relación con el target para todas las variables, por tanto, esta aproximación no aporta información adicional al problema.

3.3.2 ANÁLISIS DE CORRELACIONES

En un primer paso, se estudiará la **colinealidad**, es decir la existencia de variables input correlacionadas entre sí y por tanto que aportan la misma información. Asumiendo esta colinealidad por encima de >0,75 en valor absoluto, obtenemos este resultado:

```
correlated_features

{'caste_household_no_familias',
 'education_level_household_head_no_familias',
 'foundation_type_others',
 'gender_household_head_Male',
 'gender_household_head_no_familias',
 'height_ft_pre_eq',
 'income_level_household_no_familias',
 'land_surface_condition_No Flat',
 'legal_ownership_status_Private',
 'other_floor_type_Timber',
 'plan_configuration_other',
 'position_Not attached',
 'roof_type_Bamboo/Timber-Light roof'}
```

Ilustración 47 Variables colineadas

Sin embargo, antes de eliminarlas, es buena práctica entender si tiene sentido o por el contrario esta correlación no implica causalidad. Para ello, se visualiza un cluster map con todo el conjunto de variables:

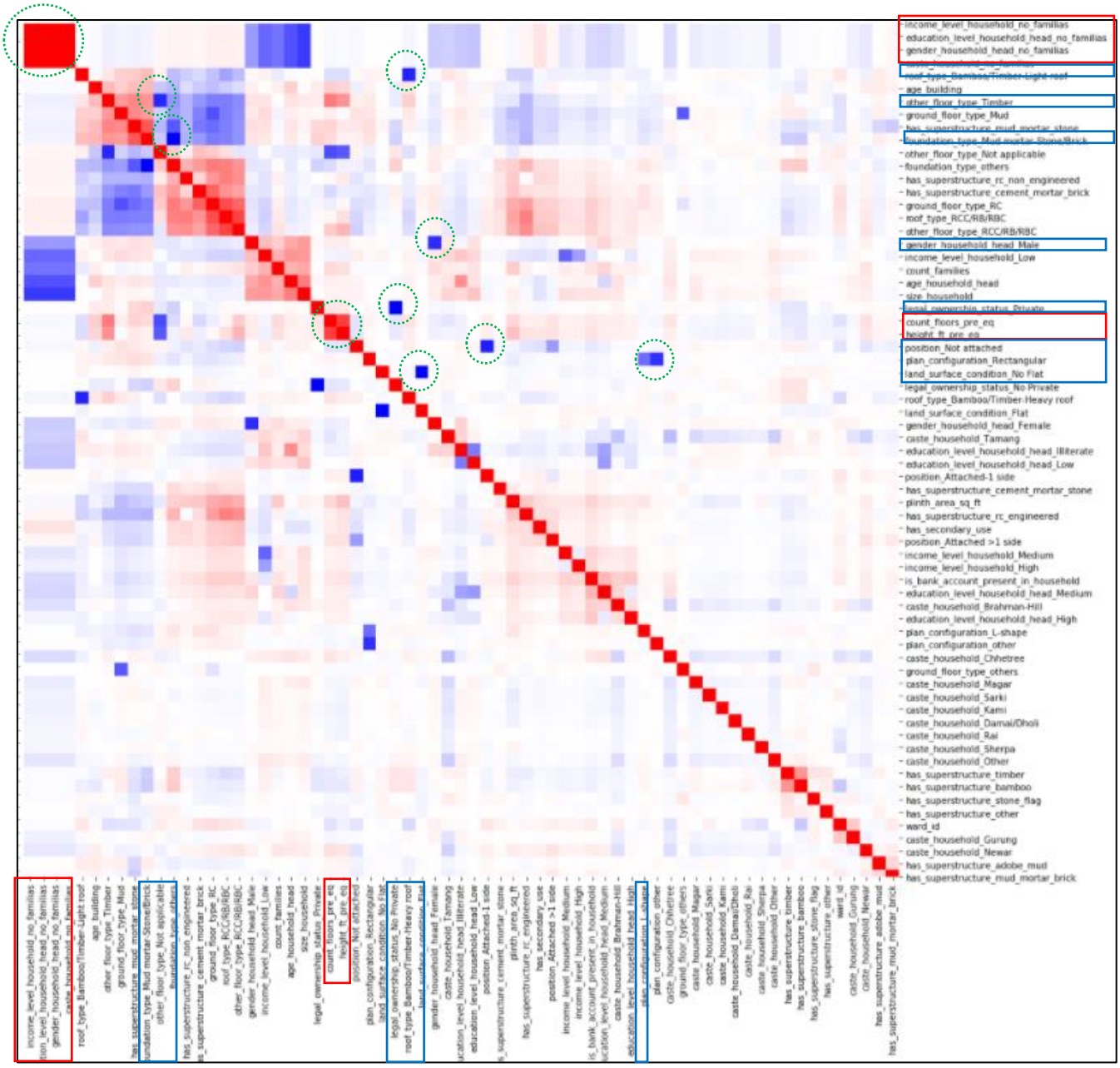


Ilustración 48 Matriz de correlación

- *"Casta_hoseuhold_no familias", "Education_level_no familias", "gender_household_no familias" e "income_level_no_familias",* están correlacionadas de manera positiva igual a 1 porque de hecho toman el mismo valor. Recordemos que la categoria "no familias" en las variables socioeconomicas, hacen referencia a valores no existentes por ser edificios no destinados a la vivienda.

- *"Heigt_ft_pre_eq",* altamente correlacionada con *"count_floors_pre_eq".* Tiene cierta lógica ya que el número de plantas y altura del edificio de alguna manera dan la misma información.

- El resto de las variables están correlacionadas en negativo con su categoría complementaria.

Por tanto, se eliminan esas variables del conjunto de datos ya que son variables redundantes y penalizarían al modelo.

En un segundo paso se estudiarán las **correlaciones de cada variable con el target**, eliminando las que tengan una correlación menor que 5% en valor absoluto.

Ilustración 49 Correlación con el target

Sin embargo, se observa que se eliminarían demasiadas variables, por tanto, no sería una buena aproximación a priori.

3.3.3 RECURSIVE FEATURE ELIMINATION CROSS VALIDATION

Un tercer paso consistiría en aplicar una técnica más avanzada como el **REFCV**, [13]obteniendo un ranking de las variables con más valor predictivo.

En el ranking de variables seleccionadas en primera position tienen cierto sentido en base a todo el análisis anterior, sin embargo, hay otras que quizás a priori pueden llamar la atención.

[13] *https://machinelearningmastery.com/rfe-feature-selection-in-python/*

rank 1	"ward_id",
rank 1	"count_floors_pre_eq",
rank 1	"age_building",
rank 1	"plinth_area_sq_ft",
rank 1	"has_superstructure_mud_mortar_stone",
rank 1	"has_superstructure_timber",
rank 1	"has_superstructure_bamboo",
rank 1	"has_secondary_use",
rank 1	"size_household",
rank 1	"is_bank_account_present_in_household",
rank 1	"ground_floor_type_Mud",
rank 1	"other_floor_type_Not applicable",
rank 1	"position_Attached-1 side",
rank 1	"caste_household_Brahman-Hill",
rank 1	"caste_household_Chhetree",
rank 1	"caste_household_Newar",
rank 1	"caste_household_Sarki",
rank 1	"education_level_household_head_High",
rank 1	"income_level_household_Medium",

rank 2	"caste_household_Magar",
rank 3	"plan_configuration_Rectangular",
rank 4	"age_household_head",
rank 5	"roof_type_RCC/RB/RBC",
rank 6	"caste_household_Tamang",
rank 7	"has_superstructure_stone_flag",
rank 8	"education_level_household_head_Low",
rank 9	"other_floor_type_RCC/RB/RBC",
rank 10	"ground_floor_type_others",
rank 11	"legal_ownership_status_No Private",
rank 12	"has_superstructure_cement_mortar_stone",
rank 13	"position_Attached >1 side",
rank 14	"income_level_household_Low",
rank 15	"caste_household_Kami",
rank 16	"gender_household_head_Female",
rank 17	"ground_floor_type_RC",
rank 18	"land_surface_condition_Flat",
rank 19	"has_superstructure_cement_mortar_brick",
rank 20	"plan_configuration_L-shape",
rank 21	"caste_household_Rai",
rank 22	"caste_household_Damai/Dholi",
rank 23	"caste_household_Gurung",
rank 24	"caste_household_Sherpa",
rank 25	"count_families",
rank 26	"has_superstructure_rc_non_engineered",
rank 27	"has_superstructure_rc_engineered",
rank 28	"education_level_household_head_Illiterate",
rank 29	"has_superstructure_adobe_mud",
rank 30	"foundation_type_Mud mortar-Stone/Brick",
rank 31	"has_superstructure_mud_mortar_brick",
rank 32	"has_superstructure_other",
rank 33	"education_level_household_head_Medium",
rank 34	"caste_household_Other",
rank 35	"income_level_household_High",
rank 36	"roof_type_Bamboo/Timber-Heavy roof",

Ilustración 50 Resultado RFECV

Llegados a este punto, se realizaron 6 estrategias diferentes de selección de variables, aplicando unos modelos de predicción sencillos (RF y XGBoost) con el objetivo de entender las variaciones en el score de cada modelo, en base a cada de las estrategias seguidas.

En esta tabla resumen se puede observar cada una de ellas y cómo las estrategias 3 y 4 son la que ofrecen un mayor accuracy. El elemento común en ambas es el tratamiento de la colinealidad, mientras que es mejor no eliminar las variables correlacionadas (de manera independiente) con el target.

Estrategia	Agrupación de categorías	Colinealidad	Correlación con el Target	RFECV	Random Forest	Xgboost
1	Si	Si	Si	Si	73,06	72,92
2	Si	Si	Si	No	71,18	73,93
3	Si	Si	No	Si	77,46	77,25
4	Si	Si	No	No	78,06	77,64
5	Si	No	No	No	74,16	74,10
6	Si	No	No	Si	69,7	69,7

Ilustración 51 Estrategias de selección de variables

En cuando a la aplicación de la técnica RFECV o no, se observa que el modelo no varía mucho, por tanto, se opta por lleva a cabo la estrategia 4, dado que el desarrollo es más sencillo y rápido computacionalmente.

3.4 OTRAS TÉCNICAS: PCA Y CLUSTERING

Una de las grandes aplicaciones de **PCA** es la selección de variables, sin embargo, no se recomienda en este problema ya que como se ha podido intuir hasta el momento, cada variable aporta un poco en la explicación del resultado total.

En componentes principales se observa que la tres primeras combinaciones lineales de variables explican en torno a un 24% de la varianza, siendo a partir de la cuarta donde la explicación de es menor. Visualizándolo en un gráfico, efectivamente el 80% de la varianza se consigue en torno a 30 componentes principales.

