



INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA

ESCUELA DE INGENIERÍA Y GESTIÓN

Singularidades en Curvas Poblacionales para el AMBA 1991-2022

Fernando Meseri

1 de septiembre de 2024

AUTOR: MESERI, Fernando (Leg. N° 503.801)

TUTOR/ES: GOMEZ, Leticia I.

*TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE ESPECIALISTA
EN CIENCIA DE DATOS*

BUENOS AIRES

PRIMER CUATRIMESTRE, 2024

1. Introducción

La información estadística que brindan las proyecciones de población representa un insumo vital en la implementación de políticas estatales. Dichas proyecciones es común encontrarlas a nivel País o Provincia, pero resulta particularmente importante poder contar con dichas proyecciones con un mayor nivel de desagregación (Municipios). En el presente trabajo se analiza en particular los municipios de la zona AMBA (Área Metropolitana de Buenos Aires, Argentina), para el período 1991-2022. El AMBA técnicamente incluye 40 municipios: Almirante Brown, Avellaneda, Berazategui, Berisso, Brandsen, Campana, Cañuelas, Ensenada, Escobar, Esteban Echeverría, Exaltación, Ezeiza, Florencio Varela, Gral. Las Heras, Gral. Rodríguez, Gral. San Martín, Hurlingham, Ituzaingó, José C. Paz, La Matanza, La Plata, Lanús, Lomas de Zamora, Luján, Malvinas Argentinas, Marcos Paz, Merlo, Moreno, Morón, Quilmes, Pilar, Presidente Perón, San Fernando, San Isidro, San Miguel, San Vicente, Tigre, Tres de Febrero, Vicente López, Zárate. Ocupa un territorio de aproximadamente 3.833km^2 y concentra 35 % de la población nacional, siendo el área geográfica más poblada del país y configurándose históricamente, como el núcleo central del sistema urbano argentino.

El Instituto Nacional de Estadísticas y Censos (INDEC) en general trabaja diferenciadamente 24(veinticuatro) de estos municipios del AMBA , siendo los que se analizarán en este trabajo. A saber: Almirante Brown, Avellaneda, Berazategui, Esteban Echeverría Ezeiza, Florencio Varela, General San Martín, Hurlingham, Ituzaingó, José C. Paz, La Matanza, Lanús, Lomas de Zamora, Malvinas Argentinas Merlo, Moreno, Morón, Quilmes, San Fernando, San Isidro, San Miguel, Tigre, Tres de Febrero y Vicente López.

Justamente el INDEC ha estimado la población de dichos municipios para el período 2010-2025 (INDEC, 2015)[12]. Si se analiza los errores de estas proyecciones respecto al valor arrojado para el censo 2022 (INDEC, 2022)[10] , surge que La Matanza presenta una desviación importante respecto al error promedio encontrado para otros departamentos.

En el presente trabajo, a partir del análisis de datos censales y variables indirectas, se analizarán las curvas poblacionales del AMBA y se buscará estimar la población para el año 2022, para luego relevar el error respecto al censo 2022, según las distintas metodologías aplicadas. Se analizará puntualmente también La Matanza.

2. Contexto

2.1. Marco Conceptual

La información estadística que brindan las proyecciones de población en general es utilizada en la planificación de políticas públicas de corto, mediano y largo plazo. Permite estimar demanda potencial de bienes y servicios en distintas áreas como Salud, Educación, entre otras (INDEC, 2015)[12].

De esta forma, el Estado puede determinar los recursos presupuestarios necesarios para satisfacer estas demandas. En la provincia de Buenos Aires ciertos aspectos del presupuesto son asignados en base a la población de cada municipio. Es necesario entonces contar con la información en nivel de desagregación espacial municipal (Departamentos).

2.2. Marco Teórico

La elaboración de proyecciones de población es una tarea compleja que debe ser realizada a través de un análisis exhaustivo que permita considerar los censos anteriores como también registros vitales y estimaciones de migración (INDEC, 2013)[13].

En general, se ha utilizado el Método de las Componentes para elaborar dichas proyecciones. Mas esta metodología no ha podido ser replicada al nivel de las jurisdicciones más elementales (departamentos), por cuanto la información no es suficientemente confiable y la inestabilidad de la migración interna no admite formulación de hipótesis a mediano plazo (Álvarez et al., 2001)[1]. Una forma de realizar estas predicciones ha sido mediante métodos matemáticos de extrapolación en base a la información censal previa (Álvarez et al., 2001)[1].

El INDEC provee proyecciones de población por departamento para el período 2010-2025 (INDEC, 2013)[13], particularmente para todos los municipios del AMBA. Dichas estimaciones se apoyan en el crecimiento intercensal, que comprende la evolución en conjunto de las tres variables básicas del análisis demográfico: la fecundidad, la mortalidad y la migración del período 2001-2010. Se destaca que el crecimiento de la población en Argentina observado en este

período a nivel departamental pone en evidencia las diferencias geográficas que existen en la dinámica poblacional, con un comportamiento heterogéneo.

2.3. Estado del Arte

Históricamente se observa conceso en la utilización del Método de las Componentes para la determinación de proyecciones poblacionales a nivel País o Provincia. El mismo contempla el crecimiento poblacional intercensal y proyecta cada una de las variables determinantes de forma independiente -fecundidad, mortalidad y migración (Álvarez et al., 2001)[1].

Ciertamente, la Serie de Análisis Demográfico de INDEC utiliza este método para la proyecciones Nacionales y Provinciales (INDEC, 2015)[12]. Asimismo, para población de países desarrollados, también se ha utilizado el modelo de Regresión Logística para este tipo de predicciones (Gupta, Bhattacharya y Chattyopadhyay, 2012)[6]. Pero este modelo tiene ciertas limitaciones cuando se aplica a data censal dispersa en el tiempo, especialmente para países en desarrollo. Generalmente, las tasas de crecimiento relativo presentan tendencias inusuales, distintas a la tendencia decreciente de la regresión logística. Gupta et al., (2012) proponen modelos simplificados y variantes de Tsoularis and Wallace Model (TWM) que han proporcionado mejores resultados. A mayor nivel de desagregación, se trabaja con métodos alternativos, como puede ser extrapolación matemática, Ratio-Correlation Method, Housing Unit Method, entre otros (Hoque, 2012)[7].

Por otra parte, el centro Latinoamericano y Caribeño de Demografía (CELADE) ha promovido la utilización de otras técnicas para mejorar las estimaciones poblaciones derivadas de la extrapolación matemática. Se utiliza la metodología de variables sintomáticas, que permite establecer correlaciones a las tendencias poblacionales con información de variables indirectamente asociadas al fenómeno de crecimiento poblacional, a saber: nacimientos y defunciones, matrícula escolar, permisos de construcción, otros (Álvarez et al., 2001)[1].

En lo que respecta a técnicas propias de ciencias de datos para análisis de información censal, se desatacan los siguientes usos: la utilización de modelos para búsqueda de patrones en la información censal, predicciones y forecasting utilizando modelos ARIMA e inducción con árboles de decisión (Chawda, Rane y Giri, 2018)[4]. También se destaca el uso de árboles de regresión y clasificación para el agrupamiento o clustering en distintas clases, tomando como input información censal.

3. Definición del Problema

El municipio de La Matanza presenta una singularidad en su curva de crecimiento poblacional, tanto en número de habitantes como en tasas intercensales, respecto a los municipios aledaños del AMBA para el período 1991-2022.

