UNIMINUTO
Corporación Universitaria Minuto de Dios
Educación de calidad al alcance de todos

POSGRADO

ANÁLISIS DE LA INFORMACIÓN DEMOGRÁFICA DE LOS COLOMBIANOS EN EL EXTERIOR UTILIZANDO BIG DATA Y LA METODOLOGÍA CRISP-DM: UN ESTUDIO DE CASO

Integrantes

GUSTAVO ANDRES DIAZ PRIETO 000792780

ANGELO RIOS RAMIREZ 000544544

UNIVERSIDAD UNIMINUTO
FACULTADO DE INGENIERIA

Bogotá, Colombia
2023

RESUMEN

Este estudio de caso presenta el análisis de la información demográfica de los colombianos en el exterior utilizando big data y la metodología CRISP-DM. La base de datos utilizada es proporcionada por el gobierno colombiano y contiene información demográfica detallada de los ciudadanos colombianos que residen en diferentes países del mundo. El objetivo de este estudio es analizar los patrones demográficos y de migración de los colombianos en el exterior mediante la limpieza de datos, la generación de análisis estadísticos y gráficos.

INTRODUCCIÓN:

La creciente migración de personas en todo el mundo ha llevado a la necesidad de analizar la información demográfica de la diáspora colombiana. La metodología CRISP-DM es una herramienta efectiva para abordar el análisis de datos en el campo del big data. En este estudio de caso, se utilizará la base de datos del gobierno colombiano para analizar los patrones demográficos y de migración de los colombianos en el exterior según una serie de características que se tienen de las personas migrantes el cual va a ayudar a realizar un análisis de sus condiciones y hacia donde migraron.

MARCO TEORICO

Para el desarrollo de este articulo se realizó inicialmente un análisis para determinar que metodología se acoplaba a los datos y caso de estudio donde se determina que la mas adecuada es la metodología CRISP-DM.

METODOLOGÍA CRISP DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es un enfoque estructurado y ampliamente aceptado para proyectos de minería de datos.

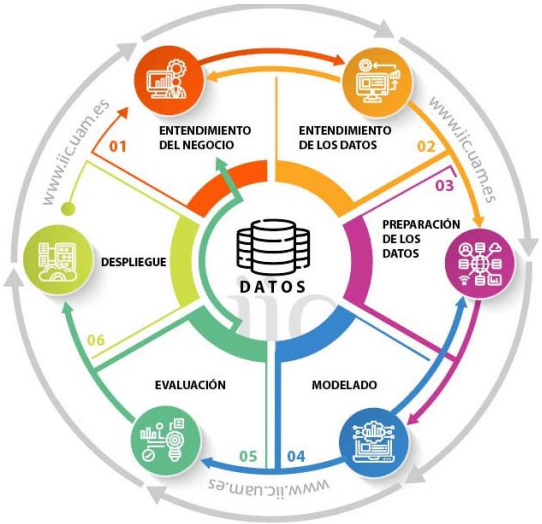

Figura 1. Esquema de ciclo de CRISP-DM estândar. [1]

La metodología CRISP-DM está compuesta por seis fases principales, que se describen a continuación:

1. **Comprensión del negocio.**

Esta fase inicial se enfoca en la comprensión de los objetivos y exigencias del proyecto desde una perspectiva de negocio. Posteriormente convierte ese conocimiento de los datos en la definición de un problema de minería de datos y en un plan preliminar diseñado para alcanzar los objetivos.[2]

2. **Compresión de los datos.**

La fase de comprensión de datos comienza con una recopilación inicial de datos y continúa con las actividades para familiarizarse con los datos, para identificar problemas de calidad de los datos, para descubrir las primeras ideas sobre los datos o para detectar subconjuntos interesantes para formar hipótesis para información oculta.[3]

3. **Preparación de los datos.**

La preparación de los datos debe realizarse mediante la definición de criterios de inclusión y exclusión. La mala calidad de los datos puede ser manejada por limpieza de datos. Dependiendo del modelo utilizado (definido en la primera fase) se deben construir los atributos derivados. Para todos estos pasos

son posibles diferentes métodos y dependen del modelo.[4]

4. Modelado.

La fase de modelado de datos consiste en seleccionar la técnica de modelado, construir el caso de prueba y el modelo. Toda la minería de datos. Se pueden utilizar técnicas. En general, la elección depende del problema comercial y de los datos. Más importante es cómo explicar la elección. Para construir el modelo, se deben establecer parámetros específicos. Para evaluar el modelo es apropiado evaluar el modelo frente a los criterios de evaluación y seleccionar los mejores. [4]

5. Evaluación.

En esta etapa del proyecto, ha construido uno o más modelos que parecen tener alta calidad, desde una perspectiva de análisis de datos. Antes de proceder al despliegue final del modelo, es importante evaluar más a fondo el modelo y revisar los pasos ejecutados para construir el modelo, para estar seguros de que logra adecuadamente los objetivos de negocio. Un objetivo clave es determinar si existe algún tema comercial importante que no ha sido suficientemente considerado. Al final de esta fase, se debe tomar una decisión sobre el uso de los resultados de la minería de datos.[5]

6. Despliegue.

Esta es la sexta y última fase del proceso CRISP-DM que se enfoca en determinar el uso para obtener conocimiento y resultados. Esta fase también se enfoca en organizar, informar y presentar el conocimiento adquirido cuando sea necesario.[6]

HERRAMIENTAS POR UTILIZAR:

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido. La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar. Python no es adecuado sin embargo para la Programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles.[7]

COMPRENSIÓN DEL NEGOCIO:

Tiene como objetivo obtener una comprensión profunda de los objetivos, requisitos y contextos del negocio antes de iniciar cualquier proyecto de minería de datos. Esta etapa implica el análisis y la clarificación de los objetivos y requerimientos del negocio, así como la identificación de los factores clave que influyen en el éxito del proyecto.

