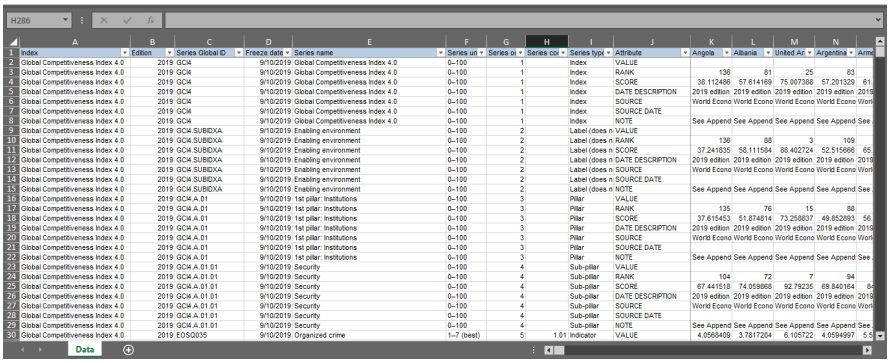
Análisis de sistemas de infraestructura Taller 2

Karol Rivera201815009

26 de septiembre de 2023

Kmeans y regresiones lineales

Para comenzar se debe ingresar a la página del Global Competitiveness Index 4.0 (GCI) y descargar la base de datos. El archivo obtenido es un excel el cual tiene 4 páginas, de las cuales la única de interés es la que contiene los datos, nombrada *data*. Por lo tanto, sobre el excel se eliminan las otras 3 ventanas. Adicionalmente, en la hoja *data* se eliminan las primeras 3 filas.



Index	Edition	Series Global ID	Freeze date	Series name	Series un	Series sc	Series col	Series type	Attribute	Angola	Albania	United Ar	Argentina	Arm
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	VALUE					
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	RANK	136	81	25	83	
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	SCORE	38.112486	57.614169	75.007388	57.201329	61
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	DATE DESCRIPTION	2019 edition	2019 edition	2019 edition	2019 edition	2019
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	SOURCE	World Econo	World Econo	World Econo	World Econo	Worl
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	SOURCE DATE					
Global Competitiveness Index 4.0	2019	GCI4	9/10/2019	Global Competitiveness Index 4.0	0–100	1		Index	NOTE	See Append	See Append	See Append	See Append	See
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	VALUE					
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	RANK	136	88	3	109	
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	SCORE	37.241835	58.111584	88.402724	52.515666	65.
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	DATE DESCRIPTION	2019 edition	2019 edition	2019 edition	2019 edition	2019
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	SOURCE	World Econo	World Econo	World Econo	World Econo	Worl
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	SOURCE DATE					
Global Competitiveness Index 4.0	2019	GCI4.SUBIDXA	9/10/2019	Enabling environment	0–100	2		Label (does n	NOTE	See Append	See Append	See Append	See Append	See
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	VALUE					
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	RANK	135	76	15	88	
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	SCORE	37.615453	51.874614	73.258837	49.852893	58.
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	DATE DESCRIPTION	2019 edition	2019 edition	2019 edition	2019 edition	2019
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	SOURCE	World Econo	World Econo	World Econo	World Econo	Worl
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	SOURCE DATE					
Global Competitiveness Index 4.0	2019	GCI4.A.01	9/10/2019	1st pillar: Institutions	0–100	3		Pillar	NOTE	See Append	See Append	See Append	See Append	See
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	VALUE					
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	RANK	104	72	7	94	
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	SCORE	67.441518	74.059868	92.79235	69.840164	84
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	DATE DESCRIPTION	2019 edition	2019 edition	2019 edition	2019 edition	2019
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	SOURCE	World Econo	World Econo	World Econo	World Econo	Worl
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	SOURCE DATE					
Global Competitiveness Index 4.0	2019	GCI4.A.01.01	9/10/2019	Security	0–100	4		Sub-pillar	NOTE	See Append	See Append	See Append	See Append	See
Global Competitiveness Index 4.0	2019	EOSQ035	9/10/2019	Organized crime	1–7 (best)	5	1.01	Indicator	VALUE	4.0568409	3.7617204	6.105722	4.0594997	5.5

Figura 1: Base de datos del GCI en excel luego de modificaciones para obtener formato deseado.

Para comenzar con el procesamiento de los datos es necesario importar algunas librerías para trabajar en el entorno dado por *jupyter notebook* en lenguaje de programación *python*:

- Numpy: Tiene como función principal el uso de arreglos tipo arrays numéricos y el uso de operaciones matemáticas.
- Pandas: Se utiliza para manejo de datos y realizar funciones estadísticas.
- Matplotlib: Tiene como función principal la visualización de datos. De esta librería se usa en particular el módulo pyplot.
- Sklearn: Herramienta para el análisis de datos, está basada en herramientas de numpy, scipy y matplotlib. Tiene funciones relacionadas con: clasificación, regresión, clustering, reducción de dimensiones, selección de modelos y preprocesamiento. En este caso se usará el módulo *preprocesing* el cual tiene funciones que permiten realizar una transformación sobre los datos. También se usará el módulo *decomposition*, en partícular la función *PCA* para realizar un análisis de componentes principales. Adicionalmente se usará el módulo *cluster* y la función *KMeans* para tratar de realizar clusterización. Otro módulo a usar es *linear_model*, en particular la función *LinearRegression* para realizar regresiones. También se usará el módulo *metrics* para calcular algunas estadísticas sobre los modelos desarrollados. Adicionalmente, otra librería a usar será *model_selection*, en particular la función *train_test_split* para dividir un conjunto de datos en un grupo de entrenamiento y un grupo de prueba. Finalmente, de esta librería también se usará el módulo *datasets*, en particular la función *make_classification*.
- Time: tiene funciones relacionadas con tiempo.

- Tensorflow: Facilita la creación de modelos de aprendizaje automático. Se usará en particular *keras* que está orientada al aprendizaje profundo.

Para comenzar se deben importar los datos mediante la función *read_excel* de la librería pandas. A continuación, se deben eliminiar algunas columnas que no contienen información importante usando la función *drop* de la librería pandas.

```
#Cargar los datos

GCI= pd.read_excel(r'WEF_GCI_4.0_2019_Dataset.xlsx')

#Eliminación de columnas con información sobrante

GCI.drop(['Index','Series Global ID','Freeze date','Series units','Series order','Series code (if applicable)'], axis=1, inplace=

GCI #Para revisar visualmente que esta pasando con nuestro arreglo de datos
```

Figura 2: Importe de datos y eliminación de columnas no relevantes.

Por otra parte, se desean construir 2 dataframes. El primero debe contener el valor del índice global de conectividad en infraestructura del año 2019. Para obtener esta información se debe usar la columna *Edition == 2019* para filtrar los datos por año, posteriormente la columna *Series name == 2nd pillar: Infraestructure* para seleccionar el indicador relacionado con infraestructura, y por último usar la columna *Attribute == SCORE* para seleccionar los valores numéricos. Además, se debe eliminar del registro todos aquellos grupos que no corresponden a países. Para finalizar, el dataframe debe transponerse de modo que los países sean las filas y el valor del segundo pilar la columna.

```
# Filtrar Dataframe para el año 2019

GCI_2019=GCI[GCI['Edition']==2019].copy()
GCI_2019.drop(['Sample average','High-income','Upper-middle-income','Lower-middle-income','Low-income','East Asia and Pacific','U
# Filtrar Dataframe para el pilar de Infraestructura

GCI_2019_Infrastructurei=GCI_2019[GCI_2019['Series name']=='2nd pillar: Infrastructure'].copy() #Selecciono la serie adecuada
GCI_2019_Infrastructurei=GCI_2019_Infrastructurei[GCI_2019_Infrastructurei['Attribute']=='SCORE'].copy() #Selecciono el atributo
GCI_2019_Infrastructurei.drop(['Edition','Series type','Attribute'], axis=1, inplace=True)
GCI_2019_Infrastructurei.set_index('Series name',inplace=True)
GCI_2019_Infrastructure=GCI_2019_Infrastructurei.transpose()
GCI_2019_Infrastructure.index.names = ['Country Name']
GCI_2019_Infrastructure.columns=['2nd pillar: Infrastructure']
GCI_2019_Infrastructure #Para revisar visualmente que esta pasando con nuestro arreglo de datos
```

Figura 3: Ajuste de la base de datos para generar el primer dataframe.