4. Justificación del Estudio

La información estadística que brindan las proyecciones de población constituye una herramienta fundamental para la planificación de políticas públicas de corto, mediano y largo plazo. Permite estimar demanda potencial de bienes y servicios en distintas áreas como Salud y Educación, entre otras (INDEC, 2013)[13].

En las proyecciones del INDEC (INDEC, 2015)[12] se ha estimado la población para los municipios del AMBA, pero el valor arrojado por el censo 2022 en el caso de La Matanza, presenta una desviación importante respecto al error promedio encontrado para el resto de los municipios. Es por esto, que se pretende analizar la singularidad en la curva poblacional de La Matanza.

5. Alcances del Trabajo y Limitaciones

El alcance del trabajo es básicamente el análisis y estudio de datos censales. Comprende principalmente la utilización de datos censales del AMBA para el período 1991-2022, para analizar las curvas poblacionales, compararlas y detectar el error en las proyecciones INDEC respecto a lo arrojado en el censo 2022 (INDEC, 2022)[10]. El trabajo se limita a demostrar la singularidad o no en el dato poblacional de la Matanza, sin intentar explicar las causas del fenómeno demográfico que pudiese estar detrás de esta singularidad.

6. Hipótesis

Es posible demostrar que, para el período 1991-2022, la curva de crecimiento poblacional de la Matanza presenta una singularidad respecto a los municipios aledaños (AMBA) en situaciones socio-demográficas similares. Se trata de una hipótesis multivariable donde las variables como población, tasa de natalidad y de mortalidad, sexo de las personas, entre otras, podrían tener una relación causa-efecto sobre el fenómeno analizado.

Fundamento

1. Utilizando estadística descriptiva, se puede estimar una tasa de crecimiento promedio de la población urbana/-suburbana en base a los cuatro Censos anteriores.
2. Dado que el censo incluye la desagregación por departamento/municipio, es posible individualizar las tasas de crecimiento por municipio, y luego agruparlos en sectores de interés, ej. AMBA.
3. Es esperable que municipios aledaños con iguales características socio-demográficas muestren patronales similares y presenten un crecimiento en el mismo orden de magnitud
4. Considerando las curvas de crecimiento reales de los municipios y las proyecciones realizadas, dado un intervalo de confianza, se puede llegar a determinar que la Matanza presenta una singularidad, aunque las causas de este fenómeno no son de interés del presente trabajo. El fenómeno demográfico es complejo, multicausal. Aun así, se pueden establecer valores de referencia acordes a la región, donde las características socio-demográficas son similares (conurbano).

7. Objetivos

7.1. Objetivo Principal

El objetivo del presente trabajo consiste en demostrar que la tasa de crecimiento y curva poblacional de la Matanza desde 1991 hasta 2022 presenta una singularidad respecto a aquella desarrollada por los otros municipios de conurbano bonaerense. Tomando como base los datos obtenidos en los 4 últimos censos nacionales 1991, 2001, 2010 y 2022.

7.2. Objetivos Específicos

1. Procesar y unificar la información censal desde 1991 a 2022 para distintos niveles de granularidad, ya que en algunos casos no se presentan todas las variables. Se requiere un pre procesamiento y limpieza de datos importante.
2. Exploratory Data Analysis: Construir las curvas poblaciones y comparaciones geográficas de las mismas.
3. Realizar proyecciones para población censo 2022 con metodologías tradicionales modernas utilizando las variables en forma individual o combinadas, tomando como base los 3 censos anteriores (1991-2001-2010).
4. Realizar proyecciones para el censo 2022 mediante data mining (decision tree regression algoritm, others), utilizando los 3 censos anteriores (1991-2001-2010)
5. Comparar las curvas censales y su ajuste con las proyecciones realizadas para 2022.-
6. Determinar el error de las distintas proyecciones para 2022 para todos los municipios, comparar con la situación de la Matanza.
7. Inferir si existe alguna metodología que ajuste mejor las proyecciones para el caso del AMBA.

8. Metodología

En esta sección se detallarán las metodologías utilizadas para realizar predicciones del valor poblacional de todos los departamentos del AMBA para el Año 2022, basados en los 3 censos anteriores y, dependiendo del método, variables sintomáticas típicas del fenómeno demográfico. Es importante destacar que ciertas metodologías tradicionales no son susceptibles de ser utilizadas debido al nivel de agregación analizado (Departamento).

8.1. Variables

8.1.1. Variables Censales

De los censos poblacionales se obtuvieron las siguientes atributos o variables.

1. Código Depto: código de identificación de una unidad administrativa (departamento) según INDEC.
2. Población: cantidad de personas que habitan un determinado sector del territorio, en distintos niveles de aglomeración.
3. Varones: cantidad de personas de sexo masculino de un determinado sector del territorio.
4. Mujeres: cantidad de personas de sexo femenino de un determinado sector del territorio.
5. Viviendas Particulares Totales: cantidad de viviendas particulares de un determinado sector del territorio.
6. Viviendas Colectivas Totales: cantidad de viviendas colectivas de un determinado sector del territorio.
7. Índice de Masculinidad: cociente entre el número de hombres y el número de mujeres de un determinado sector del territorio.

8.1.2. Variables Sintomáticas

En este caso, para algunas metodologías se utilizan variables sintomáticas como input en las estimaciones de población a realizar. A saber:

1. **Tasa Bruta de Natalidad(TBN)**: cociente entre el número de nacimientos ocurridos durante un período determinado, generalmente un año calendario, y la población media del período.
2. **Tasa Bruta de Mortalidad (TBM)**: cociente entre el número de defunciones ocurridas durante un período determinado, generalmente un año calendario, y la población media del período.
3. **Tasa de Crecimiento Vegetativo(TCV)**: diferencia entre entre la Tasa Bruta de Natalidad y la Tasa Bruta de Mortalidad de un período determinado, generalmente un año calendario
4. **Tasa Global de Fecundidad(TGF)**: número de hijos que en promedio tendría una mujer de una cohorte hipotética de mujeres que durante su vida fértil tuvieran sus hijos de acuerdo a las tasas de fecundidad por edad del período en estudio y no estuvieran expuestas al riesgo de mortalidad desde el nacimiento hasta el término de su período fértil.
5. **Tasa de Mortalidad Infantil(TMI)**: cociente entre el número de muertes de menores de un año acaecidas en la población de un área geográfica durante un período determinado, generalmente un año calendario, y los nacidos vivos en esa área durante el mismo período.
6. **Matrícula en ciclo primario común(Mat1ria)**: representa la cantidad de alumnos que ingresan en primer año del ciclo primario.

A partir del relevamiento de datos públicos gubernamentales, de fuentes como el Programa de Análisis Demográfico (INDEC, 2022)[11] e información del Ministerio de Educación Nacional (Educación, 2022)[5] se confeccionó un compendio de estas variables para los años de interés. En el Cuadro 1 puede observarse el valor de las mismas para cada año analizado.