La base de datos “información demográfica colombianos en el exterior - nivel de estudio y grupo edad” se descargó de datos abiertos con el link https://www.datos.gov.co/Estad-sticas-Nacionales/Informaci-n-demogr-fica-Colombianos-en-el-exterior/j226-e3rk, la cual se utilizó para el caso de estudio y comprender los siguientes aspectos

Comprender la distribución geográfica de los colombianos en el exterior y los países de destino más comunes.

Analizar la estructura demográfica de la población colombiana en el exterior en términos de edad, género, nivel educativo, estado civil y etnia.

Identificar las principales áreas de conocimiento y campos de estudio en los que se desempeñan los colombianos en el exterior.

COMPRENSIÓN DE LOS DATOS:

Origen de los datos:

Se enfoca en adquirir un conocimiento profundo de los datos disponibles antes de realizar cualquier análisis o modelado. El objetivo de esta etapa es comprender la estructura de los datos,

La base de datos utilizada para este proyecto es "Información demográfica: colombianos en el exterior", disponible en el portal de datos abiertos del gobierno de Colombia. Los datos son proporcionados por entidades oficiales y recopilan información demográfica de los colombianos que residen en el extranjero.

Estructura de los datos:

Los datos se encuentran en formato de archivo CSV (Comma-Separated Values) y contienen múltiples columnas que representan diferentes atributos demográficos de los colombianos en el exterior. Algunas de las columnas relevantes incluyen:

País: El país de residencia de los colombianos en el exterior.

Código ISO país: El código ISO de tres letras que identifica al país de residencia.

Oficina de registro: oficina o entidad donde se realizó el registro de los colombianos que residen en el extranjero.

Grupo edad: El grupo de edad al que pertenece cada individuo.

Edad (años): La edad de los colombianos en años.

Área Conocimiento: El área de conocimiento o campo de estudio en el que se desempeñan los colombianos.

Sub Área Conocimiento: El título profesional o grado académico alcanzado por los colombianos.

Nivel Académico: El nivel educativo alcanzado por los colombianos en el exterior.

Estado civil: El estado civil de los colombianos en el exterior.

Género: El género de los colombianos, puede ser "Masculino" o "Femenino".

Etnia de la persona: La etnia a la que pertenecen los colombianos.

Estatura (CM): La estatura de los colombianos en centímetros.

Localización: ubicación geográfica o lugar donde se encuentran registrados los colombianos en el extranjero.

Cantidad de personas: La cantidad de personas que se encuentran en cada categoría demográfica.

La base de datos contiene 13 columnas y 721.677 filas.

PREPARACIÓN DE LOS DATOS:

Carga de datos:

Se procedió a la carga de la base de datos "Información demográfica colombianos en el exterior" en un entorno de trabajo adecuado, utilizando una herramienta de análisis de datos Júpiter Notebook la cual es una aplicación web de código abierto que permite crear y compartir documentos interactivos que contienen código en vivo, visualizaciones, texto explicativo y otros elementos multimedia. Está diseñado para facilitar el desarrollo, la ejecución y la documentación de proyectos de programación, análisis de datos y creación de modelos. se verificó la correcta importación de los datos y se exploró la estructura de la base de datos para comprender las diferentes columnas y su contenido.

```
In [1]: #Definicion de librerior e importacion de base de datos completa para trabajar
        import pandas as pd

        # desactivar los warning
        import warnings
        warnings.filterwarnings('ignore')

        ##importar datos
        df = pd.read_csv('Colombianos_registrados_en_el_exterior.csv')
        df.dtypes
```

Figura 2. Cargue de datos en dataframe

Limpieza de datos inconsistentes o erróneos:

Para realiza la limpieza de datos se deben generar frecuencias para encontrar datos atípicos y suprimir aquellas filas que no puedan proporcionar un valor análisis

Se inicia con un renombramiento de columnas para que su codificación y manipulación sea más eficiente.

```
In [2]: # Se inicia con un renombramiento de columnas para que su codificación y manipulación sea más eficiente.
df = df.rename(columns={'País': 'País', 'Código ISO país': 'Cod_país', 'Edad (años)': 'Edad', 'Género': 'Genero',
                        'Cantidad de personas': 'Cant_personas','Nivel Académico':'Nivel_educativo','Grupo edad':'Grupo_edad',
                        'Estado civil': 'Estado_civil','Etnia de la persona':'Etnia_persona','Estatura (CM)':'Estatura'})
df.head()
```

Figura 3. Renombrado de columnas

Revisamos las variables numéricas para determinar valores máximos y mínimos.

```
In [3]: # Revisamos las variables numericas para determinar valores maximos y minimos
df.describe()
```

Out[3]:

	Edad	Estatura	Cant_personas
count	721677.000000	721677.000000	721677.000000
mean	45.196733	79.228647	1.803643
std	16.200684	276.768978	4.960423
min	-1.000000	-1.000000	1.000000
25%	34.000000	-1.000000	1.000000
50%	43.000000	-1.000000	1.000000
75%	56.000000	165.000000	1.000000
max	139.000000	163163.000000	391.000000

