CODER HOUSE

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

**Profesor:** Damian Dapueto

**Tutor:** Héctor Alonso

**Grupo de Trabajo:** Lucia Buzzeo, Lucia Hukovsky, Jose Saint German, Juan Martín Carini

# Introdución

# Objetivos

- Descubrir las principales variables intervinientes en el nivel máximo educativo alcanzado por la población de la Ciudad Autónoma de Buenos Aires (CABA).
- 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.

# 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.

# Estructura de los trabajos

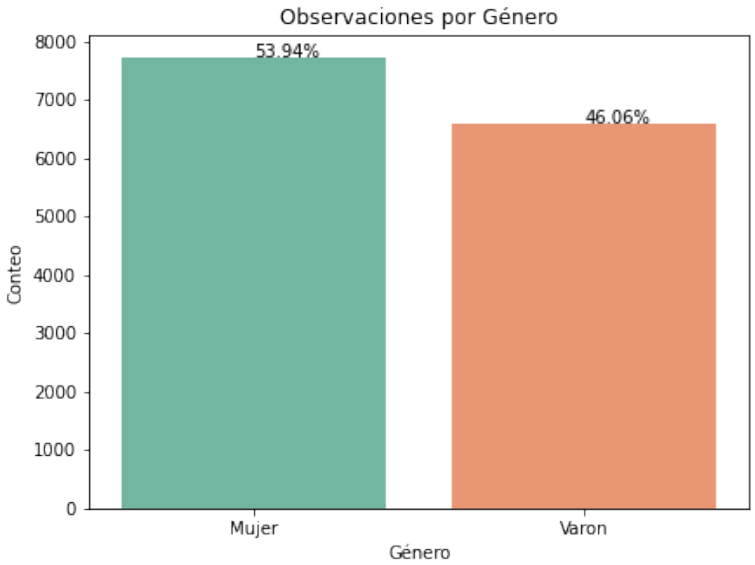
Este trabajo se ha dividido en 3 partes:

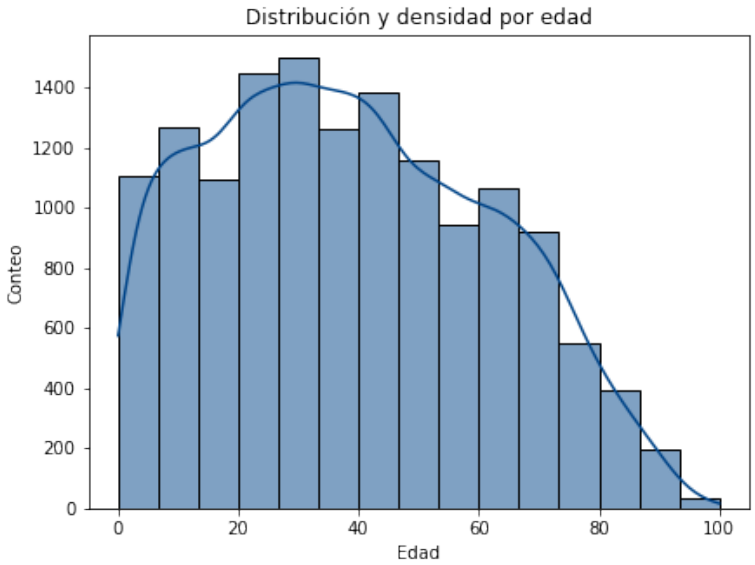
1. **Introducción a las variables del problema:** Se realiza un análisis de las variables del dataset. En el mismo se busca conocer su performance dentro del dataset. A la vez, se investiga cómo las variables interactúan entre sí. Esta parte es lo que se conoce como análisis univariado, bivariado y multivariado.
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.

# Análisis exploratorio de los datos (EDA)

# Análisis univariado

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.