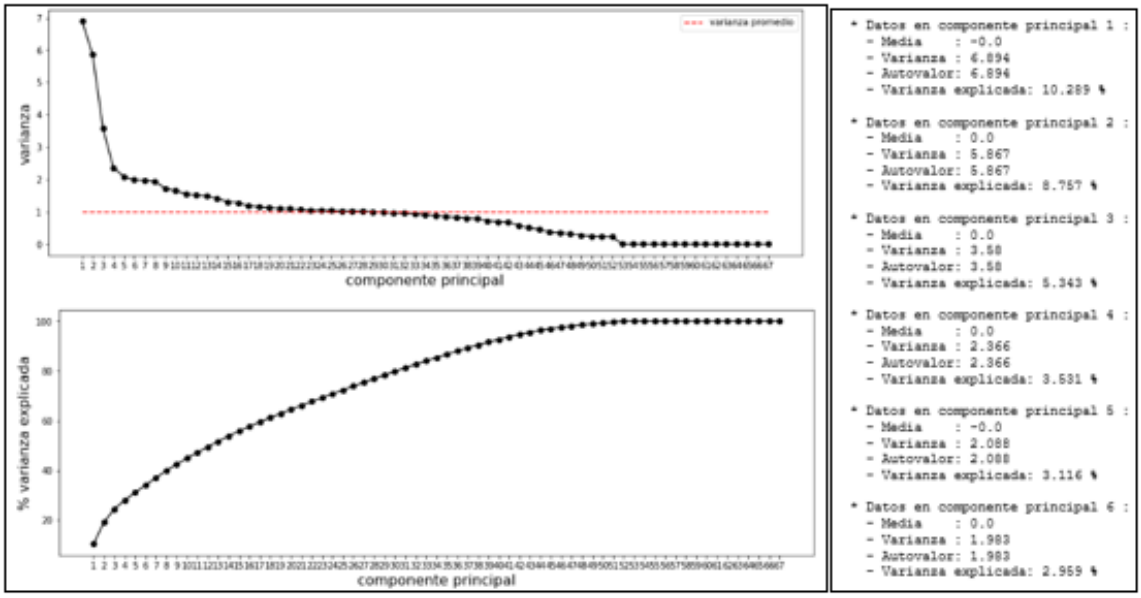


Ilustración 52 Componentes principales

Sin embargo, PCA si puede ayudar en este caso a entender cómo de separadas están las clases. En los gráficos inferiores se observa cómo los daños grandes y daños pequeños se encuentran muy solapados, siendo la clase 1, en los extremos lo que a priori se podrá predecir mejor.

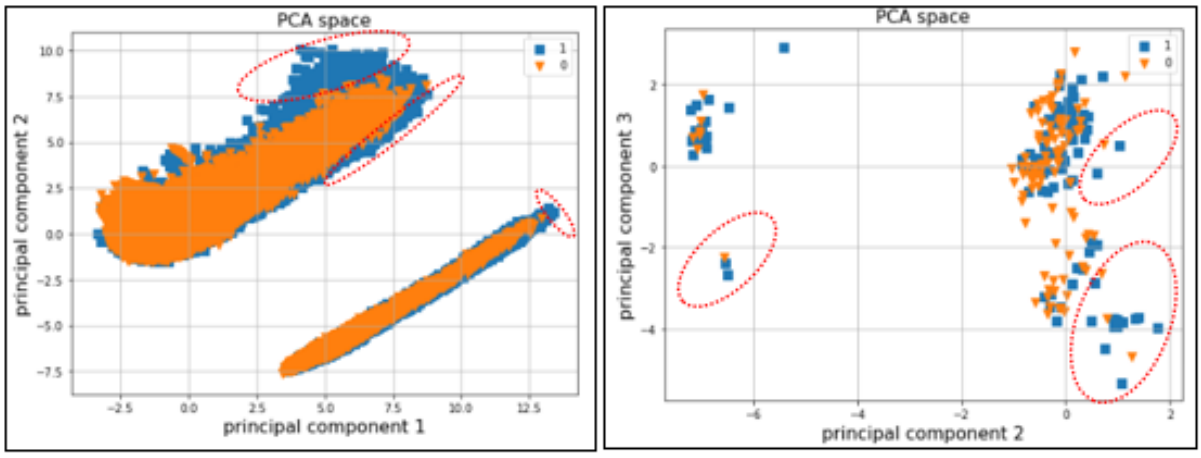


Ilustración 53 PCA para dos componentes

Otra técnica que se puede aplicar para visualizar e intuir las clases sería un **Clustering**, en este caso un K-means. Como se observa, hay 2 grupos separados, pero en cada grupo existen puntos difusos en los que se mezclan las clases, por tanto, parece que la predicción de los extremos será mejor que la predicción de los puntos intermedios

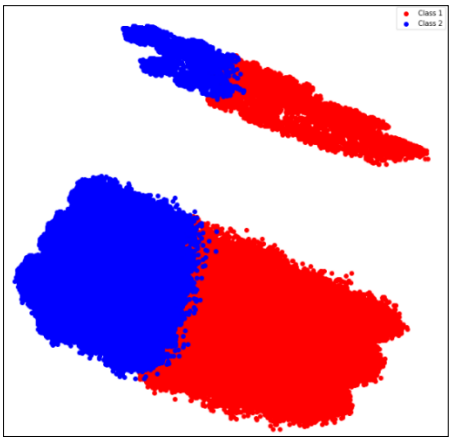


Ilustración 54 Clustering

3.5 CREACIÓN DEL CONJUNTO DE CONSTRUCCIÓN Y EXPLOTACIÓN PARA PRUEBA DE CONCEPTO

Tras el resultado del análisis exploratorio, se ha divido el conjunto de datos de la siguiente forma para la prueba de concepto:

- *Datos de construcción:* conjunto de datos utilizado para entrenar los modelos, siendo aleatoriamente el 80% del total de los datos. Éstos a su vez se han dividido en train y test.

- *Datos de explotación:* conjunto de datos utilizado para la Prueba de concepto, siendo el 20% restante.

A partir de ese conjunto de datos se han generado las predicciones representadas en la herramienta de visualización.

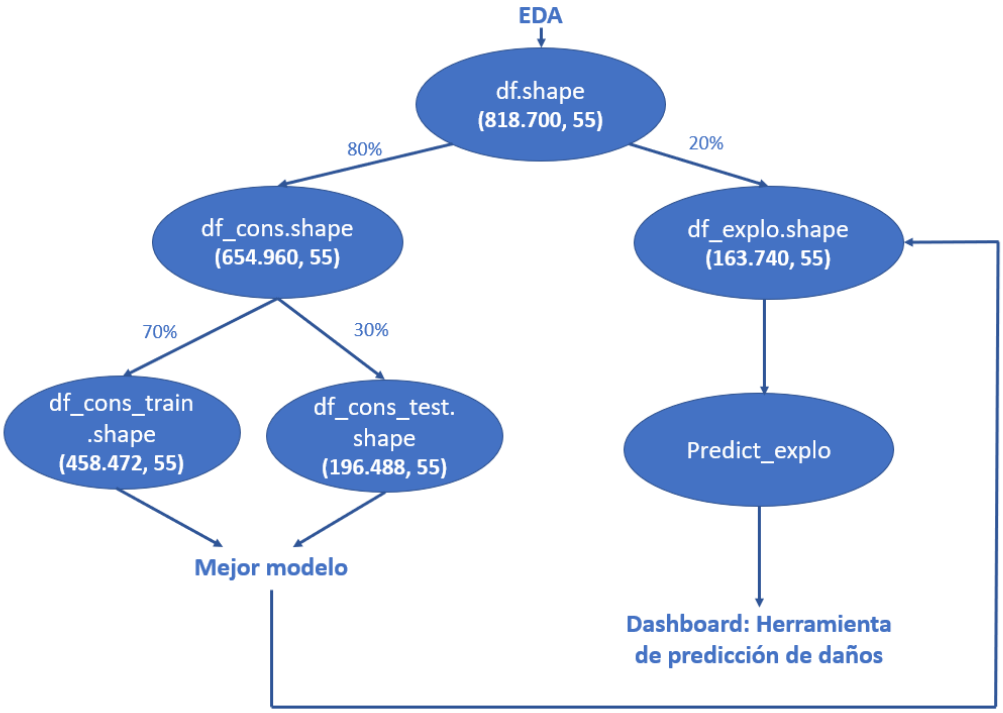


Ilustración 55 Creación del conjunto de datos para la PoC

3.6 ALGORITMOS DE APRENDIZAJE SUPERVISADO

A lo largo de este apartado se van a numerar los distintos modelos entrenados y los resultados obtenidos. Se trata de un problema de aprendizaje supervisado, el algoritmo recibe una secuencia de entradas y las salidas asociadas, donde el objetivo es aprender a producir una salida generalizada y correcta dada una nueva entrada.

Se considerarán las medidas básicas de rendimiento derivadas de la matriz de confusión[14] para la evaluación de los modelos:

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = Σ True Positive / Σ Test Outcome Positive
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = Σ True Negative / Σ Test Outcome Negative
		Sensitivity = Σ True Positive / Σ Condition Positive	**Specificity** = Σ True Negative / Σ Condition Negative	

Ilustración 56 Matriz de confusión

[14] *https://www.unite.ai/what-is-a-confusion-matrix/*

- Verdaderos Positivos (TP) / Verdaderos Negativos (TN)
- Falsos Positivos (FP) (Error Tipo I) / Falsos Negativos (FN) (Error Tipo II)
- Sensibilidad//Recall/: TP/(TP+FN)
- Razón de Falsos Positivos: FP/(FP+TN)
- Accuracy: (TP+TN)/(TP+TN+FP+FN)
- Precision: TP/(TP+FP) Especificidad: TN/(TN+FP) = 1 – FPR

Se cuenta con predicciones binarias, "*High-Severe*" '1' y "*Low-Medium*" '0'. Dada la naturaleza de los datos y su aplicación práctica, el objetivo es intentar maximizar el *recall* en la clase 1, (minimizando el número de falsos negativos), así como tratando de evitar posibles sobreajustes para encontrar un modelo que generalice lo máximo posible.

3.6.1 MODELO DUMMY

Como punto de partida en el modelado, se utiliza un clasificador dummy que siempre predice la clase mayoritaria. El score del dummy serviría de base para trabajar ya que cualquier clasificador que estuviera por debajo sería automáticamente descartado.

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
1	**Dummy**	strategy='most_frequent'	0,6	0,6

	precision	recall	f1-score	support
0	0.00	0.00	0.00	77745
1	0.60	1.00	0.75	118743
accuracy			0.60	196488
macro avg	0.30	0.50	0.38	196488
weighted avg	0.37	0.60	0.46	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	0	77745
True High-Severe	0	118743

Ilustración 57 Matriz de confusión del test1: Clasificador Dummy

3.6.2 REGRESIÓN LOGÍSTICA

La regresión logística se utiliza para clasificación y trata de encontrar la función que relaciona las variables independientes y dependientes.