Jurisdicción	Año	TMI	TGF	TBN	TBM	TCV	Mat1ria
BUENOS AIRES	1980	28.4	3.0	22.1	8.1	14.0	
BUENOS AIRES	1991	24.2	2.6	18.4	7.9	10.5	1752994.0
BUENOS AIRES	2001	15.0	2.3	16.9	8.2	8.7	1658221.0
BUENOS AIRES	2010	12.0	2.5	18.9	8.4	10.5	1667278.0
BUENOS AIRES	2022	7.9	1.89	9.7	8.8	0.899	1767473.0

Cuadro 1: Variables Sintomáticas para cálculo de estimaciones en Provincia de Buenos Aires

8.2. Metodologías Tradicionales

Resumen de la Regresión Lineal En una primera aproximación se utilizó la regresión lineal. Esta técnica estadística se utiliza para entender la relación entre dos variables. Una de estas variables es reconocida como la variable independiente (X), y la otra es la variable dependiente (Y). La idea principal es encontrar la línea recta que mejor se ajuste a los datos, denominada "línea de regresión". La regresión lineal busca minimizar la distancia vertical entre cada punto de datos y la línea de regresión. Esto significa que la línea de regresión pasa lo más cerca posible de todos los puntos de datos.

En este caso se utilizó como variable independiente el **Año** del censo y como variable dependiente la **Población**.

8.2.1. Estimaciones INDEC

Se considerará también las proyecciones realizadas por el INDEC que indican la "Población estimada al 1 de julio de cada año calendario por sexo, según partido. Provincia de Buenos Aires. Años 2010-2025" (INDEC, 2015)[12].

Estas proyecciones se apoyan en el crecimiento intercensal que comprende la evolución en conjunto de las tres variables básicas del análisis demográfico: la fecundidad, la mortalidad y la migración del período 2001–2010. Los resultados de las estimaciones departamentales de población son coherentes y consistentes con las proyecciones de población nacionales Serie de análisis demográfico N 35 (INDEC, 2013a) y provinciales vigentes (INDEC, 2013b) (INDEC, 2015)[12].

8.3. Técnicas Data Mining

8.3.1. CART, Random Forest y LightGBM

Si bien en este caso los datos son dispersos ("sparse"), se aplicarán árboles de decisión simples y ensamblados, y otros algoritmos de clasificación tal cual lo expuesto en el artículo "Exploring New Models for Population Prediction in Detecting Demographic Phase Change for Sparse Census Data" (Chawda, Rane y Giri, 2018)[4].

Los árboles de decisión son una técnica de aprendizaje automático y análisis de datos que se utiliza tanto para problemas de clasificación como de regresión (CART: Classification And Regression Tree). Su objetivo es tomar decisiones o hacer predicciones basadas en una serie de preguntas o condiciones. Debido al sesgo habitual en los árboles individuales, se suele combinar múltiples árboles de decisión, en pos de mejorar la precisión y la robustez de las predicciones, obteniéndose una técnica de aprendizaje automático denominada Random Forest. Es un método de ensamble, lo que significa que construye varios modelos y combina sus resultados para obtener una predicción final.

Por último, LightGBM es un framework de aprendizaje automático que utiliza el algoritmo de boosting para construir modelos predictivos. Boosting es una técnica de ensamble que crea un fuerte modelo predictivo a partir de una combinación de modelos más débiles, generalmente árboles de decisión. Se trata de una implementación del algoritmo de gradient boosting diseñado para ser eficiente y escalable, tanto en términos de tiempo de entrenamiento como de uso de memoria.

8.4. Herramientas

Como herramientas se contemplan los lenguajes Python y SQL. Los datos se almacenan y manipulan en una base de datos POSTGRES, para visualización geográfica se utilizará QGIS. En cuanto al scripting para el modelado de estas metodologías se gestionará mediante Jupyter Notebooks.

Adicionalmente, se desarrolló un proceso de ETL en Python y SQL, mediante una primera instancia de creación de todas las tablas necesarias, para luego producir la ingestión y transformación de los diferentes inputs que componen el modelo de datos.

9. Obtención de Datos Fuente y Análisis Exploratorio de Datos

En primera instancia se buscó recopilar los datos de los censos poblacionales elaborados por el INDEC (Instituto Nacional de Estadísticas y Censos, Argentina) desde 1991. Desde los repositorios web del INDEC se pudieron obtener los datos correspondientes a los censos de 1991, 2001, 2010 y 2022 (UN-CEPAL, 2023)[2] (CEPAL, 2023)[3] (INDEC,

2022)[10]. En general el INDEC dispone la información en formatos CSV o bien archivos Microsoft Excel de difícil procesado para su ingesta por parte de un proceso de ETL.

En primera instancia los archivos no cuentan con estructura de encabezados en la primera línea, presentan espacios en blanco no uniformas, así como descripciones y comentarios justamente en las primeras líneas del archivo. En general no respetan formatos uniformes según la dimensión de análisis (población por sexo, población por edad , etc) de los datos dentro del mismo período censal. Asimismo los formatos de presentación cambian de censo a censo. Por otra parte, el formato de presentación de los archivos ”.csv” así como las variables relevadas se han visto modificadas desde el censo 1991 hasta el 2022. Por ejemplo el censo 1991 no relevó la cantidad de habitantes en viviendas colectivas y particulares.

Una vez analizada las variables comunes a todos estos censos, fue necesario un pre-proceso de los archivos para lograr uniformidad y consistencia, de forma de que sean tomados por el proceso de ETL. Se removieron espacios en blanco y comentarios en la parte superior, y se eliminaron columnas nulas. Luego fue necesario unificar y adaptar todos los archivos a la misma nomenclatura y cantidad de columnas.

Es importante aclarar que existe una base consolidada - (CEPAL, 2023)[3]- (CEPAL, 2023)[3] con toda la información censal para los censos 2001 y 2010, pero lamentablemente, la misma no está disponible para los censos de 1991 y 2022. Esta base sólo puede accederse mediante REDATAM, un software para procesamiento estadístico especializado en microdatos de censos de población y vivienda, encuestas y estadísticas vitales, desarrollado por el CELADE-División de Población de la CEPAL, de las Naciones Unidas. Durante la fase de relevamiento de fuentes, se analizó y manipuló este software sin resultados alentadores.

Al finalizar el proceso de obtención de datos, las variables censales definidas, comunes al período 1991-2022, son las que se indican en el Cuadro 2.

Variable	Description
Censo	Año del censo indicado como entero de 4 dígitos
CodigoDepto	Código de 4 dígitos con el INDEC indica los municipios / departamentos censales
Población	Cantidad de personas que habitan un determinado sector del territorio, en distintos niveles de agregación
Varones	Cantidad de personas de sexo masculino
Mujeres	Cantidad de personas de sexo femenino
VivPartTot	Cantidad de viviendas particulares totales para ese nivel de agregación
VivColectTot	Cantidad de viviendas colectivas totales para ese nivel de agregación
IndMasc	Índice de masculinidad
Superficie	Superficie total del partido a la fecha del censo

Cuadro 2: Descripción de las variables comunes al período 1991-2022

9.1. Dimensión Departamento: Slowly Changing Dimension

Al analizar los distintos censos desde 1991 hasta 2022 se observa claramente que los departamentos del AMBA han modificado su división político-administrativa en este período. Es decir, se han creado partidos nuevos, así como en otros casos, hay partidos que cedieron superficie o se fusionaron con otros. Este fenómeno genera que la dimensión ‘departamento’ cambie, particularmente desde el censo 1991 al censo 2001.

Para tratar con estas modificaciones, se analizó el enfoque Slowly Changing Dimensions (SCD) para bases de datos. Este concepto fue introducido por Kimball (Kimball, 1996)[14], y refiere a casos donde las dimensiones sufren pequeñas modificaciones que afectan el modelo lógico del almacén de datos.

