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

Principales hitos:

- Al estudiar los índices correspondientes a uno de los ejes de mayor importancia para la sociedad, educación, en la Ciudad Autónoma de Buenos Aires, se ha encontrado una gran limitación relacionada con su acceso inequitativo para los diferentes actores de la sociedad.
- 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.
- 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.

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.

Estructura de los trabajos

Este trabajo se ha dividido en 3 partes:

1. **Introducción a las variables del problema:** Se realiza un análisis EDA de las variables del dataset para conocer su performance dentro del dataset y cómo interactúan entre sí.
2. **Modelos analíticos:** En esta sección se entrenan diversos modelos analíticos y algoritmos que sirven para alcanzar los objetivos seteados para el proyecto. Como la variable objetivo es categórica, se realizan modelos de clasificación.
3. **Conclusión:** Se extraen 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)

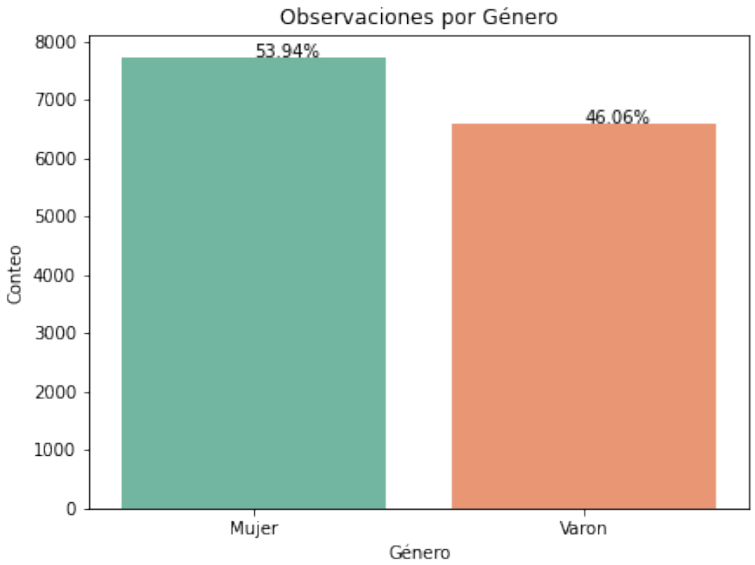
Introducción a las variables:
Encontramos 31 variables dentro de dataset, de las cuales 10 son numéricas y 21 son categóricas. A su vez estas se relaciona con:

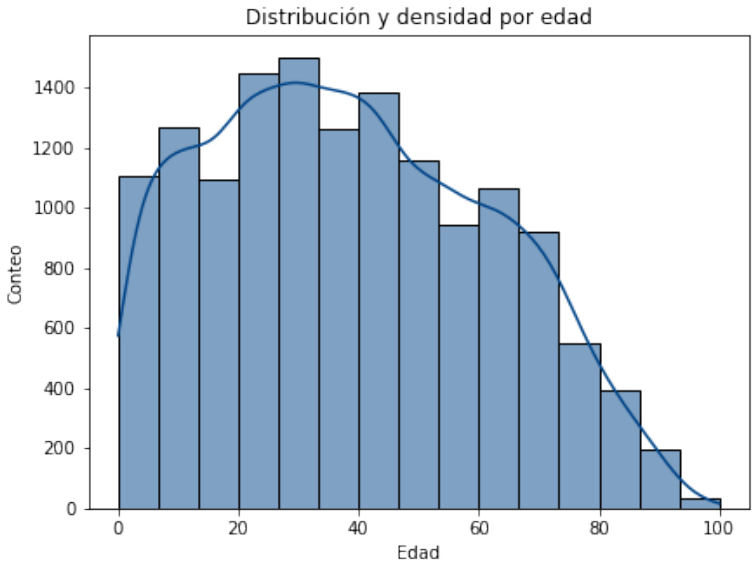
- Ingresos
- Sector educativo
- Factores geográficos
- Salud
- Descripción del grupo familiar

Identificación del Target: En primer lugar transformamos la variable Target (nivel máximo educativo) para reducir su dimensionalidad. Quedando cuatro valores dentro de la variables: **Inicial**, **Prim. Completo**, **Sec. Completo** y **Superior**

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.