En este caso, se ha aplicado una regresión logística obteniendo los siguientes resultados:

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
2	**Regresión logística**	solver='lbfgs' C=10	0,74	0,742

	precision	recall	f1-score	support
0	0.80	0.47	0.59	77745
1	0.73	0.92	0.81	118743
accuracy			0.74	196488
macro avg	0.76	0.70	0.70	196488
weighted avg	0.75	0.74	0.72	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	36383	41362
True High-Severe	9153	109590

Ilustración 58 Matriz de confusión del test2: Regresión Logística

Se puede observar que el modelo muestra un buen recall en la clase 1 pero no en la clase 0.

3.6.3 NAÏVE BAYES

Se trata de un clasificador[15] probabilístico fundamentado en el teorema de Bayes, donde se asume que las variables predictoras son independientes entre sí. Es decir, que la presencia de una cierta característica en un conjunto de datos no está en absoluto relacionada con la presencia de cualquier otra característica.

Los resultados obtenidos con los parámetros por defecto:

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
3	**Naive Bayes**	default	0,634	0,634

	precision	recall	f1-score	support
0	0.70	0.13	0.22	77745
1	0.63	0.96	0.76	118743
accuracy			0.63	196488
macro avg	0.67	0.55	0.49	196488
weighted avg	0.66	0.63	0.55	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	10052	67693
True High-Severe	4219	114524

Ilustración 59 Matriz de confusión del test 3: Naive Bayes

En este caso, está prediciendo casi por defecto la clase 1 y no mejora prácticamente el score respecto al modelo dummy. Por lo tanto, esta aproximación no sería válida ya que no es capaz de predecir la clase 0.

3.6.4 KNN

Este algoritmo "perezoso" delega todo el cálculo a la fase de clasificación. Cuando se recibe una instancia nueva que se debe clasificar, se calcula su distancia a todas las instancias del conjunto de entrenamiento y nos quedamos con los k puntos más cercanos.

[15] *https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f*

Los resultados obtenidos para los dos test realizados con este algoritmo son los siguientes:

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
4	**KNN**	n_neighbors=3	0,878	0,781
5		n_neighbors=5	0,851	0,789

Este algorimo muestra buenos resultados y se ha probado con 3 y 5 neighbors, siendo los resultados de éste segundo modelo (5 vecinos) los que se muestran a continuación:

	precision	recall	f1-score	support
0	0.75	0.69	0.72	77745
1	0.81	0.85	0.83	118743
accuracy			0.79	196488
macro avg	0.78	0.77	0.78	196488
weighted avg	0.79	0.79	0.79	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	54006	23739
True High-Severe	17685	101058

Ilustración 60 Matriz de confusión del test 5: KNN

3.6.5 DECISION TREE CLASSIFIER

El objetivo de este algoritmo es crear un modelo que prediga el valor de una variable mediante el aprendizaje de reglas simples de decisión inferidas a partir de las características de los datos. Tiene una estructura de árbol donde un nodo interno representa una característica, la rama representa una regla de decisión y cada nodo hoja representa un resultado.

Usando los parámetros por defecto (test nº6), se obtiene un mayor score en training, pero se puede observar que hay una gran diferencia en test, con lo que el modelo parece estar sobre ajustando.

Para intentar corregirlo, se opta por modificar los parámetros y realizar otro test, nº7, el cual ha bajado el score, pero el modelo deja de sobre ajustar.

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
6	**Decission Tree**	default	0,999	0,768
7		min_samples_split=10 min_samples_leaf=5 max_depth=3	0,742	0,743

	precision	recall	f1-score	support
0	0.81	0.46	0.59	77745
1	0.73	0.93	0.81	118743
accuracy			0.74	196488
macro avg	0.77	0.70	0.70	196488
weighted avg	0.76	0.74	0.72	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	35985	41760
True High-Severe	8542	110201

Ilustración 61 Matriz de confusión del test 7: Decission Tree

En base a la matriz de confusión, el recall en la clase 0 no es bueno, se deja más de un 50% de registros sin predecir correctamente. A pesar de que el principal objetivo es maximizar la clase 1, no se puede perder de vista los resultados en la clase 0 por lo que se va a intentar mejorar el resultado.

3.6.6 RANDOM FOREST[16]

Se trata de una técnica de aprendizaje supervisado en la cual se generan un gran número de árboles de decisión individuales, que se combinan con el fin de obtener un único modelo, más robusto se compara con los resultados de cada árbol por separado.

Cada árbol generado contiene un grupo de observaciones aleatorias extraídas de los datos y generadas mediante bootstrapping que es una técnica estadística para obtener muestras de población donde una observación se puede considerar en más de una muestra. Las salidas de todos los árboles se combinan en una salida final que se obtiene mediante alguna regla. Cuando las salidas de los árboles del ensamblado son numéricas se utiliza el promedio y cuando son categóricas, el conteo de votos.

El Random Forest es el algoritmo que mejores resultados estaba mostrando, por tanto, se decidió realizar diversos tests ajustando parámetros para intentar mejorar el modelo.

16

https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting

N0. test	Clasificador	Parámetros	Scoring training	Scoring test
8	**Random Forest**	**n_estimators=100 random_state=0 max_features=None n_jobs=-1 max_depth=15**	**0,861**	**0,82**
9		n_estimators=200 random_state=0, min_samples_split=2 min_samples_leaf=2 max_features=None n_jobs=-1	0,983	0,823
10		n_estimators=300, random_state=0, min_samples_split=4, min_samples_leaf=2, max_features=None, n_jobs=-1	0,983	0,823
11		n_estimators=200, random_state=0, min_samples_split=4, min_samples_leaf=2, max_features=None, n_jobs=-1	0,983	0,823
12		n_estimators=500, random_state=0, min_samples_split=4, min_samples_leaf=2, max_features=None, n_jobs=-1	0,984	0,824

Los test nº 9, 10 y 11 muestran mejor accuracy en training, sin embargo, existe una gran diferencia con el test por lo que parece que esos modelos estén sobreajustando. A priori, no se seleccionan ya que el objetivo es encontrar un modelo lo más genérico posible.

Por lo tanto, el modelo que se elegiría con este algoritmo es el nº 8, que es el que mejor resultados presenta sin sobreajustar. Además, no solo muestra unos mejores resultados globales, sino que mejora también el recall en la clase 1 y 0.

```
              precision    recall  f1-score   support

           0       0.79      0.74      0.77     77745
           1       0.84      0.87      0.85    118743

    accuracy                           0.82    196488
   macro avg       0.81      0.81      0.81    196488
weighted avg       0.82      0.82      0.82    196488

                 Predicted Low-Medium  Predicted High-Severe
True Low-Medium                 57733                  20012
True High-Severe                15195                 103548
```

Ilustración 62 Matriz de confusión del test8: RF

3.6.7 MODELOS BOOSTING

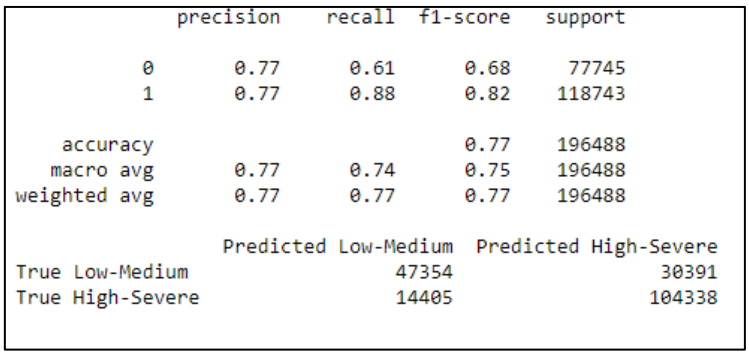
Los algoritmos mostrados a continuación se basan en el boosting, cuya idea es generar múltiples modelos de predicción “débiles” secuencialmente, y que cada uno de estos tome los resultados del modelo anterior, para generar un modelo más “fuerte”, con mejor poder predictivo y mayor estabilidad en sus resultados

Adaboost

AdaBoost[17] (adaptive boosting) fue propuesto por (Freund and Schapire 1995) y consiste en crear varios predictores sencillos en secuencia, de tal manera que el segundo ajuste bien lo que el primero no ajustó, que el tercero ajuste un poco mejor lo que el segundo no pudo ajustar y así sucesivamente.

A continuación, se muestran los resultados obtenidos para este algoritmo, no aportando una gran mejora en los parámetros respecto a los obtenidos anteriormente.

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
13	**Adaboost**	n_estimators=100 random_state=0	0,77	0,7



	precision	recall	f1-score	support
0	0.77	0.61	0.68	77745
1	0.77	0.88	0.82	118743
accuracy			0.77	196488
macro avg	0.77	0.74	0.75	196488
weighted avg	0.77	0.77	0.77	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	47354	30391
True High-Severe	14405	104338

Ilustración 63 Matriz de confusión del test 13: Adaboost

Gradient Boosting[18]

La diferencia con Adaboost es que ya no se pesa cada punto independientemente, sino que se propone una función de error cuyo gradiente hay que minimizar.

El mejor caso es el test nº 16, que como se puede observar, predice bastante bien la clase 1 aunque no es el algoritmo que mejores resultados muestra en la clase 0.

NO. test	Clasificador	Parámetros	Scoring training	Scoring test
14	**Gradient boosting**	n_estimators=5	0,74	0,741
15		learning_rate=0.01 n_estimators=200 max_features=0.8 min_samples_split=4	0,752	0,754
16		learning_rate=0.02 n_estimators=500 max_features=0.8 min_samples_split=4	0,775	0,776

[17] *https://fhernanb.github.io/libro_mod_pred/adaboost.html*

[18] *https://spainml.com/blog/como-funciona-gradient-boosting/*

	precision	recall	f1-score	support
0	0.77	0.61	0.69	77745
1	0.78	0.88	0.83	118743
accuracy			0.78	196488
macro avg	0.78	0.75	0.76	196488
weighted avg	0.78	0.78	0.77	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	47762	29983
True High-Severe	13932	104811

Ilustración 64 Matriz de confusión del test16: Gradient Boosting

XGBoost [19] (eXtreme Gradient Boosting)

En este algoritmo, también basado en árboles de decisión, los parámetros de cada modelo son ajustados iterativamente tratando de encontrar el mínimo de una función objetivo, que puede ser la proporción de error en la clasificación, el área bajo la curva (AUC), la raíz del error cuadrático medio (RMSE) o alguna otra.

Cada modelo es comparado con el anterior. Si un nuevo modelo tiene mejores resultados, entonces se toma éste como base para realizar modificaciones. Si por el contrario, tiene peores resultados, se regresa al mejor modelo anterior y se modifica de una manera diferente.

Este proceso se repite hasta llegar a un punto en el que la diferencia entre modelos consecutivos es insignificante, lo cual indica que se ha encontrado el mejor modelo posible, o cuando se llega al número de iteraciones máximas definido por el usuario.