El segundo dataframe se construye a partir de los indicadores utilizados para generar el pilar de infraestructura. Por lo tanto, nuevamente se filtra por el año utilizando *Edition == 2019*, y se utiliza el nombre del indicador igualado a la columna *Series name*:

- *Series name* = Road connectivity.
- *Series name* = Quality of road infraestructure.
- *Series name* = Railroad density.
- *Series name* = Efficiency of train services.
- *Series name* = Airport connectivity.
- *Series name* = Efficiency of air transport services.
- *Series name* = Liner shipping connectivity.
- *Series name* = Efficiency of seaport services.
- *Series name* = Electricity supply quality.
- *Series name* = Exposure to unsafe drinking water.
- *Series name* = Reliability of water supply.

Finalmente, en todos los casos se utiliza *Attribute == SCORE* para seleccionar los valores numéricos. Finalmente, se eliminan las columnas sin información relevante. Una vez filtrada toda la información, se obtienen arrays para cada uno de los indicadores. Estos se unen en un solo dataframe utilizando la función *merge()* utilizando el nombre del país como punto de unión.

```
df_indicadores=pd.merge(lista_indicadores[0], lista_indicadores[1], on='Country Name')

for i in range(2,len(lista_indicadores)):
    df_indicadores=pd.merge(df_indicadores, lista_indicadores[i], on='Country Name')

df_indicadores #Para revisar visualmente que esta pasando con nuestro arreglo de datos
```

Figura 4: Paso final para generar el segundo dataframe.

A continuación, se debe realizar el procesamiento básico de los datos. Teniendo en cuenta que en estos datos no son series de tiempo no se pueden completar los datos faltantes ya que no existen correlaciones aparentes a partir de las cuales realizar alguna clase de interpolación o extrapolación. Por lo tanto, se decide preliminarmente que los datos faltantes de los indicadores relacionados con infraestructura ferrea indican que el país no tiene este tipo de infraestructura. Para realizar esto se utiliza el método *fillna(0)* indicando que todos los valores faltantes serán reemplazados con ceros en los indicadores relacionados con infraestructura ferrea. Para terminar, como se desea tener una base de datos con países que tengan toda la información de indicadores completa, se procede a eliminar todos los países que tengan datos incompletos con respecto a algún indicador, usando la función *dropna*, y ajustando los parámetros *how=any* y *inplace=True*.

```
# A juicio del analista de de los datos: se considera que si no hay registro de indicadores relacionados a trafico ferreo, el pa

df_indicadores['Railroad density'] = df_indicadores['Railroad density'].fillna(0)
df_indicadores['Efficiency of train services'] = df_indicadores['Efficiency of train services'].fillna(0)
df_indicadores['Liner shipping connectivity'] = df_indicadores['Liner shipping connectivity'].fillna(0)
df_indicadores['Efficiency of seaport services'] = df_indicadores['Efficiency of seaport services'].fillna(0)

#Eliminar los países que no contienen información de algun indicador

df_indicadores.dropna(how='any', inplace=True)

df_indicadores #Para revisar visualmente que esta pasando con nuestros arreglos de datos
```

Figura 5: Completar los datos de infraestructura férrea y eliminar países incompletos.

Ahora se procede a realizar clusterización utilizando el algoritmo de kmeans sobre el dataframe con los indicadores que permiten construir el puntaje en el pilar de infraestructura. Para aplicar este algoritmo se debe realizar una normalización a los datos de modo que se creen variables nuevas sin unidades y que conserven el orden en una escala relativa:

$$Z_i = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

```
Datos_pilar_infr_Array=df_indicadores.to_numpy() # Array de los indicadores
Datos_pilar_infr_Array=(Datos_pilar_infr_Array-np.min(Datos_pilar_infr_Array))/(np.max(Datos_pilar_infr_Array,axis=0)-np.min(Datc
Datos_pilar_infr=pd.DataFrame(Datos_pilar_infr_Array,index=df_indicadores.index)
Datos_pilar_infr.columns = lista_nombres_ind
Datos_pilar_infr #Para revisar visualmente que esta pasando con nuestro arreglo de datos
```

Figura 6: Normalización de los datos para uso en el algoritmo de clusterización.

Para continuar, el algoritmo se ejecuta al instanciar la función, definirle la cantidad de clusters y pasarle los datos a clasificar. Como se quiere clasificar en 3 categorías de acuerdo la información de infraestructura, en el algoritmo de clusterización se define la cantidad de clusters como 3, mientras que los datos de ingreso son los datos normalizados de los indicadores. Vale la pena mencionar que *kmeans* es un algoritmo de aprendizaje no supervisado para clasificar en k categorías basándose en la similitud entre datos, generalmente basándose en la distancia a los clusters.

Una vez entrenado el algoritmo de clusterización, se le pasan nuevamente los datos del pilar de infraestructura para que realice la predicción del cluster al que pertenece cada país. Con el resultado de la predicción se agrupan los datos:

- Cluster 0: Todos los resultados cuya predicción sea menores o iguales a cero.
- Cluster 1: Todos los resultados que son mayores a cero y menores a dos.
- Cluster 3: Todos los resultados cuta predicción sea mayor o igual a dos.

```
kmeans = KMeans(n_clusters=3).fit(Datos_pilar_infr_Array)

centroids = kmeans.cluster_centers_

df_cluster=pd.DataFrame(kmeans.predict(Datos_pilar_infr_Array),index=Datos_pilar_infr.index, columns=['cluster'])

df_cluster_cero = df_cluster.drop(df_cluster[df_cluster['cluster']>0].index)
df_cluster_uno= df_cluster.drop(df_cluster[df_cluster['cluster']==0].index)
df_cluster_uno=df_cluster_uno.drop(df_cluster_uno[df_cluster_uno['cluster']==2].index)
df_cluster_dos= df_cluster.drop(df_cluster[df_cluster['cluster']<2].index)

df_cluster_cero #Para revisar visualmente que esta pasando con nuestro arreglo de datos
```

Figura 7: Ejecución del algoritmo kmeans.

Una primera observación con respecto a los clusters parece indicar que los valores del pilar de infraestructura se encuentra correlacionados con el cluster obtenido, esto puede deberse principalmente a que los indicadores usados para clasificar los países alimentan el cálculo del pilar de infraestructura . En otras palabras, parece que tener valores más bajos en el pilar de infraestructura se correlaciona con estar en el cluster cero, mientras que valores medios en el cluster 1 y valores altos en el cluster 2. Si se calculan los valores límite asociados con cada cluster se obtienen los resultados de la figura 8.

```
#Valores mínimos y máximos

max_cero=max(Comparacion_cero['2nd pillar: Infrastructure'])
min_cero=min(Comparacion_cero['2nd pillar: Infrastructure'])

max_uno=max(Comparacion_uno['2nd pillar: Infrastructure'])
min_uno=min(Comparacion_uno['2nd pillar: Infrastructure'])

max_dos=max(Comparacion_dos['2nd pillar: Infrastructure'])
min_dos=min(Comparacion_dos['2nd pillar: Infrastructure'])
```

```
print('El primer cluster está conformado por países con GCI en el rango', min_cero,max_cero, '\n','El se
```

```
El primer cluster está conformado por países con GCI en el rango 26.878243786231074 66.85168818474732
 El segundo cluster está conformado por países con GCI en el rango 55.554451438923365 88.48554173222215
 El tercer cluster está conformado por países con GCI en el rango 70.33613446834686 94.33942042551192
```

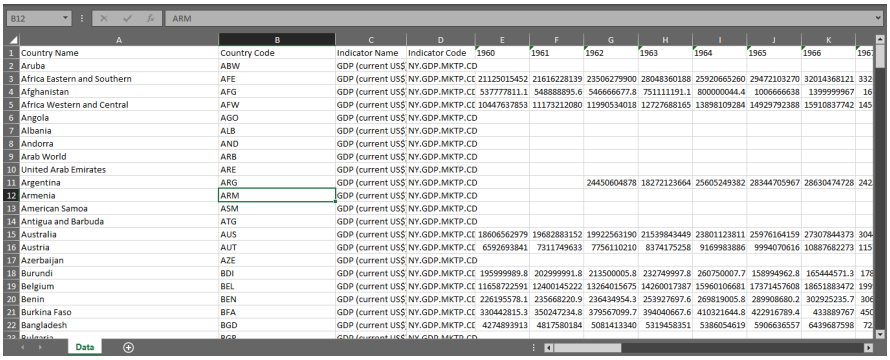
Figura 8: Valores límite para los clusters encontrados con el algoritmo kmeans.