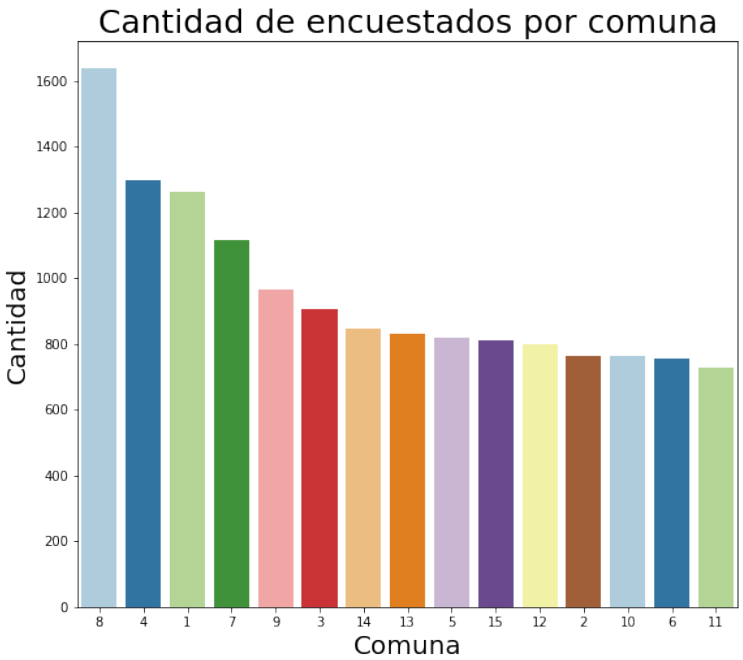


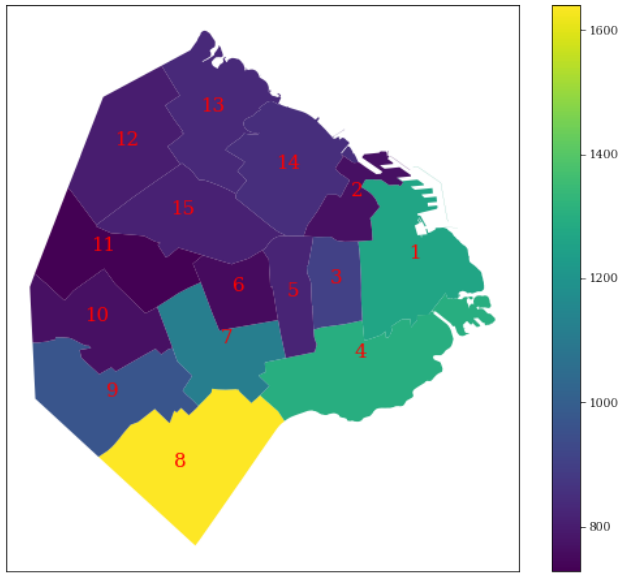




En la variable género los datos parecen equilibrados en las categorías. Para el caso de la variable "edad", la distribución se asemeja a la de una normal.

# Análisis univariado

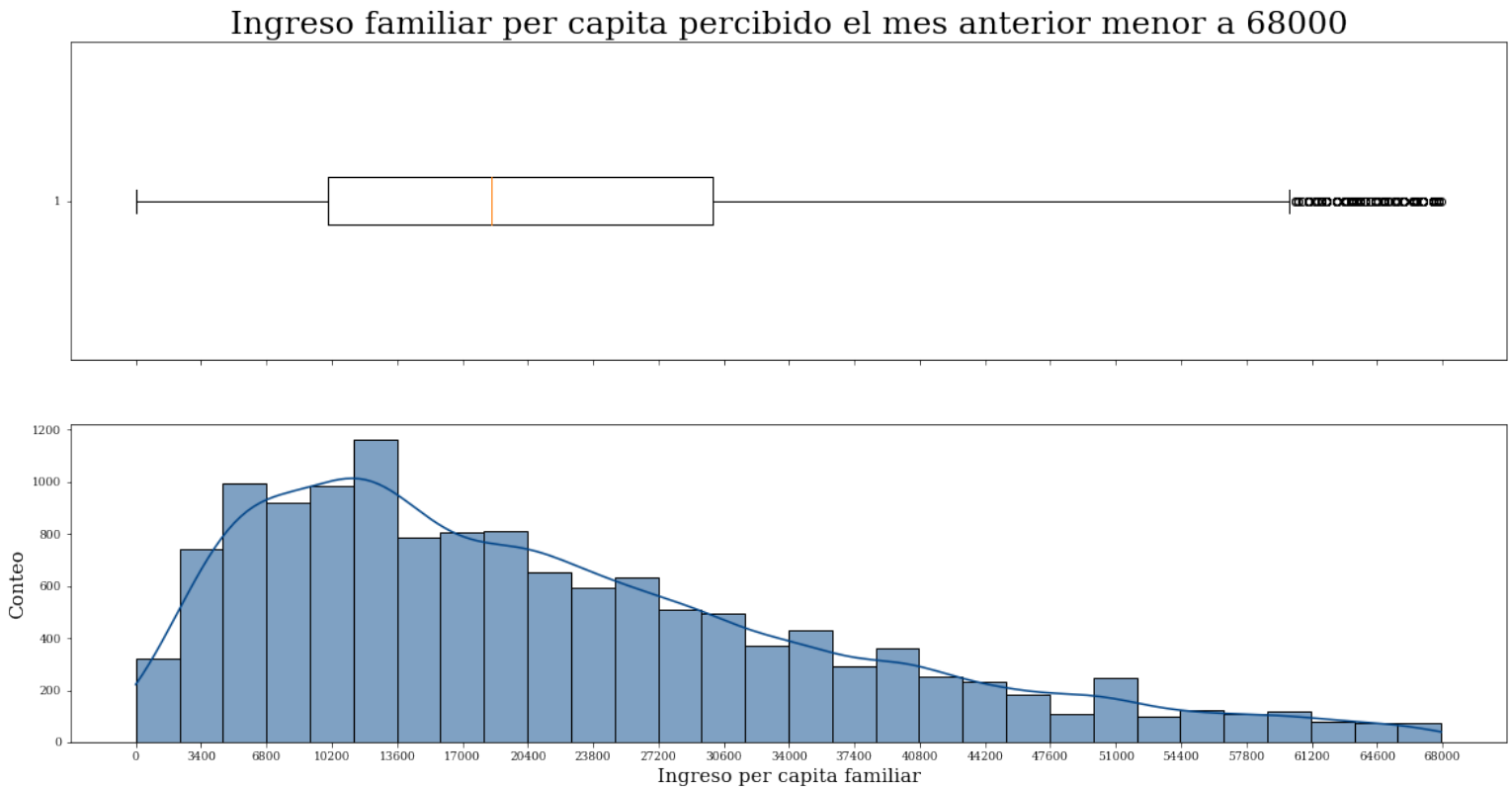
Seguimos observando la variable “comuna”:



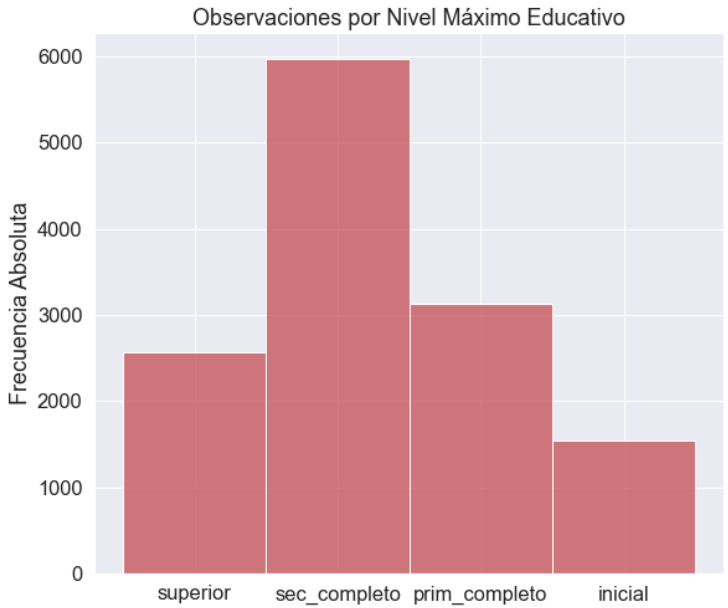



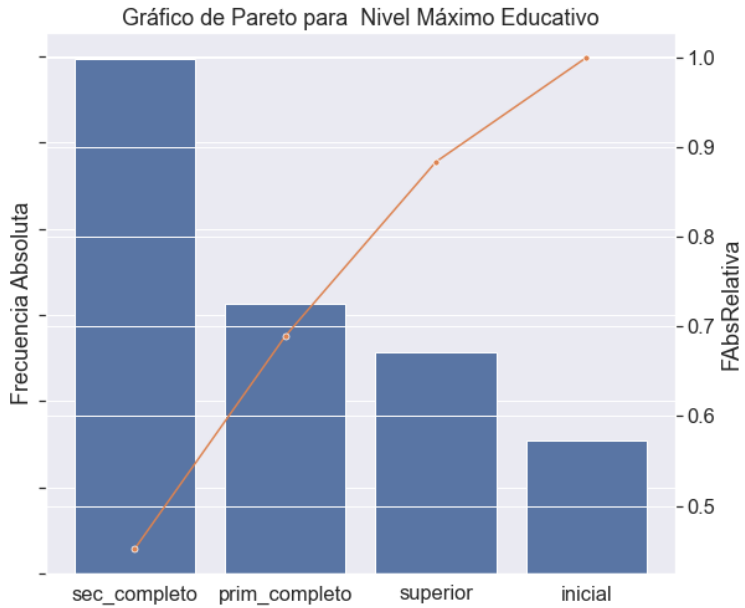

Observando ambos gráficos vemos que las comunas 1,4,7 y 8 tienen mayor cantidad de casos.

# Análisis univariado



# Análisis univariado

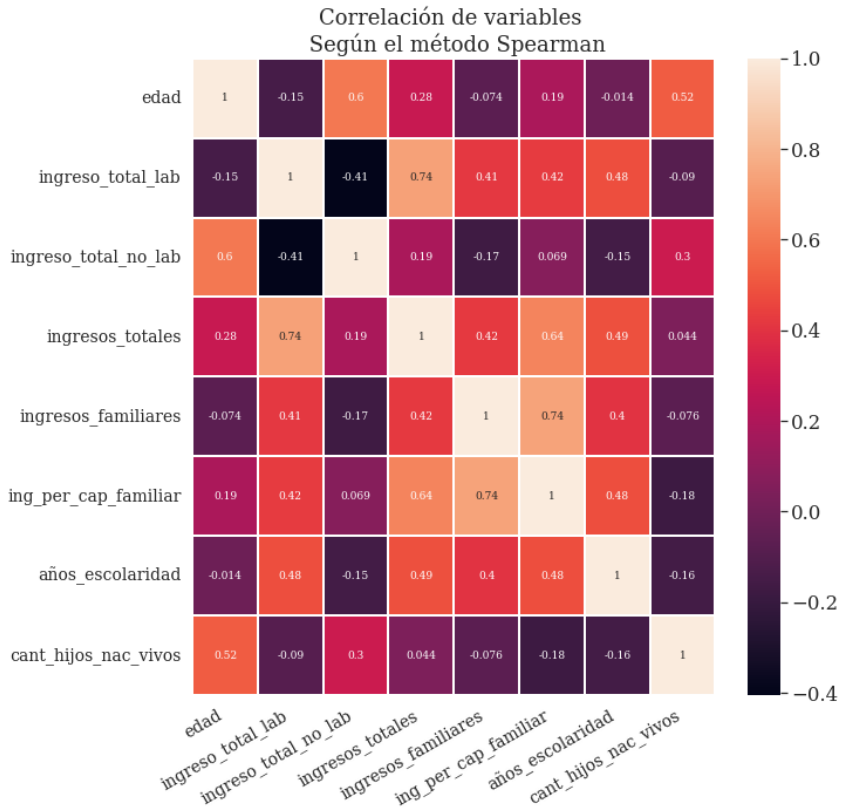



Podemos observar que el nivel máximo educativo más alcanzado es el secundario completo, seguido por el primario. Contrario de lo que habíamos intuido anteriormente, el nivel superior quedó en tercer lugar. Adicionalmente, el nivel secundario y primario explican casi el 77 % de los datos.

# 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.




- No se observan fuertes correlaciones.
- "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").