A priori dado la cantidad de datos y número de variables disponibles, era el algoritmo que podría arrojar mejores resultados. Como se puede observar tras realizar el proceso de ajuste de los hiperparámetros[20], el modelo mejora significativamente respecto a los parámetros por defecto:

N0. test	Clasificador	Parámetros	Scoring training	Scoring test
17	**XGBoost**	default	0,773	0,774
18		colsample_bytree=0.7771636084915482 gamma=6.583804065504037 max_depth= 11 min_child_weight= 4.0 reg_alpha= 69 reg_lambda= 0.2942349631872772	0,808	0,806

[19] *https://rpubs.com/jboscomendoza/xgboost_en_r*

[20] A Guide on XGBoost hyperparameters tuning

	precision	recall	f1-score	support
0	0.78	0.70	0.74	77745
1	0.82	0.87	0.84	118743
accuracy			0.81	196488
macro avg	0.80	0.79	0.79	196488
weighted avg	0.80	0.81	0.80	196488

	Predicted Low-Medium	Predicted High-Severe
True Low-Medium	54796	22949
True High-Severe	15137	103606

Ilustración 65 Matriz de confusión del test19: XGBoost

3.6.8 TABLA RESUMEN DE RESULTADOS

A continuación, se muestra una tabla con el resumen de los distintos algoritmos probados:

N0. test	Clasificador	Parámetros	Scoring training	Scoring test	AUC	Recall clase 0	Recall clase 1
	Dummy	strategy='most_frequent'	0,6	0,6	0,5	0	1
2	Regresión logística	solver='lbfgs' C=10	0,74	0,742	0,77	0,47	0,92
3	Naive Bayes	default	0,634	0,634	0,72	0,13	0,96
5	KNN	n_neighbors=5	0,851	0,789	0,85	0,69	0,85
7	Decission Tree	min_samples_split=10 min_samples_leaf=5 max_depth=3	0,742	0,743	0,78	0,46	0,93
8	Random Forest	**n_estimators=100** **random_state=0** **max_features=None** **n_jobs=-1** **max_depth=15**	**0,861**	**0,82**	**0,9**	**0,74**	**0,87**
13	Adaboost	n_estimators=100 random_state=0	0,77	0,772	0,85	0,61	0,88
16	Gradient boosting	learning_rate=0.02 n_estimators=500 max_features=0.8 min_samples_split=4	0,775	0,776	0,86	0,61	0,88
18	XGBoost	colsample_bytree= 0.7771636084915482 gamma= 6.583804065504037 max_depth= 11 min_child_weight= 4.0 reg_alpha= 69 reg_lambda=0.2942349631872772	0,808	0,806	0,89	0,7	0,87

El mejor modelo y el utilizado para la prueba de concepto es el Random Forest ya que es el que mejor score alcanza, sin sobreajustar, y además presenta unos buenos números en cuanto al recall de la clase 1, no existiendo diferencias significativas en esta métrica respecto a otros modelos.

El modelo XGBoost también tiene unos buenos resultados, pero no se va a utilizar ya que es más costoso computacionalmente.

3.6.9 CURVA PRECISION RECALL

La prueba de concepto se ha llevado a cabo aplicando el resultado del mejor modelo a los datos de explotación, obteniendo así una probabilidad de colapso de todos los edificios.

Para poder hacer una representación gráfica, se han fijado *tres umbrales* considerando la curva *precisión-recall de la clase 1*: edificios con daño alto, medio y bajo.

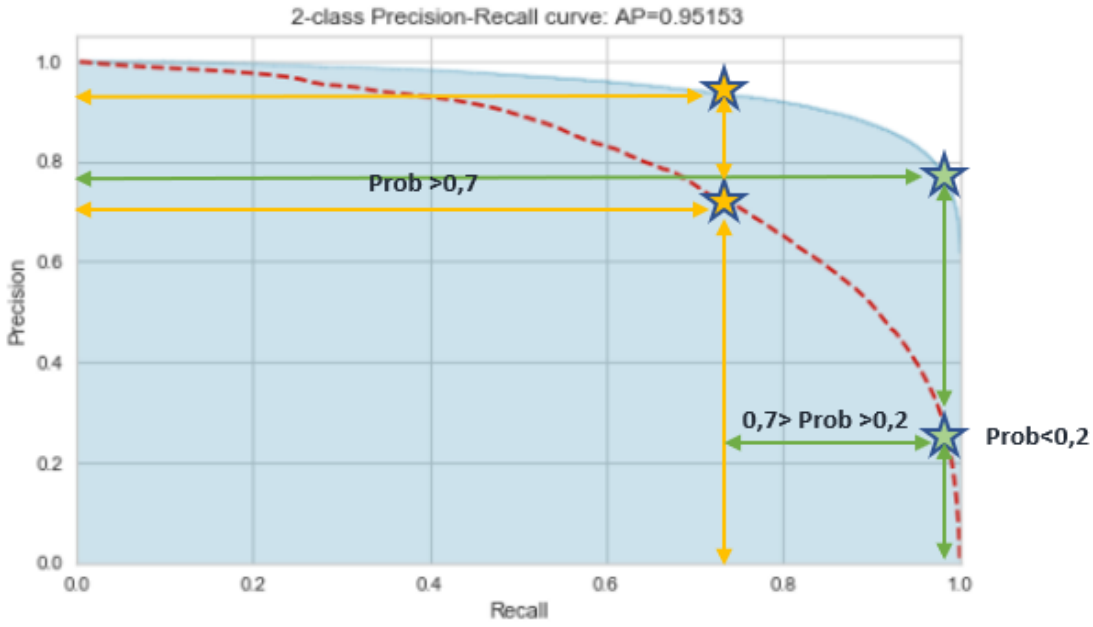


Ilustración 66 Curva Precision-Recall

- **Edificios con daño alto**: Probabilidad de clase 1 >70%.
  Se estarían detectando en torno al 70% de los edificios que realmente se van a caer y se conseguiría una precisión mayor que el 0.9. Se ha intentado encontrar un compromiso entre detectar el mayor número posible de edificios que se van a caer sin obtener muchos falsos positivos.

- **Edificios con daño medio:** 20%< Probabilidad de clase 1 <70%.
  De esta forma se catalogarían como riesgo medio un 25% (~0.95-0.7) de edificios que se van a caer.

- **Edificios con daño bajo**: 0%< Probabilidad de clase 1 <20%.
  Es decir, no se estarían detectando un 5% de edificios que realmente son clase 1.

3.7 PRUEBA DE CONCEPTO: HERRAMIENTA DE VISUALIZACIÓN

En el desarrollo del PoC, se ha tratado de demostrar la viabilidad en una aplicación de **preparación** y **respuesta** a desastres naturales. El objetivo es validar la funcionalidad de la aplicación y su integración en el proceso actual, identificando puntos bloqueantes y determinar futuros pasos.

Se ha intentado integrar la funcionalidad de la herramienta dentro del ciclo de eventos adversos que se muestra a continuación, esencialmente como ya se ha comentado, durante la fase de preparación y respuesta.

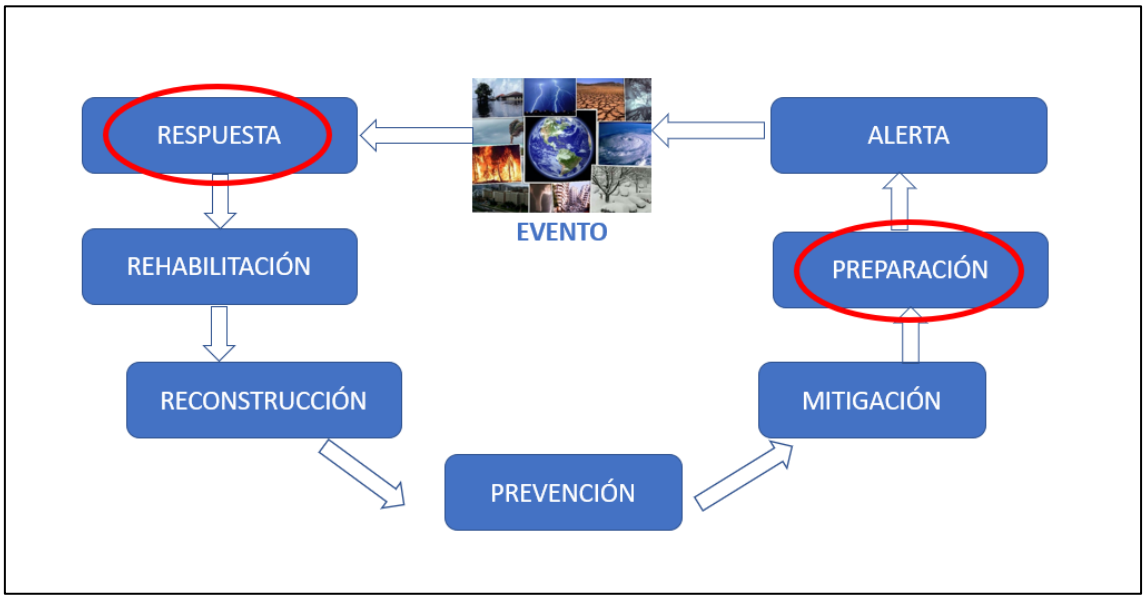


Ilustración 67 Ciclo de eventos adversos

Para entender el proceso, lo primero que hay que explicar es el modelo de explotación y las necesidades previas.

Un primer paso necesario para poder explotar la solución propuesta sería crear un censo con las características de las viviendas y otro tipo de variables input (barrio, renta per cápita…), a definir con ayuda de expertos en técnicos en edificación, personal de emergencia… y disponer así de variables lo más representativas posibles. Las variables analizadas a lo largo de este TFM podrían servir de base para ese proceso, que se integraría dentro de la fase de preparación del ciclo de eventos adversos.

Una vez que se produce el terremoto, dentro de la fase de respuesta, se debería realizar un muestreo de edificios de la zona afectada y clasificarlos de acuerdo con el daño que se ha producido. Con este muestreo, se entrenaría de forma rápida el modelo, obteniendo predicciones de colapso para la totalidad de edificios censados.

3.7.1 COMPONENTES DEL DASHBOARD

El desarrollo del dashboard se ha llevado a cabo con *Dash*[21]. Se han utilizado los principales *Dash* [22]*Core components* y *Dash HTML components* para poder hacer la herramienta interactiva, así como *Boostrap* para mejorar el layout.

[21] *https://dash.plotly.com/*

[22] *https://dash-bootstrap-components.opensource.faculty.ai/*

Para la representación geográfica de los datos, se ha utilizado *Choropleth Maps* [23] que permite cruzar el mapa de Nepal en formato geojson con los datos de las encuestas e integrarlo en Dash.

La herramienta está formada las tres pestañas que se describen a continuación.

Pestaña 1: Sobre el Proyecto

Esta pestaña contiene información general del TFM: motivación, objetivo y fuente de los datos.