Estos límites muestran que hay zonas donde se puede pertenecer a uno u otro cluster, lo que podría indicar que existen algunos indicadores que son más influyentes que otros. Esto quiere decir que el valor global de infraestructura obtenido no refleja totalmente la influencia de los indicadores más influyentes para la clasificación, generando así que no exista un valor límite entre cada cluster en el pilar de infraestructura.

Otra observación es que se tienen similitudes en condiciones socioeconómicas y de desarrollo entre los países obtenidos en cada cluster. Por ejemplo, en el cluster 0 se obtienen países reconocidos por ser de tercer mundo o subdesarrollados, ubicados en África, reconocidos por estar subdesarrollados con respecto a la región en la que se encuentran o que han padecido problemas sociales o económicos (P.E. Venezuela, Madagascar, Haiti, Senegal, Nigueria, etc.). En el cluster 2, por el contrario, se encuentran países reconocidos como de primer mundo, desarrollados y con un nivel socioeconómico destacado (P.E. Francia, España, Estados Unidos, Taiwan, Japón, etc.).

Para continuar con el análisis, se pueden descargar los datos del Banco mundial sobre el GDP. Estos datos nuevamente requieren un procesamiento previo sobre su estructura en excel de modo que

solo se conserve la hoja *Data*, en la cual se eliminan las primeras 3 filas.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	1966	1967
2	Aruba	ABW	GDP (current US$)	NY.GDP.MKTP.CD								
3	Africa Eastern and Southern	AFE	GDP (current US$)	NY.GDP.MKTP.CD	21125015452	21616228139	23506279900	28048360188	25920665260	29472103270	32014368121	332
4	Afghanistan	AFG	GDP (current US$)	NY.GDP.MKTP.CD	537777811.1	548888895.6	546666677.8	751111191.1	800000044.4	1006666638	1399999967	16
5	Africa Western and Central	AFW	GDP (current US$)	NY.GDP.MKTP.CD	10447637853	11173212080	11990534018	12727688165	13898109284	14929792388	15910837742	145
6	Angola	AGO	GDP (current US$)	NY.GDP.MKTP.CD								
7	Albania	ALB	GDP (current US$)	NY.GDP.MKTP.CD								
8	Andorra	AND	GDP (current US$)	NY.GDP.MKTP.CD								
9	Arab World	ARB	GDP (current US$)	NY.GDP.MKTP.CD								
10	United Arab Emirates	ARE	GDP (current US$)	NY.GDP.MKTP.CD								
11	Argentina	ARG	GDP (current US$)	NY.GDP.MKTP.CD			24450604878	18272123664	25605249382	28344705967	28630474728	242
12	Armenia	ARM	GDP (current US$)	NY.GDP.MKTP.CD								
13	American Samoa	ASM	GDP (current US$)	NY.GDP.MKTP.CD								
14	Antigua and Barbuda	ATG	GDP (current US$)	NY.GDP.MKTP.CD								
15	Australia	AUS	GDP (current US$)	NY.GDP.MKTP.CD	18606562979	19682883152	19922563190	21539843449	23801123811	25976164159	27307844373	304
16	Austria	AUT	GDP (current US$)	NY.GDP.MKTP.CD	6592693841	7311749633	7756110210	8374175258	9169983886	9994070616	10887682273	115
17	Azerbaijan	AZE	GDP (current US$)	NY.GDP.MKTP.CD								
18	Burundi	BDI	GDP (current US$)	NY.GDP.MKTP.CD	195999989.8	202999991.8	213500005.8	232749997.8	260750007.7	158994962.8	165444571.3	178
19	Belgium	BEL	GDP (current US$)	NY.GDP.MKTP.CD	11658722591	12400145222	13264015675	14260017387	15960106681	17371457608	18651883472	199
20	Benin	BEN	GDP (current US$)	NY.GDP.MKTP.CD	226195578.1	235668220.9	236434954.3	253927697.6	269819005.8	289908680.2	302925235.7	306
21	Burkina Faso	BFA	GDP (current US$)	NY.GDP.MKTP.CD	330442815.3	350247234.8	379567099.7	394040667.6	410321644.8	422916789.4	433889767	450
22	Bangladesh	BGD	GDP (current US$)	NY.GDP.MKTP.CD	4274893913	4817580184	5081413340	5319458351	5386054619	5906636557	6439687598	72

Figura 9: Base de datos del banco mundial sobre el GDP.

Además de las modificaciones en el banco mundial se deben eliminar algunas columnas que no aportan información redundante como lo son: 'Country code', 'Indicator name', 'Indicator code'. Esto se hace con la función *drop*. Como se trata una serie de tiempo, inicialmente se deben eliminar las series que no tengan ningún registro. Esto se realiza con la función *dropna* ajustando los parámetros *how=all* y *inplace=True*. Para continuar, y con el objetivo de completar los datos, se realiza una interpolación usando la función *interpolate* ajustando los parámetros para tener un método lineal que interpola en ambas direcciones (*method=Linear, limit_direction=both*).

```
df_GDP.dropna(how='all', inplace=True) #eliminar los paises para los cuales no existe ningún dato en la serie de tiempo
df_GDP.interpolate(method ='linear', limit_direction ='both', axis=1, inplace=True) #Interpolar
df_GDP #Para revisar visualmente que esta pasando con nuestro arreglo de datos
```

Figura 10: Base de datos del GDP cargada.

Adicionalmente, como la base de datos anterior cuenta con datos de países únicamente, en esta nueva base de datos se eliminan todas las agrupaciones. Además, se realiza la normalización de los datos de acuerdo con la ecuación previamente presentada para este procedimiento.

El siguiente paso es realizar una regresión lineal para cada uno de los clusters identificados previamente. La regresión lineal se hará entre los indicadores obtenidos de la base de datos para el pilar de infraestructura, y los datos del GDP. De este modo, las variables X de la regresión son los indicadores, y la variable Y los datos del GDP.

- X_1: Road connectivity
- X_2: Quality of road infrastructure
- X_3: Railroad density
- X_4: Efficiency of train services
- X_5: Airport connectivity
- X_6: Efficiency of air transport services
- X_7: Liner shipping connectivity
- X_8: Efficiency of seaport services
- X_9: Electricity supply quality
- X_{10}: Exposure to unsafe drinking water
- X_{11}: Reliability of water supply