# Análisis bivariado

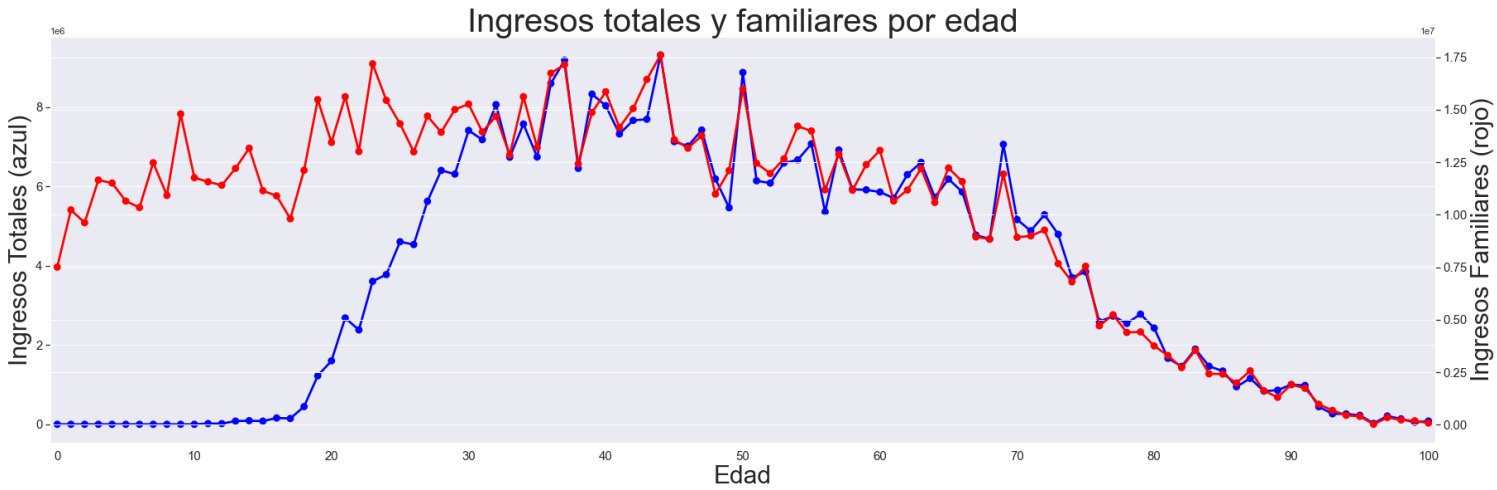
Corremos una tabla de correlación y filtramos las de valores más altos:

| | 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 |

- 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.

# Análisis bivariado

Observaremos la relación entre los ingresos totales de cada hogar y los ingresos familiares por edad:

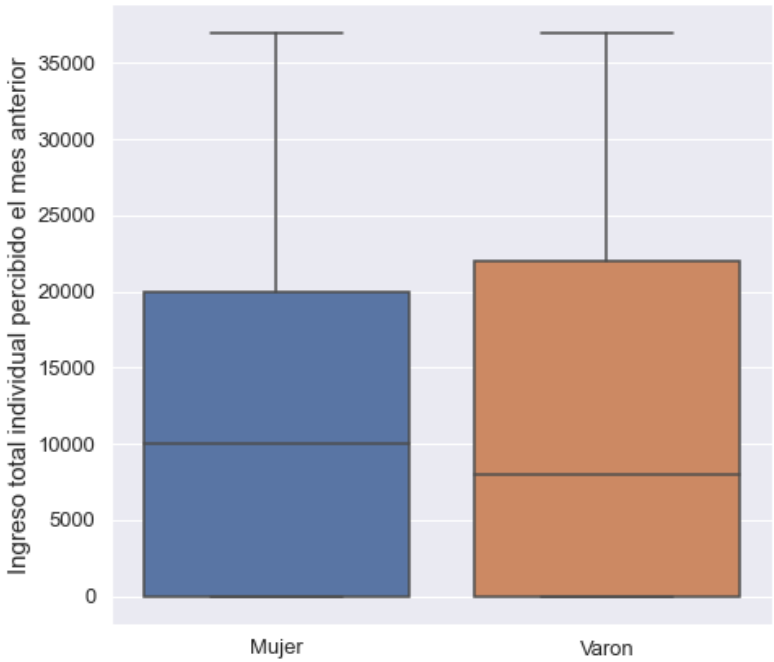


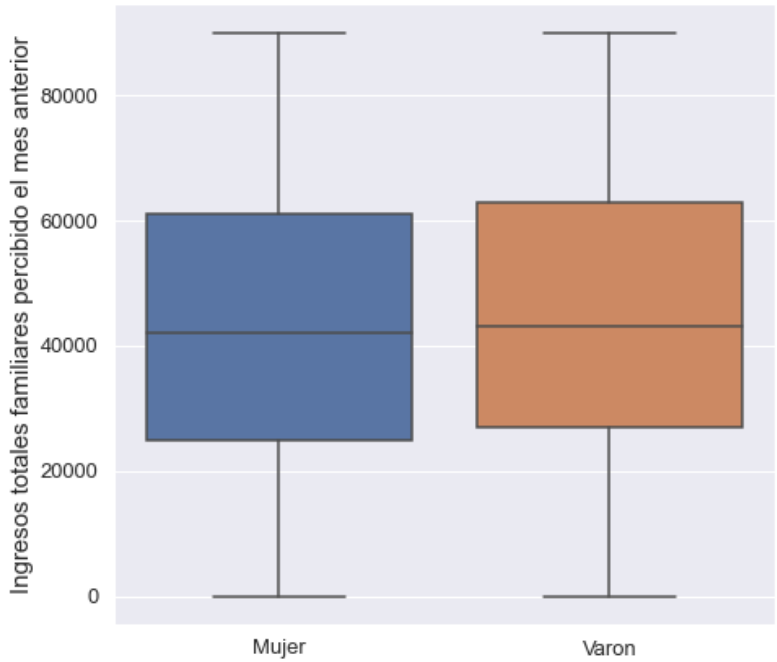

Se puede ver que desde los 30 años en adelante el ingreso total de la persona se corresponde con el ingreso familiar. Por ende suele haber un único ingreso fuerte por grupo familiar.