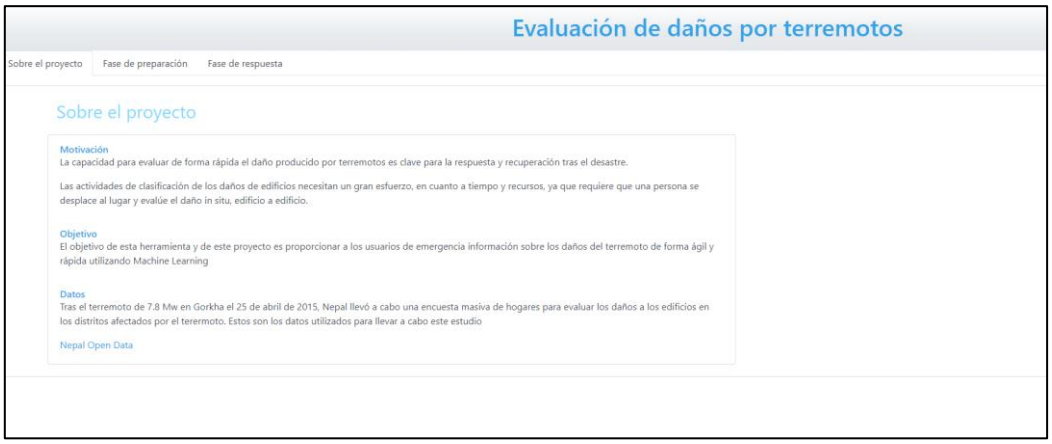


Ilustración 68 Pestaña 1: Sobre el proyecto

Pestañas 2 Y 3: Fase de Preparación y Respuesta

Las pestañas 2 y 3 contienen elementos comunes, ya que el layout que se muestra es igual, sin embargo, la información es distinta.

Los elementos comunes son:

- *Dropdown distrito:* aplica un filtro geográfico que permite seleccionar los distintos distritos afectados. De esta forma se puede añadir granularidad al proceso. Sin aplicar ningún filtro se mostrarían todos los distritos afectados y al aplicar el filtro se muestra sólo el distrito seleccionado y los municipios que lo forman.

- *Dropdown variable:* filtra la variable que se desea mostrar.

Ilustración 69 Dropdown Dashboard

[23] *https://plotly.com/python/choropleth-maps/*

- *Mapa*: muestra la información representada geográficamente de acuerdo con los valores de los dropdown. Debido a la naturaleza de las distintas variables a representar, se han utilizado varias métricas para la visualización. Las variables numéricas se han representado con la media y las categóricas con el porcentaje de cada nivel.

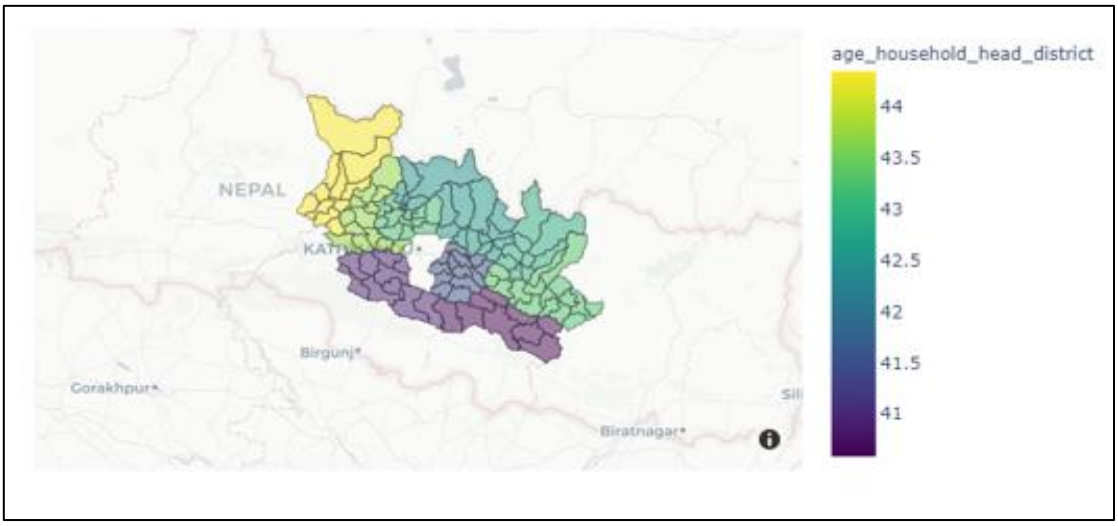


Ilustración 70 Ejemplo Mapa

- *Tabla*: representa la información de los dropdown seleccionados en forma tabular. Se puede interactuar y ordenar en función de las columnas que más interesen en cada momento. En ambas pestañas, siempre se muestra el distrito, columna "District", y el municipio, columna "N_ID".

N_ID	DISTRICT	age_household_head_municipality
kispang-nuwakot	Nuwakot	44.68512624655209
meghang-nuwakot	Nuwakot	43.443174862273004
tarkeshwar-nuwakot	Nuwakot	45.50444726810674
bidur-nuwakot	Nuwakot	41.81349813375757
tadi-nuwakot	Nuwakot	46.72650862068966
dupcheshwar-nuwakot	Nuwakot	42.949371277299804
panchakanya-nuwakot	Nuwakot	45.17818671454219
kakani-nuwakot	Nuwakot	42.16177465169755
shivapuri-nuwakot	Nuwakot	43.05734874643447
belkotgadhi-nuwakot	Nuwakot	42.454211171423566

« ‹ 1 / 2 › »

Ilustración 71 Ejemplo tabla

Pestaña 2: Fase de Preparación

El objetivo de esta pestaña es mostrar de manera interactiva, las variables más significativas generadas en el censo de edificios. De esta forma, los usuarios podrían visualizar esas variables, obtener información y sacar conclusiones para prepararse ante posibles incidentes.

Las variables que se pueden representar en la versión actual son:

- Edad media del cabeza de familia
- Edad media del edificio
- Tamaño medio de los hogares
- Porcentaje de hogares con suelo de barro
- Porcentaje de hogares cuyos ingresos son menores a 20.000 rupias
- Porcentaje de hogares cuyo cabeza de familia no tiene educación escolar

La pestaña también contiene dos cuadros de texto que muestran la población y los hogares censados.

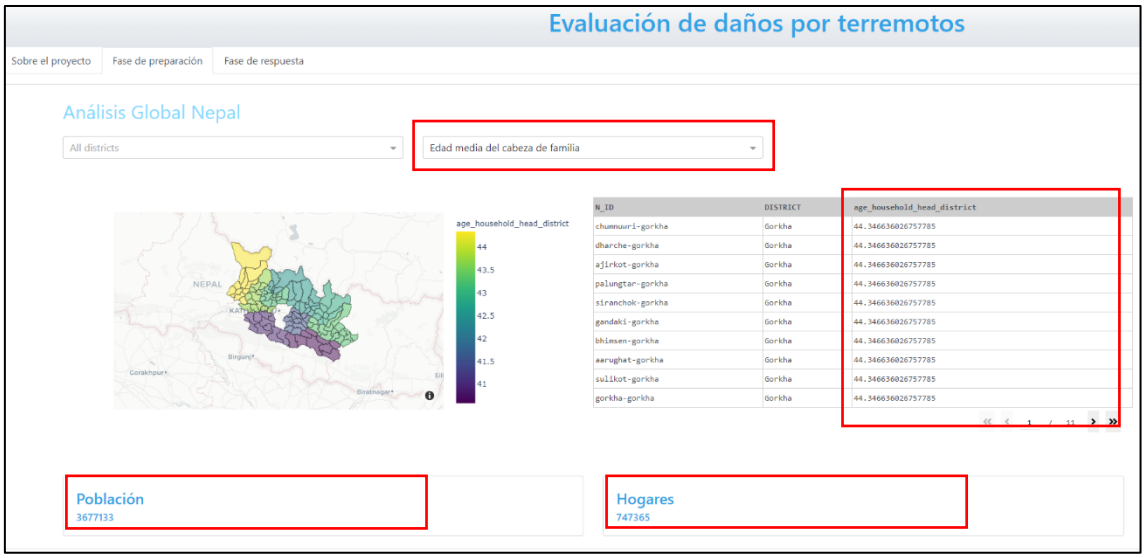


Ilustración 72 Pestaña 2: Fase Preparación

Pestaña 3: Fase de Respuesta

Una vez que se ha producido el terremoto, se proporcionaría a los usuarios la información que se ha conseguido tras entrenar el modelo.

En este caso, las variables a representar son el porcentaje de edificios con *daño alto, medio* y *bajo*, considerando los umbrales definidos anteriormente.

La tabla también refleja la información detallada por edificio:

- *Building_id:* Este identificador tendría asociada la información del edificio, número de familias e incluso sus contactos.

- *Probabilidad colapso edificio*: probabilidad de colapso del edificio determinada por el modelo.

- *Porcentaje edificios con daño alto/medio/bajo:* porcentaje de edificios con un daño determinado de acuerdo con la selección de los dropdown.

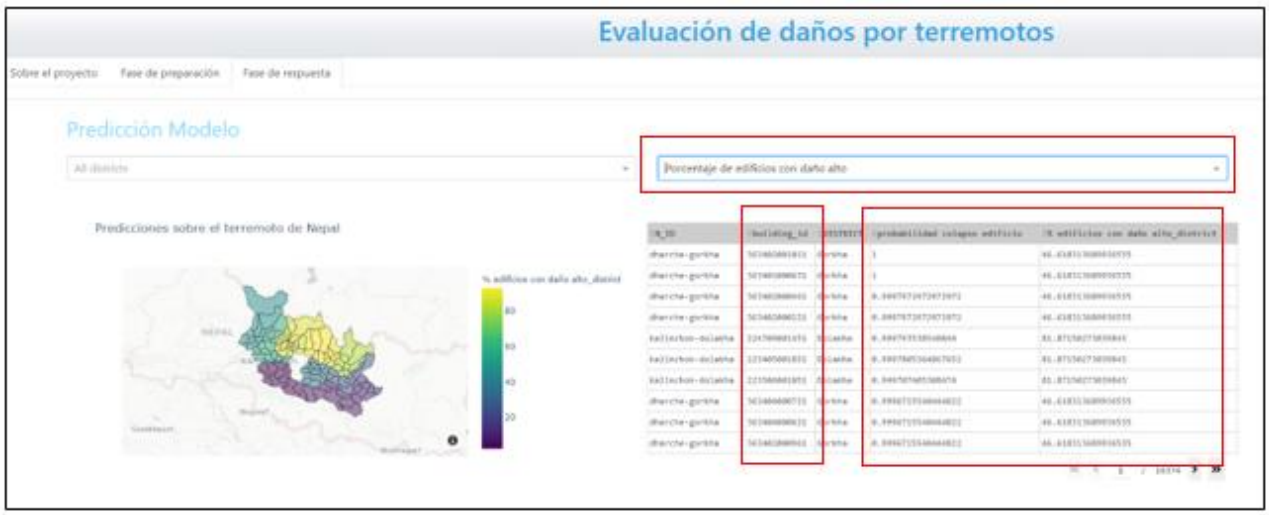


Ilustración 73 Ilustración 73 Ejemplo pestaña fase respuesta

3.7.2 NAVEGACIÓN A TRAVÉS DEL DASHBOARD

A continuación, se incluyen una serie de ejemplos de navegación aplicando distintos filtros:

1.- **Fase de preparación:** en el ejemplo se ha seleccionado el distrito "*Rasawa"* y la variable *"Edad media del cabeza de familia".*

Tal y como se observa en la imagen, el mapa representa solamente el distrito seleccionado. La tabla se ha actualizado mostrando los 5 municipios del distrito y la variable se corresponde a la Edad media del cabeza de familia en cada uno de ellos.