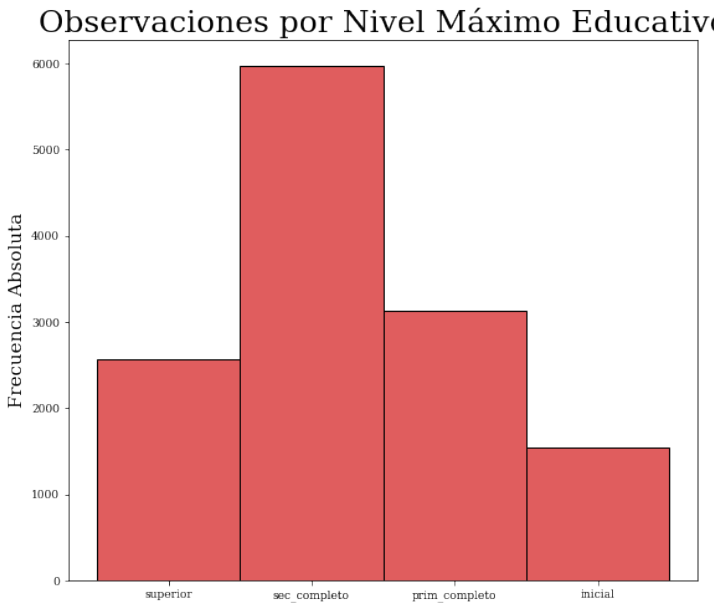


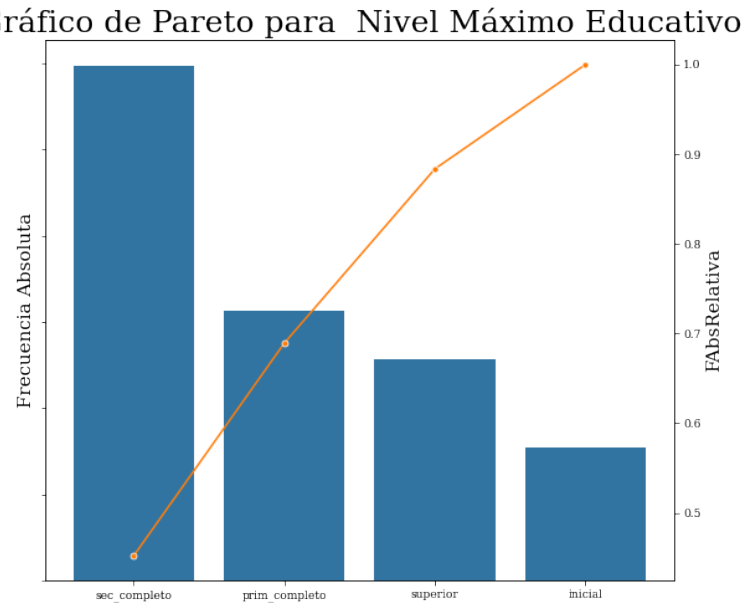


En la variable género los datos parecen equilibrados en las categorías. Para el caso de la variable "edad", la distribución se asemeja a la de una normal.

Análisis univariado

Cuando observamos el **nivel máximo educativo** encontramos que el 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.

