

CoderHouse

Curso Data Science

Informe del Proyecto Final

**Análisis socioeducativo de los habitantes de la Ciudad de
Buenos Aires**

Profesor: Damian Dapuetto

Tutor: Héctor Alonso

Grupo de Trabajo:

Lucia Buzzeo, Lucia Hukovsky,
Jose Saint German, Juan Martín Carini

10 de septiembre de 2022



Índice

| | |
|--|-----------|
| 1. Introducción | 3 |
| 1.1. Resumen del proyecto | 3 |
| 1.2. Objetivos del proyecto | 3 |
| 1.3. Definición de la fuente de información | 3 |
| 2. Planificación | 4 |
| 3. Introducción a las variables: Análisis exploratorio de los datos | 5 |
| 3.1. Análisis univariado | 7 |
| 3.1.1. Género y edad | 7 |
| 3.1.2. Comuna | 7 |
| 3.1.3. Ingreso familiar per cápita | 8 |
| 3.1.4. Años de escolaridad | 8 |
| 3.1.5. Máximo nivel educativo (Target) | 9 |
| 3.2. Análisis bivariado | 10 |
| 3.2.1. Comparación entre variables numéricas | 11 |
| 3.2.2. Comparación de variables categóricas con numéricas | 11 |
| 3.2.3. Variable numéricas con comuna | 14 |
| 3.3. Análisis multivariado | 16 |
| 4. Modelos analíticos | 19 |
| 4.1. Tratados de nulos | 19 |
| 4.2. Target | 20 |
| 4.2.1. Borrado de variables | 20 |
| 4.3. Procesamiento | 21 |
| 4.4. Árbol de decisión | 21 |
| 4.4.1. Primer modelo | 21 |
| 4.4.2. Segundo modelo | 22 |
| 4.4.3. Gridsearch con CV | 22 |
| 4.5. Random Forest Classifier | 23 |
| 4.5.1. Primer modelo | 23 |
| 4.5.2. Segundo modelo | 24 |
| 4.5.3. Gridsearch con CV | 24 |
| 5. Conclusiones | 25 |

1. Introducción

1.1. Resumen del proyecto

La ciudadanía es un concepto jurídico, filosófico y político que ha sido creado para designar a una persona física que constituye una sociedad o entidad territorial. Para las personas que forman parte de una comunidad, ciudadanos, resulta de suma importancia sentirse representados por los demás integrantes de la misma, mediante políticas públicas que abarquen sus necesidades y requerimientos.

La toma de datos demográficos y la estadística son dos herramientas primordiales a la hora de identificar requerimientos de los integrantes de una comunidad. Dichas herramientas describen, de forma cuantitativa, a la sociedad bajo estudio. Precisamente, los censos y la estadística son la fuente primaria de información para la planificación económica y social de una población, por parte de sus representantes.

En el caso particular de Argentina, el Instituto Nacional de Estadística y Censos (INDEC) es el organismo público que ejerce la dirección superior de todas las actividades estadísticas oficiales que se realizan en el país. La información que produce el INDEC es una herramienta básica para la planificación de políticas públicas, así como para las investigaciones y proyecciones que se realizan en los ámbitos académico y privado.

Al adentrarse y estudiar los índices correspondientes a uno de los ejes principales, educación, en un territorio delimitado, Ciudad Autónoma de Buenos Aires, se ha encontrado una gran limitación relacionada con su acceso inequitativo para los diferentes actores de la sociedad.

Este hecho tiene consecuencias de índole social y económico para la población. Sin embargo, la principal problemática se da a nivel individual, y radica en el impedimento al acceso educativo para un porcentaje de la sociedad. Esto no ha resultado una novedad para el grupo, pero sí ha dado el pie a la búsqueda de una respuesta teórica a dicha disparidad, en concreto, a descubrir las principales variables que afectan el nivel educativo.

El análisis realizado en el marco del presente proyecto podría establecer una base de requerimientos que permitan generar políticas públicas efectivas, no solo en el ámbito educativo, sino en el económico, cultural, social y geográfico, entre otros.

1.2. Objetivos del proyecto

En el presente proyecto se persigue el objetivo de descubrir las principales variables intervinientes en el nivel máximo educativo alcanzado por la población de la Ciudad Autónoma de Buenos Aires (CABA).

En concreto, con este proyecto se plantea el análisis de variables que podrían contribuir al nivel máximo educativo alcanzado por cada individuo en CABA. Las mismas se estudian tanto de forma independiente como entrelazadas. El alcance de dicho objetivo permitiría identificar posibles causas que derivan en la desigualdad de acceso educativo, y, en última instancia, generar políticas públicas eficaces que permitan subsanar dicha problemática.

Y por último intentaremos generar un modelo de predicción aplicado a nuestra variable target “Nivel Máximo Educativo”, esto lo haremos implementando los siguientes modelos de clasificación:

- **Árbol de decisión:** que construye un árbol durante el entrenamiento que es el que aplica a la hora de realizar la predicción.
- **Bosque Aleatorio:** que es un conjunto (ensemble) de árboles de decisión combinados con bagging.

A partir de la obtención del mejor árbol de decisión y el mejor bosque aleatorio tomaremos la decisión de cuál de los dos es el mejor algoritmo para lograr los objetivos de este trabajo.

1.3. Definición de la fuente de información

Para trabajar esta problemática, se ha recurrido a la [Encuesta Anual de Hogares](#) del Gobierno de la Ciudad de Buenos Aires para el año 2019. El dataset está disponible en la base de datos abiertos del GCBA.

Esta encuesta contiene información demográfica, social, económica, educativa y de salud de 14319 habitantes de la Ciudad, la cual es una muestra representativa que permite obtener un vistazo de la población de la Ciudad.

2. Planificación

Estructura de los trabajos

Este trabajo se ha dividido en 3 partes:

1. **Introducción a las variables del problema:** Se realiza un análisis de las variables del dataset. En el mismo se busca conocer su performance dentro del dataset. A la vez, se investiga cómo las variables interactúan entre sí. Esta parte es lo que se conoce como análisis univariado, bivariado y multivariado.
2. **Modelos analíticos:** En esta sección se entrenan diversos modelos analíticos y algoritmos que sirven para alcanzar los objetivos seteados para el presente proyecto. Como la variable objetivo es categórica, se realizan modelos de clasificación.
3. **Conclusión:** Se alcanzan conclusiones finales sobre los hallazgos. Además, se discuten posibles limitaciones y se plantean futuras líneas de análisis, a partir del análisis presente.

3. Introducción a las variables: Análisis exploratorio de los datos

Una vez cargado el dataset con el que vamos a trabajar, miramos sus variables, el tipo que son y si tienen nuls:

```
RangeIndex: 14319 entries, 0 to 14318
Data columns (total 31 columns):
#   Column                                Non-Null Count  Dtype
0   id                                     14319 non-null  int64
1   nhogar                               14319 non-null  int64
2   miembro                             14319 non-null  int64
3   comuna                              14319 non-null  int64
4   dominio                             14319 non-null  object
5   edad                                14319 non-null  int64
6   sexo                                14319 non-null  object
7   parentesco_jefe                     14319 non-null  object
8   situación_conyugal                 14318 non-null  object
9   num_miembro_padre                  14319 non-null  object
10  num_miembro_madre                  14319 non-null  object
11  estado_ocupacional                 14319 non-null  object
12  cat_ocupacional                    14319 non-null  object
13  calidad_ingresos_lab                14319 non-null  object
14  ingreso_total_lab                  14319 non-null  int64
15  calidad_ingresos_no_lab             14319 non-null  object
16  ingreso_total_no_lab                14319 non-null  int64
17  calidad_ingresos_totales            14319 non-null  object
18  ingresos_totales                   14319 non-null  int64
19  calidad_ingresos_familiares         14319 non-null  object
20  ingresos_familiares                14319 non-null  int64
21  ingreso_per_capita_familiar         14319 non-null  int64
22  estado_educativo                   14319 non-null  object
23  sector_educativo                   14316 non-null  object
24  nivel_actual                        14319 non-null  object
25  nivel_max_educativo                 13265 non-null  object
26  años_escolaridad                   14257 non-null  object
27  lugar_nacimiento                   14318 non-null  object
28  afiliacion_salud                   14315 non-null  object
29  hijos_nacidos_vivos                6535 non-null  object
30  cantidad_hijos_nac_vivos           14319 non-null  object
dtypes: int64(10), object(21)
memory usage: 3.4+ MB
```

Ahora, en base a los datos arrojados por la tabla de arriba, generamos diversas transformaciones de variables, así como la creación de la variable “Target”, pues es la que usaremos para todo el análisis:

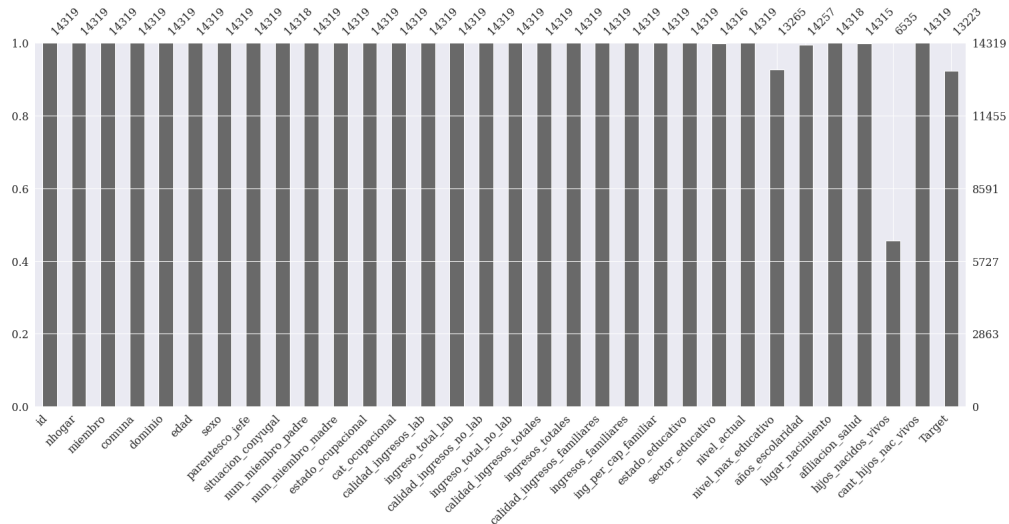
- Creamos la variable “Target” y le asignamos la variable “nivel_max_educativo”.
- En la variable “Target”, reducimos su dimensionalidad intercambiando los valores:
 - “Secundario/medio común” y “EGB (1° a 9° año)” por “sec_completo”,
 - “Primario especial” y “Primario común” por “prim_completo”,
 - “Sala de 5” por “inicial”,
 - “Otras escuelas especiales” por “superior”,
 - y por último a “No corresponde” por nulos.
- Reemplazamos los valores de “años_escolaridad” para que todos sean numéricos.
- En la variable “cantidad_hijos_nac_vivos” cambiamos el valor “no corresponde” como nulo, para luego cambiar el tipo de variable a entero.
- Las variables “comuna”, “id”, “nhogar” y “miembro” son de tipo numérico, pero deberían ser categóricas, por lo tanto transformamos su tipo a string.
- Por último renombramos algunas variables para que sean más cortas:

- “dominio_Villas_de_emergencia” por “dominio_villas”,
- “ingreso_per_capita_familiar” por “ing_per_cap_familiar”,
- “cantidad_hijos_nac_vivos” por “cant_hijos_nac_vivos”.

A continuación detallamos un diccionario de las variables:

- **“id”** : Clave que identifica la vivienda,
- **“nhogar”** : La variable id + nhogar = clave que identifica a cada hogar”,
- **“miembro”** : Variables id + nhogar + miembro = clave que identifica cada persona”,
- **“comuna”** : Comuna donde reside la persona encuestada,
- **“edad”** : Edad de la persona encuestada,
- **“sexo”** : Sexo de la persona encuestada,
- **“parentesco_jefe”** : Parentesco entre la persona encuestada y el jefe de hogar,
- **“situacion_conyugal”** : Situación conyugal de la persona encuestada,
- **“num_miembro_padre”** : Número de miembro que corresponde al padre,
- **“num_miembro_madre”** : Número de miembro que corresponde a la madre,
- **“estado_ocupacional”** : Situación ocupacional de la persona encuestada,
- **“cat_ocupacional”** : Categoría ocupacional de la persona encuestada,
- **“calidad_ingresos_lab”** : Calidad de la declaración de ingresos laborales totales,
- **“ingreso_total_lab”** : Ingreso total laboral percibido el mes anterior,
- **“calidad_ingresos_no_lab”** : Calidad de la declaración de ingresos no laborable totales”,
- **“ingreso_total_no_lab”** : Ingreso total no laboral percibido el mes anterior,
- **“calidad_ingresos_totales”** : Calidad de ingresos totales individuales,
- **“ingresos_totales”** : Ingreso total individual percibido el mes anterior,
- **“calidad_ingresos_familiares”** : Calidad de ingresos totales familiares,
- **“ingresos_familiares”** : Ingresos totales familiares percibido el mes anterior,
- **“ing_per_cap_familiar”** : Ingreso familiar per capita percibido el mes anterior,
- **“estado_educativo”** : Asistencia (pasada o presente) o no a algún establecimiento educativo”,
- **“sector_educativo”** : Sector al que pertenece el establecimiento educativo a que asiste”,
- **“nivel_actual”** : Nivel cursado al momento de la encuesta,
- **“nivel_max_educativo”** : Máximo nivel educativo que se cursó,
- **“años_escolaridad”** : Años de escolaridad alcanzados,
- **“lugar_nacimiento”** : Lugar de nacimiento de la persona encuestada,
- **“afiliacion_salud”** : Afiliación de salud de la persona encuestada,
- **“hijos_nacidos_vivos”** : Tiene o tuvo hijos nacidos vivos,
- **“cant_hijos_nac_vivos”** : Cantidad de hijos nacidos vivos,
- **“dominio”** : ¿La vivienda se ubica en una villa de emergencia?,
- **“Target”** : Nivel máximo educativo.

Y comenzamos el análisis EDA del mismo, primero analizando los nulos:

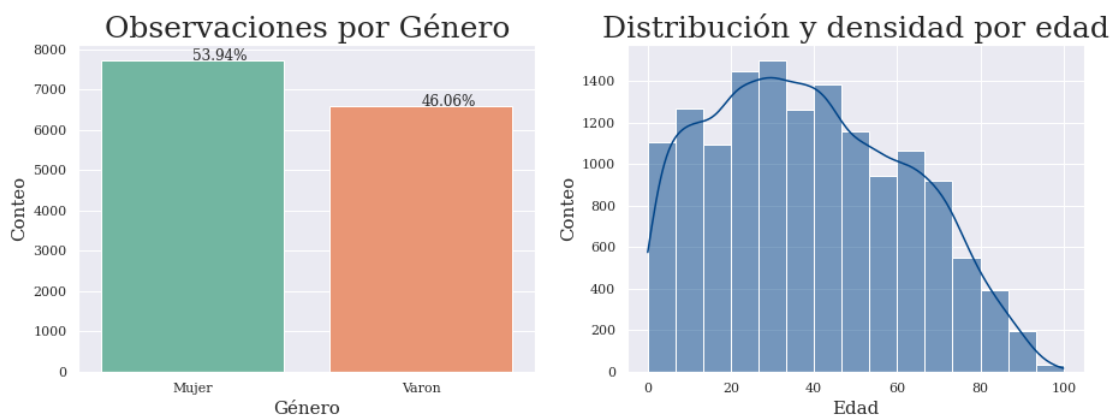


Así, detectamos que nuestra variable target tiene 1054 valores nulos. Es importante tener este dato presente cuando queramos correr un algoritmo de clasificación.

3.1. Análisis univariado

3.1.1. Género y edad

Comenzamos con un pantallazo general sobre las primeras cualidades de los datos, como muestra representativa para la EPH, sobre quiénes son los ciudadanos representados en el dataset.

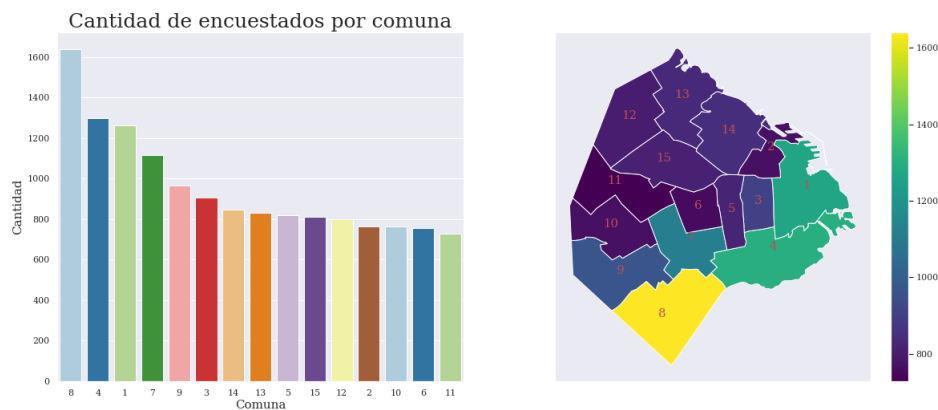


En la variable género los datos parecen equilibrados en las categorías. Para el caso de la variable “edad”, la distribución se asemeja a la de una normal.

3.1.2. Comuna

Seguimos observando la variable “comuna”. En la misma se muestra la comuna de la Ciudad de Buenos Aires del entrevistado, de manera de tener una ubicación geográfica. Consideramos importante revisar esta variable ya que tenemos como hipótesis que el nivel educativo alcanzado puede estar dependiendo de la zona geográfica de la ciudad en la que se encuentra el entrevistado.

Para esto vamos a generar un mapa, así que utilizaremos el mapa de comunas de la Ciudad de Buenos Aires, transformamos las variables que vamos a usar para joinear el mapa con la base de manera que coincidan, transformamos la base para contabilizar la frecuencia con la que aparece cada comuna en la base. Y por último unimos ambos datasets y generamos una nueva variable con las coordenadas para poder agregar etiquetas en el centro geográfico de cada comuna, que nos da como resultado los siguientes gráficos:

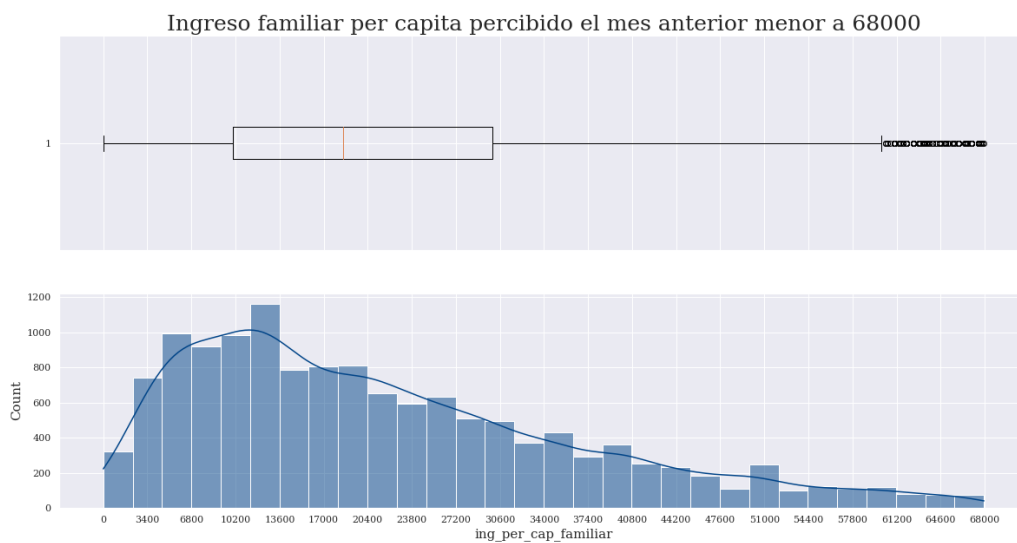


Observando ambos gráficos vemos que las comunas 1,4,7 y 8 tienen mayor cantidad de casos. Queda por verse si en posteriores análisis es necesario abordar esta diferencia para evitar sesgos. Para eso, será necesario tomar en cuenta el porcentaje de la población total de cada comuna.