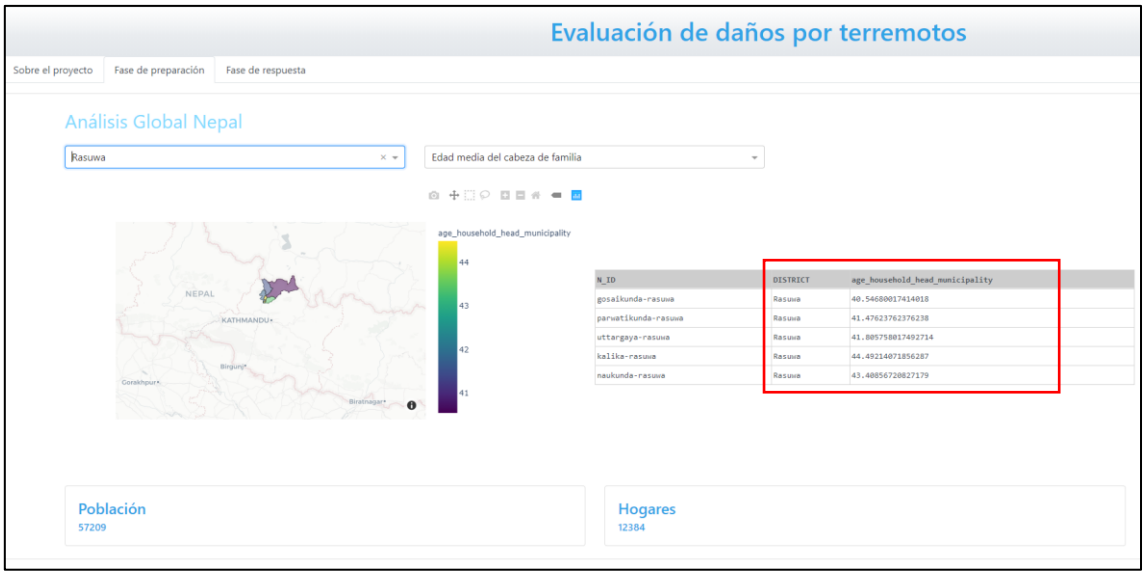


Ilustración 74 Ejemplo pestaña fase preparación con filtro distrito

2.-**Fase de respuesta**: No se ha seleccionado ningún distrito en particular, así que se observan todos los distritos afectados. Se ha filtrado por "*porcentaje de edificios con daño alto*". El mapa muestra que las zonas amarillas tienen el porcentaje de daños más elevado.

Además, la tabla aporta información detallada de los edificios, por ejemplo, el edificio con building_id *363402000461* tiene una probabilidad de 0.99979 de colapso.

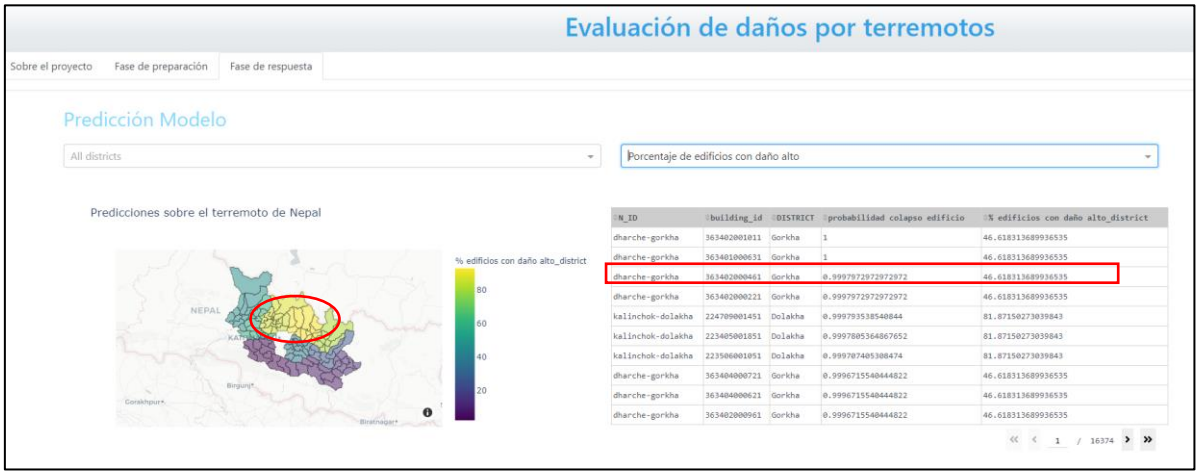


Ilustración 75 Ejemplo pestaña fase respuesta sin filtro distrito y porcentaje alto

3.-**Fase de respuesta**: En este caso, se ha seleccionado el distrito "*Nuwakot*" y porcentaje de edificios con daño bajo.

Este ejemplo muestra una funcionalidad adicional del mapa: posando el cursor sobre el municipio, se obtiene la información de la variable seleccionada. En el ejemplo, se observa que el 84.9% de los edificios del municipio "*Chisankhugadi*" tienen daños bajos.

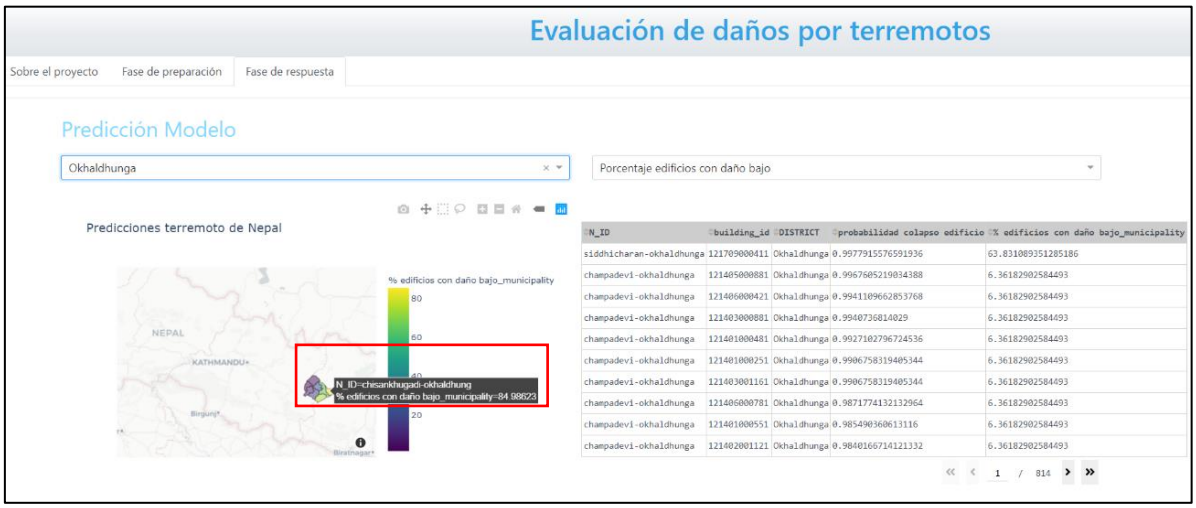


Ilustración 76 Ejemplo pestaña fase respuesta con filtro distrito e información del municipio

4 DEEP LEARNING: TRATAMIENTO DE IMÁGENES

En la aplicación de Deep Learning, se han utilizado Redes Neuronales Convolucionales (CNN). Se trata de un algoritmo utilizado en Aprendizaje Automático para el procesado de imágenes, es decir darle una visión al ordenador. Con ello se desea conseguir que el algoritmo sea capaz de clasificar una imagen según el daño que "vea".

4.1 CONJUNTO DE DATOS

La recolección de imágenes para el experimento ha sido a través de la página web CrisisNLP[24]. Estos recursos están puestos a disposición para ayudar a investigadores y técnicos en el avance del desarrollo de nuevos modelos computacionales, técnicas innovadoras y sistemas útiles para la ayuda humanitaria.

Son imágenes pertenecientes al Terremoto de Nepal de 2015, y están etiquetadas en 3 categorías de daños:" mild*", "none" y "severe"*. El conjunto de datos contiene un total de 10.962 imágenes divididas en 718, 2.500 y 7.744 respectivamente.

Se partía inicialmente de un conjunto de imágenes mucho mayor, pero debido a que había un número elevado de imágenes que no estaban en buen estado, finalmente se redujo a 10.962 imágenes.

En el análisis de imágenes se ha tenido las siguientes dificultades:

- Las imágenes de desastres naturales no son imágenes bien definidas como un perro, gato, hombre, etc… Son imágenes que se tienen que interpretar en su conjunto y por ello dificulta aún más la elección de un algoritmo que sea capaz de interpretar esas circunstancias en los datos.

- Mucho ruido en el conjunto de datos y problemas con imágenes corruptas, ya que se ha encontrado con un gran número de imágenes inservibles, teniendo que limpiar y eliminar un gran número para poder obtener un resultado aceptable.

- La interpretación del daño en las imágenes tiene un carácter subjetivo, por lo que esta subjetividad humana se le traslada al algoritmo. Esto se ve reflejado a la hora de que el algoritmo clasifique y tenga "dudas" en distinguir cuando es una clase y cuando es otra.

La clasificación de un daño u otro se basa en la magnitud del daño en cuanto a la destrucción física que refleja las imágenes. Es decir, se está buscando un daño estructural como edificios derrumbados, carreteras levantadas, puentes partidos, etc… Dentro de cada categoría de daño encontramos:

- *Daños severos:* Imágenes que muestran una destrucción importante de una infraestructura. Por ejemplo, un edificio no habitable o no utilizable, un puente no cruzable o una carretera no conducible.

- *Daños leves:* Daños generalmente superiores a los menores, con hasta el 50% de la infraestructura. Por ejemplo, un edificio sin techo, o un puente que aún puede ser usado.

[24] *https://crisisnlp.qcri.org/*

- *Poco o ningún daño:* Imágenes que muestran una infraestructura libre de daños estructurales.