- Y: GDP

Para aplicar el algoritmo de la regresión lineal se realizó una función. Inicialmente se debe realizar un merge entre los datos de los indicadores, el pilar y los datos de GDP, tomando como referencia el nombre del país (*Country name*). A continuación, se deben normalizar los valores al interior del dataframe creado con el merge. Una vez se tienen las variables normalizadas, se procede a realizar la regresión lineal. Para esto, se utiliza la función *train_test_split* de manera que se toma una muestra aleatoria de las variables X y Y para utilizarla como datos de entrenamiento de la regresión. La regresión como tal se realiza instanciando el regresor *LinearRegression()* y solicitándole realizar un fit sobre los datos de prueba. La función genera un resultado dependiendo del parámetro recibido por la función. Estas posibles respuestas se observan en la figura 11, sin embargo, para este análisis se tiene interés en los valores de los coeficientes que acompañan cada indicador, por lo que la función se llama con *'theta'*.

```
def caso_cero(tipo):

        r_s=4
        #Caso cero
        Caso_cero=pd.merge(df_cluster_cero, Datos_pilar_infr, on='Country Name', suffixes=('_left', '_right')) #Une por el nombre
        Caso_cero=pd.merge(Caso_cero, df_GDP, on='Country Name', suffixes=('_left', '_right')) #Une por el nombre del país result
        Caso_cero=Caso_cero.drop(['cluster'], axis=1)

        #Normalización
        Caso_cero_array=Caso_cero.to_numpy()
        Caso_cero_array=(Caso_cero_array-np.min(Caso_cero_array))/(np.max(Caso_cero_array,axis=0)-np.min(Caso_cero_array))
        Caso_cero=pd.DataFrame(Caso_cero_array,index=Caso_cero.index)


        list_columns=lista_nombres_ind.copy()
        list_columns.append('GDP')
        Caso_cero.columns = list_columns
        X = Caso_cero.iloc[:,0:11]
        Y = Caso_cero['GDP']

        #Regresión lineal

        X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state=r_s)
        lin_model = LinearRegression()
        lin_model.fit(X_train, Y_train)
        if tipo=='graf':
            plt.scatter(X_train['Quality of road infrastructure'],Y_train)
            plt.xlabel('Quality of road infrastructure') # Establecer el título del eje x
            plt.ylabel("GDP") # Establecer el título del eje y

        if tipo=='stad':

            # Evaluacion del modelo para el training set

            t=time.time()
            y_train_predict = lin_model.predict(X_train)
            elapsed=time.time()-t

            rmse = (np.sqrt(mean_squared_error(Y_train, y_train_predict)))
            r2 = r2_score(Y_train, y_train_predict)

            print('The elapsed time was {}'.format(elapsed))
            print("The model performance for training set")
            print("--------------------------------------")
            print('RMSE is {}'.format(rmse))
            print('R2 score is {}'.format(r2))
            print("\n")

            # evaluacion del modelo para el test set

            y_test_predict = lin_model.predict(X_test)
            rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
            r2 = r2_score(Y_test, y_test_predict)

            print("The model performance for testing set")
            print("--------------------------------------")
            print('RMSE is {}'.format(rmse))
            print('R2 score is {}'.format(r2))
            return(r2)

        if tipo=='theta':

            Theta=[lin_model.intercept_,lin_model.coef_]
            return(Theta)

        if tipo=='pred':

            # make a prediction
            rta_new = lin_model.predict(X)
            rta_new=pd.DataFrame(rta_new, index=Caso_cero.index)
            comp_cero=pd.merge(rta_new, Y, on='Country Name', suffixes=('_left', '_right'))
            comp_cero.columns=['Predicción','GDP Real']
            return(comp_cero)

        if tipo=='modelo':

            return(lin_model)
```

Figura 11: Función para regresión lineal en el cluster cero. En los otros cluster la función es la misma variando el listado de países en el cluster.

En la regresión lineal los parámetros que acompañan a cada X pueden dar un indicio de qué tan

influyente es el parámetro para la regresión. En este caso se puede realizar este tipo de inferencia ya que todos los indicadores se encuentran normalizados.

Variable	Cluster 0	Cluster 1	Cluster 2
X_1	-0.076	0.0784	-1.0133
X_2	0.037	0.0782	0.0942
X_3	0.173	0.0316	-1.2181
X_4	0.042	0.0853	-0.1308
X_5	0.303	0.0120	0.2516
X_6	-0.112	0.0793	0.2584
X_7	0.058	-0.0376	0.4770
X_8	0.051	-0.0998	-1.1492
X_9	0.152	-0.0225	1.3513
X_{10}	-0.244	-0.0594	1.3219
X_{11}	0.042	-0.0936	-0.0250
R^2	0.218	0.0086	-24.309

Cuadro 1: Coeficientes que acompañan las variables X y R^2

Inicialmente se analizará el valor de R^2, este parámetro habla sobre la varianza explicada por el modelo de regresión generado. Se observa que en todos los casos el R^2 es mucho menor a 1. Esto indica que el ajuste no es bueno ya que no logra explicar la variabilidad de los datos del GDP. Se observa en particular que en el cluster 2, que corresponde a los países desarrollados, el valor de R^2 es negativo. El obtener un valor negativo es un fuerte indicador de que las variables independientes no son buenos predictores de la variable dependiente, ya que no logran explicar nada de la variabilidad de los datos. El modelo generado para los cluster 0 y 1 es más positivo ya que el valor de R^2 es positivo. El caso del cluster 0 es el que mejor ya que se explica cerca del 21,8 % de la varianza de los datos, mientras que en el caso del cluster 1 se explica menos del 1 % de la varianza de los datos.

En el caso del cluster 0, los indicadores con mayor incidencia sobre el GDP son: Airport connectivity (X_5), Exposure to unsafe drinking water (X_{10}), Railroad density (X_3), Electricity supply quality (X_9), respectivamente. Es razonable que el GDP dependa de la conectividad del transporte aéreo y de la densidad de vías de ferrocarril, puesto que contar con mejor infraestructura de transporte potencia las actividades económicas primarias de tipo extractivas y de importación, las principales en los países de este cluster (subdesarrollados o en vías de desarrollo). El hecho de que la exposición al agua no potable tenga signo negativo y que sea el segundo indicador más importante no es sorprendente ya que el consumo de agua potable tiene una incidencia directa en la mano de obra disponible: mayor exposición al agua no potable genera mayores enfermedades y por lo tanto un retraso en la calidad de vida de la población y su nivel de desarrollo. Finalmente, que la calidad del suministro de electricidad sea relavante también es esperable si se tiene en cuenta que el desarrollo de algunas industrias depende de un suministro de electricidad constante y confiable.

En el caso del cluster 1, los indicadores con mayor incidencia sobre el GDP son: Efficiency of seaport services (X_8), Reliability of water supply (X_{11}), Efficiency of train services (X_4), y Efficiency of air transport services (X_6). En este caso la mayor parte de los indicadores están relacionados con la eficiencia de de los servicios de transporte, lo que es evidente ya que la infraestructura de transporte es necesaria para el desarrollo de las demás industrias. Por otra parte, resulta sorpresivo que la confiabilidad en el suministro de agua sea relevante para el GDP, y que su signo sea negativo. Se esperaría que tuviera un signo positivo ya que a mayor confiabilidad en el suministro de agua, mayor GDP.

Finalmente, en el caso del cluster 2, los indicadores con mayor incidencia sobre el GDP son: Electricity supply quality X_9, Exposure to unsafe drinking water X_{10}, Railroad density X_3, y Efficiency of seaport services X_8. Nuevamente aparece la calidad del sumiistro de electricidad, pero en un papel más relevante lo que tiene sentido teniendo en cuenta que muschas industrias requieren de un suministro eléctrico constante. Adicionalmente, se repite la eficiencia de los servicios de transporte y la densidad de vías férreas puesto que la infraestructura de transporte resulta fundamental para una industria desarrollada. Para finalizar, aparece nuevamente la exposición a agua no potable, la cual tiene signo

positivo, lo que no tiene sentido y muestra por qué el modelo no es bueno para predecir ni explicar los datos del GDP.

Se observa que en todos los casos el acceso al agua, y la infraestructura de transporte aparecen como indicadores relevantes. Aunque en los casos del cluster 1 y 2 el comportamiento del parámetro asociado no es el más adecuado. Por otra parte, también aparece la infraestructura eléctrica, que se relaciona fuertemente con el desarrollo de las industrias. En general, se podría decir que, aunque los modelos no sean representativos, si pueden indicar los indicadores a tener en cuenta para un análisis diferente. Vale la pena mencionar que en este caso los parámetros de los indicadores tienen valores pequeños por lo que es razonable pensar que, aunque hay algunos más relevantes que otros, la realidad es que la diferencia en influencia no es muy marcada.