3.1.3. Ingreso familiar per cápita

Ahora probamos con observar los ingresos familiares. Creemos que puede ser un indicador interesante del nivel educativo.

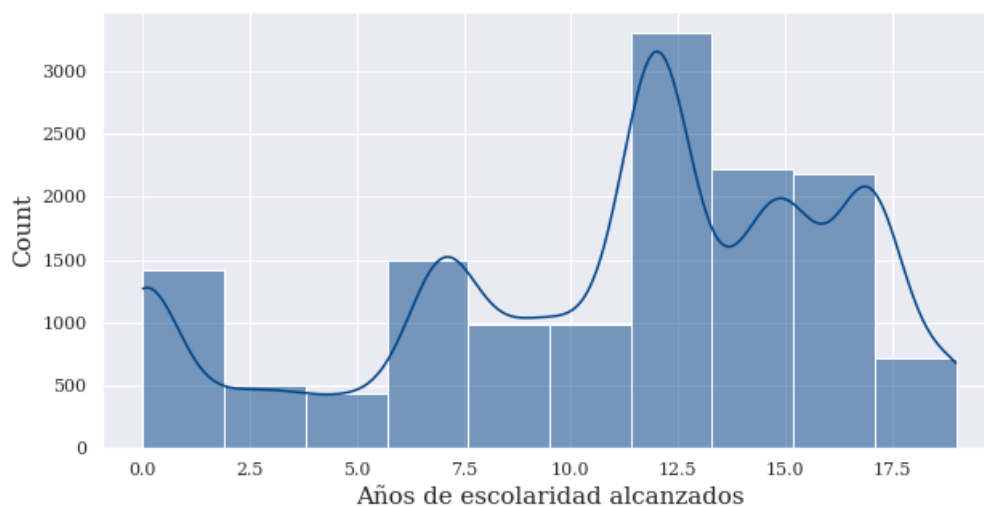
Para esto, armamos una función para graficar y jugar con el nivel del filtrado de la variable y obtener un histograma que permita apreciar mejor la distribución de la variable sin tantos outliers:



Y luego de remover los outliers la distribución de los ingresos familiares sigue estando **sesgada**.

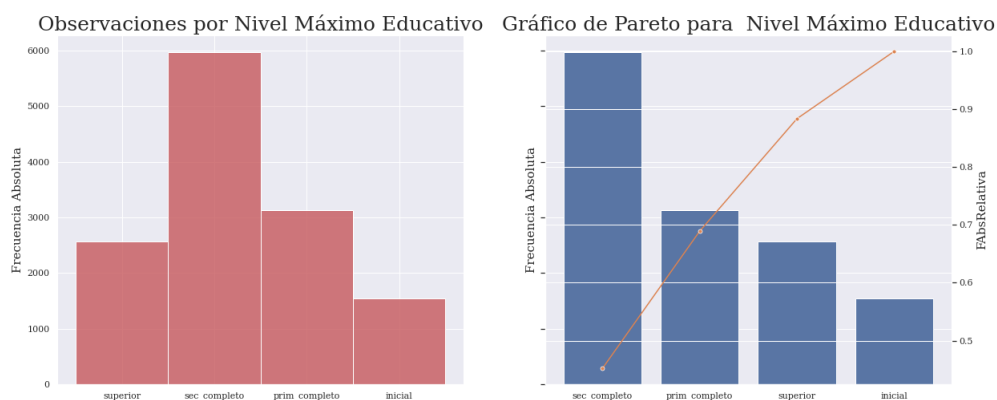
3.1.4. Años de escolaridad

Analizamos mediante un gráfico de barras los años de escolaridad alcanzados por los encuestados:



A simple vista se observan tres "picos": en el valor mínimo, alrededor del 7.5 y alrededor del 12.5. Podemos inferir que estos tres casos corresponden a no tener estudios, solo haber transcurrido el primario y haber transcurrido hasta la educación secundaria, respectivamente.

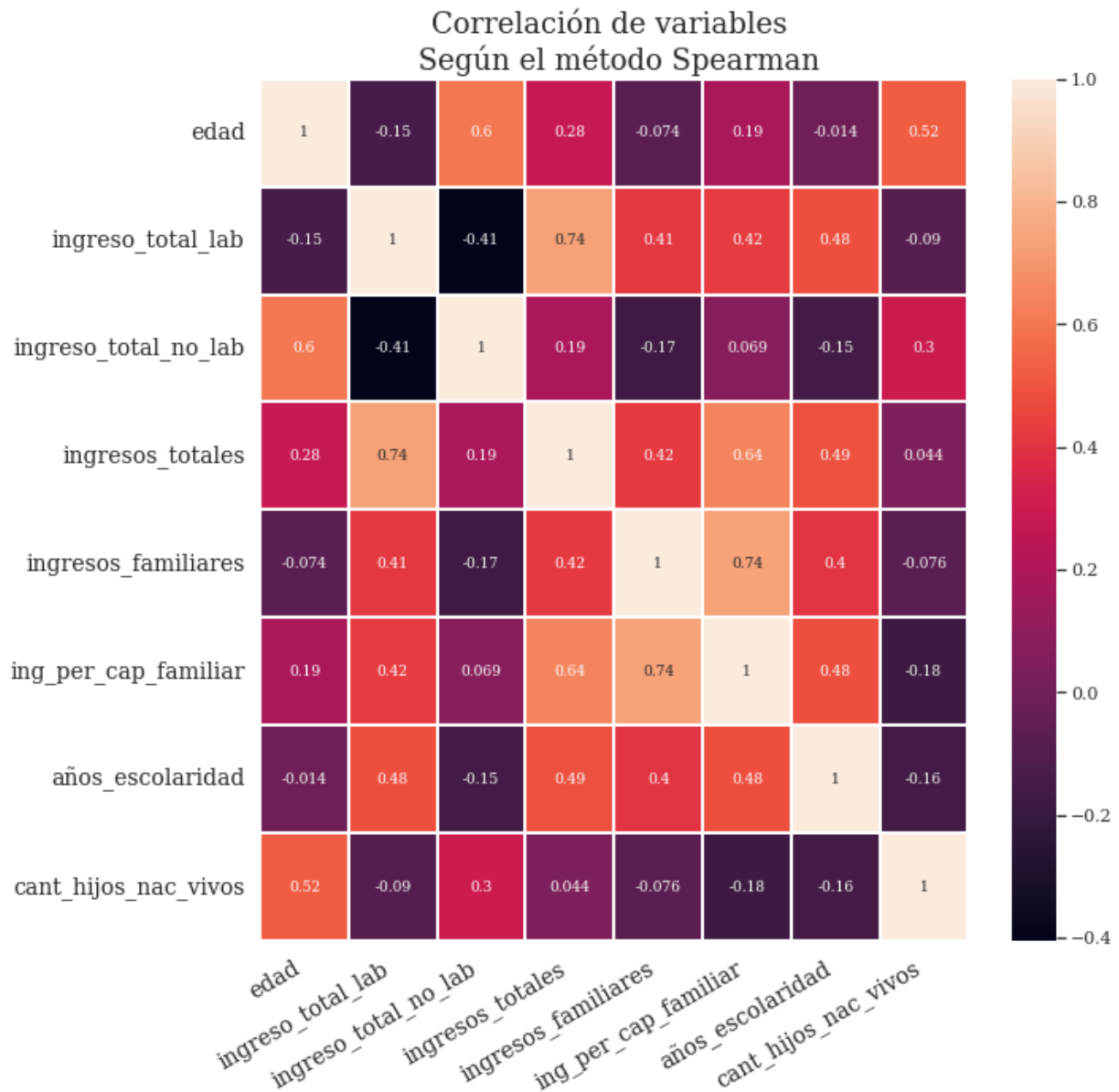
3.1.5. Máximo nivel educativo (Target)



Podemos observar que el nivel máximo educativo más alcanzado es el secundario completo, seguido por el primario. Contrario de lo que habíamos intuido anteriormente, el nivel superior quedó en tercer lugar. Adicionalmente, el nivel secundario y primario explican casi el 77 % de los datos.

3.2. Análisis bivariado

Para comenzar el análisis bivariado del problema, realizamos diferentes heat maps para ver si algo nos llama la atención entre las variables numéricas.