El caso de SCD Tipo 1 corresponde a cambios esporádicos, no permanentes, que implican una modificación en algun

atributo de la dimensión. En definitiva es una corrección de este atributo, se ejecuta mediante un UPDATE de las filas involucradas sin conservar el valor anterior del atributo. La instancia de SCD Tipo 2 se basa en la teoría de datos temporales, donde para la versión que es afectada se produce un versionado. Es decir, se inserta una nueva tupla cada vez que se produce un cambio. En estos casos, generalmente se extiende la dimensión con columnas "FROM" y "TO" que indican fechas de validez para los atributos correspondientes a ese valor de la dimensión. Obviamente, esto afecta las consultas y todos los procesos de ETL involucrados en el almacén de datos. Por último, se habla de SCD Tipo 3 cuando, si bien se desea mantener el cambio y relevan el mismo sin sobrescribir la tupla, no se mantiene el historial de la dimensión. En este caso sólo se almacena el estado actual y pasado de la misma, para lo cual, simplemente se agrega una columna por cada atributo sujeto a cambios.

En nuestro caso, se utiliza el enfoque SCD Tipo 3 para la dimensión Departamento, donde se impacta el cambio sufrido en un departamento, conservando en el registro el valor anterior, de qué otro departamento proviene y la superficie correspondiente asociada. Esto es necesario para poder luego procesar información consistente en cuanto a la granularidad utilizada, y resulta particularmente importante para analizar las superficies de los departamentos.

En lo que respecta a las variables poblacionales se han reclasificado los datos de 1991 para corresponder con la división político-administrativa de año 2001. De hecho, el INDEC consigna: "Nota: con el fin de posibilitar la comparación entre los Censos 1991 y 2001, los datos que corresponden al año 1991 fueron reprocesados según la división político - administrativa vigente al año 2001" (INDEC, 2023)[9].

En el Cuadro 3 puede verse la dimensión departamento con los campos agregados y los comentarios correspondientes en cada caso.

9.2. Proyecciones INDEC 2010-2025

Se incorpora también al conjunto de datos a analizar, las proyecciones realizadas por el INDEC "Población estimada al 1 de julio de cada año calendario por sexo, según partido. Provincia de Buenos Aires. Años 2010-2025" (INDEC, 2015)[12]. Dichas estimaciones se apoyan en el crecimiento intercensal que comprende la evolución en conjunto de las tres variables básicas del análisis demográfico: la fecundidad, la mortalidad y la migración del período 2001-2010. Las mismas se utilizarán para comparar los resultados de los censos, acompañadas de proyecciones realizadas por otros métodos.

La descripción de este conjunto puede verse en el Cuadro 4, mientras que en el Cuadro 5 se muestra la estructura típica del dataset. En forma gráfica se dividen los departamentos según la población inicial para el año 2010. Se grafica, por un lado aquellos departamentos con una población menor a 300.000 habitantes (Figura 1), luego aquellos con una población de entre 300.000 y 700.000, (Figura 2) y por último La Matanza cuya población supera ampliamente los 700.000 habitantes (Figura 3).

9.3. Componente Geográfico

Con el objetivo de enriquecer el dataset se incorpora una tabla con los polígonos georeferenciados correspondientes a cada departamento. La misma se obtiene a través de capa SIG de todos los departamentos de la República Argentina del Instituto Geográfico Nacional (IGM, 2024)[8]. Luego, dichos datos se cruzan con el listado de departamentos del AMBA, generando un dataset enriquecido con las características geográficas, que completan los datos censales recopilados desde 1991 a 2022. La capa geográfica incorpora las variables que se detallan en el Cuadro 6.

9.4. Población

Se detallan a continuación los datos recopilados según nivel de agregación. En primera instancia se generó un dataset con todos los departamentos del AMBA, sus características geográficas, así como los valores de las variables censales correspondientes a los censos 1991-2001-2010-2022. En el cuadro 7 se puede observar el esquema del dataset.

En el Cuadro 8 se muestra una agregación de todos los campos del dataset a nivel global. En el mismo, podemos notar que para los campos correspondientes a población (pob, var, muj, ind_masc, dens_pob) hay una cantidad menor de valores, debido al hecho de los valores nulos que se corresponden con los casos de municipios que han desaparecido en la reorganización administrativa desde 1991 a 2001, a saber: Esteban Echeverría, Ezeiza, Florencio Varela, Hurlingham, Ituzaingó, José C. Paz, Malvinas Argentinas, Morón, San Miguel y General Sarmiento.

Codigo	Departamento	PartidoFrom	Codigo From	Sup1991	Sup2001	Comentarios
6028	Almirante Brown			122.0	122.0	
6035	Avellaneda			55.0	55.0	
6091	Berazategui			188.0	188.0	
6260	Esteban Echeverría			377.0	120.0	(1) Superficie modificada, cede tierras a Cañuelas y San Vicente y para la creación de Ezeiza y Presidente Perón. Leyes provinciales 11.550 del 20/10/1994 y 11.480 del 25/11/1993.
6270	Ezeiza	Esteban Echeverría	6260	0.0	223.0	(2) Se crea con tierras del partido de Esteban Echeverría. Ley provincial 11.550 del 20/10/1994.
6274	Florencio Varela			206.0	190.0	(3) Sup.modificada, cede tierras a Presidente Perón. Ley 11.480 del 25/11/1993.
6371	General San Martín			56.0	56.0	
6408	Hurlingham	Morón	6568	0.0	36.0	(4) Se crea con tierras del partido de Morón. Ley provincial 11.610 del 28/12/1994.
6410	Ituzaingó	Morón	6568	0.0	39.0	(5) Se crea con tierras del partido de Morón. Ley provincial 11.610 del 28/12/1994.
6412	José C. Paz	General Sarmiento	6005	0.0	50.0	(6) Se crea con tierras del partido de General Sarmiento. Ley provincial 11.551 del 20/10/1994.
6427	La Matanza			323.0	323.0	
6434	Lanús			45.0	45.0	
6490	Lomas de Zamora			89.0	89.0	
6515	Malvinas Argentinas	General Sarmiento	6005	0.0	63.0	(7) Se crea con tierras del partido de General Sarmiento e incorpora un sector del partido de Pilar. Ley provincial 11.551 del 20/10/1994.
6539	Merlo			170.0	170.0	
6560	Moreno			180.0	180.0	
6568	Morón			131.0	56.0	(8) Superficie modificada, cede tierras para la creación de los partidos de Hurlingham e Ituzaingó. Ley 11.610 del 28/12/1994.
6658	Quilmes			125.0	125.0	
6749	San Fernando			924.0	924.0	
6756	San Isidro			48.0	48.0	
6760	San Miguel	General Sarmiento	6005	0.0	80.0	(9) Se crea con tierras del partido de General Sarmiento. Ley provincial 11.551 del 20/10/1994.
6805	Tigre			360.0	360.0	
6840	Tres De Febrero			46.0	46.0	
6861	Vicente López			39.0	39.0	
6005	General Sarmiento			196.0	0.0	Desaparece para el año 2001

Cuadro 3: Dimensión Departamento

Column	Non-Null Count	Dtype
id	384	non-null
CodigoDpto	384	non-null
ano	384	non-null
Departamento	384	non-null
Poblacion	384	non-null
Varones	384	non-null
Mujeres	384	non-null

Cuadro 4: Summary del dataset public.proyecciones.