Un análisis que podría realizarse es modificar uno de los indicadores para observar su impacto sobre el GDP. En este caso se realizará un aumento del 30 % en la calidad de las vías. Para el cluster 0 se observa un aumento significativo en el GDP para todas los países. En el caso del cluster 1 se obtienen los mismos valores para el GDP proyectado sin el aumento en la calidad de vías. Esto es un fuerte indicio de que este indicador no es tan relevante en los países en vías de desarrollo. Finalmente, en el caso del cluster 2, los países también presentan un incremento en el GDP estimado, aunque este es menor que en el caso de los países del cluster 0. Esto indica que la calidad de vías es menos relevante para los países del cluster 2, considerados con una buena infraestructura de vías. Esto puede responder al hecho de que, cambios pequeños en la infraestructura de países subdesarrollados suponen mejoras significativas, mientras que cambios pequeños en la infraestructura de países desarrollados suponen mejoras casi imperceptibles. Con el objetivo de mostrar esto, se calcula el promedio del GDP estimado con y sin el aumento en la calidad de vías.

```
#Comparaciones

Comparacion_caso_cero=pd.merge(aumento('caso 0'), caso_cero('pred')['Predicción'], on='Country Name', suffixes=('_left', '_right'
Comparacion_caso_cero.columns=['Aumentando la calidad de vías', 'Normal']

Comparacion_caso_uno=pd.merge(aumento('caso 1'), caso_uno('pred')['Predicción'], on='Country Name', suffixes=('_left', '_right'))
Comparacion_caso_uno.columns=['Aumentando la calidad de vías', 'Normal']

Comparacion_caso_dos=pd.merge(aumento('caso 2'), caso_dos('pred')['Predicción'], on='Country Name', suffixes=('_left', '_right'))
Comparacion_caso_dos.columns=['Aumentando la calidad de vías', 'Normal']

print('Caso cero: Aumentando la calidad de vías', Comparacion_caso_cero['Aumentando la calidad de vías'].mean(), 'Sin modificar',
      'Caso uno: Aumentando la calidad de vías', Comparacion_caso_uno['Aumentando la calidad de vías'].mean(), 'Sin modificar', Co
      'Caso dos: Aumentando la calidad de vías', Comparacion_caso_dos['Aumentando la calidad de vías'].mean(), 'Sin modificar', Co
```

```
Caso cero: Aumentando la calidad de vías -0.0008201995485222311 Sin modificar 0.07076931780023789
 Caso uno: Aumentando la calidad de vías 0.027930138163521405 Sin modificar 0.027930138163521405
 Caso dos: Aumentando la calidad de vías 0.08966663463064145 Sin modificar 0.0969304812796909
```

Figura 12: Valor promedio del GDP para cada cluster con y sin el aumento en la calidad de vías.

Finalmente, para complementar el análisis puede realizarse una nueva regresión lineal, la cual no considere la clusterización sino que incluya todos los países.

Esta permitirá realizar una comparación más objetiva. En este caso, el R^2 obtenido es 0.039, lo que significa que la regresión explica el 3,9 % de la varianza. No es significativo pero es mayor al caso de los clusters 1 y 2. Por otra parte, los coeficientes obtenidos son: -0.0058, 0.0008, 0.0163, 0.0214, 0.0935, 0.0003, 0.0095, -0.0144, -0.1115, -0.0111, y 0.0021 respectivamente. Esto muestra que los indicadores más relevantes son: Electricity supply quality (X_9), Airport connectivity (X_5), Efficiency of train services (X_4), y Railroad density (X_3). Nuevamente salen como significativos los indicadores relacionados con el transporte, en particular, la conectividad aérea, la eficiencia en los servicios férreos y la densidad de redes férreas. Este resultado es predecible teniendo en cuenta que el GDP es un indicador económico y se relaciona con el transporte fuertemente por su incidencia en el desarrollo de las industrias. El otro indicador que sale relevante es el indicador de calidad del suministro eléctrico, el cual es requerido para desarrollar industrias más especializadas. Sopresivamente, los indicadores relacionados con el agua no salen entre los más relevantes, a pensar de que para todos los clusters había un indicador relacionado.

Si se realiza la comparación con el modelo de mejor resultado, el del cluster 0, se observa que la precisión del modelo con todos los países es menor, pero que se tienen variables muy similares, a

```
X = datos_completos.iloc[:,0:11]
Y = datos_completos['GDP']

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state=5)
lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)

# Evaluacion del modelo para el training set
t=time.time()
y_train_predict = lin_model.predict(X_train)
elapsed=time.time()-t
rmse = (np.sqrt(mean_squared_error(Y_train, y_train_predict)))
r2 = r2_score(Y_train, y_train_predict)
print('The elapsed time was {}'.format(elapsed))
print("The model performance for training set")
print("--------------------------------------")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
print("\n")
# evaluacion del modelo para el test set
y_test_predict = lin_model.predict(X_test)
rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
r2 = r2_score(Y_test, y_test_predict)
print("The model performance for testing set")
print("--------------------------------------")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

Figura 13: Regresión lineal para calcular el GDP con todos los países.

excepción de la exposición a agua no potable. Esto indica que para todos los países la infraestructura de transporte es lo más relevante dentro de los indicadores propuestos, mientras que el agua entra a jugar un papel menos. El indicador menos relevante para el modelo es el relacionado a la eficiencia del transporte aéreo, seguido por la calidad y conectividad de las vías. Resulta curioso que entre los indicadores más importantes se encuentre la infraestructura férrea, y en los menos importantes la infraestructura de vías. Este comportamiento es similar para los países del cluster 0.

Redes neuronales y regresión logistica

Para esta parte del taller, se requiere importar una librería adicional: *import_piynb*. La función de esta librería es importar algunos dataframes creados en la primera parte del taller:

- df_GDP: Dataframe con la información del GDP de los países en el año 2019. Base de datos normalizada.
- Datos_pilar_infr: Dataframe con la información de los indicadores relacionados con el segundo pilar de sostenibilidad: la infraestructura.
- lista_nombres_ind: Dataframe que contiene el listado de los nombres de los indicadores.

Para comenzar, se pueden agrupar los países en 4 categorías de acuerdo a los cuartiles del GDP. Esto se realiza utilizando el método *quantile()* ingresándole como parámetro el valor de corte asociado a cada cuartil.

A partir de los valores encontrados se pueden definir los siguientes grupos de clasificación:

- Cuartil superior: Países con GDP ¿0.010498. Se definen con clasificación = 3.
- Cuartil 3: Países con GDP entre 0.001233 y 0.010498. Se definen con clasificación = 2.
- Cuartil 2: Países con GDP entre 0.000299 y 0.001233. Se definen con clasificación = 1.
- Cuartil 1: Países con GDP ¡0.000299. Se definen con clasificación = 0.

```
longitud=len(df_GDP) # Numero de paises

# Obtenermos Q1, Q2, Q3

mediana=df_GDP.median()
Q3=df_GDP.quantile(0.75) #Cuartil 3
Q1=df_GDP.quantile(0.25) #Cuartil 1

df_GDP=df_GDP.sort_values(by=0,ascending=False) #Ordenar

print('Q1 es', Q1)
print('La mediana de los datos es', mediana)
print('Q3 3s', Q3)
```

```
La mediana de los datos es 0    0.001233
dtype: float64
Q1 es 0    0.000299
Name: 0.25, dtype: float64
Q3 0    0.010498
Name: 0.75, dtype: float64
```

Figura 14: Cálculo de los cuartiles en la base de datos del GDP.

A continuación se crea cada uno de los grupos de clasificación, primero ordenando los datos, luego contando la cantidad de países que van en dicho cuartil, y luego agregando dichos países en un arreglo de datos al que se le define una columna llamada clasificación, con la clasificación respectiva. A continuación se presenta el código ejemplo para los primeros dos cuartiles.