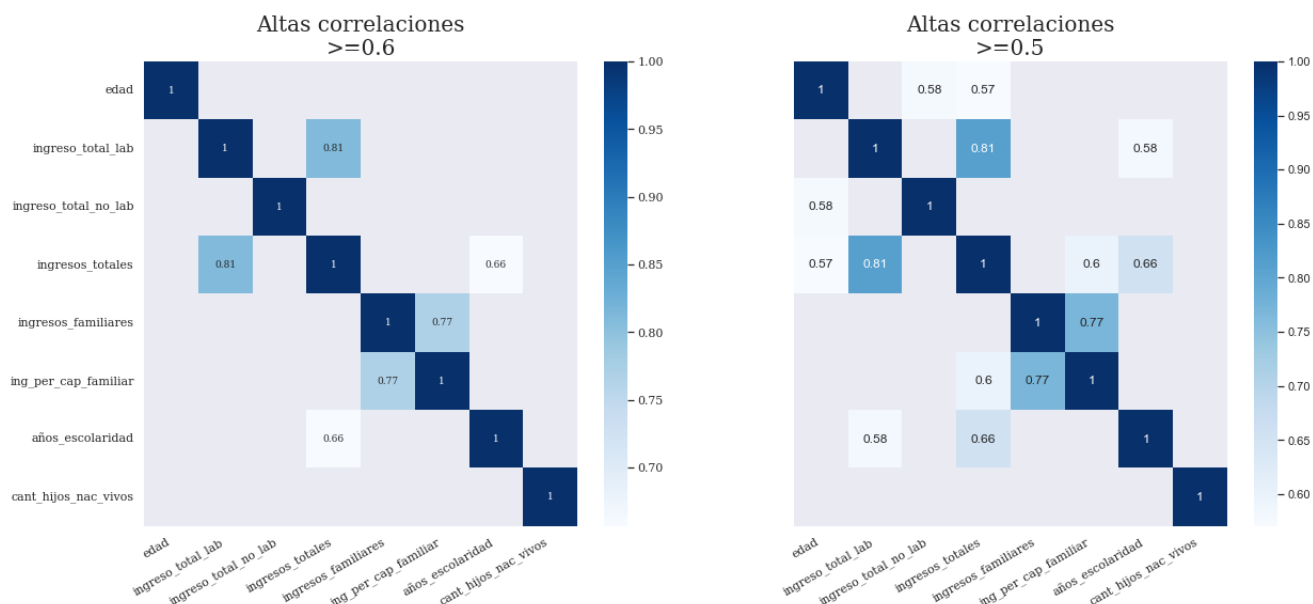


A simple vista, no se observan fuertes correlaciones.

Podemos notar que la variable “años.escolaridad” correlaciona moderadamente bien con variables relacionadas al ingreso.

La principal correlación positiva es “años.escolaridad” con ingreso familiar per cápita (“ing_per_cap.familiar”), lo cual hace sentido teórico.

Ahora, utilizando el método para solo graficar en base a un threshold vemos que los años de escolaridad alcanzados por los entrevistados tienen algo relación (66 %) con la variable “ingresos_totales”



Por último corremos una tabla de correlación y filtramos las de valores más altos

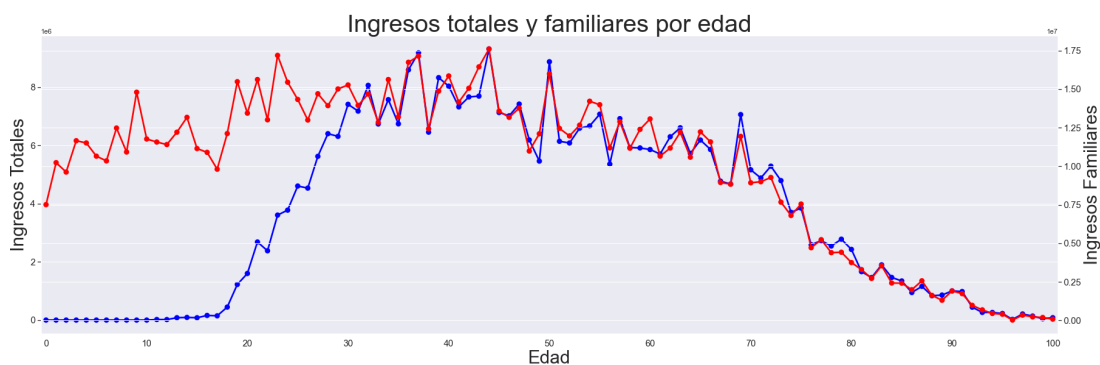
| | Variable 1 | Variable 2 | Correlación |
|---|---------------------|----------------------|-------------|
| 2 | ingreso_total_lab | ingresos_totales | 0.80 |
| 6 | ingresos_familiares | ing_per_cap_familiar | 0.76 |
| 4 | ingresos_totales | ing_per_cap_familiar | 0.62 |
| 5 | ingresos_totales | años_escolaridad | 0.60 |
| 1 | edad | ingreso_total_no_lab | 0.57 |
| 7 | años_escolaridad | Target | 0.57 |
| 3 | ingreso_total_lab | años_escolaridad | 0.54 |

En conclusión podemos ver que:

- Como es esperable, hay alta correlación entre las variables relacionadas al ingreso.
- A su vez, encontramos una alta correlación (66 %) entre los ingresos y los años de escolaridad.
- Y también observamos una relación positiva entre la edad y los ingresos totales.

3.2.1. Comparación entre variables numéricas

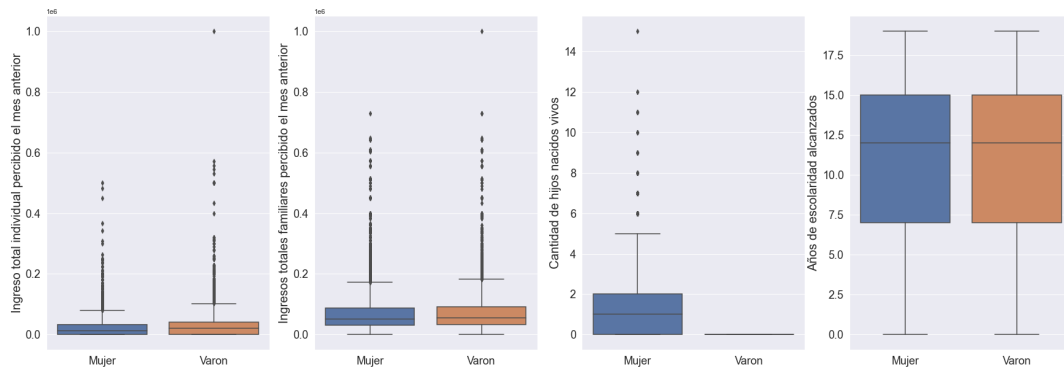
En primer lugar, observaremos la relación entre los ingresos totales de cada hogar y los ingresos familiares por edad:



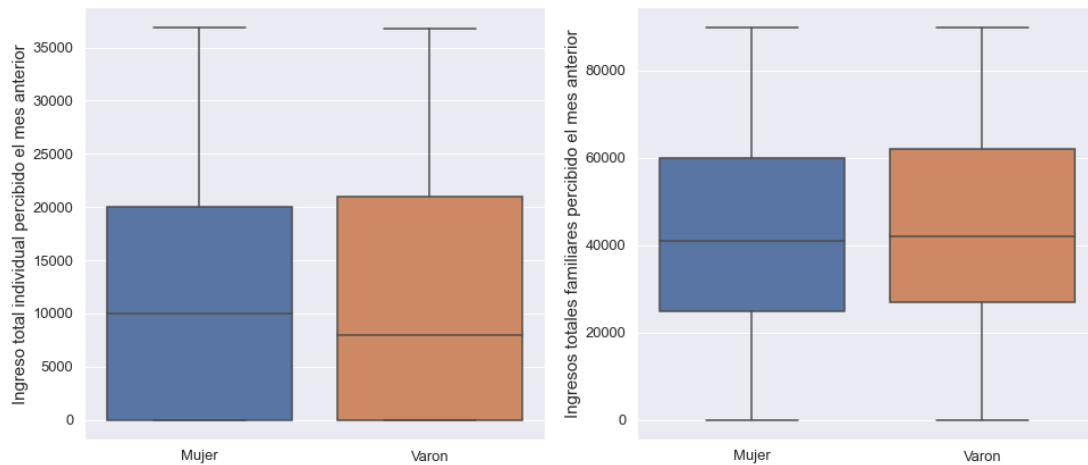
Se puede ver que desde los 30 años en adelante el ingreso total de la persona se corresponde con el ingreso familiar. Por ende suele haber un único ingreso fuerte por grupo familiar.

3.2.2. Comparación de variables categóricas con numéricas

Adicionalmente, vamos a comparar algunas variables con nuestro target, comenzando con los ingresos totales, pero primero veamos como se relacionan algunas categorías con el genero de los encuestados:

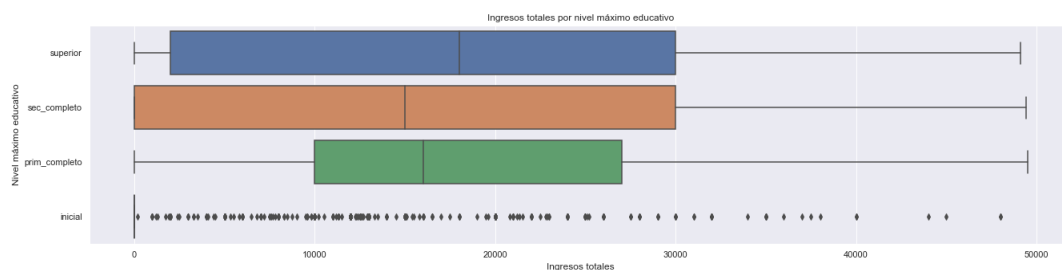


Probemos quitando outliers, a excepción de la cantidad de hijos nacidos vivos (puesto que no arrojará ningún dato nuevo) y de años de escolaridad (que no tiene outliers)