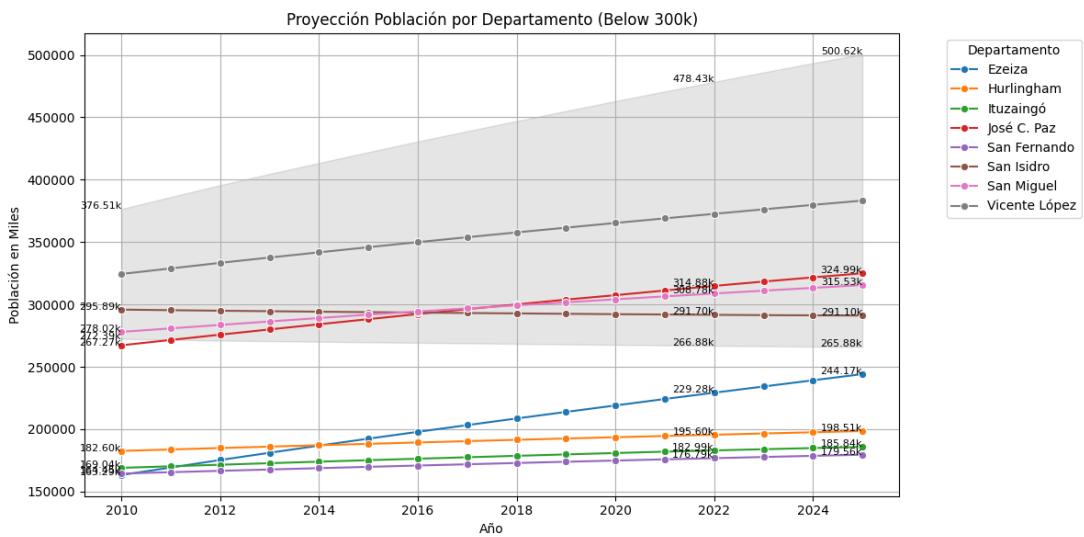


Figura 1: Población por departamento 2010-2025.INDEC

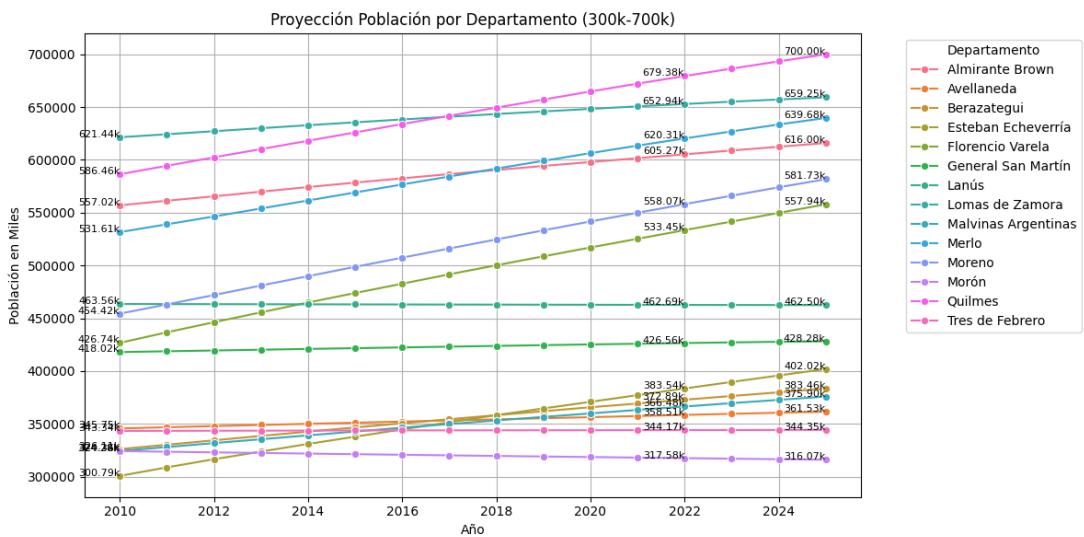


Figura 2: Población por departamento 2010-2025.INDEC

id	CodigoDpto	ano	Departamento	Poblacion	Varones	Mujeres
1	6028	2010	Almirante Brown	557025	273352	283673
25	6028	2011	Almirante Brown	561349	275570	285779
49	6028	2012	Almirante Brown	565509	277794	287715
73	6028	2013	Almirante Brown	569911	279980	289931
97	6028	2014	Almirante Brown	574263	282143	292120
121	6028	2015	Almirante Brown	578513	284281	294232
145	6028	2016	Almirante Brown	582541	286295	296246
169	6028	2017	Almirante Brown	586564	288288	298276
193	6028	2018	Almirante Brown	590418	290157	300261
217	6028	2019	Almirante Brown	594270	292051	302219
.....						

Cuadro 5: Proyecciones del valor población por departamento en el período 2010 a 2025 .Fuente INDEC

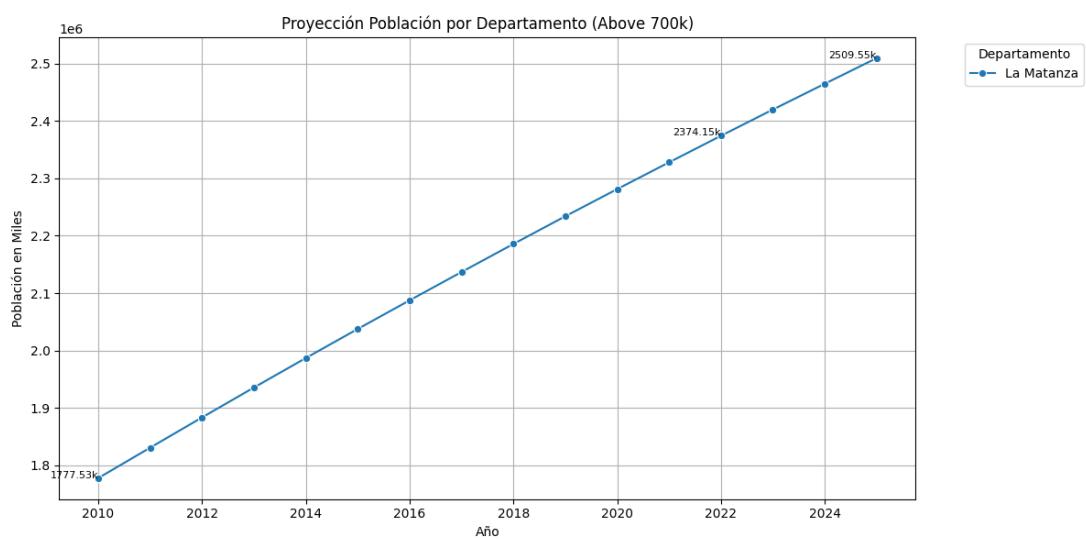


Figura 3: Población por departamento 2010-2025.INDEC

También hay una pérdida de registros vivpart y vivtotal, dado que para el censo 1991 no se detalla el total de viviendas particulares y colectivas.

9.4.1. Población: Geolocalización

En este caso se ofrece una visualización geográfica del dataset. Utilizando el software QGIS conectado directamente a la base de datos AMBA (POSTGRES) se pudo disponer la información de cada Departamento con su geolocalización. Se detalla en la Figura 4 la población total para cada departamento (Censos 1991-2022), donde se observa claramente que el distrito más poblado es La Matanza.