```
#Datos por encima del Q3 - Clasificacion 3

cont=(Q3<df_GDP).sum() #Numero de valores por encima del cuartil Q3

Cuartil_superior=np.zeros(int(cont))
for n in range (int(cont)):
    Cuartil_superior[n]=df_GDP.iloc[n]

Cuartil_superior=pd.DataFrame(Cuartil_superior, index=df_GDP[:int(cont)].index)
Cuartil_superior['clasificación']=3

# Datos entre el Q2 y Q3

cont1=(mediana<df_GDP).sum() #Numero de valores entre Q2 y Q3
cont1=cont1-cont

Cuartil_Q3=np.zeros(int(cont1))
for n in range (int(cont1)):
    Cuartil_Q3[n]=df_GDP.iloc[n+int(cont)]

df_GDP_v=df_GDP[int(cont):]
df_GDP_v=df_GDP_v[:int(cont1)]

Cuartil_Q3=pd.DataFrame(Cuartil_Q3, index=df_GDP_v.index)
Cuartil_Q3['clasificación']=2
```

Figura 15: Proceso de clasificación de acuerdo con los cuartiles para los dos primeros grupos.

Finalmente, para generar un solo dataframe con la clasificación se utiliza la función *concat* sobre una lista que contiene los dataframes de cada cuartil.

Con el dataframe generado, es posible tratar de replicar la categorización realizada utilizando una red neuronal que reciba como entrada los indicadores del segundo pilar de la sostenibilidad. Por lo tanto, se requiere inicialmente definir los parámetros que ingresan a la red. Para ello se realiza un *merge* entre los datos clasificados y los datos de los indicadores, utilizando como punto de unión el nombre de los países. Adicionalmente, se debe realizar una transformación a los datos de modo que los indicadores se vuelvan comparables entre sí al escalarlos. La transformación se realiza con la función *MinMaxScaler()* del módulo *preprocessing*:

$$X_{std} = \frac{X - min(X)}{max(X) - min(X)} X_{scaled} = X_{std} * (max_{rango} - min_{rango}) + min_{rango}$$

```
# Un solo Dataframe con los datos clasificados

frames = [Cuartil_superior, Cuartil_Q3, Cuartil_Q1, Cuartil_inferior]
datos_GDP_clasificados=pd.concat(frames)
datos_GDP_clasificados.columns=['GDP','clasificacion']
datos_GDP_clasificados
```

Country Name	GDP	clasificacion
United States	1.000000	3
China	0.667881	3
Japan	0.239369	3
Germany	0.181852	3
South Asia	0.171085	3
...	...	...
Palau	0.000011	0
Marshall Islands	0.000008	0
Kiribati	0.000006	0
Nauru	0.000003	0
Tuvalu	0.000000	0

Figura 16: Arreglo de datos con la clasificación de los países de acuerdo con los cuartiles del GDP.

los valores de min_{rango} y max_{rango} vienen del rango definido para el escalamiento de datos, que por defecto es de 0 a 1. Para continuar, se separan los datos en datos de entrenamiento y datos de prueba utilizando la función *train_test_split* tomando para el test el 20 % de los datos aproximadamente.

```
data=pd.merge(Datos_pilar_infr, datos_GDP_clasificados, on='Country Name', suffixes=('_left', '_right'))

X = data.iloc[:,0:11].values
Y = data['clasificacion'].values

min_max_scaler = preprocessing.MinMaxScaler()
X_scaler = min_max_scaler.fit_transform(X)

X_train, X_test, Y_train, Y_test = train_test_split(X_scaler, Y, test_size = 0.2)

print(X_train.shape, X_test.shape, Y_train.shape,Y_test.shape)
```

Figura 17: Dataframe definitivo y escalamiento de los indicadores.

Para generar la red neuronal se crea una función la cual utiliza la función (models.Sequential()) de la librería *keras* ejecutada sobre *tensorflow* para obtener un mode secuencial. Los parámetros de esta función son:

- keras.layers.Flatten(input_shape =(11,)): Esta capa se encarga de realizar un reshape a los datos de entrada de modo que queden ajustados para trabajarse en la red neuronal.
- keras.layers.Dense(100, activation='relu', input_shape=(11,)): Esta capa recibe los datos transformados, realiza el cálculo lineal y posteriormente aplica una función de activación de ripo *relu* que filtra todos los valores negativos asignandoles un valor de cero, mientras que a los positivos los incrementa linealmente.

  Esta capa se encuentra compuesta por 100 neuronas.
- keras.layers.Dense(4, activation='softmax'): Esta es la última capa, cuenta con 4 neuronas porque hay 4 posibles clasificaciones. Se encarga de calcular la probabilidad de que cada país (cuantificado por su grupo de indicadores) corresponda a una clasificación. La función de activación que utiliza es *softmax* que tiene un comportamiento similar a la sigmoide.

Para continuar, se usa la función *compile* sobre el mode instanciado previamente, ajustando los parámetros *optimizer = .ªdam", loss = "sparse_categorical_crossentropy", metrics = ['accuracy']*. Los

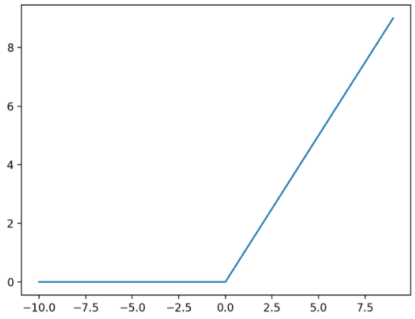
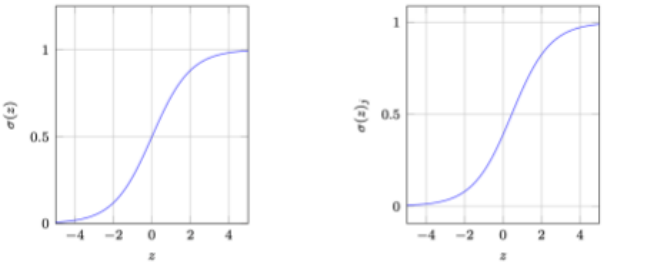


Figura 18: Función de activación capa 2.



(a) Sigmoid activation function. (b) Softmax activation function.

Figura 19: Función de activación capa 3.

parámetros ajustados indican que la optimización se realizará por el método .ªdam", se minimizará la función crossentropy relacionada con datos distribuidos normalmente, y se maximizará la métrica accuracy que se relaciona con la cantidad de clasificaciones correctas sobre el total de las clasificaciones.

```
# Define a simple sequential model

def create_model():

    model = keras.models.Sequential([
        keras.layers.Flatten(input_shape=(11,)),
        keras.layers.Dense(100, activation='relu', input_shape=(11,)),
        keras.layers.Dense(4, activation='softmax'), #4 neuronas de salida porque tenemos 4 clasificaciones posibles
      ])
    model.compile(optimizer='adam',
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])
    return model

# Create a basic model instance

model = create_model() #Creacion del modelo de red neuronal

# Display the model's architecture

model.summary() #Resum del modelo creado
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_1 (Flatten)         (None, 11)                0

 dense_2 (Dense)             (None, 100)               1200

 dense_3 (Dense)             (None, 4)                 404

=================================================================
Total params: 1604 (6.27 KB)
Trainable params: 1604 (6.27 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figura 20: Función para generar la red neuronal.

Los resultados del modelo muestran que la red tiene un total de 1604 parámetros para entrenar, de los cuales 1200 corresponden a la capa 2 y 404 a la capa 3. Evidentemente la capa 1 no tendrá

parámetros puesto que solo reordena los datos. Una vez instanciado el modelo se procede a entrenar la red neuronal con el conjunto de datos de entrenamiento y a evaluarlo.

```
t=time.time()
model.fit(X_train,
          Y_train,
          epochs=70,
          validation_data=(X_test,Y_test)); #Entrenamiento del modelo y validacion
elapsed=time.time()-t
elapsed_r=round(elapsed,3)
print('The elapsed time was {}'.format(elapsed_r))
```

Figura 21: Entrenamiento y validación de la red neuronal.

El tiempo gastado en el entrenamiento de la red neuronal fueron 7.579s. Vale la pena mencionar que, en este caso, el parámetro *epoch=70* representa la cantidad de repeticiones realizadas para el entrenamiento. Adicionalmente, también se observa que el accuracy de la red sobre el grupo de datos para testear es 0.653.