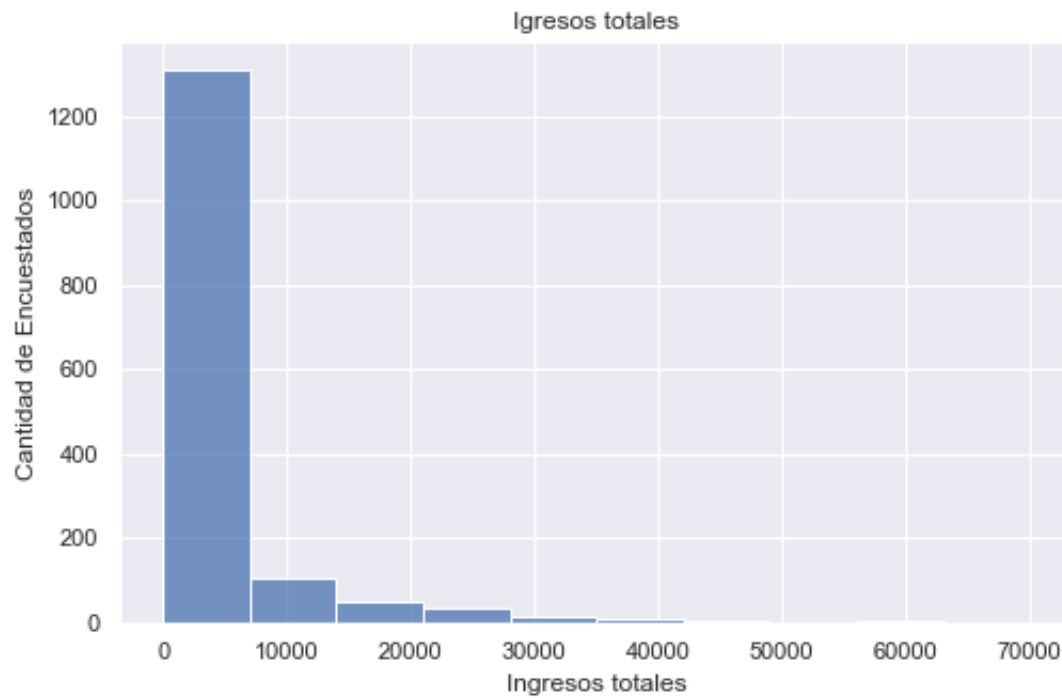


Por parte de las variables de ingreso, no parece haber nada disruptivo. La distribución por ingreso y años de escolaridad pareciera ocurrir pero no en un orden lineal.

Llama la atención la variable “sexo”: que por algún motivo, todos los encuestados hombres figuran sin hijos nacidos vivos. Alternativamente, se podría investigar la metodología de la encuesta para ver si hay alguna respuesta. Adicionalmente, los hombres parecieran tener ingresos totales y familiares mayores que las mujeres, pero no pareciera que haya distribuciones desiguales en los años de escolaridad.

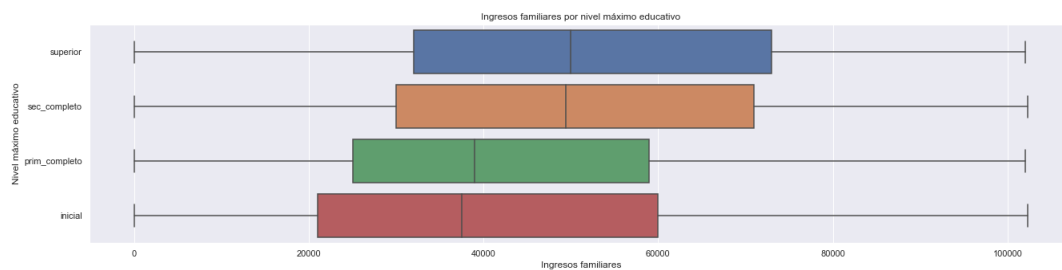


Parece que para el nivel inicial la remoción de outliers en otra categoría sigue siendo insuficiente para mostrar la distribución real de la variable. Echemos un vistazo a los valores de esta categoría:



Lógicamente, la enorme mayoría de los ingresos tienen el valor inicial de 0, puesto que incluye a personas que en ese momento estaban cursando su educación inicial, por lo que tenían entre 2 y 6 años.

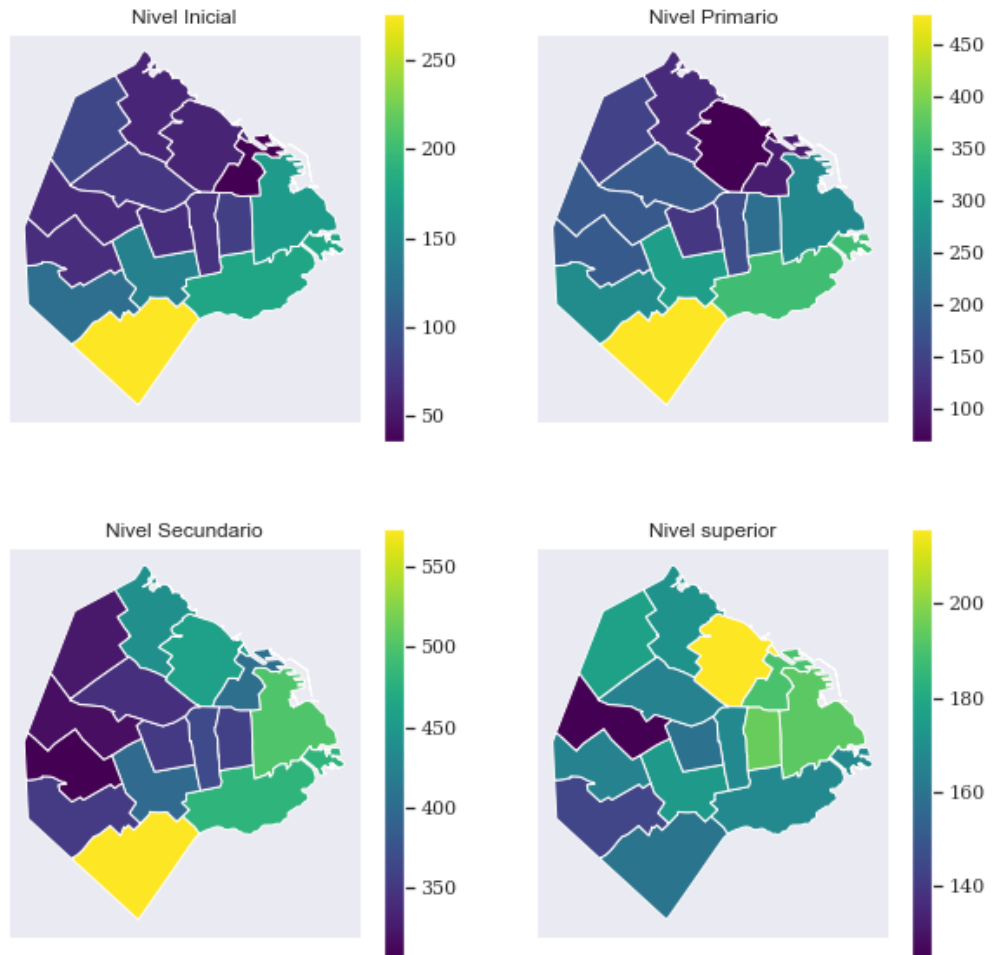
Entonces, es correspondiente analizar como se distribuyen los ingresos familiares con respecto a nuestro target:



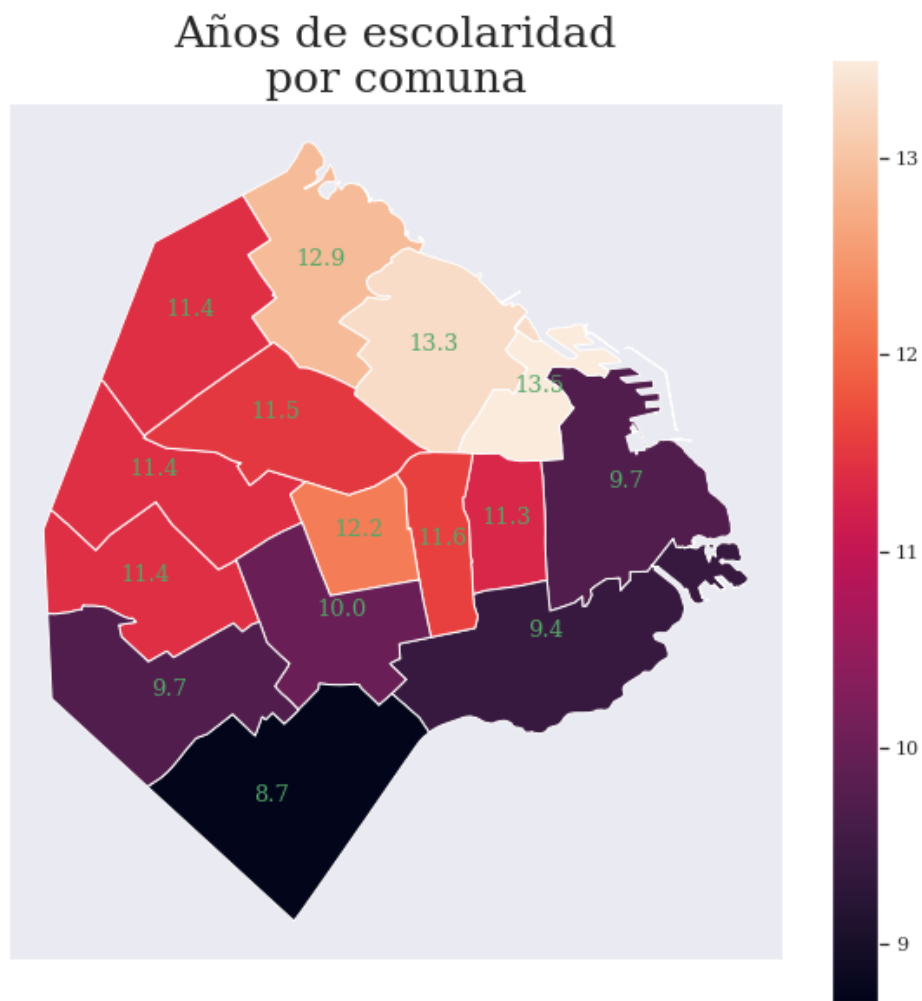
En definitiva, se observa un corrimiento de los valores centrales (dentro de la caja) hacia la izquierda a medida que aumenta el nivel educativo.