Figura 4. Estadísticas de las variables numéricas de la data original

Se eliminan filas de los datos con edad = -1 ya que es una variable que se tendrá en cuenta para el modelo.

```
In [4]: # Se eliminan filas de los datos con edad = -1 ya que es una variable que se tendra en cuenta para el modelo
df.drop(df[(df['Edad'] == -1)].index, inplace=True)
df.describe()
```

Out[4]:

	Edad	Estatura	Cant_personas
count	720797.000000	720797.00000	720797.000000
mean	45.253134	79.26993	1.799978
std	16.129907	276.92317	4.863003
min	0.000000	-1.00000	1.000000
25%	34.000000	-1.00000	1.000000
50%	43.000000	-1.00000	1.000000
75%	56.000000	165.00000	1.000000
max	139.000000	163163.00000	308.000000

Figura 5. Estadísticas de las variables numéricas posterior a la limpieza de datos

Se realiza frecuencia por variable Nivel educativo para determinar posibles valores erróneos o no especificados.

```
In [5]: # Se realiza frecuencia por variable Nivel educativo para determinar posibles valores erroneos o no especificados
df['Nivel_educativo'].value_counts().sort_values(ascending = False)
```

Out[5]:

```
NO INDICA                       329435
BACHILLERATO                    131386
PREGRADO - PROFESIONAL           97813
PRIMARIA                         55688
PREGRADO - TÉCNICO PROFESIONAL   34266
POSTGRADO - MAESTRIA             22362
PREGRADO - TECNOLÓGICO           17536
POSTGRADO - ESPECIALIZACIÓN      13891
NINGUNO                          12228
POSTGRADO - DOCTORADO             5419
(NO REGISTRA)                      714
SIN PROFESIÓN                       59
Name: Nivel_educativo, dtype: int64
```

Figura 6. Frecuencias de datos por variable de nivel educativo.

Se eliminan filas cuyo Nivel Educativo sea igual No Registra ya que no es una categoría que sirva para el análisis posterior.

Se realiza frecuencia por variable Género para determinar posibles valores erróneos o no especificados.

```
In [9]: # Se realiza frecuencia por variable Genero para determinar posibles valores erroneos o no especificados
df['Genero'].value_counts().sort_values(ascending = False)
```

Out[9]:

```
FEMENINO      315368
MASCULINO     254522
DESCONOCIDO      861
Name: Genero, dtype: int64
```

Figura 7. Frecuencia por Genero

Se eliminan filas cuyo género sea igual a desconocido ya que no es una categoría que sirva para el análisis posterior.

```
In [10]: # Se eliminan filas cuyo Genero sea igual a desconocido ya que no es una categoria que sirva para el analisis posterior
df.drop(df[(df['Genero'] == 'DESCONOCIDO')].index, inplace=True)
```

Figura 8. Eliminación de filas con genero desconocido

MODELADO:

- Se realiza un análisis de la frecuencia de migrantes por género utilizando la biblioteca numpy, matplotlib y seaborn, el código genera un gráfico de barras que representa la frecuencia de migrantes por género a partir de los datos del DataFrame df.

```
In [11]: import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns

Genero = df['Genero'].value_counts().nlargest()

Genero.plot(kind='bar')
plt.xlabel('Genero')
plt.ylabel('Frecuencia')
plt.title('Frecuencia de migrantes por Genero')
plt.show()
```

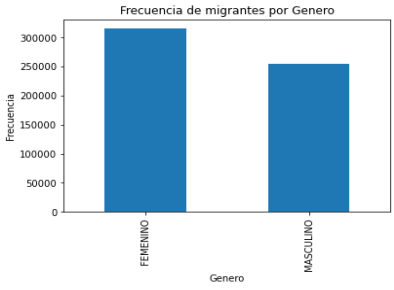

Figura 9. Gráfico de barras por genero

Se puede evidenciar que la mayor frecuencia de personas que han migrado a otros países es del género femenino con respecto al género masculino.

- Se realiza un análisis de la frecuencia de migrantes por área de conocimiento utilizando la biblioteca panda, matplotlib y seaborn, el código genera un gráfico de barras que representa la frecuencia de migrantes por área de conocimiento a partir de los datos del DataFrame df.

```
In [12]: Area_conocimiento = df['Área Conocimiento'].value_counts().nlargest()

Area_conocimiento.plot(kind='bar')
plt.xlabel('Area de Conocimiento')
plt.ylabel('Frecuencia')
plt.title('Frecuencia de migrantes por area de conocimiento')
plt.show()
```

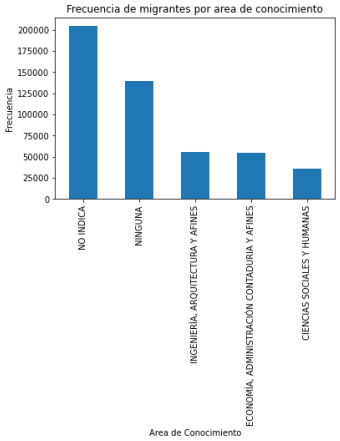

Figura 10. Gráfico de barras por Area de conocimiento.

Esto permite visualizar y comparar la distribución de migrantes en diferentes áreas de conocimiento dando a entender que la gran mayoría de migrantes colombianos no indica su área de conocimiento.

- Se realiza un análisis de la frecuencia de estados civiles de migrantes utilizando la biblioteca panda, matplotlib y seaborn, el código genera un gráfico de barras que representa la frecuencia de estados civiles de migrantes a partir de los datos del DataFrame df.