# Análisis bivariado

# Análisis bivariado

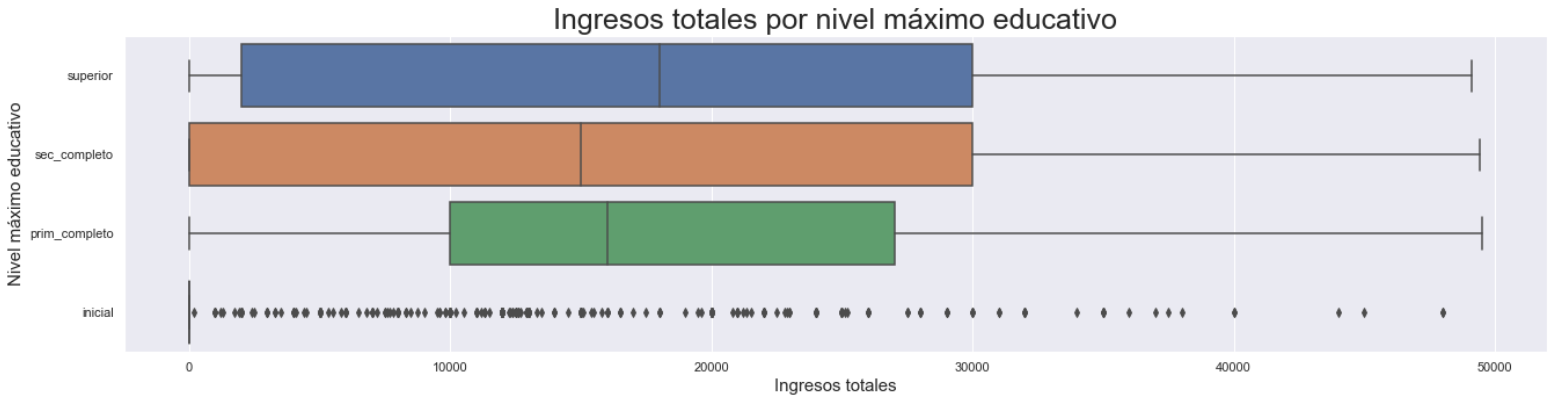
Veamos como se relacionan las categorías de ingresos con el genero de los encuestados:




Y resulta que con las variables de ingreso, no parece haber nada disruptivo. Salvo que los hombres parecieran tener ingresos totales y familiares mayores que las mujeres, pero no pareciera que haya distribuciones desiguales en los años de escolaridad.

# Análisis bivariado



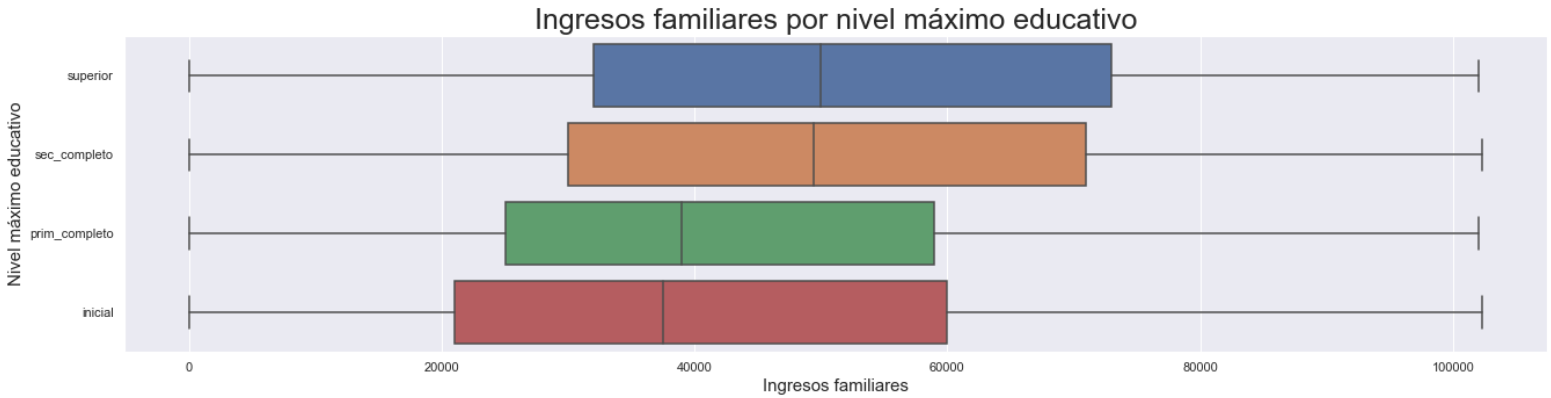

Parece que para el nivel inicial la remoción de outliers en otra categoría sigue siendo insuficiente para mostrar la distribución real de la variable. Echemos un vistazo a los valores de esta categoría:

# Análisis bivariado


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.