Otra metodología que se puede implementar para tratar de replicar la clasificación mediante el GDP es usando una regresión logística. Para desarrollar el algoritmo de regresión logística se deben escalar los datos nuevamente, además de que se agregan unas variables cuadráticas al conjunto de indicadores. Para realizar la regresión nuevamente se toman el 80 % de los datos para entrenamiento y el restante paratestear. Se crea la instancia del modelo de regresión logística usando la función *LogisticRegression()* y se realiza un fit sobre los datos de entrenamiento, de modo que se entrena el modelo. Adicionalmente, usandola matriz de errores se calcula el accuracy de modo que sea comparable con los resultados de la red neuronal.

```
X_log_regression= data.iloc[:,0:11].values
Y_log_regression = data['clasificacion'].values

min_max_scaler_reg = preprocessing.MinMaxScaler()
X_scaler_log_regression = min_max_scaler.fit_transform(X_log_regression)

X_scaler_log_regression=np.append(X_scaler_log_regression,X_scaler_log_regression**2,axis=1) #Se añaden variables cuadradas
X_log_regression_train, X_log_regression_test, Y_log_regression_train, Y_log_regression_test = train_test_split(X_scaler_log_regr
lr = LogisticRegression() #Se genera el modelo logistico
t=time.time()
lr.fit(X_log_regression_train, Y_log_regression_train) #Se entrena el modelo
elapsed=time.time()-t
print('The elapsed time was {}'.format(elapsed))
y_pred = lr.predict(X_log_regression_train) #Se realiza la prediccion del train set
confusion=confusion_matrix(Y_log_regression_train, y_pred)/np.sum(confusion_matrix(Y_log_regression_train, y_pred)) #generacion d
accuracy=np.trace(confusion) #calculo de la exactitud
print("The model performance for training set")
print("----------------------------------------")
print('Accuracy of {}'.format(accuracy))
y_pred = lr.predict(X_log_regression_test)
confusion=confusion_matrix(Y_log_regression_test, y_pred)/np.sum(confusion_matrix(Y_log_regression_test, y_pred))
accuracy=np.trace(confusion)
print("The model performance for testing set")
print("----------------------------------------")
print('Accuracy of {}'.format(accuracy))
```

Figura 22: Entrenamiento y validación de la regresión logística.

Se observa que el tiempo para el entrenamiento de la regresión logísstica fueron solo 0.1135s, mientras que el accuracy es de 0.61538.

Para comparar ambos métodos inicialmente se hablará de los tiempos de entrenamiento. Por el tipo de algoritmo que se utiliza para realizar redes neuronales, es razonable que este tome más tiempo que la elaboración de una regresión logística. Para este caso en específico, el entrenamiento de la red neuronal tarda 66.76 veces lo que tarda la regresión logística, lo que marca una diferencia importante entre ambas metodologías para la clasificación, principalmente porque muchas veces se buscan algoritmos más rápidos para realizar el procesamiento de los datos. Por otra parte, en términos de la métrica usada para evaluar el desempeño de ambas metodologías, se encuentra que la red neuronal cuenta con un mejor desempeño (teniendo en cuenta que al accuracy es la razón entre predicciones acertadas y todas las predicciones, sin embargo, la mejora entre el modelo de regresión logística y el modelo de red neuronal no es tan significativa como la diferencia en tiempo. De lo anterior se puede concluir que

la red neuronal puede alcanzar un mayor nivel de exactitud pero con un costo alto en tiempo, sobre todo si se tiene en cuenta que se pueden agregar capas para mejorar su desemepeño). Mientras que la regresión logística puede dar resultados aceptables con un bajo coste temporal y computacional.

Para continuar, es posible realizar un análisis de los indicadores y la clasificación obtenida para Colombia, considerando la red neuronal, que es el modelo con mayor exactitud. En primer lugar se debe conocer la clasificación actual de Colombia de acuerdo con los cuartiles, 3. Posteriormente, se debe calcular la clasificación de acuerdo con la red neuronal, de modo que se pueda verificar si prefice adecuadamente para el país. El resultado muestra que el modelo de red neuronal permite replicar la clasificación de Colombia.

```
X_2=pd.DataFrame(X_scaler,index=data.index)
X_2.columns = lista_nombres_ind

originales_Col=np.zeros(11)

for n in range (0,11):
    originales_Col[n]=X_2[lista_nombres_ind[n]]['Colombia']

# Numero de fila de Colombia
ind=0
for i in range(0, len(X_2)):
    if X_2['Quality of road infrastructure'].iloc[i]==X_2['Quality of road infrastructure']['Colombia']:
        ind=i

valor_calidad_vias_Col=X_2['Quality of road infrastructure']['Colombia']
predictions = model.predict(X_2) #Predicciones
categoria_prediccion_Col=np.argmax(predictions[ind])
print('De acuerdo con la red neuronal (predicion) Colombia esta en categoria', categoria_prediccion_Col)
print('De acuerdo con la clasificacion real por cuartiles Colombia esta en la categoria', datos_GDP_clasificados['clasificacion']
```

```
4/4 [==============================] - 0s 7ms/step
De acuerdo con la red neuronal (predicion) Colombia esta en categoria 3
De acuerdo con la clasificacion real por cuartiles Colombia esta en la categoria 3
```

Figura 23: Clasificación original y generada por la red neuronal.

Ahora, si se realiza una variación en el índice de calidad de vías, se pueden ingresar los valores de Colombia a la red neuronal y observar la nueva clasificación. En este caso se decide poner a prueba el caso límite en el que Colombia obtuviera un indicador = 1, de modo que tiene la mejor calidad de vías posible. Al evaluar con la red neuronal se encuentra que la clasificación para el GDP sigue siento la misma, es decir, 3. Esto significa que el indicador de calidad de vías no es tan relevante para la clasificación. Este resultado concuerda relativamente con el resultado anterior donde la calidad de vías nunca fue un indicador relevante para la estimación del GDP. Adicionalmente, para el caso de Colombia tiene sentido que mejorar las vías no incremente su clasificación puesto que más allá de tener mejores vías, el país necesita mayor redundancia vial, es decir, incrementar la cantidad de vías.

Para finalizar, como se encuentra que para Colombia el indicador de calidad de vías no realiza un cambio significativo en la clasificación recibida, se puede realizar la prueba para los otros indicadores, es vez realizando un pequeño cambio en cada uno. La intensión de realizar el mismo cambio pequeño es que los resultados puedan compararse en términos de la variación de las probabilidades.

La siguiente tabla resume el resultado para los indicadores (siguen el mismo orden que cuando se analizaron en la regresión lineal).

De estos resultados la primera conclusión es que en todos los casos la probabilidad más alta es estar en el grupo de clasificación 3, o en el cuartil superior. Sin embargo, si se observan algunas variaciones en las probabilidaes:

- Incrementar el indicador 'Road connectivity' incrementa la probabilidad de estar en las clasificaciones 2 y 3.
- Incrementar el indicador 'Quality of road infrastructure' disminuye la probabilidad de estar en la clasificación 3, mientras que aumenta la probabilidad de estar en la clasificación 2.
- Incrementar el indicador 'Railroad density' incrementa la probabilidad de estar en las clasificaciones 2 y 3.
- Incrementar el indicador 'Efficiency of train services' incrementa la probabilidad de estar en la clasificación 3, y disminuye todas las demás. Es el tercer indicador que más incrementa la

```
X_2=pd.DataFrame(X_scaler,index=data.index)
X_2.columns = lista_nombres_ind

originales_Col=np.zeros(11)

for n in range (0,11):
    originales_Col[n]=X_2[lista_nombres_ind[n]]['Colombia']

# Numero de fila de Colombia
ind=0
for i in range(0, len(X_2)):
    if X_2['Quality of road infrastructure'].iloc[i]==X_2['Quality of road infrastructure']['Colombia']:
        ind=i

valor_calidad_vias_Col=X_2['Quality of road infrastructure']['Colombia']
predictions = model.predict(X_2) #Predicciones
categoria_prediccion_Col=np.argmax(predictions[ind])
print('De acuerdo con la red neuronal (predicion) Colombia esta en categoria', categoria_prediccion_Col)
print('De acuerdo con la clasificacion real por cuartiles Colombia esta en la categoria', datos_GDP_clasificados['clasificacion'])
```

```
4/4 [==============================] - 0s 7ms/step
De acuerdo con la red neuronal (predicion) Colombia esta en categoria 3
De acuerdo con la clasificacion real por cuartiles Colombia esta en la categoria 3
```

Figura 24: Metodología para incrementar un 15 % a cada uno de los indicadores por separado, para evidenciar su efecto sobre las probabilidades de clasificación.