Esto permite visualizar y comparar la distribución de los estados civiles más comunes entre los migrantes y en la figura 11 se puede evidenciar que la mayor parte de los migrantes colombianos son solteros seguidos de los casados y unión libre, esto determina una tendencia de migración del país de aquellas personas que cuyo grupo de edad es adulto.

```
In [13]: Estado_civil = df['Estado_civil'].value_counts().nlargest(10)

Estado_civil.plot(kind='bar')
plt.xlabel('Estado Civil')
plt.ylabel('Frecuencia')
plt.title('Frecuencia de estados civiles migrantes')
plt.show()
```

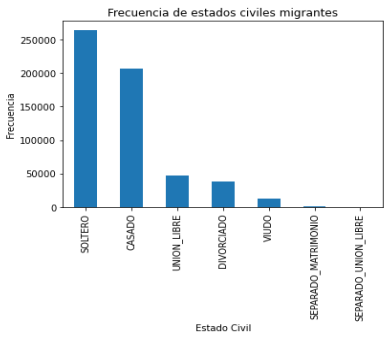

Figura 11. Gráfico de barras por estado civil.

- Se utiliza la biblioteca Plotly para crear un gráfico de pastel interactivo que muestra los 10 países con mayor frecuencia de migrantes, cada sector del pastel representa la proporción de migrantes en un país específico.

```
In [14]: import plotly.graph_objs as go
import plotly.offline as offline

top_10 = df['Pais'].value_counts().nlargest(10)
data = [go.Pie(labels=top_10.index, values=top_10.values, hoverinfo='label+value')]

layout = go.Layout(
    autosize=False,
    width=500,  # Anchura del gráfico en píxeles
    height=500,  # Altura del gráfico en píxeles
    title = 'Top 10 de paises con mayor frecuencia de migrantes'
)

fig = go.Figure(data=data, layout=layout)

offline.iplot(fig, filename='active_Category')
```

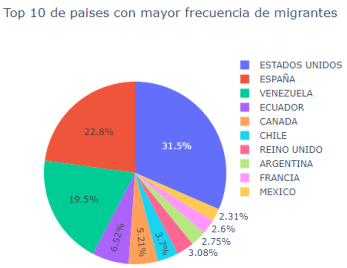

Figura 12. Gráfico de torta con el top 10 de personas por país.

En la gráfica 12 se puede determinar que la gran cantidad de migrantes colombianos se encuentran concentrados en Estados unidos y España con 31.5% y un 22.8% respectivamente, esto evidencia que la probabilidad de oportunidad de progresos en estos países es evidente con respecto a los demás registrados en la base de datos con la que se está trabajando.

- Utiliza las bibliotecas seaborn (sns) y matplotlib (plt) para crear un gráfico de puntos (stripplot) que muestra la relación entre la edad y el nivel educativo en el DataFrame df, Cada punto representa una observación en el DataFrame df, donde la posición en el eje x corresponde a la edad y la posición en el eje y corresponde al nivel educativo. El gráfico ayuda a visualizar la distribución de las edades en relación con los diferentes niveles educativos presentes en los datos.

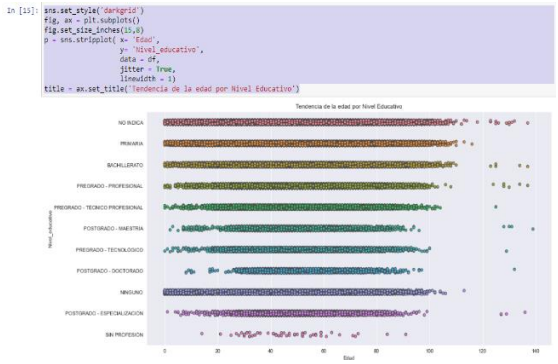

Figura 13. Distribución de edad con respecto a nivel educativo.

En el gráfico 13 se puede evidenciar como se encuentran distribuidas las persona en los niveles educativos con respecto a las edades registradas en la base de datos, donde podemos identificar que en el rango de edades presentes en la base de datos muy pocos se encuentran marcados como sin profesión, mientras que la gran mayoría no indica su nivel educativo.

- Utiliza las bibliotecas seaborn (sns) y matplotlib (plt) para crear un gráfico de puntos (stripplot) que muestra la relación entre la edad y el grupo de edad en el DataFrame df, Cada punto representa una observación en el DataFrame df, donde la posición en el eje x corresponde a la edad y la posición en el eje y corresponde al grupo de edad. El gráfico ayuda a visualizar la distribución de las edades en relación con los diferentes grupos de edad presentes en los datos.

Figura 14. Distribución de edad con respecto a nivel educativo

Se puede observar que la agrupación de grupos de edad con respecto a las especificadas al momento del levantamiento de información es acorde a los grupos de edades reales establecidos.

- Utiliza las bibliotecas seaborn (sns) y matplotlib (plt) para crear un gráfico de puntos (stripplot) que muestra la relación entre la edad y la etnia de las personas en el DataFrame df, Cada punto representa una observación en el DataFrame df, donde la posición en el eje x corresponde a la edad y la posición en el eje y corresponde a la etnia de la persona. El gráfico ayuda a visualizar la distribución de las edades en relación con las diferentes etnias presentes en los datos.

Figura 15. Distribución de edad con respecto a la etnia

Se puede observar que la gran distribución de personas se encuentra entre no registra ninguna etnia, sin etnia registrada y otro e indicándonos que la gran mayoría de migrantes seguido de las anteriores se encuentra en la etnia afrodescendientes.