# Análisis bivariado

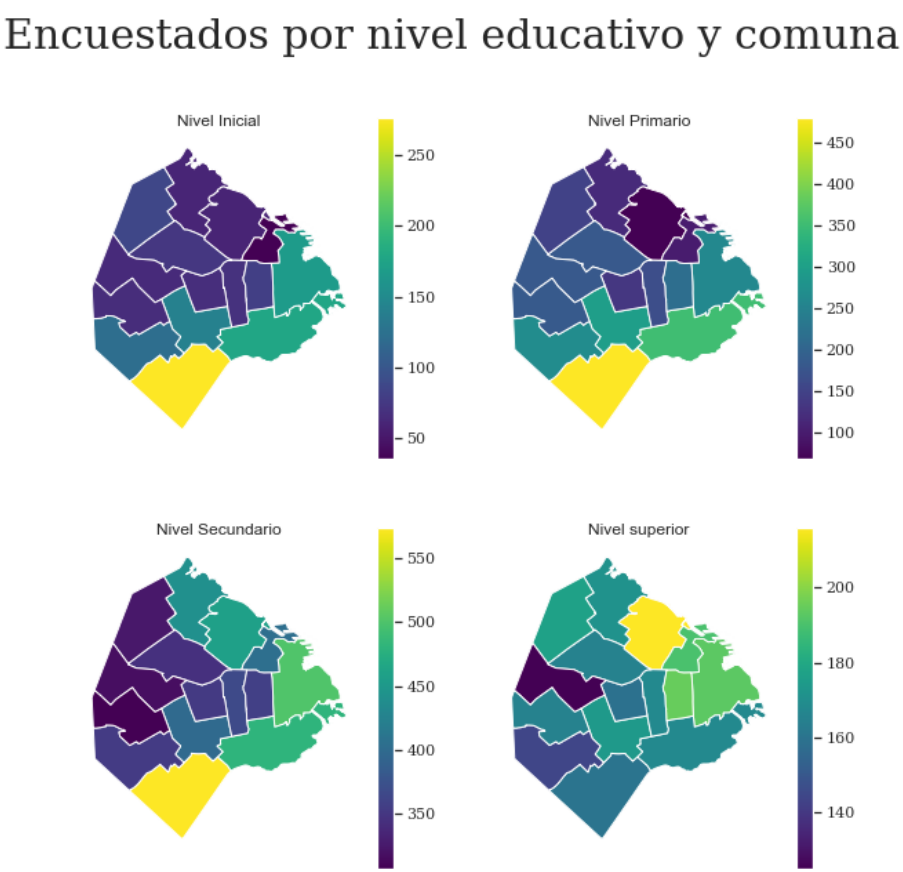
Entonces, es correspondiente analizar como se distribuyen los ingresos familiares con respecto a nuestro target:




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

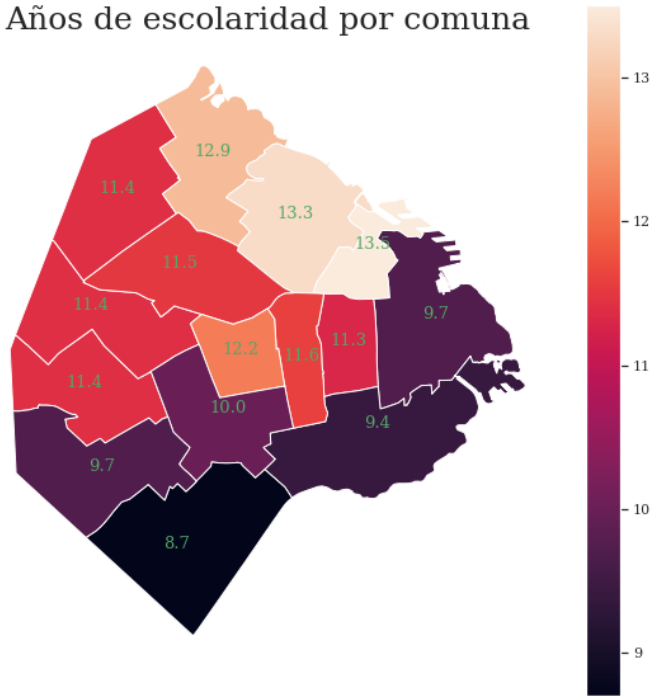
# Análisis bivariado

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.