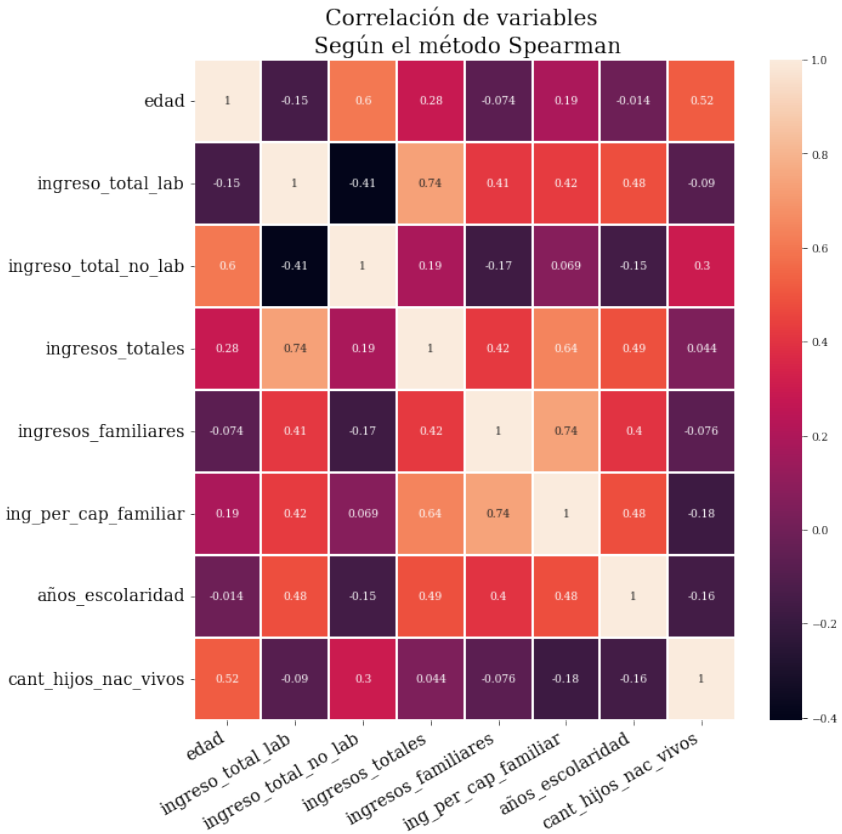




Análisis bivariado

Variable numéricas:

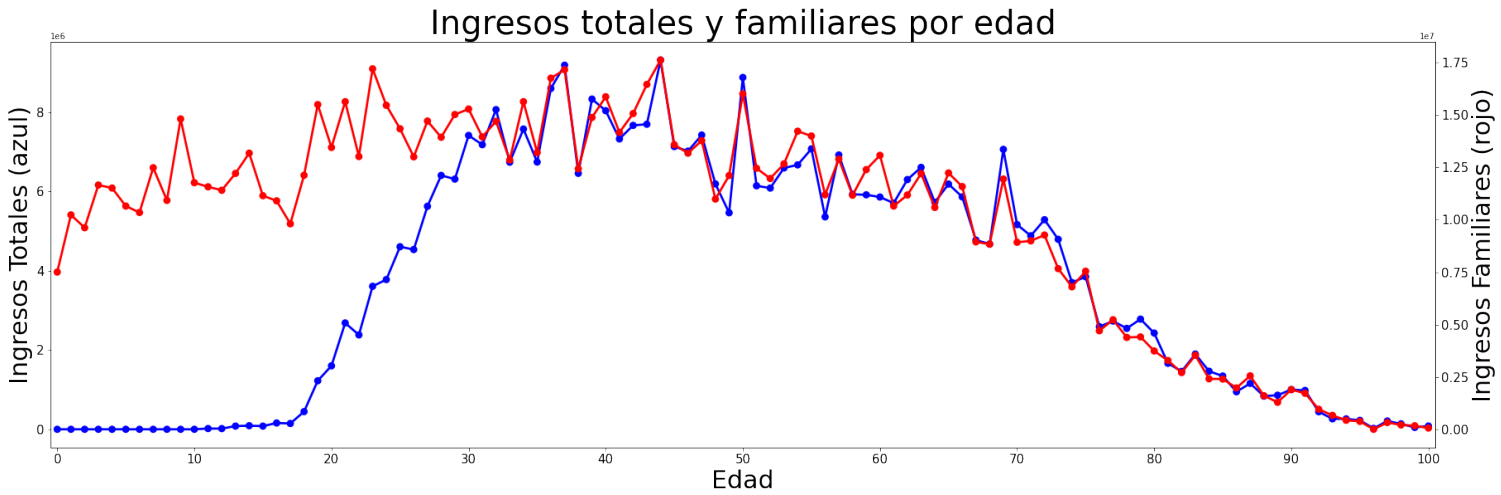
Realizamos un mapa de calor para ver la interacción entre las variables numéricas.