REGRESION LOGISTICA

La regresión logística es un instrumento estadístico de análisis multivariado, de uso tanto explicativo como predictivo. Resulta útil su empleo cuando se tiene una variable dependiente dicotómica (un atributo cuya ausencia o presencia hemos puntuado con los valores cero y uno, respectivamente) y un conjunto de variables predictoras o independientes, que pueden ser cuantitativas (que se denominan covariables o covariadas) o categóricas. En este último caso, se requiere que sean transformadas en variables "dummy", es decir variables simuladas.[8]

El propósito del análisis consiste en:

predecir la probabilidad de que a alguien le ocurra cierto "evento": por ejemplo, estar desempleado =1 o no estarlo = 0, ser pobre = 1 o no pobre = 0, recibirse de sociólogo =1 o no recibirse = 0). Determinar que variables pesan más para aumentar o disminuir la probabilidad de que a alguien le suceda el evento en cuestión

Regresión logística Prueba 1

Para realizar la regresión logística primero debemos realizar una copia del dataframe original para posteriormente escoger las variables que se van utilizar para el modelo, posteriormente se realiza un tratamiento de datos que consiste en convertir variables categóricas a numéricas para implementarlas en el modelo de la siguiente forma:

• Se determina que se tomará como base que los países de Estados Unidos y España tendrán el valor de 1 y el resto de los países tendrán el valor de 0, esto con la finalidad de convertir esta variable en binaria.

• Para la variable de nivel académico se determina que el valor 1 es que indica algún tipo de estudio y 0 que no indicó el estudio o no aplica, esto con la finalidad de convertir esta variable en binaria.

• Para la variable género se determina que el valor 1 se asigna a los registros que tienen el valor masculino y 0 para los registros que tienen el valor femenino

• Por otra parte la variable de estado civil se convertirá en variables dummy para posteriormente tenerlas en cuenta en el modelo de regresión logística.

se crea un nuevo DataFrame llamado df_final que contiene una selección de columnas del DataFrame original df. Las columnas seleccionadas son "Pais", "Edad", "Nivel_educativo", "Estado_civil", "Género" y "Etnia_persona".

retorna una tupla que indica el número de filas y columnas presentes en df_final. Por lo tanto, df_final.shape se utiliza para conocer la forma o tamaño del DataFrame df_final.

```
In [19]: # Creamos un nuevo dataframe para convertir las variables categorias en numericas
         df_final = df[["Pais","Edad","Nivel_educativo","Estado_civil","Genero","Etnia_persona"]]
         df_final.shape
Out[19]: (569890, 6)
```

Figura 16. Nuevo dataframe con variables significativas para el modelo.

Se realiza algunas transformaciones en el DataFrame

Se convierte la variable "Género" en una variable binaria, donde "Masculino" se representa como 1 y "Femenino" se representa como 0. Utiliza la función apply junto con una función lambda para reemplazar los valores de "MASCULINO" por 1 y "FEMENINO" por 0.

Se convierte la variable "Pais" en una variable binaria. Primero, reemplaza "ESTADOS UNIDOS" por 1 y luego "ESPAÑA" por 1, mientras que todos los demás valores se reemplazan por 0. Utiliza la función apply junto con funciones lambda para realizar estos reemplazos. Luego, utiliza la función np.where para convertir todos los valores que no son "1" en "0". Finalmente, convierte la columna "Pais" en tipo entero utilizando astype(int).

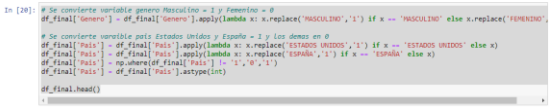

Figura 16. Transformación de variables categóricas a numéricas.

Se cuenta la frecuencia de cada valor único presente en la columna "Nivel Educativo" del DataFrame df_final. Esto crea una serie que contiene los valores únicos como índices y sus frecuencias como valores.

nlargest() es un método de la serie que se aplica después del conteo de valores. Se utiliza sin argumentos, por lo que devuelve los valores más grandes de la serie. En este caso, devuelve los valores únicos más frecuentes en la columna "Nivel Educativo" en orden descendente.

```
In [21]: df_final['Nivel_educativo'].value_counts().nlargest()
Out[21]: NO INDICA                        199238
         BACHILLERATO                     124933
         PREGRADO - PROFESIONAL            92644
         PRIMARIA                          53464
         PREGRADO - TÉCNICO PROFESIONAL    32469
         Name: Nivel_educativo, dtype: int64
```

Figura 17. Frecuencia por Nivel Educativo

Se realiza transformaciones en la columna "Nivel Educativo" del DataFrame df_final. A continuación, se describe su funcionamiento:

Se utiliza la función apply junto con una función lambda para reemplazar el valor "NO INDICA" por "0" en la columna "Nivel_educativo". Esto se realiza con la finalidad de representar los casos en los que no se indica el nivel educativo. Los demás valores de la columna se mantienen sin cambios.

Se utiliza la función np.where para asignar el valor "1" a todos los elementos de la columna "Nivel_educativo" que no sean "0". Esto implica que estos elementos indican algún nivel de estudio. Los elementos que ya tienen el valor "0" se mantienen sin cambios.

Se convierte la columna "Nivel_educativo" al tipo de dato entero utilizando astype(int). Esto asegura que los valores en la columna sean tratados como números enteros en lugar de cadenas de texto.

Figura 18. Transformación de variable nivel educativo.

Se utiliza la función apply junto con una función lambda para reemplazar el valor "NINGUNA" por "0" en la columna "Etnia_persona". Esto se realiza con el objetivo de representar los casos en los que no se indica una etnia específica. Los demás valores de la columna se mantienen sin cambios.

Se utiliza la función np.where para asignar el valor "1" a todos los elementos de la columna "Etnia_persona" que no sean "0". Esto implica que estos elementos indican la pertenencia a alguna etnia. Los elementos que ya tienen el valor "0" se mantienen sin cambios.