# Análisis bivariado

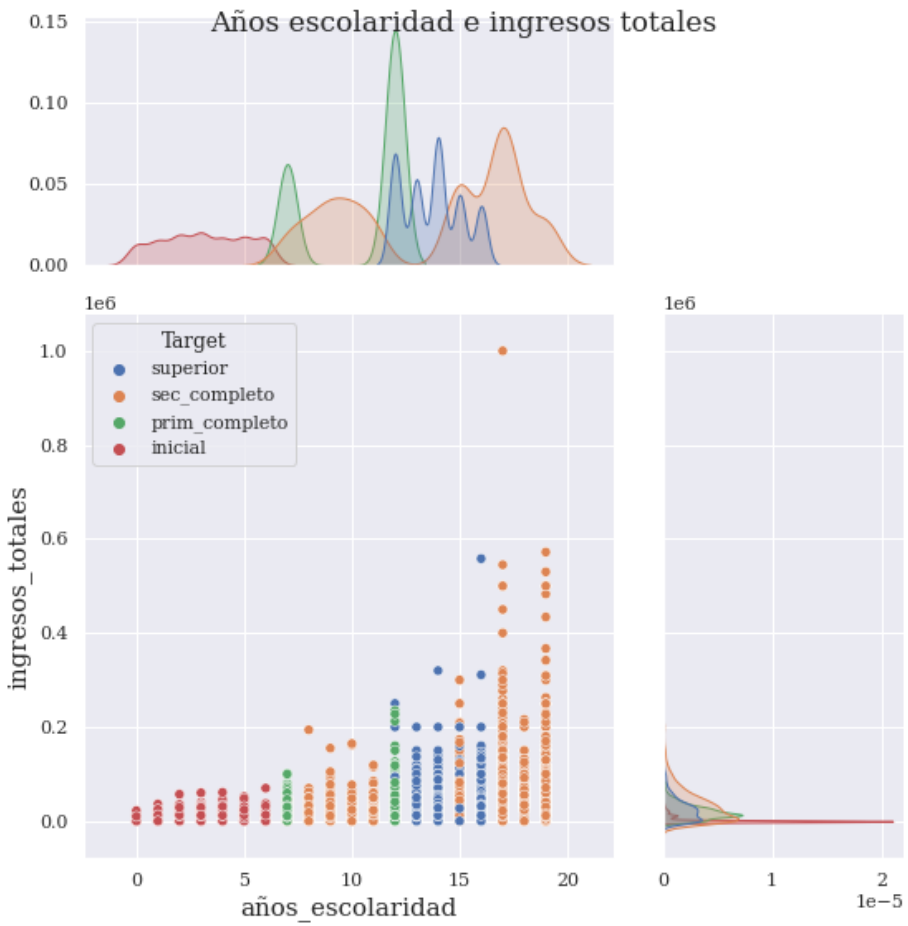



Lo que podemos observar en los últimos dos gráficos es una clara división geográfica del nivel educativo:

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

# Análisis multivariado

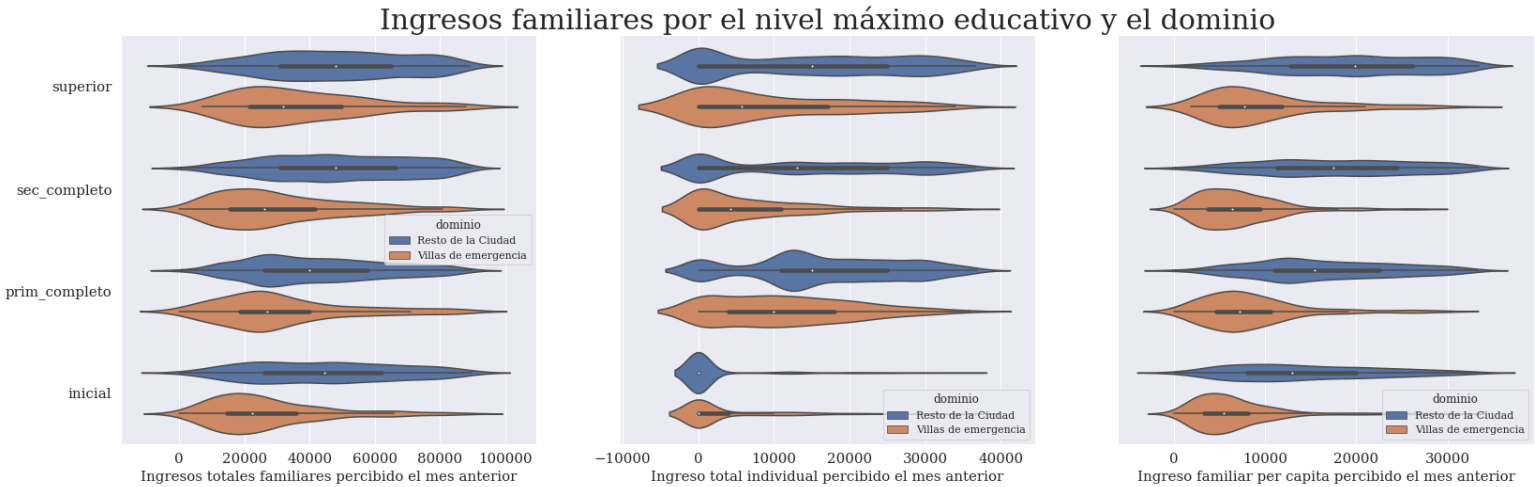
Probamos de cruzar años de escolaridad, nivel máximo educativo y los ingresos totales.




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.