Indicador	P(Clas=0)	P(Clas=1)	P(Clas=2)	P(Clas=3)
Sin incremento	0.014667	0.168285	0.326614	0.490533
$1.15X_1$	0.013784	0.163055	0.327555	0.495606
$1.15X_2$	0.014360	0.165502	0.329669	0.490469
$1.15X_3$	0.014513	0.167234	0.327184	0.491069
$1.15X_4$	0.014219	0.163212	0.321638	0.500931
$1.15X_5$	0.011705	0.125419	0.282346	0.580529
$1.15X_6$	0.014007	0.177279	0.327703	0.481010
$1.15X_7$	0.012330	0.140882	0.297547	0.549241
$1.15X_8$	0.013893	0.175204	0.340523	0.470380
$1.15X_9$	0.014653	0.189468	0.340748	0.455131
$1.15X_{10}$	0.014655	0.178601	0.342882	0.463862
$1.15X_{11}$	0.014385	0.162008	0.329924	0.493684

Cuadro 2: Cambio en la propabilidad de pertenecer a alguna categoría dependiendo del incremento en los indicadores.

probabilidad de estar en el cuartil superior.

- Incrementar el indicador 'Airport connectivity' incrementa la probabilidad de estar en la clasificación 3. Es el indicador que más incrementa la probabilidad de estar en el cuartil superior.
- Incrementar el indicador 'Efficiency of air transport services' disminuye la probabilidad de estar en la clasificación 3 y aumenta la probabilidad de estar en la clasificación 2.
- Incrementar el indicador 'Liner shipping connectivity' incrementa la probabilidad de estar en la clasificación 3. Es el segundo indicador que más incrementa la probabilidad de estar en el cuaartil superior.
- Incrementar el indicador 'Efficiency of seaport services' disminuye la probabilidad de estar en la clasificación 3, mientras que aumenta la probabilidad de estar en la clasificación 1 y 2.
- Incrementar el indicador 'Electricity supply quality' disminuye la probabilidad de estar en la clasificación 3 y aumenta la probabilidad de estar en las clasificaciones 1 y 2.
- Incrementar el indicador 'Exposure to unsafe drinking water' disminuye la probabilidad de estar en una clasificación 3, mientras que incrementa la probabilidad de estar en una clasificación 1 o 2.
- Incrementar el indicador 'Reliability of water supply' aumenta la probabilidad de estar en una clasificación 2 o 3.

De las variaciones de probabilidad se puede concluir que el indicador más influyente en el GDP de Colombia es la conectividad aérea. Este resultado es razonable teniendo en cuenta que en el país

existen muchas zonas con un nivel de desconexión importante. La inversión en infraestructura aérea mejoraría varios aspectos de la economía, desde ser una vía competitiva para transporte de mercancía hasta ser un medio de transporte público competitivo para las personas. Además, tener una infraestructura aérea más amplia genera conectividad y puede contribuir a desarrollar zonas que se encuentran desconectadas del resto del país (o cuya conexión podría mejorar, p.e. los llanos) por lo que se pueden generar nuevas industrias o fortalecer las existentes.

Otros indicadores que generan un aumento en la probabilidad de estar en el cuartil superior del GDP son la conectividad del transporte marítimo y la eficiencia en los servicios férreos. Que incrementarla eficiencia de los servicios férreos sea un factor importante tiene sentido puesto que la infraestructura de trenes, así como los servicios prestados por la misma, es una de las más atrasadas en todo el país. Las características geofráficas del país han dificultado la creación de sistemas férreos eficientes para el tansporte de mercancias o para el desplazamiento de personas. Adicionalmente, se ha observado que en países con un nivel de desarrollo mayor al de Colombia, este tipo de infraestructura juega un papel fundamental para los sistemas de transporte público, así como servicio complementario de transporte de marcancías al interior del país.

Por otra parte, la conectividad del transporte marítimo es un resultado esperable teniendo en cuenta que la mayor parte de la carga que ingresa al país proviene únicamente de dos puertos (Buenaventura y Cartagena), lo que genera una dependencia de zonas muy puntuales y no explota el potencial de toda la zona marítima con la que cuenta el país. Además, que el transporte marítimo se concentre en sitios puntuales ocasiona que otros sectores (como el transporte carretero) se concentren en las mismas zonas, relegando aún más los lugares desconectados.

De los resultados anteriores resulta evidente que un país como Colombia requiere trabajar en su infraestructura aérea, marítima y férrea. Por lo tanto, sería conveniente proponer una estrategia que permita al país trabajar en los puntos que le permiten llegar a un mejor nivel de desarrollo cuantificado mediante el GDP.

Evidentemente la inversión más urgente para el país resulta ser los aeropuertos, seguida por la ifnraestructura férrea, y por último la conectividad marítima. En este orden de ideas, el país podría concentrar sus esfuerzos en desarrollar un mayor nivel de conectividad a nivel aéreo, principalmente mejorando la infraestructura existente y creando nueva infraestructura en lugares con el potencial de generar nuevas industrias o ser nuevos centros de desarrollo. Por ejemplo, una opción podría ser ubicar más aeropuertos en la zona de los llanos, se trata de una región con gran potencial agrícola y con una exploración y aprovechamiento bajo debido a la presencia de ganadería. La existencia de infraestructura aérea podría incentivar otras activades que generen tránsito aéreo. Además, facilitaría la llegada de suministros y podría presentarse como una alternativa a los problemas de transporte ocasionados por los cierres en la vía al llano.

La segunda inversión que debería realizar el país es en infraestructura férrea. La razón es que este tipo de transporte permite llevar cargamentos de mayor volumen, y podría disminuir los costos de operación, teniendo en cuenta que el transporte carretero (principal en el país), es costoso debido a las vías que deben sortear la geografía. Teniendo en cuenta que el país ya cuenta con una infraestructura de ferrocarriles, la cual se encuentra deteriorada en este momento, la inversión podria permitir reactivar este sistema y comenzar a posicionar sistemas complementarios. Además, la reactivación de este sistema traerá mejoras a nivel de desarrollo en las poblaciones cercanas, lo que genera un beneficio más allá de lo económico.

La tercera inversión debería ser en conectividad marítima, por ejemplo mediante el desarrollo de más puertos o aprovechando las redes fluviales del país para el transporte de mercancía. La aparición de nuevos puertos contribuye a descongestionar las zonas activas y a llevar desarrollo a nuevas regiones probablemente abandonadas por el estado (como es el caso de la costa pacífica). Adicionalmente, la aparición de nuevos puertos contribuye a generar nuevas rutas internas en el país para el tránsito y transporte interno, lo que podría empalmar bien con la infraestructura desarrollada para sistemas férreos y aéreos. Por otra parte, la opción de generar un sistema de transporte fluvial nace de aprove-

char los ríos de gran envergadura (como el Magadalena), que permitirían sortear la geografía del país de forma más eficiente, de modo que se abaraten algunos costos relacionados con transporte. La medida del transporte fluvial también contribuiría a generar nuevos centros de desarrollo y complementar el plan de acción para tener mayor eficiencia y conectividad al interior del país.

Finalmente, una cuarta inversión requerida por el país debería ser en infraestructura vial para transporte carretero. La razón es que, más allá de que existan otros modos de transporte que puedan ser más eficientes, la conectividad a nivel de vías y la redundancia vial es un problema latente en el país, y mientras las nuevas infraestructuras (como la aérea o la marítima) se desarrollan, el país debe aprovechar la infraestructura existente para sacar el mejor provecho de lo que se tiene y de lo que se está creando con nuevos proyectos. En este orden de ideas, esta inversión, más que ser la cuarta inversión, puede verse como una inversión transversal al plan de acción planteado. En esta vía el país se encuentra en buen camino ya que se encuentra mejorando su sistema de concesiones y construyendo vías que, por lo menos, conecten los sitios más importantes para la economía del país (como es el caso de la ruta del sol).

Evidentemente la estrategia planteada se construye a partir del análisis de los indicadores más influyentes sobre el GDP, y de una observación sobre el estado actual del país. Por lo tanto, se espera que siguiendo esta ruta sea posible incrementar el indicador del país, de modo que se consiga mantener en el cuartil superior.