4.2 PROCESADO Y APLICACIÓN DE ALGORITMOS PARA TRATAMIENTO DE IMÁGENES

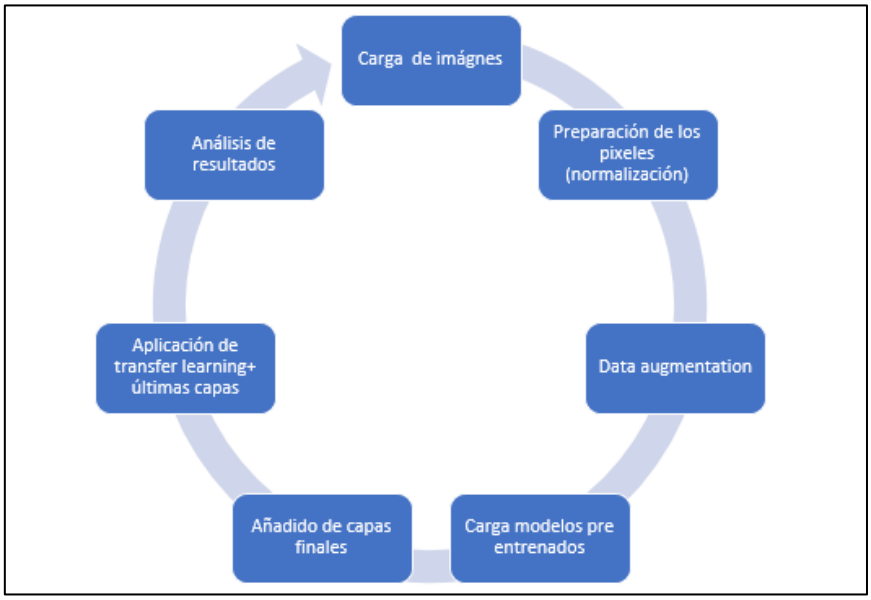


Ilustración 66 Proceso de tratamiento de imágenes

El procesado de imágenes se ha desarrollado en un notebook de Collaboratory. Esta herramienta ha permitido ejecutar y programar Python en el navegador local, sin necesidad de una configuración previa, pudiendo acceder a GPUs de forma gratuita y pudiendo compartir los notebooks de forma fácil y sencilla.

El proceso se inicia con la carga de imágenes desde dos directorios divididos en train y test, dentro de cada uno está subdividido en subdirectorios denominados como las clases "*none*", "*mild*" y "*severe*". Los números son los siguientes:

- TRAIN consta de 8.953 imágenes: "*mild*" 654, "*none*" 1.929 y "*severe*" 6.370
- TEST consta de 2.079 imágenes: "*mild*" 134, "*none*" 571, y "*severe*" 1.374

4.2.1 NORMALIZACIÓN

Cada imagen tiene una dimensión distinta por lo que se debía pasar todas las imágenes a una misma dimensión.

Se sabe que los pixeles de cada imagen no tienen signo, son enteros y se mueven en el rango entre sin color y con color, es decir de 0 a 255.

Un buen punto de partida es normalizar los valores de los píxeles, por ejemplo, reajustarlos al rango [0,1]. Esto implica por un lado convertir el tipo de datos de enteros sin signo a decimales, y luego dividir los valores de los píxeles por el valor máximo.

El motivo es que la red toma como entrada los pixeles de cada imagen, y esta entrada tiene que ser igual para todas las imágenes. Ejemplo: una imagen con 35×35 pixeles de alto y ancho

equivale a 1225 neuronas. En este caso también hay imágenes a color por lo que se necesita también los 3 canales de color (red, green, blue) 35x35x3 = 3675 neuronas de entrada. Esta sería la capa de entrada, pero multiplicada por todas las imágenes.

4.2.2 DATA AUGMENTATION

El data augmentation implica hacer copias de las imágenes con pequeñas modificaciones, pero iguales en esencia, lo que permite a la red desenvolverse mejor en la fase de inferencia.

Esto tiene un efecto regularizador ya que amplía el conjunto de datos de entrenamiento y permite que el modelo aprenda las mismas características, aunque de manera más generalizada.

Hay muchos tipos de data augmentation que podrían aplicarse. De esta forma se está evitando que la red acabe memorizando la imagen si se está entrenando demasiado, mediante la aplicación de transformaciones de forma aleatoria, cada vez que se vuelva a introducir la imagen a la red. Así se conseguirá que la red no sobreajuste y generalice mejor.

Ejemplos de transformaciones son voltear la imagen en horizontal/ vertical, rotar la imagen X grados, recortar, añadir relleno, redimensionar, etc.

4.2.3 CARGA DE MODELOS PRE-ENTRENADOS

En el experimento se ha utilizado tres modelos “Resnet50”, “InceptionV3” y “MobileNet50”, siendo “Resnet50” e “InceptionV3” las que mejor resultado ha obtenido. A continuación, se indica una breve introducción de cada modelo, extraído de deeplearningitalia[25]

Resnet50

“ResNet, nació de una simple observación: *“¿por qué agregar más capas a las redes neuronales profundas no mejora la precisión, o incluso empeora?”*

Intuitivamente, las redes neuronales más profundas no deben funcionar peor que las superficiales, o al menos no durante el entrenamiento cuando no existe el riesgo de overfitting. Tomando una red de ejemplo con *n* capas que alcanzan cierta precisión, como mínimo, una red con *n + 1* capas deberían ser capaz de alcanzar el mismo grado de precisión copiando las primeras n capas y ejecutando un mapeo de identidad para la última capa. Del mismo modo, las redes de *n + 2, n + 3* y *n + 4* capas pueden, con el mismo método, obtener la misma precisión. Sin embargo, a medida que la profundidad de la red crece esto no siempre es cierto.

InceptionV3

Si “ResNet” se concentra en la profundidad, “Inception Family” se enfoca en la extensión. Los desarrolladores de Inception estaban interesados en la eficiencia computacional de entrenar redes más grandes. En otras palabras: *“¿cómo se podría aumentar el ancho de las redes neuronales sin exceder la capacidad computacional de una computadora?”* [17]

4.2.4 CAPAS ÚLTIMAS AÑADIDAS

Como punto de partida del proceso de entrenamiento, y para adaptarlo a este problema, se importan los pesos de las capas reutilizadas.

[25] www.deeplearningitalia.com/guia-para-arquitecturas-de-redes-profundas/

Las 4 capas se han adaptado al modelo “Resnet50” (mejor ROC). En la última capa es donde se especifica las neuronas de salida, en nuestro caso son 3 y la función de activación para multiclase *softmax (se han probado distintas arquitecturas y esta última arquitectura es la que mejor resultado ha dado)*

4.2.5 ANÁLISIS DE RESULTADOS Y MÉTRICAS

Las dos métricas más importantes que se utilizan son:

“Sensibilidad”: estructuras destruidas acertadas
“Especificidad”: utilizada para calcular los Falsos Positivos Reales (FPR =1 -especificidad), estructuras no dañadas falsamente evaluadas como dañadas.
Con estas dos métricas se obtiene la gráfica de relación “Receiving Operator Curve (ROC)”, en la que se observa que la relación optima tiene que ser inversa: cuanto mayor es la sensibilidad menor es 1-espeficidad. También se ha tenido en cuenta la capacidad de acierto total o accuracy.

Tras la exposición de las métricas evaluadas y procesadas las imágenes con dos algoritmos, en uno obtenemos un mayor accuracy que en otro, pero también menores AUC (Area Under Curve).

InceptionV3

En este algoritmo se consigue un mayor accuracy respecto al Resnet50, consiguiendo en test un 76.5% de aciertos. Pero a la hora de acertar los casos positivos las AUC son inferiores respecto a ResNet50. Según el caso que se plantea, elegiremos uno u otro algoritmo.

```
130/130 [==============================] - 22s 168ms/step
Test loss: 0.147965207695961
Test accuracy: 0.765271782875061
```

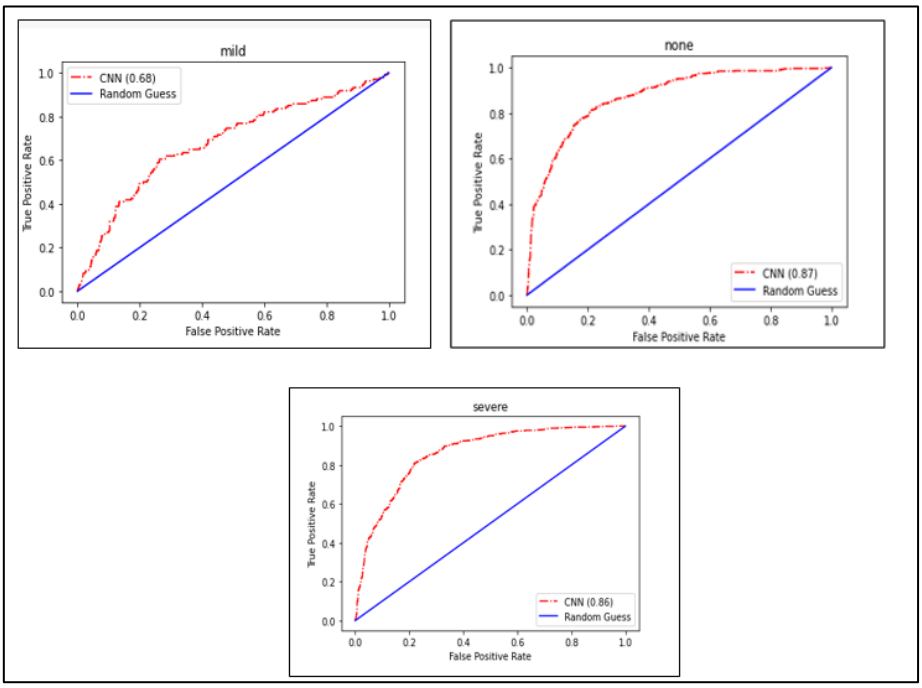


Ilustración 77 Interception V3

Resnet50

El accuracy en la ResNet50 es 74.84% frente al 76.5% de la red InceptionV3, pero las AUC son mayores. Por lo que para el caso de uso que se plantea, interesa que las AUC sean mayores y en especial para los casos severos. Esto se debe, a que el modelo es capaz de identificar mejor los casos de infraestructuras con daños severos.

```
130/130 [==============================] - 22s 165ms/step
Test loss: 0.11696960777044296
Test accuracy: 0.7484367489814758
```