# Análisis multivariado

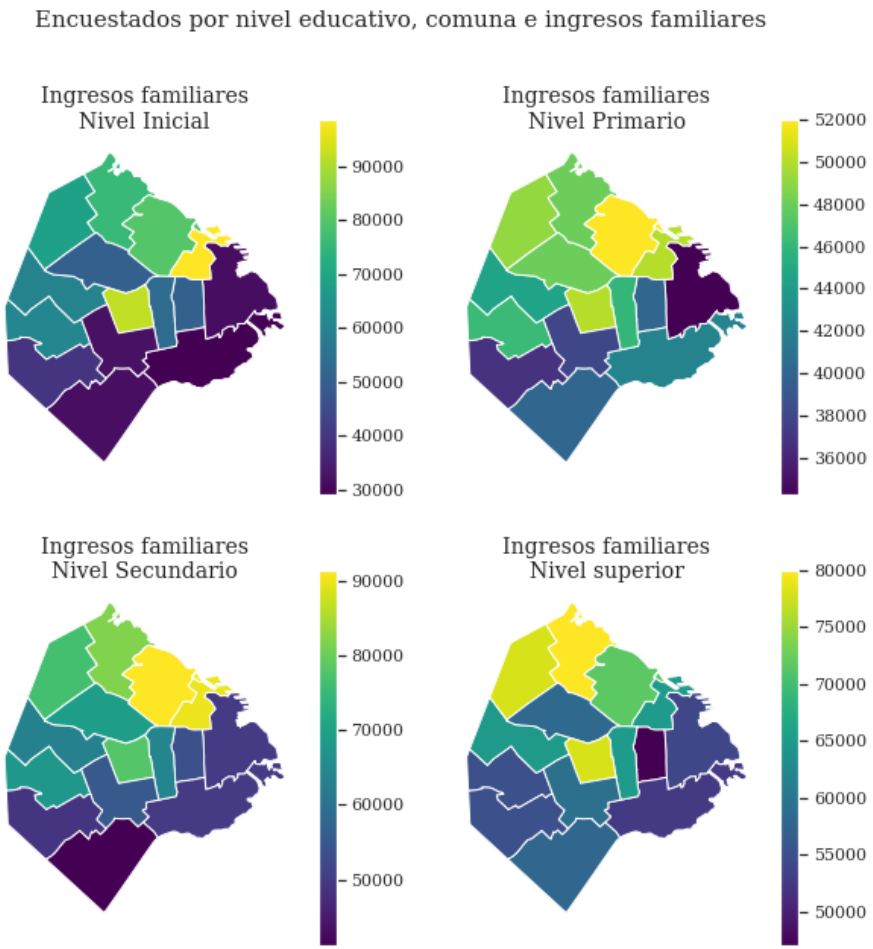


# Análisis multivariado

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.

# Análisis multivariado

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.

# 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.

## Tratados de nulos

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

| Variable | Nulos | Acción |
|---|---|---|
| situacion_conyugal | 1 | Reemplazamos con la moda |
| lugar_nacimiento | 1 | Reemplazamos con la moda |
| sector_educativo | 3 | Reemplazamos con la moda |
| afiliacion_salud | 4 | Reemplazamos con la moda |
| años_escolaridad | 62 | Reemplazamos con la mediana por comuna y sexo |
| nivel_max_educativo | 1054 | Eliminamos la variable |
| Target | 1096 | Eliminamos sus nulos y transformamos su tipo a entero |
| hijos_nacidos_vivos | 7784 | Reemplazamos con la moda |

# Modelos analíticos

Por último eliminamos algunas variables que no tienen importancia, y tenemos nuestro dataset listo para el procesamiento:

RangeIndex: 14319 entries, 0 to 14318

| # | 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

## 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 especifica.
- 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.

## Árbol de decisión

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:

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

# Modelos analíticos

Y obtenemos las siguientes métricas:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Inicial | 1.00 | 1.00 | 1.0 | 446 |
| Primario | 0.93 | 0.99 | 0.95 | 978 |
| Secundario | 0.95 | 0.96 | 0.95 | 1771 |
| Superior | 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.

# Modelos analíticos

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 constatarse 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.

# Modelos analíticos

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

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

Que nos da como resultado:

| | precision | recall | f1-score | suppo |
|---|---|---|---|---|
| Inicial | 0.83 | 0.79 | 0.81 | 446 |
| Primario | 0.45 | 0.26 | 0.33 | 978 |
| Secundario | 0.56 | 0.66 | 0.60 | 1771 |
| Superior | 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 |

# Modelos analíticos

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.

## 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".

# Modelos analíticos

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

| | precision | recall | f1-score | suppo |
|---|---|---|---|---|
| Inicial | 0.99 | 0.74 | 0.85 | 446 |
| Primario | 0.49 | 0.21 | 0.30 | 978 |
| Secundario | 0.55 | 0.87 | 0.69 | 1771 |
| Superior | 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 %.

## Random Forest Classifier

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

## 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.

# Modelos analíticos

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

| | precision | recall | f1-score | suppo |
|---|---|---|---|---|
| Inicial | 1.00 | 0.96 | 0.98 | 446 |
| Primario | 0.86 | 0.95 | 0.90 | 978 |
| Secundario | 0.91 | 0.93 | 0.9 | 1771 |
| Superior | 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.

## Conclusión

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.

# 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 |
|---|---|---|---|---|
| Inicial | 0.95 | 0.78 | 0.86 | 446 |
| Primario | 0.54 | 0.23 | 0.32 | 978 |
| Secundario | 0.56 | 0.88 | 0.69 | 1771 |
| Superior | 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 |

# Modelos analíticos

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.

## 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".

# Modelos analíticos

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

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Inicial | 0.95 | 0.79 | 0.86 | 446 |
| Primario | 0.56 | 0.23 | 0.33 | 978 |
| Secundario | 0.56 | 0.89 | 0.69 | 1771 |
| Superior | 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 **OVERFITING**.

# 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 | f1_inicial | f1_pri | f1_sec | f1_sup |
|---|---|---|---|---|---|---|---|
| árbol_default | 0.53 | 1.00 | 0.46 | 0.81 | 0.33 | 0.60 | 0.42 |
| árbol_mejorado | 0.60 | 0.65 | 0.05 | 0.85 | 0.30 | 0.69 | 0.53 |
| bosque_default | 0.62 | 0.89 | 0.27 | 0.86 | 0.32 | 0.69 | 0.53 |
| bosque_mejorado | 0.62 | 0.91 | 0.29 | 0.86 | 0.33 | 0.69 | 0.54 |

# Conclusiones

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, asì como su capacidad de ser visualizada en un gráfico.