3.2.3. Variable numéricas con comuna

En primer lugar, analizaremos la relación de nuestro target con cada comuna:



Se observa que en el sur de la ciudad hay mayor cantidad de encuestados con niveles de inicial, primario y secundario completo, mientras que el norte (particularmente el barrio de Palermo) tiene mayor cantidad de personas con estudios superiores. En menor medida también las comunas del este (comúnmente llamado el “centro de la ciudad”) destacan por la cantidad de encuestados con nivel superior.

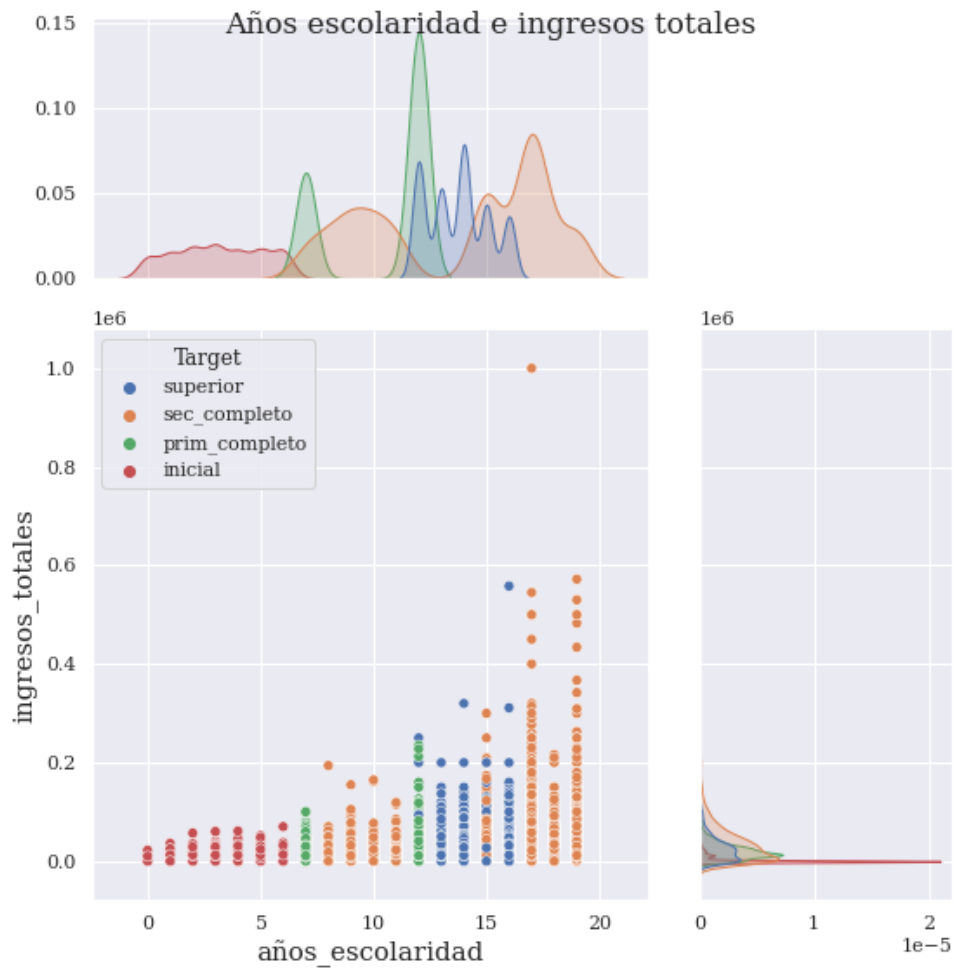


Lo que podemos observar en los últimos dos gráficos es una clara división geográfica del nivel educativo:

- Las comunas del norte son las que tienen mayor nivel educativo.
- Las comunas del centro tienen niveles medios.
- Las comunas del sur (con la comuna 6 en el centro de la ciudad como outlier) y la comuna 1 en el este son las que tienen niveles más bajos.

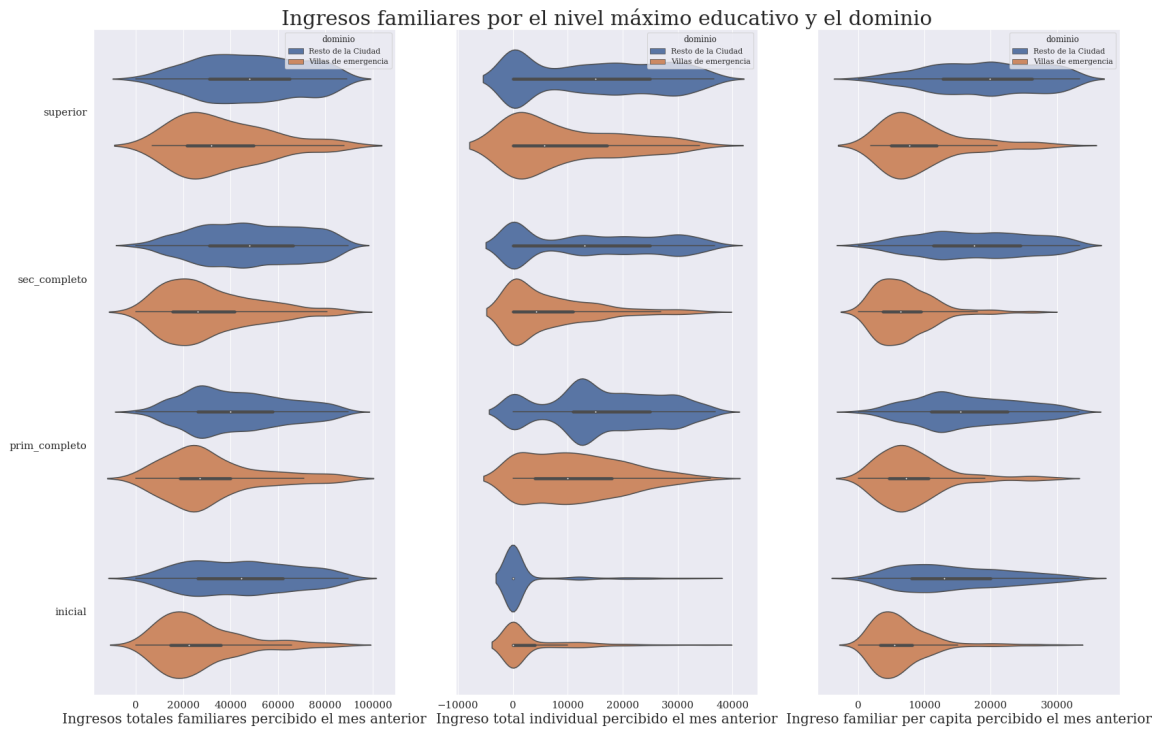
3.3. Análisis multivariado

Probamos de cruzar años de escolaridad, nivel máximo educativo y los ingresos totales.



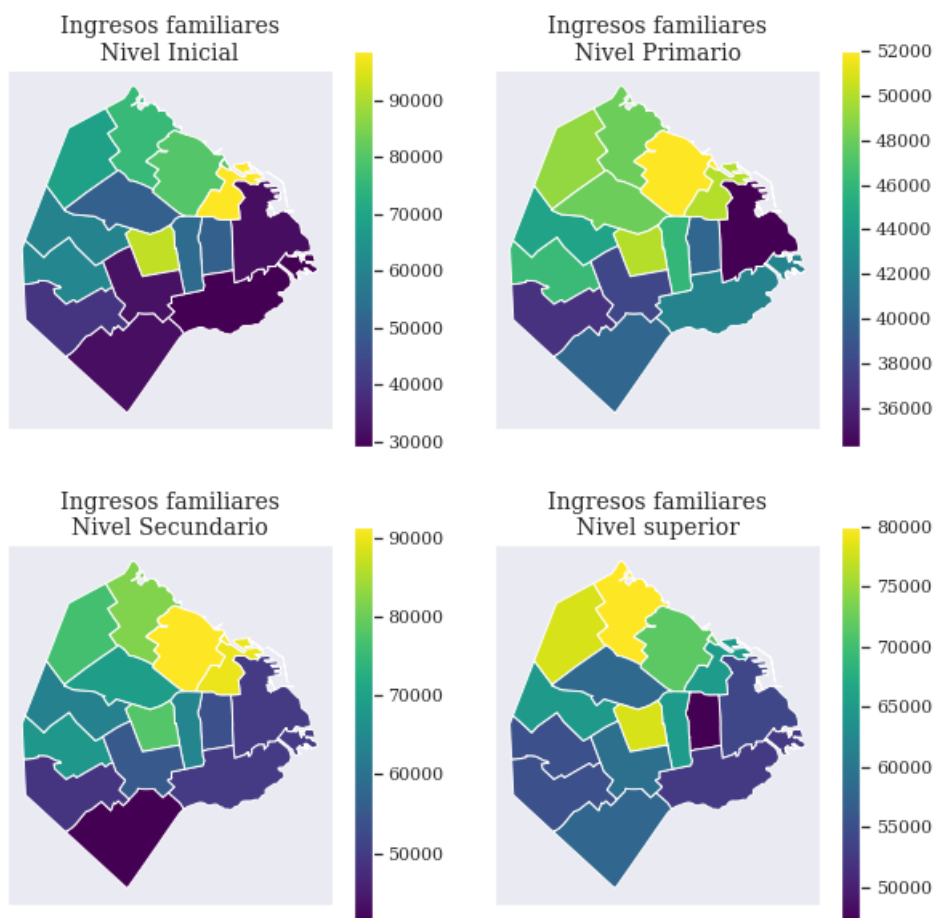
Al visualizar el gráfico podemos observar que:

- Hasta los 6 años, como era esperable, todos los casos llegan al nivel inicial.
- Vemos dos años en que aparece el primario completo: 7 y 12 años. Estimamos que se debe a la división entre los que comenzaron su educación en la primaria y los que comenzaron en el nivel inicial.
- A partir de los 12 años vemos un aumento consistente de los ingresos totales.