- 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

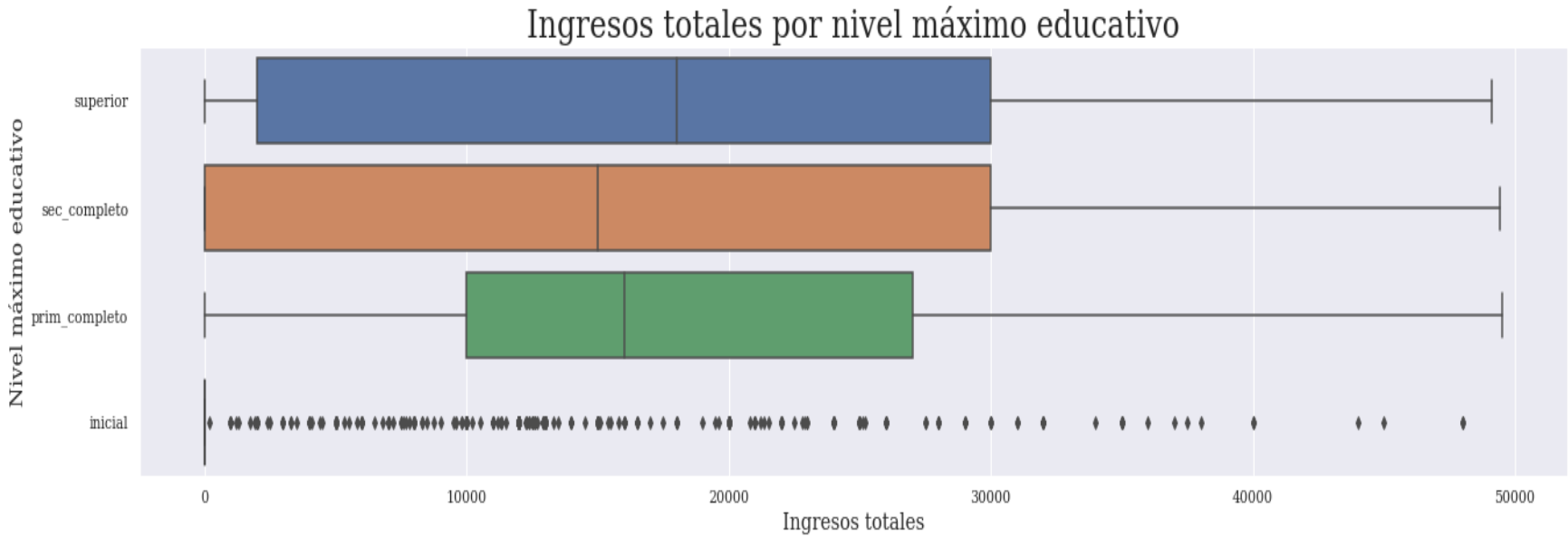
Observamos la relación entre los ingresos totales de cada hogar y los ingresos familiares por edad:



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.