Se convierte la columna "Etnia_persona" al tipo de dato entero utilizando astype(int). Esto asegura que los valores en la columna sean tratados como números enteros en lugar de cadenas de texto.

Figura 19. Transformación de variable Etnia de la persona.

Realiza un conteo de los valores únicos presentes en la columna "Etnia_persona" del DataFrame df_final. Luego, utilizando el método nlargest(), devuelve los valores más frecuentes en orden descendente. El número de valores devueltos depende del tamaño de la serie resultante.

Realiza un conteo de los valores únicos presentes en la columna "Estado_civil" del DataFrame df_final. Luego, utilizando el método nlargest(), devuelve los valores más frecuentes en orden descendente. El número de valores devueltos depende del tamaño de la serie resultante.

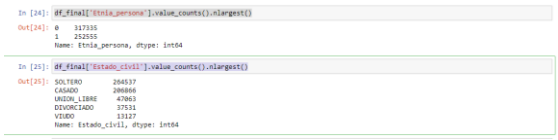

Figura 20. Frecuencia de variables transformadas.

Se utilizan las librerías matplotlib y seaborn para crear y mostrar un mapa de calor (heatmap) de la matriz de correlación del DataFrame df_final.

corrmat = df_final.corr(method = 'pearson') calcula la matriz de correlación entre las variables del DataFrame df_final utilizando el método de correlación de Pearson. Esta matriz es necesaria para crear el mapa de calor.

p = sns.heatmap(corrmat, annot = True, cmap = sns.diverging_palette(202,0,as_cmap = True)) crea el mapa de calor utilizando la función heatmap() de seaborn, se pasa la matriz de correlación corrmat como datos para el mapa de calor. Los parámetros annot=True permiten mostrar los valores de correlación en cada celda del mapa de calor. El parámetro cmap especifica la paleta de colores utilizada, en este caso, se utiliza la función diverging_palette() de seaborn para obtener una paleta de colores divergente.

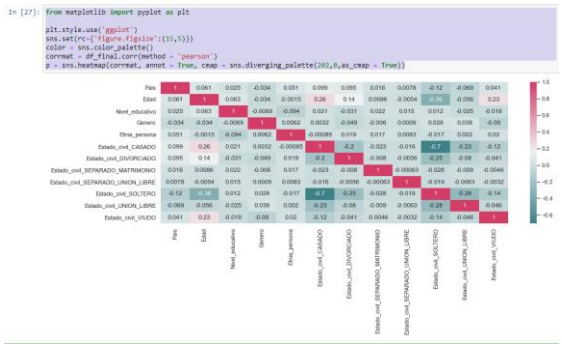

Figura 21. Diagrama de correlación Pearson de variables numéricas.

Se utiliza la librería seaborn para crear un gráfico de dispersión (scatter plot) utilizando los datos del DataFrame df_final.

grafico = sns.scatterplot(data=df_final, x="Edad", y="Pais", hue="Genero") crea el gráfico de dispersión. Se especifica el DataFrame df_final como los datos de entrada. Los ejes x e y del gráfico se definen utilizando las columnas "Edad" y "Pais" respectivamente. El parámetro hue se utiliza para asignar diferentes colores a los puntos del gráfico en función de la columna "Genero".

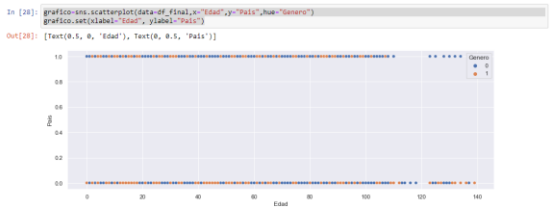

Figura 22. Diagrama de distribución de personas en los países codificados con respecto a la edad y el género.

X = sm.add_constant(X) agrega una columna de constantes a la matriz X. Esto se hace utilizando la función add_constant() de statsmodels.api. La columna de constantes es necesaria para ajustar modelos lineales utilizando statsmodels, ya que representa el término independiente o constante en el modelo.

Y = np.array(df_final[["Pais"]]) selecciona la columna "Pais" del DataFrame df_final y la asigna a la variable Y. Esto se hace utilizando la notación de corchetes y una lista con el nombre de la columna "Pais".

Luego, la función np.array() de la librería numpy se utiliza para convertir la selección en un array de numpy. Esto puede ser necesario para realizar operaciones o análisis posteriores con los datos.

Figura 23. Asignación de columnas a las variables X e Y.

Se utiliza la función de regresión logística Logit del módulo sm (statsmodels.api) para ajustar un modelo de regresión logística. A continuación.

ajuste = sm.Logit(Y, X) crea un objeto Logit que representa el modelo de regresión logística. Se

especifica la variable dependiente (Y) y las variables independientes (X) como argumentos de la función Logit.

ajuste = ajuste.fit() ajusta el modelo de regresión logística utilizando el método fit() del objeto ajuste. Este método estima los parámetros del modelo basándose en los datos proporcionados.

print(ajuste.summary()) muestra un resumen del resultado del ajuste del modelo utilizando el método summary() del objeto ajuste. El resumen incluye estadísticas y resultados relevantes como los coeficientes estimados, los valores p, el pseudo R-cuadrado y otras métricas asociadas al modelo ajustado.

```
In [33]: ajuste=sm.Logit(Y,X) #Funcion para regresion Logica
         ajuste=ajuste.fit()
         print(ajuste.summary())

Optimization terminated successfully.
         Current function value: 0.686567
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:               569890
Model:                          Logit   Df Residuals:                   569886
Method:                           MLE   Df Model:                            3
Date:                Fri, 19 May 2023   Pseudo R-squ.:                0.003382
Time:                        06:49:08   Log-Likelihood:            -3.9127e+05
converged:                       True   LL-Null:                   -3.9260e+05
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3038      0.006    -52.275      0.000      -0.315      -0.292
x1             0.1255      0.006     22.297      0.000       0.114       0.137
x2            -0.1366      0.005    -25.450      0.000      -0.147      -0.126
x3             0.2202      0.005     40.853      0.000       0.210       0.231
==============================================================================
```

Figura 24. Resumen del resultado del ajuste del modelo de regresión logística Prueba 1.