Aquí obtuvimos un descubrimiento interesante: no importa el nivel máximo educativo, los casos que no provienen de villas de emergencia (dominio=“villas_de_emergencia”) obtienen en promedio ingresos más altos en todos los niveles educativos. El alcanzar estudios superiores no parece homogeneizar ambos conjuntos. Esto se puede observar en el segundo gráfico, ya que el violín naranja acumula mayor cantidad de casos hacia la derecha, en comparación con los violines azules que tienen una mayor distribución.

Luego, observamos los ingresos familiares según el máximo nivel educativo alcanzado:



Aquí, podemos observar que a medida que avanza el nivel educativo máximo se atenúan levemente las diferencias de ingresos familiares entre comunas. Queda pendiente cruzar estos datos con la edad, para saber si el hecho de incluir a menores de edad está sesgando los valores para nivel inicial, primario y secundario.

4. Modelos analíticos

Comenzamos transformando algunas variables para poder trabajar con los algoritmos:

- recategorizando la variables “Target” en variables numéricas, es decir, a cada nivel educativo le asignamos un valor numérico del 1 al 4:
 - `inicial= 1`,
 - `prim_completo= 2`,
 - `sec_completo= 3`,
 - `superior= 4`,
- reagrupamos la variable “comuna” por regiones para reducir la dimensionalidad (en norte, centro, sur y oeste),
- y por último renombramos algunas variables para que sean más cortas.

Como resultado nos quedan las siguientes variables:

```
RangeIndex: 14319 entries, 0 to 14318
Data columns (total 33 columns):
#   features                                types    non_null_counts
0   id                                       object   14319
1   nhogar                                  object   14319
2   miembro                                object   14319
3   comuna                                 object   14319
4   dominio                                object   14319
5   edad                                   int64    14319
6   sexo                                   object   14319
7   parentesco_jefe                        object   14319
8   situacion_conyugal                     object   14318
9   num_miembro_padre                      object   14319
10  num_miembro_madre                      object   14319
11  estado_ocupacional                     object   14319
12  cat_ocupacional                        object   14319
13  calidad_ingresos_lab                   object   14319
14  ingreso_total_lab                      int64    14319
15  calidad_ingresos_no_lab                 object   14319
16  ingreso_total_no_lab                   int64    14319
17  calidad_ingresos_totales                object   14319
18  ingresos_totales                       int64    14319
19  calidad_ingresos_familiares             object   14319
20  ingresos_familiares                    int64    14319
21  ing_per_cap_familiar                    int64    14319
22  estado_educativo                       object   14319
23  sector_educativo                       object   14316
24  nivel_actual                           object   14319
25  nivel_max_educativo                    object   13265
26  años_escolaridad                       float64   14257
27  lugar_nacimiento                       object   14318
28  afiliacion_salud                       object   14315
29  hijos_nacidos_vivos                    object    6535
30  cant_hijos_nac_vivos                    int64    14319
31  Target                                 Int64    13223
32  region                                 object   14319
dtypes: Int64(1), float64(1), int64(7), object(24)
memory usage: 3.6+ MB
```

4.1. Tratados de nulos

En primer lugar, armamos una función para tener una lista limpia de variables con nulos, que nos da como resultado:

| | |
|---------------------|------|
| situacion_conyugal | 1 |
| lugar_nacimiento | 1 |
| sector_educativo | 3 |
| afiliacion_salud | 4 |
| años_escolaridad | 62 |
| nivel_max_educativo | 1054 |
| Target | 1096 |
| hijos_nacidos_vivos | 7784 |

Entonces, para eliminar los valores nulos de la variable “años_escolaridad” reemplazamos los nulos con la mediana por comuna y sexo.

Por otro lado, a los nulos las variables:

- “lugar_nacimiento”,
- “situacion_conyugal”,
- “afiliacion_salud”,
- “sector_educativo”
- “hijos_nacidos_vivos”

los reemplazamos con la moda.

Luego, eliminamos la variable “nivel_max_educativo” ya que no la vamos a utilizar y por último eliminamos los nulos de nuestro target y pasamos el tipo de dato de la misma a entero.

4.2. Target

4.2.1. Borrado de variables

Hay muchas variables que consideramos que no es necesario sumarlas al algoritmo de clasificación dado que brindan información repetida o que no suma para la clasificación. A continuación se comparten las categoría que se descartarán para correr el algoritmo:

- **id:** no suma información para la clasificación,
- **nhogar:** no suma información para la clasificación,
- **parentesco_jefe:** no suma información para la clasificación,
- **miembro:** no suma información para la clasificación,
- **num_miembro_padre:** no suma información para la clasificación,
- **num_miembro_madre:** no suma información para la clasificación,
- **cat_ocupacional:** brinda la misma información que estado_ocupacional,
- **calidad_ingresos_lab:** brinda la misma información que ingreso_total_lab,
- **calidad_ingresos_no_lab:** brinda la misma información que ingreso_total_no_lab,
- **calidad_ingresos_totales:** brinda la misma información que ingresos_totales,
- **calidad_ingresos_familiares:** brinda la misma información que ingreso_familiares,
- **estado_educativo:** no aporta información para la clasificación,
- **nivel_actual:** no aporta información para la clasificación,
- **hijos_nacidos_vivos:** brinda la misma información que cant_hijos_nac_vivos,
- **comuna:** variable ya abordada en la variable 'región'.

Y como resultado, tenemos nuestro dataset listo para el procesamiento:

```
RangeIndex: 14319 entries, 0 to 14318
Data columns (total 18 columns):
#   features                types    non_null_counts
0   dominio                 object   13223
1   edad                   int64    13223
2   sexo                   object   13223
3   situacion_conyugal     object   13223
4   estado_ocupacional     object   13223
5   ingreso_total_lab      int64    13223
6   ingreso_total_no_lab   int64    13223
7   ingresos_totales       int64    13223
8   ingresos_familiares    int64    13223
9   ing_per_cap_familiar   int64    13223
10  sector_educativo        object   13223
11  años_escolaridad       float64  13223
12  lugar_nacimiento       object   13223
13  afiliacion_salud       object   13223
14  cant_hijos_nac_vivos    int64    13223
15  Target                 int32    13223
16  region                 object   13223
dtypes: Int64(1), float64(1), int64(7), object(8)
memory usage: 2.0+ MB
```

4.3. Procesamiento

Para preparar los datos para el modelado generamos una función que:

- Divide el dataframe en X_train, y_train, X_test e y_test, haciendo la división entre test y el train en un 30 % y un 70 % respectivamente, con una semilla específica.
- Procesa el X_train y el X_test con un pipeline generado previamente, el cual convierte las variable numéricas con el minmaxscaler y las categóricas con one hot encoding.

Una vez aplicada dicha función a nuestro dataframe, tenemos ya lista la partición (con la misma cantidad de columnas) del mismo en X_train, y_train, X_test e y_test.

4.4. Árbol de decisión

4.4.1. Primer modelo

Como primera aproximación, vamos a usar un árbol de clasificación usando con parámetros:

- random_state = 50,
- max_depth=8,
- criterion='gini',

para saber como performa y mejorarlo a partir de ahí.

Como resultado obtenemos que el Accuracy score para el test es de: 0.940 y la matriz de confusión nos da:

| | Predicc. Inicial | Predicc. Primario | Predicc. Secundario | Predicc. Superior |
|------------|------------------|-------------------|---------------------|-------------------|
| Inicial | 445 | 0 | 1 | 0 |
| Primario | 0 | 961 | 8 | 9 |
| Secundario | 0 | 19 | 1693 | 59 |
| Superior | 1 | 57 | 84 | 630 |

Y obtenemos las siguientes métricas:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1 | 1.00 | 1.00 | 1.0 | 446 |
| 2 | 0.93 | 0.99 | 0.95 | 978 |
| 3 | 0.95 | 0.96 | 0.95 | 1771 |
| 4 | 0.90 | 0.82 | 0.86 | 772 |
| accuracy | | | 0.94 | 3967 |
| macro avg | 0.94 | 0.94 | 0.94 | 3967 |
| weighted avg | 0.94 | 0.94 | 0.94 | 3967 |

A simple vista, parece que el modelo performa muy bien, dado su accuracy. Veamos más en detalle y calculemos su sesgo y su varianza:

- **Bias o sesgo:** 96.89 % que nos indica que tengo poco error, lo que indica que tenemos un sesgo bajo,
- **Variance=Test_Score - Bias=** 2.89 %, lo que nos indica que la varianza también es baja.

Entonces, el modelo tiene una **buena relación** de sesgo y varianza.

De aquí, vemos necesario ver cuáles son las variables más importantes para el armado del modelo. Esto nos permitirá volver el modelo más robusto, al quitar las mismas.

Y como resultado, tenemos que la variable “años_escolaridad” tiene una importancia del 84 %, por mucho superior al resto de variables.