Variable	Nombre	Descripción
geom	geometry	Polígono WKT con los límites del departamento
fna	Nombre geográfico	Nombre completo que se utiliza para designar un objeto en un mapa o carta. Está formado por el término genérico y el término específico. Ejemplo: río Mendoza.
gna	Término genérico	Parte del nombre geográfico que indica el tipo de objeto que identifica. Ejemplo: río, monte, glaciar, establecimiento.
nam	Término específico	Parte de un nombre geográfico que acompaña al término genérico y que identifica e individualiza un objeto geográfico determinado. Ejemplo: Paraná en río Paraná; Upsala en glaciar Upsala; Las Marías en establecimiento Las Marías; Esperanza en el caso de bahía Esperanza.
in1	Código INDEC	Código único de vías de circulación asignado por el Instituto Nacional de Estadística y Censos de la República Argentina.
fdc	Fuente de captura	Identificación del nombre y tipo de fuente utilizada para capturar la información. Puede incluir fecha y otros datos adicionales.
sag	Autoridad de fuente	Nombre de la autoridad responsable de la información utilizada.

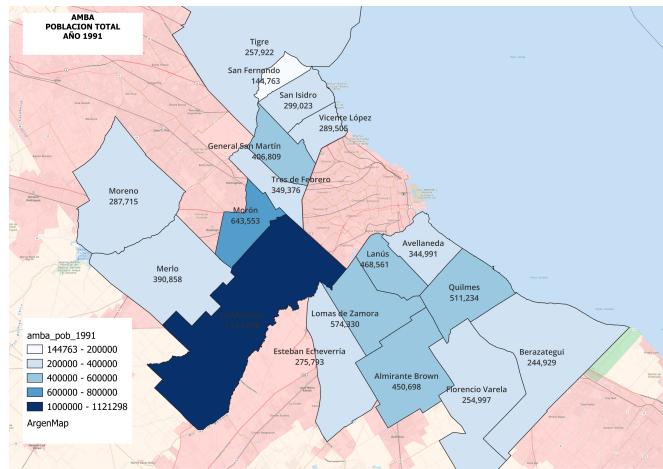
Cuadro 6: Descripción de Variables Geográficas

nam	cod_dept	anio	pob	var	muj	vivpart	vivtotal	sup	ind_masc	dens_pob
Almirante Brown	06028	1991	450698.0	222042.0	228656.0	nan	nan	157.87	97.1	2854.87
Almirante Brown	06028	2001	515556.0	252454.0	263102.0	143543.0	88.0	157.87	96.0	3265.70
Almirante Brown	06028	2010	552902.0	270247.0	282655.0	156218.0	78.0	157.87	95.6	3502.26
Almirante Brown	06028	2022	585852.0	281842.0	301779.0	184403.0	60.0	157.87	93.4	3710.98
Avellaneda	06035	1991	344991.0	164243.0	180748.0	nan	nan	68.54	90.9	5033.43
Avellaneda	06035	2001	328980.0	155450.0	173530.0	117200.0	59.0	68.54	89.6	4799.82
Avellaneda	06035	2010	342677.0	162264.0	180413.0	121307.0	68.0	68.54	89.9	4999.66
Avellaneda	06035	2022	370939.0	174572.0	194911.0	144988.0	64.0	68.54	89.6	5412.01
.....										

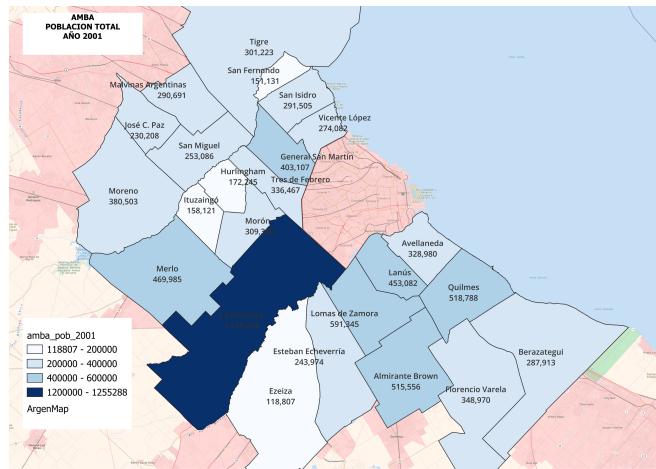
Cuadro 7: Dataset Censos AMBA

Column	Non-Null Count	Dtype
nam	96	object
cod_dept	96	object
anio	96	object
pob	90	float64
var	90	float64
muj	90	float64
vivpart	72	float64
vivtotal	72	float64
sup	96	object
ind_masc	90	object
dens_pob	90	object

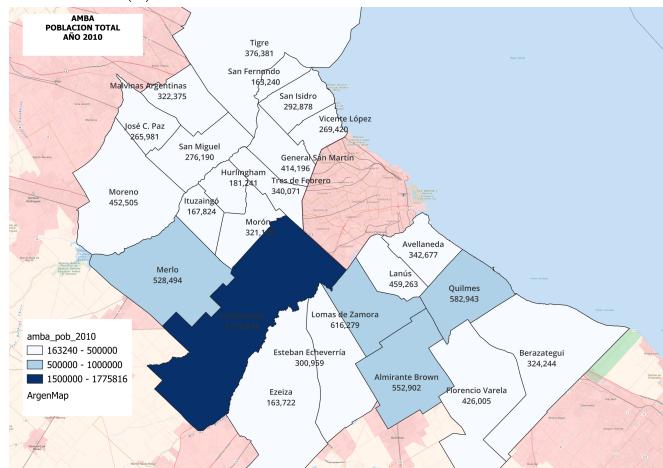
Cuadro 8: Resumen de columnas de Censos AMBA



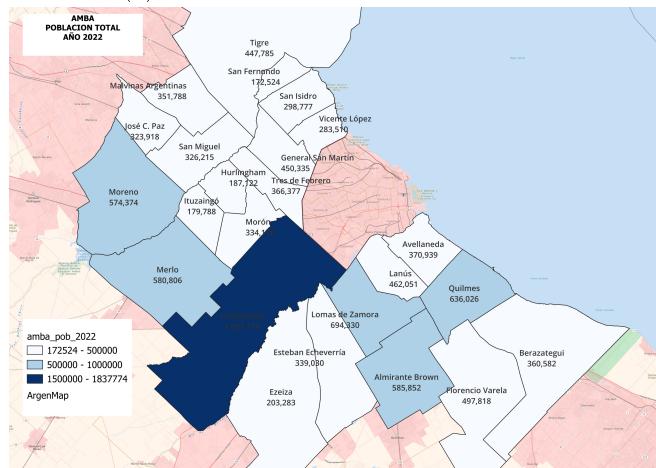
(a) AMBA- Población total Censo 1991



(b) AMBA- Población total Censo 2001



(c) AMBA- Población total Censo 2010



(d) AMBA- Población total Censo 2022

Figura 4: Población Total Censos 1991-2022

9.4.2. Tasa de Crecimiento en Población

Se determinaron los ratios de crecimiento entre censos consecutivos para cada departamento. Para este análisis se trató de manera diferenciada aquellos municipios que han sufrido modificaciones administrativas desde 1991 a 2001: **Esteban Echeverría, Ezeiza, Florencio Varela, Hurlingham, Ituzaingó, José C. Paz, Malvinas Argentinas, Morón, San Miguel y General Sarmiento.** En los mismos, al desmembrarse en varios partidos, resignando superficie, algunas tasas o ratios de crecimiento se muestran negativos. Asimismo, los partidos que han visto incrementada su superficie, presentan tasas de crecimiento no representativas del fenómeno demográfico.