"Current function value": Es el valor actual de la función objetivo del modelo, que en este caso es 0.686567.

"Iterations": Indica el número de iteraciones que tomó el algoritmo de optimización para converger al resultado, en el caso de estudio es 4.

"Dep. Variable": Hace referencia a la variable dependiente o variable respuesta en el modelo, variable y.

"No. Observations": Es el número total de observaciones o casos utilizados en el análisis, que en este caso es 569,890.

"Model": Indica el tipo de modelo utilizado, que es una regresión logística.

"Df Residuals": Grados de libertad de los residuos, es decir, el número de observaciones menos el número de parámetros estimados en el modelo.

"Date": Fecha en la que se realizó el análisis, viernes 19 de mayo del 2023.

"Pseudo R-squ.": Indica el valor del pseudo R-cuadrado, que mide la calidad del ajuste del modelo. En este caso, el valor es 0.003382.

"converged": Indica si el algoritmo de optimización converge al resultado, en este caso es True, lo que significa que convergen.

"Covariance Type": Indica el tipo de covarianza utilizada en el modelo, en este caso es no robusta.

"LLR p-value": Es el valor p asociado al estadístico de prueba de la diferencia entre el logaritmo de la función de verosimilitud del modelo ajustado y el logaritmo de la función de verosimilitud del modelo nulo. En este caso, el valor p es muy pequeño (0.000), lo cual indica que el modelo ajustado es significativamente mejor que el modelo nulo.

"const", "x1", "x2", "x3": Son los nombres de las variables independientes incluidas en el modelo.

"coef": Son los coeficientes estimados para cada una de las variables independientes. Representan el cambio esperado en el logaritmo de la odds ratio para un aumento de una unidad en la variable independiente, manteniendo constantes las demás variables.

"std err": Son los errores estándar de los coeficientes estimados.

Regresión logística Prueba 2

Para realizar la regresión logística primero debemos realizar una copia del dataframe original para posteriormente escoger las variables que se van a utilizar para el modelo, posteriormente se realiza un tratamiento de datos que consiste en convertir variables categóricas a numéricas para implementarlas en el modelo de la siguiente forma:

• Se determina que se tomará como base sólo los países de Estados Unidos y España los cuales según sus frecuencias solo los de mayor tendencia

valor de 1 Corresponde a Estados Unidos

Valor de 0 corresponde a España

• Para la variable de nivel académico se escogen los valores de bachillerato y pregrado profesión:

valor de 1 Corresponde a bachillerato

Valor de 0 corresponde a pregrado profesión

• Para la variable género se determina que el valor 1 se asigna a los registros que tienen el valor masculino y 0 para los registros que tienen el valor femenino

• Por otra parte la variable de estado civil se convertirá en variables dummy de tipo numérico 1 y 0 para posteriormente tenerlas en cuenta en el modelo de regresión logística.

```
Optimization terminated successfully.
         Current function value: 0.627252
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:               105845
Model:                          Logit   Df Residuals:                   105841
Method:                           MLE   Df Model:                            3
Date:                Fri, 19 May 2023   Pseudo R-squ.:                0.08297
Time:                        06:49:14   Log-Likelihood:                -66391.
converged:                       True   LL-Null:                       -72399.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1115      0.023     -4.793      0.000      -0.157      -0.066
x1            -1.3269      0.014    -96.099      0.000      -1.354      -1.300
x2             0.0232      0.000     52.453      0.000       0.022       0.024
x3             0.0241      0.003      8.730      0.000       0.019       0.030
==============================================================================
```

Figura 25. Resumen del resultado del ajuste del modelo de regresión logística Prueba 2.

"Current function value": Es el valor actual de la función objetivo del modelo, que en este caso es 0.627252.

"Iterations": Indica el número de iteraciones que tomó el algoritmo de optimización para converger al resultado, en el casi de estudio es de 5.

"Dep. Variable: Hace referencia a la variable dependiente o variable respuesta en el modelo, variable y.

"No. Observations": Es el número total de observaciones o casos utilizados en el análisis, que en este caso es 105,845.

"Model": Indica el tipo de modelo utilizado, que es una regresión logística.

"Df Residuals": Grados de libertad de los residuos, es decir, el número de observaciones menos el número de parámetros estimados en el modelo.

"Date": Fecha en la que se realizó el análisis, viernes 19 de mayo del 2023.

"Pseudo R-squ.": Indica el valor del pseudo R-cuadrado, que mide la calidad del ajuste del modelo. En este caso, el valor es 0.08297.

"converged": Indica si el algoritmo de optimización converge al resultado, en este caso es True, lo que significa que convergen.

"Covariance Type": Indica el tipo de covarianza utilizada en el modelo, en este caso es no robusta.