Por lo tanto, vamos a tener que desarrollar un nuevo modelo sin esta variable. El principal motivo es que los años de escolaridad es un dato que puede contastarse de forma conjunta con el nivel máximo educativo, por lo que tiene sentido que si no tenes la variable target, tampoco tengas la variable de los años de escolaridad.

Así, creamos un dataset nuevo (llamado “df2”) sin la variable “años_escolaridad”, para volver a aplicar la función “procesador” para dividir nuevamente el mismo y generar nuevos modelos.

4.4.2. Segundo modelo

Esta vez al correr el modelo, utilizaremos el “DecisionTreeClassifier” solo con los parámetros:

- random_state = 50,
- criterion='entropy'.

Que nos da como resultado:

| | precision | recall | f1-score | suppo |
|--------------|-----------|--------|----------|-------|
| 1 | 0.83 | 0.79 | 0.81 | 446 |
| 2 | 0.45 | 0.26 | 0.33 | 978 |
| 3 | 0.56 | 0.66 | 0.60 | 1771 |
| 4 | 0.39 | 0.45 | 0.42 | 772 |
| accuracy | | | 0.53 | 3967 |
| macro avg | 0.56 | 0.59 | 0.54 | 3967 |
| weighted avg | 0.53 | 0.53 | 0.52 | 3967 |

Luego, analizamos el sesgo y la varianza:

- **Bias o sesgo:** 99.78 % que nos indica que tengo poco error, es decir, un sesgo bajo,
- **Variance=Test_Score - Bias=** 46.39 %, lo que nos indica un nivel de varianza alto,

Lo que nos da como resultado, que este modelo esta haciendo **OVERFITTING**.

Por lo que se observa, el árbol performa bastante peor sin esta variable, aumentando especialmente la varianza. Por lo tanto optamos probar mejorar nuestro modelo con un grid search.

4.4.3. Gridsearch con CV

En la grilla de parámetros para el Gridsearch elegimos los siguientes:

- 'max_depth': range(5,11),
- 'max_features': range(1,14),

- 'criterion': ['gini','entropy','log_loss'];

como estimador el “DecisionTreeClassifier” con el random_state=50, con el cross-validation =10, usando todos los procesadores.

Y nos da como resultado, que el mejor árbol de decisión posible obtiene 0.642. Y para eso el árbol debe tener una profundidad de 6, utilizar 10 variables y usar el método “gini”.

Entonces, entrenamos el modelo bajo estos mismos parámetros y obtenemos el siguiente reporte de clasificación:

| | precision | recall | f1-score | suppo |
|--------------|-----------|--------|----------|-------|
| 1 | 0.99 | 0.74 | 0.85 | 446 |
| 2 | 0.49 | 0.21 | 0.30 | 978 |
| 3 | 0.55 | 0.87 | 0.69 | 1771 |
| 4 | 0.78 | 0.41 | 0.54 | 772 |
| accuracy | | | 0.60 | 3967 |
| macro avg | 0.70 | 0.56 | 0.59 | 3967 |
| weighted avg | 0.63 | 0.60 | 0.57 | 3967 |

Luego, analizamos el sesgo y la varianza:

- **Bias o sesgo:** 65.27 % que nos indica que tengo bastantes errores \Rightarrow high bias,
- **Variance=Test_Score - Bias==** 5.14 % \Rightarrow low variance.

Por lo tanto, el modelo esta haciendo **UNDERFITTING**

Conclusión

Utilizar el grid search nos permitió mejorar bastante el modelo que había perdido bastante accuracy al retirar los años de escolaridad. La métrica que más pudimos mejorar con este método fue la varianza, que pasó de 44.97 % a 5.14 %, pero en contra parte nuestro acuracy bajo de 100 % a 65.27 %.

4.5. Random Forest Classifier

Ahora vamos a trabajar con random forest, para saber si este algoritmo nos arroja mejores resultados. Y teniendo en cuenta todo el dataset incluido la variable con los años de escolaridad.

4.5.1. Primer modelo

Como primer modelo con el Random Forest Classifier, elegimos los siguientes parámetros:

- n_estimators=200,
- max_depth=15,
- random_state=50,
- criterion='gini'.

Que nos da los siguientes resultados en cuanto a las métricas:

| | precision | recall | f1-score | suppo |
|--------------|-----------|--------|----------|-------|
| 1 | 1.00 | 0.96 | 0.98 | 446 |
| 2 | 0.86 | 0.95 | 0.90 | 978 |
| 3 | 0.91 | 0.93 | 0.9 | 1771 |
| 4 | 0.92 | 0.76 | 0.83 | 772 |
| accuracy | | | 0.91 | 3967 |
| macro avg | 0.92 | 0.90 | 0.91 | 3967 |
| weighted avg | 0.91 | 0.91 | 0.90 | 3967 |

El random forest performa bastante bien, es decir, mucho mejor que los modelos anteriores:

- **Bias o sesgo:** 97.80 % que nos indica que tengo bastantes errores, es decir tenemos un sesgo alto,
- **Variance=Test_Score - Bias==** 7.20 %, que nos indica que la varianza es baja.

En conclusión obtenemos un buen modelo. De todas maneras buscamos cuales son las variables más importantes, y encontramos que los años de escolaridad redujo la enorme importancia (a un 43.58 %) que tenía en el random tree. Sin embargo, sigue correspondiendo quitarla del modelo.

4.5.2. Segundo modelo

En este caso elegimos los siguientes parámetros:

- `n_estimators=200`,
- `max_depth=10`,
- `random_state=50`,
- `criterion='gini'`.

Dándonos por resultado los siguientes medidas de desempeño:

| | precision | recall | f1-score | suppo |
|--------------|-----------|--------|----------|-------|
| 1 | 0.95 | 0.78 | 0.86 | 446 |
| 2 | 0.54 | 0.23 | 0.32 | 978 |
| 3 | 0.56 | 0.88 | 0.69 | 1771 |
| 4 | 0.80 | 0.40 | 0.53 | 772 |
| accuracy | | | 0.62 | 3967 |
| macro avg | 0.71 | 0.57 | 0.60 | 3967 |
| weighted avg | 0.65 | 0.62 | 0.59 | 3967 |

En este caso obtuvimos los siguientes valores del sesgo y la varianza:

- **Bias o sesgo:** 89.11 % que nos indica que tengo bastantes errores, es decir, que el sesgo es alto,
- **Variance=Test_Score - Bias== 27.4 %**, esto indica que tenemos un valor alto en la varianza.

El modelo empeora su accuracy pero está muy cercano al mejor modelo de Random Tree, mientras que crece mucho la varianza a un 27.4 % (unos 20 puntos). Ahora, al igual que con el el DesicionTree, vamos a probar mejorándolo con grid search.

4.5.3. Gridsearch con CV

En la grilla de parámetros para el Gridsearch elegimos los siguientes:

- `'max_depth': [5,7,10,15,None]`,
- `'max_features': [5,8,10,30,41]`,
- `'n_estimators': [200,300,500]`,
- `'criterion': ['gini','entropy','log_loss']`.

como estimador el “RandomTreeClassifier” que utilizamos en el último modelo, con el cross-validation =10 y usando todos los procesadores.

Y nos da como resultado, que el mejor random forest posible obtiene 0.668.

Y para eso el árbol debe tener una profundidad de 15 , utilizar 10 variables, tener 300 estimadores y utilizar el método “gini”.

Entonces, entrenamos el modelo bajo estos mismos parámetros y obtenemos el siguiente reporte de clasificación:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1 | 0.95 | 0.79 | 0.86 | 446 |
| 2 | 0.56 | 0.23 | 0.33 | 978 |
| 3 | 0.56 | 0.89 | 0.69 | 1771 |
| 4 | 0.80 | 0.40 | 0.54 | 772 |
| accuracy | | | 0.62 | 3967 |
| macro avg | 0.72 | 0.58 | 0.60 | 3967 |
| weighted avg | 0.65 | 0.62 | 0.59 | 3967 |

- **Bias o sesgo:** 90.65 % que nos indica que tengo bastantes errores, es decir, el sesgo es alto,
- **Variance=Test_Score - Bias: 28.54 %** \Rightarrow , y nuestra varianza también es alta.

Lo que nos indica que nuestro modelo esta haciendo **OVERFITTING**.

5. Conclusiones

Finalmente, nos queda elegir el mejor modelo para realizar nuestras predicciones. Para eso vamos a tomar las métricas de cada uno de ellos y hacer un cuadro comparativo:

| modelo | accuracy | sesgo | varianza |
|-----------------|----------|--------|----------|
| árbol_default | 0.5339 | 0.9978 | 0.4639 |
| árbol_mejorado | 0.6012 | 0.6527 | 0.0514 |
| bosque_default | 0.6117 | 0.8911 | 0.2740 |
| bosque_mejorado | 0.6211 | 0.9065 | 0.2854 |

Con esta información podemos decidir qué modelo nos conviene usar:

- El árbol default tiene el mejor resultado con respecto al sesgo, pero su varianza lo deja afuera de la competencia.
- Por el contrario, el árbol mejorado tiene una varianza insuperable de 5 %, aunque con el menor puntaje con respecto al sesgo.
- El bosque default tiene resultados mixtos en ambas categorías.
- El bosque mejorado destaca por bajo sesgo pero su varianza es la segunda peor.

Como era de esperarse, los finalistas son el árbol y el bosque mejorado. Sorprendentemente, ambos performan muy bien pero en métricas diferentes. A su vez, el accuracy de ambos difiere en apenas un 2 %.

En nuestra opinión, es el árbol mejorado el ganador, ya que tiene la robustez suficiente para poder generalizar en caso de agregar nuevos datos al modelo. Otra ventaja frente al bosque aleatorio es su mayor velocidad de entrenamiento, así como su capacidad de ser visualizada en un gráfico.