Por otra parte, los partidos que desaparecen o se crean en el año 2001 no se consideran en el valor del ratio para el intervalo 1991-2001.

En el Cuadro 9 se pueden observar un fragmento de los valores obtenidos. Para el año en cuestión la tasa surge de dividir el valor poblacional del censo de este año sobre el valor poblacional del censo anterior, expresado en porcentaje.

Es decir:

$$\text{Growth Ratio} = \left(\frac{\text{Población}_{(\text{Censo } X)}}{\text{Población}_{(\text{Censo } X - 1)}} - 1 \right) \times 100 \quad (1)$$

nam	anio	pob	growth_ratio
Almirante Brown	2001	515556.0	14.39
Almirante Brown	2010	552902.0	7.24
Almirante Brown	2022	585852.0	5.96
Avellaneda	2001	328980.0	-4.64
Avellaneda	2010	342677.0	4.16
Avellaneda	2022	370939.0	8.25
Berazategui	2001	287913.0	17.55
Berazategui	2010	324244.0	12.62
Berazategui	2022	360582.0	11.21
Esteban Echeverría	2001	243974.0	-11.54
Esteban Echeverría	2010	300959.0	23.36
Esteban Echeverría	2022	339030.0	12.65
....			

Cuadro 9: Extracto: Tasa de crecimiento Intercensal

A partir de un análisis estadístico simple sobre estas tasas decrecimiento, en el Cuadro 10 podemos observar una gran disparidad en los valores, a pesar de que se trata de un sector geográfico de similares características socio-demográficas.

Estadístico	Valor
Count	66
Mean	9.3
Standard Deviation	13.1
Minimum	-51.9
25 % Percentile	3.0
50 % Percentile	8.1
75 % Percentile	16.5
Maximum	41.5

Cuadro 10: Resumen Estadístico de Tasas de Crecimiento

A partir de la elaboración de un gráfico boxplot se determinaron los Outliers para estas tasas de crecimiento. En la Figura 5 se pueden observar cuatro outliers, tanto en valores positivos como negativos. Como se comentó anteriormente no se analizarán los casos con modificaciones administrativas desde 1991 a 2001. En el Cuadro 3 se detallan los cuatro outliers para estas tasas de crecimiento intercensal. Una vez desestimado el caso de Morón para el período 1991-2001 donde la tasa es negativa de 51.9 %, debido a la cesión de tierras para la creación de distintos partidos, en la Figura 6, se puede ver la curva población de los 3 outliers destacados: Florencio Varela ,La Matanza y Ezeiza.

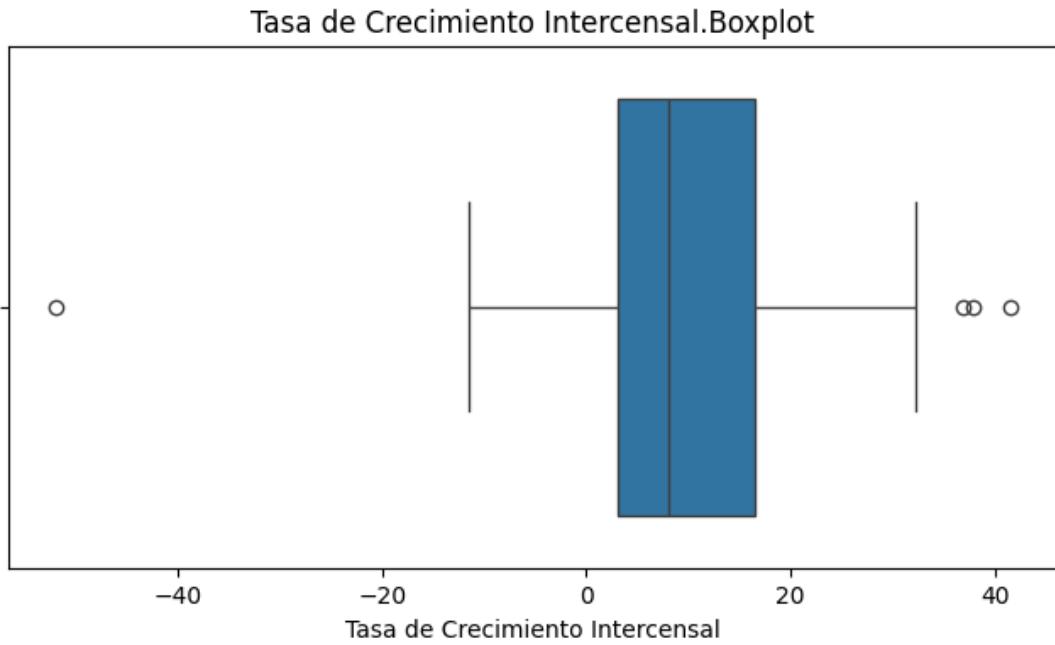


Figura 5: Box plot Tasas de crecimiento intercensal 1991-2022

id	nam	codDept	anio	pob	growthRatio
54	Morón	06568	2001	309380.0	-51.92
78	Florencio Varela	06274	2001	348970.0	36.85
13	La Matanza	06427	2010	1775816.0	41.47
29	Ezeiza	06270	2010	163722.0	37.80

Cuadro 11: Outliers para Tasas de crecimiento intercensal

9.4.3. Tasas de Crecimiento Intercensal: Coeficiente de Variación

Con el objetivo de profundizar el análisis de las tasas de crecimiento, se procedió a calcular la variación en la tasa de crecimiento por Departamento.

Sobre la media y la desviación estándar de cada subset de datos, se define el coeficiente de variación (CV) como:

$$CV_{(Departamento)} = \left(\frac{\text{Std.Dev}_{(Departamento)}}{\text{Mean}_{(Departamento)}} \right) \times 100 \quad (2)$$

El coeficiente de variación es una medida de la dispersión alrededor de la media de la población. En este caso, vemos curvas poblacionales con gran dispersión en su tasas de crecimiento para los censos del periodo analizado año 1991-2022.