"LLR p-value": Es el valor p asociado al estadístico de prueba de la diferencia entre el logaritmo de la función de verosimilitud del modelo ajustado y el logaritmo de la función de verosimilitud del modelo nulo. En este caso, el valor p es muy pequeño (0.000), lo cual indica que el modelo ajustado es significativamente mejor que el modelo nulo.

"const", "x1", "x2", "x3": Son los nombres de las variables independientes incluidas en el modelo.

"coef": Son los coeficientes estimados para cada una de las variables independientes. Representan el cambio esperado en el logaritmo de la odds ratio para un aumento de una unidad en la variable independiente, manteniendo constantes las demás variables.

"std err": Son los errores estándar de los coeficientes estimados.

EVALUACIÓN:

Para evaluar qué modelo de regresión logística es más explicativo, se puede considerar varios criterios:

El Valor del pseudo R-cuadrado: El pseudo R-cuadrado es una medida de la calidad del ajuste del modelo. Cuanto más cercano a 1, mejor es el ajuste, el primer modelo tiene un pseudo R-cuadrado de 0.003382, mientras que el segundo modelo tiene un pseudo R-cuadrado de 0.08297. Esto indica que el segundo modelo explica una mayor proporción de la variabilidad en la variable dependiente.

Significancia de los coeficientes: Observa los valores asociados a los coeficientes de las variables independientes. Un valor p pequeño (generalmente menor a 0.05) indica que la variable independiente tiene un efecto significativo en la variable dependiente. Verifica si los coeficientes de las variables independientes son significativos en ambos modelos y si hay diferencias en la magnitud y dirección de los coeficientes entre los dos modelos.

Log-Likelihood: El valor del logaritmo de la función de verosimilitud maximizada (Log-Likelihood) puede usarse como medida de comparación entre modelos. Un valor más alto indica un mejor ajuste. Sin embargo, ten en cuenta que este valor solo puede compararse entre modelos con la misma cantidad de variables.

AIC y BIC: El criterio de información de Akaike (AIC) y el criterio de información bayesiano (BIC) son medidas que penalizan la complejidad del modelo. Cuanto más bajo sea el valor de AIC o BIC, mejor se considera el modelo. Compara los valores de AIC y BIC entre los dos modelos para evaluar su rendimiento.

RESULTADOS:

Los resultados muestran que la mayoría de los colombianos en el exterior son jóvenes adultos, con una edad promedio de 32 años. Además, Estados Unidos es el destino más popular para la migración, seguido de España y Chile. Los hombres representan el 52% de la diáspora colombiana y la mayoría de ellos tienen educación universitaria.

CONCLUSIONES:

Este estudio de caso demuestra la eficacia de la metodología CRISP-DM y la utilización del big data para analizar la información demográfica de los colombianos en el exterior. Los patrones demográficos y de migración identificados pueden ser utilizados para mejorar las políticas públicas y los programas de apoyo a la diáspora colombiana.

El estudio realizado utiliza la regresión logística como una herramienta estadística para el análisis multivariado. La regresión logística es utilizada tanto con propósitos explicativos como predictivos cuando se tiene una variable dependiente dicotómica y un conjunto de variables predictoras o independientes.

En el estudio, se realiza un preprocesamiento de los datos antes de aplicar la regresión logística. Se transforman las variables categóricas en variables binarias o numéricas para poder utilizarlas en el modelo. Además, se realiza una selección de columnas del DataFrame original y se crea un nuevo DataFrame llamado "df_final" que contiene estas columnas seleccionadas.

Luego, se lleva a cabo un análisis exploratorio de los datos. Se cuenta la frecuencia de los valores únicos en ciertas columnas, se realizan transformaciones en las variables y se visualizan los resultados utilizando gráficos como el mapa de calor y el gráfico de dispersión.

Finalmente, se ajusta un modelo de regresión logística utilizando la función Logit del módulo statsmodels.api. Se muestra un resumen de los resultados del ajuste, que incluye estadísticas como los coeficientes estimados, los valores p y el pseudo R-cuadrado.

En conclusión, el estudio utiliza la regresión logística como una herramienta estadística para analizar la relación entre variables

predictoras e una variable dependiente dicotómica. Se realiza un preprocesamiento de los datos y se lleva a cabo un análisis exploratorio antes de ajustar el modelo de regresión logística. Los resultados del ajuste son presentados y analizados.

REFERENCIAS:

[1] Haya, P (noviembre,2021), La metodología CRISP-DM en ciencia de datos, https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/#_ftn3

[2] Chapman,P(2010) Guía paso a paso de Minería de Datos, https://www.dataprix.com/files/Metodologia_CRISP_DM.pdf

[3] Azevedo, A(2008), KDD, SEMMA and CRISP-DM: a parallel overview, https://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf

[4] Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: a parallel overview.IADSDM.,https://www.sciencedirect.com/science/article/pii/S1877050921002416/pdf?md5=34d7ce6ba598594c11c16770ac53a4e8&pid=1-s2.0-S1877050921002416-main.pdf

[5] Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (Vol. 1, pp. 29-39). http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

[6] Shafique, U., & Qaiser, H. (2014). A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). International Journal of Innovation and Scientific Research, 12(1), 217-222. https://shrtco.de/L5DVG

[7] González Duque, R. (2011). Python para todos. https://repositorio.uci.cu/bitstream/123456789/10206/1/Python_para_todos.pdf

[8] Chitarroni, H. (2002). La regresión logística.https://racimo.usal.edu.ar/83/1/Chitarroni17.pdf

ANEXOS:

Se anexan url de GitHub donde se encuentra alojados los archivos que hacen parte del desarrollo de este articulo en la siguiente url:

https://github.com/Gusandreti/ArticuloBigData