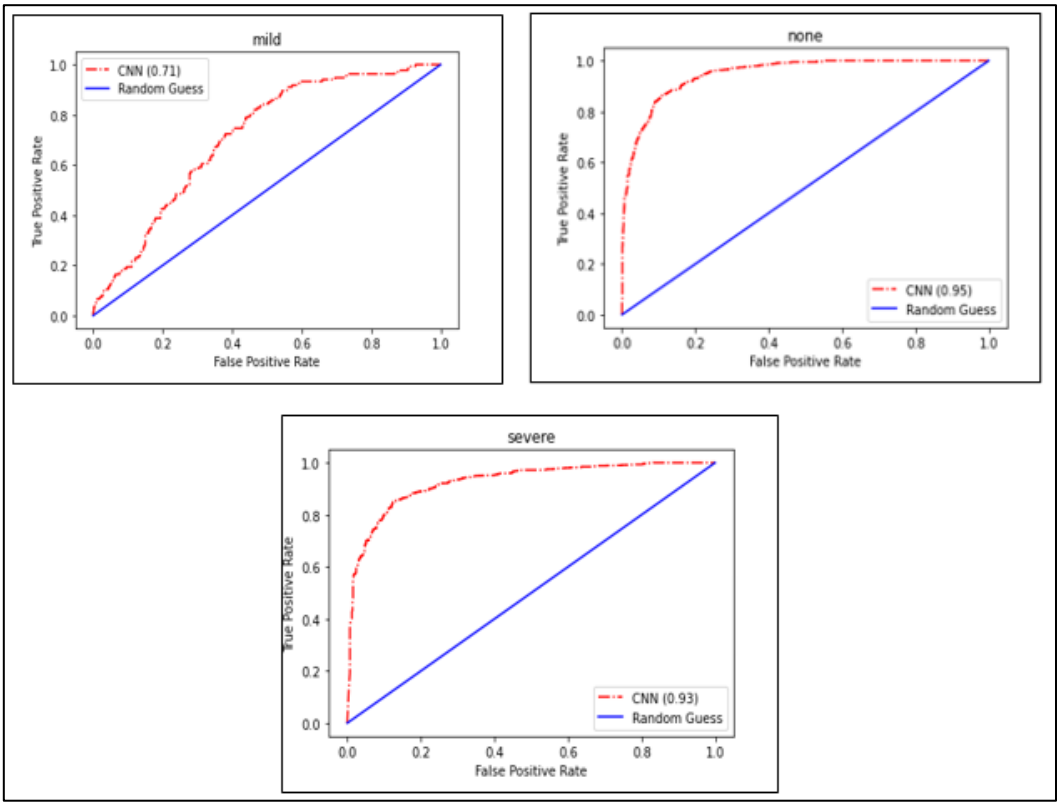


Ilustración 78 Restnet50

Para el caso de uso planteado en el procesado de imágenes, el modelo funciona, ya que es capaz de distinguir mucho mejor los casos en los que una infraestructura se va a derrumbar frente al acierto general.

5 CONCLUSIÓN

Las catástrofes naturales están perdiendo su carácter de fenómeno extraordinario. En los últimos años, el número de desastres ha aumentado siendo cada vez más devastadores, donde el 90% de ellos están relacionados con el cambio climático. Por tanto, se hace necesario que los países tomen medidas, especialmente para proteger a los más vulnerables.

Determinar a la mayor brevedad el daño que un terremoto ha causado en edificios, es un paso crítico en la respuesta al evento y en los planes de recuperación. El daño del edificio puede ir desde una grieta superficial hasta el colapso completo dependiendo de sus características: tipo de construcción, tamaño, así como su localización. Además, la distribución del impacto de un terremoto a lo largo del territorio afectado hace que la evaluación de los daños en los edificios sea un proceso complejo y caro en términos de tiempo y recursos.

El objetivo planteado en este TFM debe entenderse como una primera aproximación, cuyo fin es demostrar el potencial de la ciencia de datos en el ámbito de las catástrofes naturales y cómo se podría mejorar el proceso de respuesta actual si se integrara en los protocolos de actuación existentes.

Dos características de esta prueba de concepto son la extensibilidad y adaptabilidad. Por un lado, a pesar de que los datos utilizados pertenecen a un terremoto concreto de un país concreto, se podría extender a otro tipo de desastres naturales como huracanes, inundaciones… siempre que haya impactos en edificios.

Es adaptable en cuanto que cada país puede decidir qué variables son más relevantes para el modelo, dado que la casuística de cada región varía, y por otro lado definir esos umbrales de sensibilidad que identificarán el nivel de daño y por tanto el despliegue de sus efectivos en caso de desastre natural.

Además, este modelo generaría un alto impacto social, ya que, con una inversión mínima por parte de los gobiernos u otras entidades responsables, se conseguiría salvar la vida de muchas personas.

En cuanto a las limitaciones, el modelo debe ser entrenado con datos reales en las horas próximas en las que el evento tiene lugar. Por tanto, es necesario que se produzca un primer evento, y por otro lado se debe realizar ese muestreo con el que entrenar el modelo y obtener así las predicciones de colapso de la totalidad de edificios.

Sin embargo, el tratamiento de imágenes realizadas por particulares, como segunda solución propuesta en este TFM, debería ser integrada en una versión avanzada del modelo. Ya sea a través del envío a una plataforma concreta o bien en RRSS. Así, la fase de muestreo podría hacerse de una manera más rápida y barata.

Además, en un futuro cercano, esta recogida de datos, actualmente bastante manual y hecha por personas en su gran mayoría, podrá ser más sencilla y rápida con la incorporación de tecnología IoT en los edificios, uso de drones, imágenes de satélites, que en algunos lugares ya se está desarrollando.

En cuanto a la ejecución del TFM, nos hemos encontrado con grandes dificultades a la hora de consolidar las diferentes fuentes de información y sobre todo entenderlas. Ha habido una labor muy importante de limpieza y preprocesado del conjunto de datos, dedicándole en torno a un 70% del tiempo total. El desarrollo del dashboard también ha sido un proceso complejo, ya que

era una materia en la que no se había entrado en profundidad anteriormente. Además, la situación actual ha hecho que el trabajo en equipo haya sido 100% virtual lo cual en algunos momentos ralentizaba el avance del proyecto. Por ese motivo, hemos tenido que manejar la comunicación de una manera efectiva y estructurada, demostrando nuestra dedicación y compromiso personal en beneficio de todo el grupo.

Nos hemos encontrado con muchas situaciones en las que no sabíamos cómo resolver una problemática concreta, teniendo que estudiar y recurrir a numerosos soportes disponibles en la web. Además, hemos contado con el apoyo de Manuel que siempre ha estado dispuesto a resolver nuestras dudas. Ha sido un proceso duro pero gratificante, no solo a nivel académico sino también a nivel personal.

6 REFERENCIAS

[1] Informe Marco de Acción de Hyogo ONU.

https://www.eird.org/publicaciones/2909_OCHADisasterpreparednesseffectiveresponseSPA.pdf

[2] Driven data portal

https://www.drivendata.org/

[3] Artificial Intelligence for digital response

http://aidr.qcri.org/

[4] Natural disasters are increasing in frequency and ferocity. Here's how AI can come to the rescue

https://www.weforum.org/agenda/2020/01/natural-disasters-resilience-relief-artificial-intelligence-ai-mckinsey/

[5] 2015 Nepal Open Data Portal

https://eq2015.npc.gov.np/#/

[6] CrisisNLP Portal

https://crisisnlp.qcri.org/

[7] Mapa fronteras administrativas de Nepal

https://drklrd.github.io/adminboundaries-np/?fbclid=IwAR0XPuHCLS0JoDSj_KbPV7DLY8WqfzF1UgGDKC4ckO45YlE70-jNd8QUfvI

[8] The education system of Nepal described and compared with Dutch system

www.nuffic.nl/sites/default/files/2020-08/education-system-nepal.pdf

[9] Nepal: economía y demografía

datosmacro.expansion.com/paises/nepal#:~:text=Su%20capital%20es%20Katmand%C3%BA%20y, habitante312%24%20dolares%20por%20habitante

[10] Mangalathu, Sujith & Sun, Han & Nweke, Chukwuebuka & Yi, Zhengxiang & Burton, Henry. (2019). Classifying Earthquake Damage to Buildings Using Machine Learning. Earthquake Spectra. 36. 10.1177/8755293019878137.

[11] Cienciadedatos.net

https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting

[12] *https://www.datasource.ai/*

[13] Mangalathu, Sujith & Sun, Han & Nweke, Chukwuebuka & Yi, Zhengxiang & Burton, Henry. (2019). Classifying Earthquake Damage to Buildings Using Machine Learning. Earthquake Spectra. 36. 10.1177/8755293019878137

[14] The education system of Nepal described and compared with the Dutch system

https://www.nuffic.nl/sites/default/files/2020-08/education-system-nepal.pdf

[15]. Dat T. Nguyen, Ferda Ofli, Muhammad Imran, Prasenjit Mitra. Damage Assessment from Social Media Imagery Data During Disasters

[16] Dr. Mayank Kejriwal. Crisis management: Using Artificial Intelligence to help save lives

[17] RFE feature selection in python

https://machinelearningmastery.com/rfe-feature-selection-in-python/

[18] What is a confusion matrix

https://www.unite.ai/what-is-a-confusion-matrix/

[19] The ultimate guide to binary classification metrics

https://towardsdatascience.com/the-ultimate-guide-to-binary-classification-metrics-c25c3627dd0a

[20] Algoritmos Naive Bayes: Fundamentos e Implementación

*https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307*f

[21] Cómo funciona gradient boosting

https://spainml.com/blog/como-funciona-gradient-boosting/

[22] Adaboost

https://fhernanb.github.io/libro_mod_pred/adaboost.html

[23] Árboles de decision random forest, gradient boosting y C5.0

https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting

[24] XGboost en R

https://rpubs.com/jboscomendoza/xgboost_en_r

[25] A guide on XGBoost hyperparameters tunning

https://www.kaggle.com/prashant111/a-guide-on-xgboost-hyperparameters-tuning

[26] Guia para arquitecturas de redes profundas

www.deeplearningitalia.com/guia-para-arquitecturas-de-redes-profundas/

[27] Dash Plotly

https://dash.plotly.com/

[28] Dash boostrap components

https://dash-bootstrap-components.opensource.faculty.ai/

[29] Choropleth maps

https://plotly.com/python/choropleth-maps/

7 ANEXO I: LISTADO DE JUPYTER NOTEBOOKS

A continuación, se detallan los Jupyter Notebooks generados a lo largo del presente TFM:

- Paso_1_Cruce de Tablas y missing.ipynb
- Paso_2_Visualizaciones y agrupamientos_sin building id.ipynb
- Paso_2_ Visualizaciones y agrupamientos_con building id.ipynb
- Paso_3_Selección de variables_modelos.ipynb
- Paso_4.1_Modelos_Matriz_confusión_COLINEALIDAD.ipynb
- Paso_4.2_Mejor_Modelo_Predicciones_COLINEALIDAD.ipynb
- Paso_5.1_Tablas_Mapa.ipynb
- Paso_5.2_ Visualizacion_PoC_RF+COLINEALIDAD.ipynb
- 6.1 TFM_Earthquake_con_transfer_learning_Resntet50.ipynb
- 6.2 TFM_Earthquake_con_transfer_learning_MobileNetV2.ipynb
- 6.3 TFM_Earthquake_con_transfer_learning_InceptionV3.ipynb