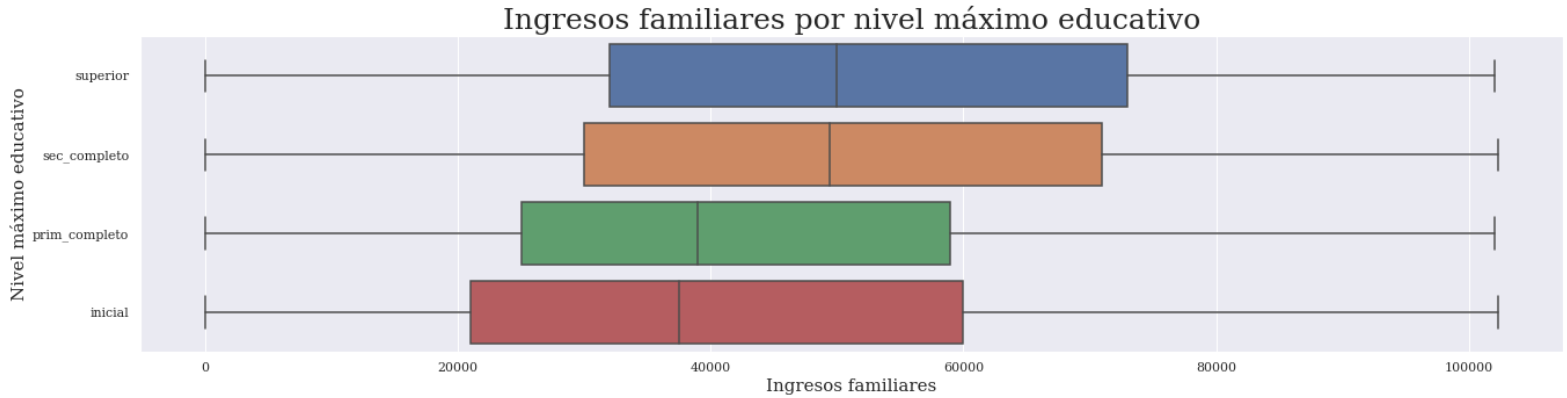
Variable numéricas y categóricas

Adicionalmente, comparamos algunas variables con nuestro target, comenzando con los ingresos totales.



Análisis bivariado

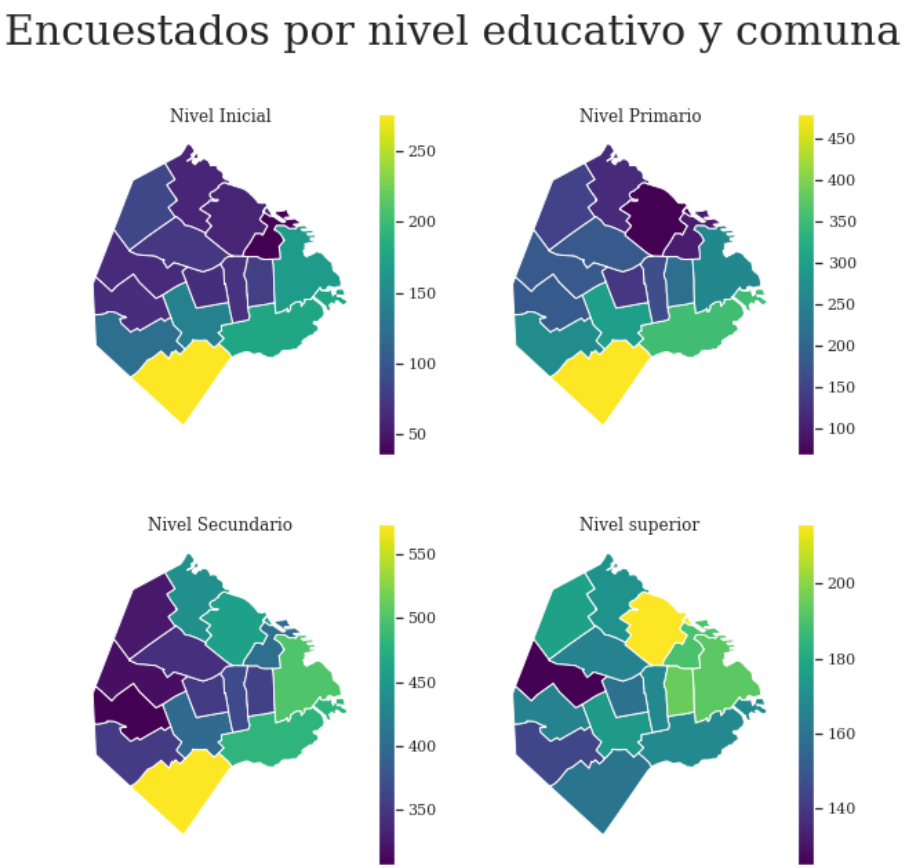
Más aún, analizamos como se distribuyen los ingresos familiares con respecto a nuestro target:



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

Análisis bivariado

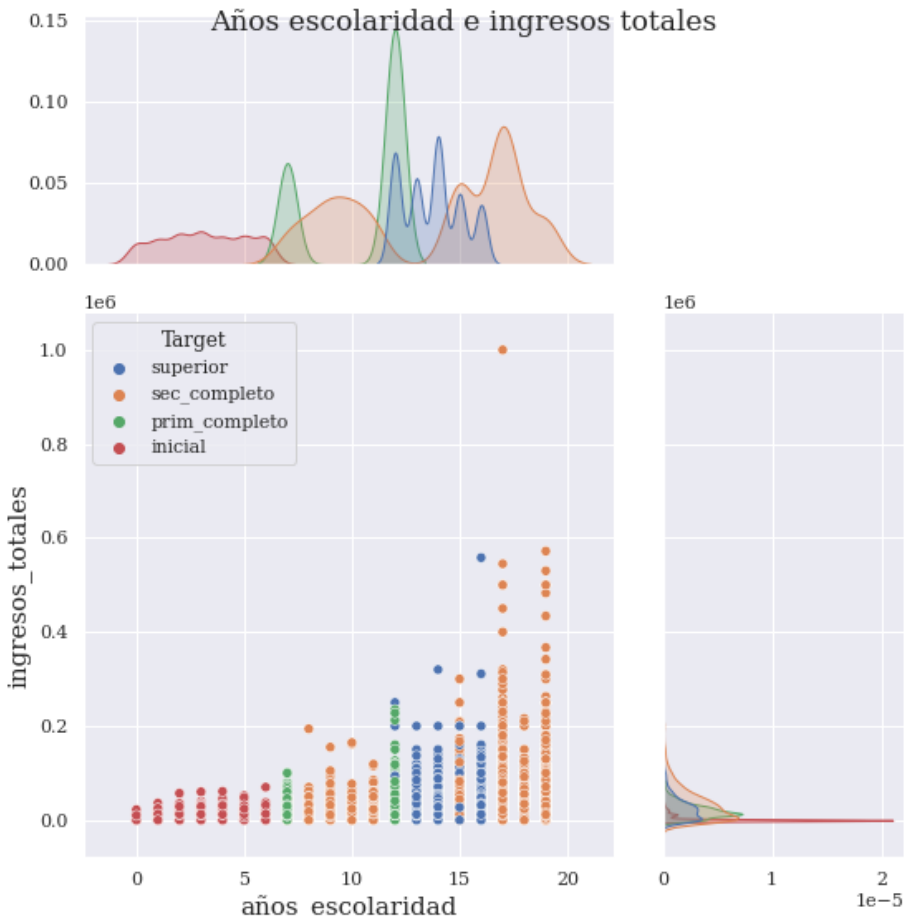
Por último, analizamos la relación de nuestro target en cada comuna:



- 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 multivariado

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

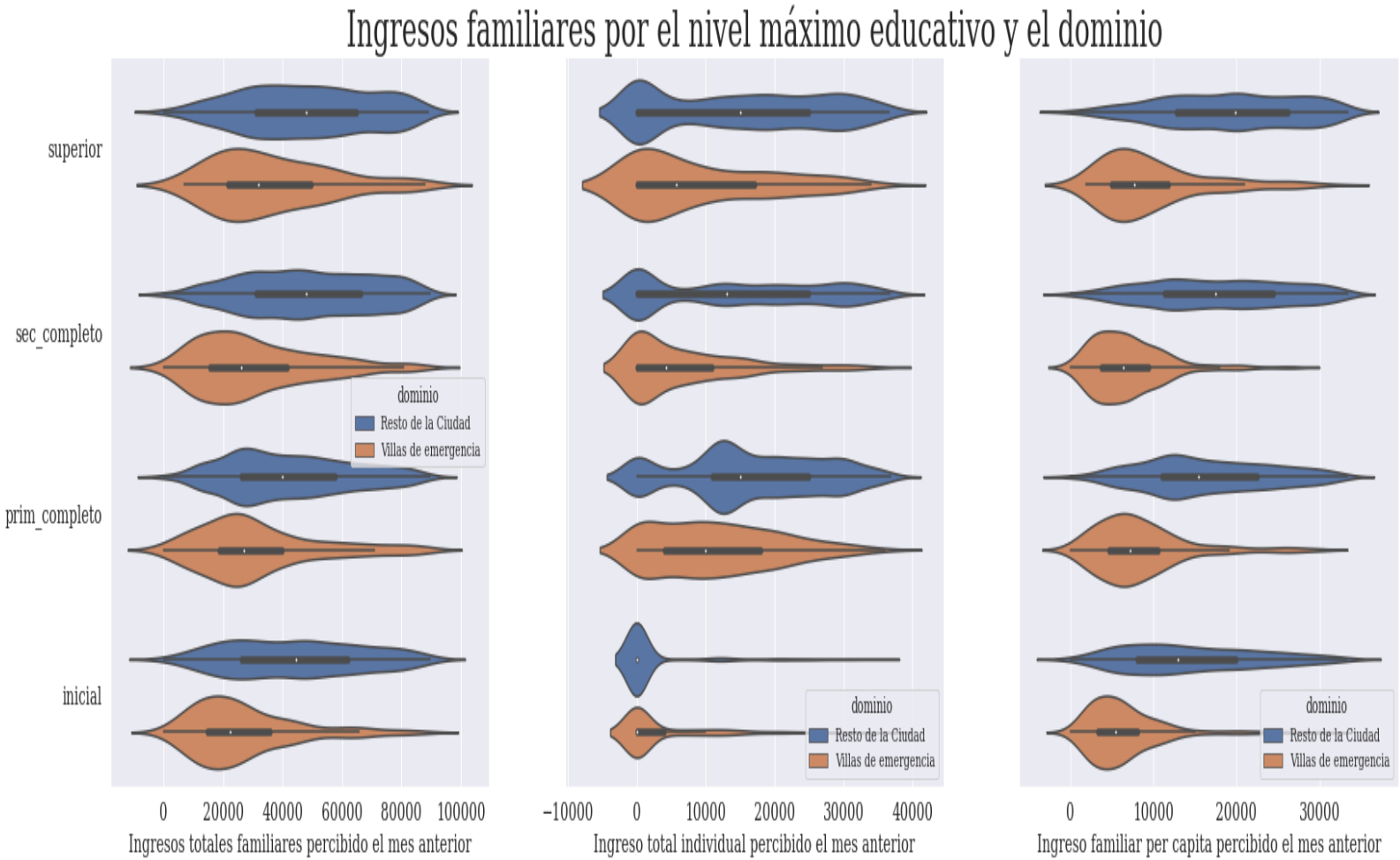


Al visualizar el gráfico podemos observar que:

- 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

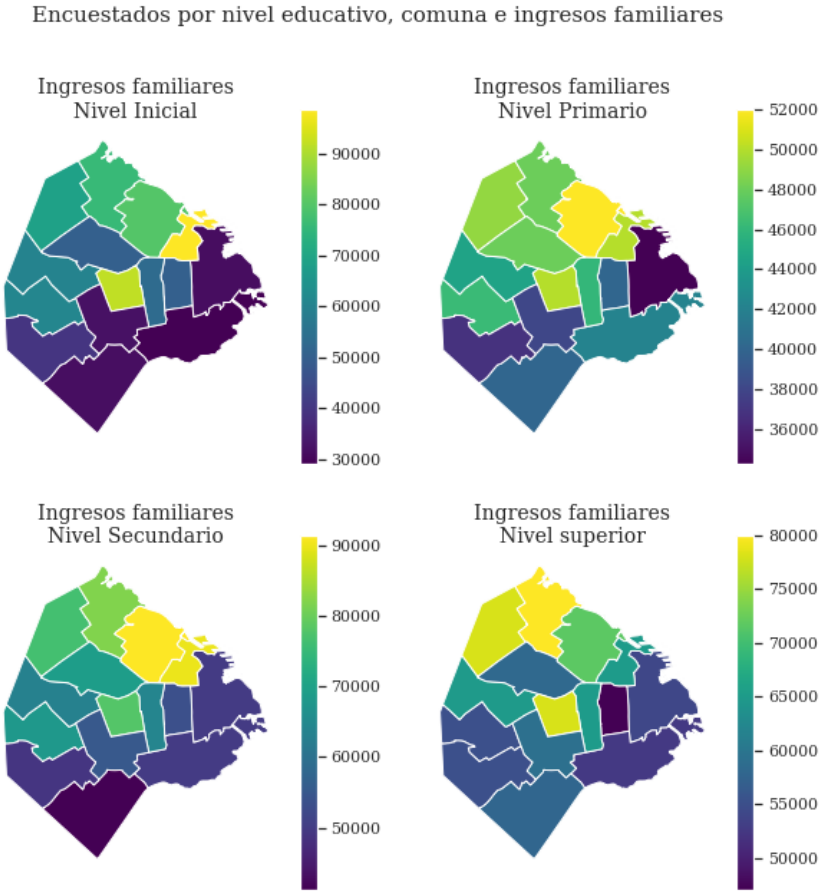
Añadimos una variable más: dominio =“villas_de_emergencia”:



Podemos ver que los casos que no provienen de villas de emergencia obtienen en promedio ingresos más altos en todos los niveles educativos. El alcanzar estudios superiores no parece homogeneizar ambos conjuntos.

Análisis multivariado

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



- 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.

Figura: Figure X

Modelos analíticos

Transformamos 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),
- eliminamos algunas variables que no resultan relevantes,
- renombramos algunas variables para que sean más cortas.

Tratados de nulos

Encontramos valores nulos en estas variables, por lo que para trabajar con el modelado los remplazamos con las siguientes métricas:

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

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.

Árbol de decisión

1. Primer modelo:

Como punto de partida usamos un árbol de clasificación con parámetros:

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

El Accuracy score para el test es de: 0.940 y las 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

Resultados

- **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.

Sin embargo, tenemos que la variable "años_escolaridad" tiene una importancia del 84 %, por mucho superior al resto de variables.

Por lo tanto, desarrollamos un nuevo modelo sin esta variable.

2. Segundo modelo:

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

Resultados

- **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 bastante 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”.

Gridsearch con CV

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

Resultados

- **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**

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

3. Tercer modelo:

En esta instancia hemos teniendo en cuenta todo el dataset incluido la variable con los años de escolaridad.

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

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

Random Forest Classifier

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.

Resultados

- **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.

Obtuvimos un buen modelo. No obstante, 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. Cuarto 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

Resultados

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

- **Bias o sesgo:** 89.11 % que nos indica que tengo pocos errores, es decir, que el sesgo es bastante bajo,
- **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”.

Gridsearch con CV

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

Resultados

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

Lo que nos indica que nuestro modelo esta haciendo **OVERFITING**.
Al utilizar random forest hemos podido mejorar el sesgo y disminuir el underfitting en 2 puntos porcentuales aproximadamente.

Sin embargo, se vio afectada la varianza en estos modelos, que pasó de estar alrededor del 5 % en el árbol de decisión mejorado a 28 %.

Conclusiones Finales

Finalmente, tomamos las métricas de cada uno de ellos y hacemos 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

- 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.

Conclusiones Finales

Los finalistas son el árbol y el bosque mejorado. Ambos performan muy bien pero en métricas diferentes.

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.
- Tiene mayor velocidad de entrenamiento
- Tiene mayor capacidad de ser visualizada en un gráfico.

En futuras líneas de investigación se debería investigar en profundidad el desbalanceo de datos propio del Target y mitigar la problemática agregando datos en categorías con deficit y eliminando datos de categorías en exceso