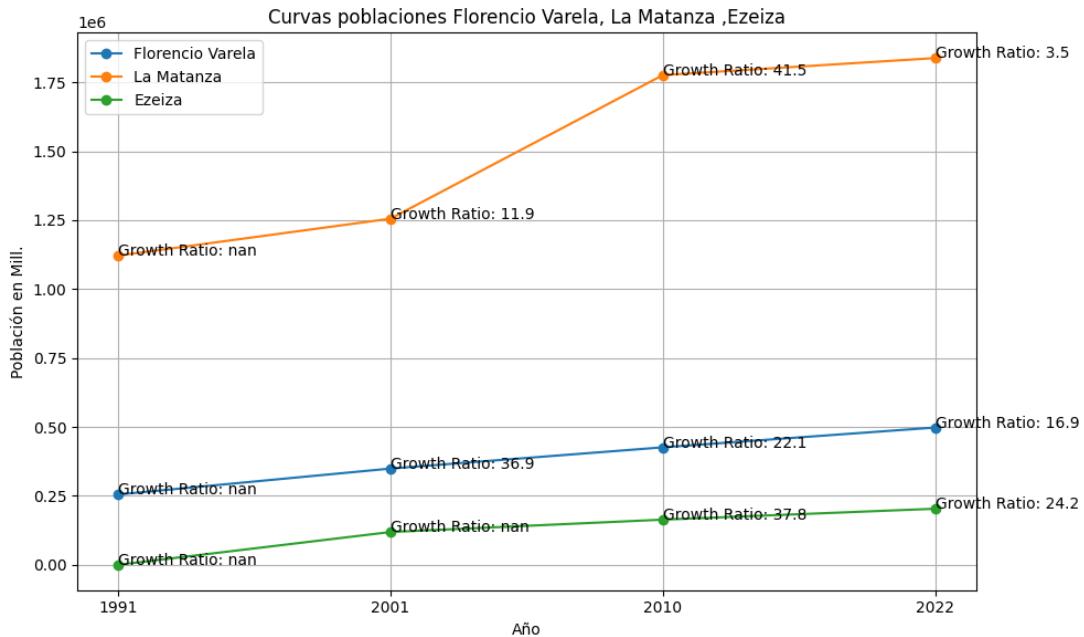


Figura 6: Curvas de Población por Departamento 1991-2022

En la Figura 7 se observa el resultado. Se destaca en particular los valores extremos tanto negativos como positivos, a saber : Vicente López, Morón, Tres de Febrero, Avellaneda, La Matanza, Esteban Echeverría y General San Martín.

Cabe recordar en este instancia que, los partidos de Esteban Echeverría, Florencio Varela y Morón sufrieron modificaciones administrativa en el período 1991-2001. En la Figuras 8 y 9 pueden observarse las curvas poblacionales de los mencionados departamentos destacados, difreenciadas para población media y alta, respectivamente.

En estos casos se prestará particular atención a las predicciones hechas por el INDEC, así como las metodologías que se implementen para la estimación de la curva poblacional, ya que es esperable un mayor error de predicción en estos departamentos cuyas curvas presentan singularidades respecto a los municipios inmediatamente aledaños.

9.5. Densidad de Población

En el dataset se incorpora la densidad de población en habitantes/km². Si analizamos la distribución de los valores de densidad para cada censo, puede observarse distribuciones homogéneas simétricas, con un incremento en la mediana de cada población censal desde 2001 a 2022. El análisis univariado de esta variable puede observarse en las Figuras 10 y 11.

Se presenta también la distribución geográfica de la densidad por departamento para los censos analizados. Si bien el distrito más poblado es La Matanza, debido su extensión, no es el departamento con mayor densidad poblacional en hab/km². Como puede verse en la Figura 12, los distritos más densamente poblados son Lanús y Vicente López.

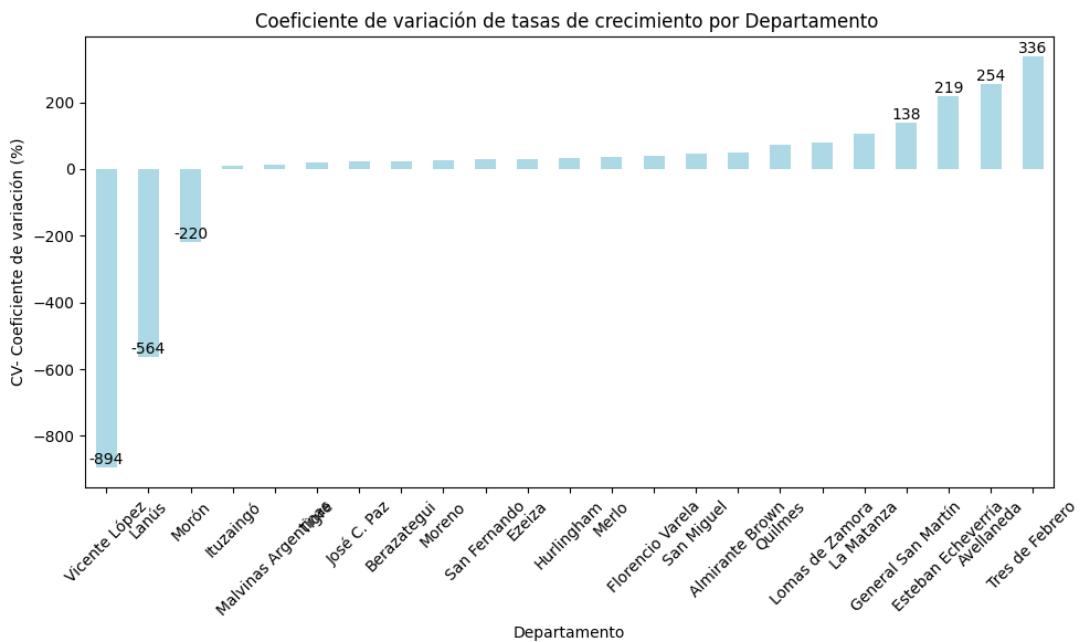


Figura 7: CV. Coeficiente de variación por departamento -1991-2022

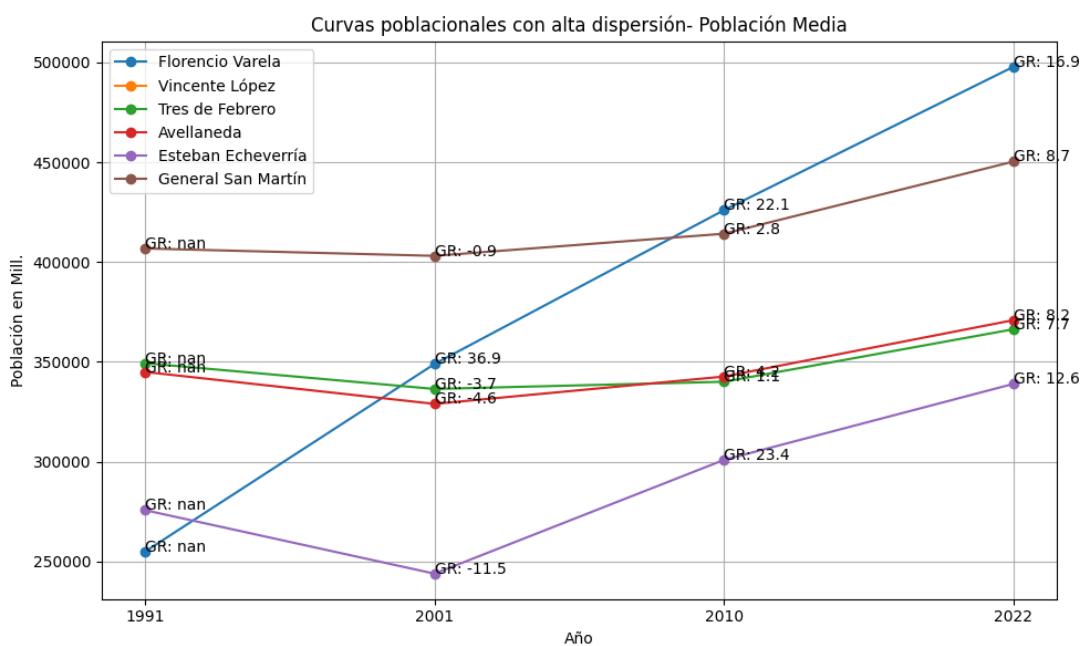


Figura 8: Curvas poblacionales de alta dispersión - Población Media

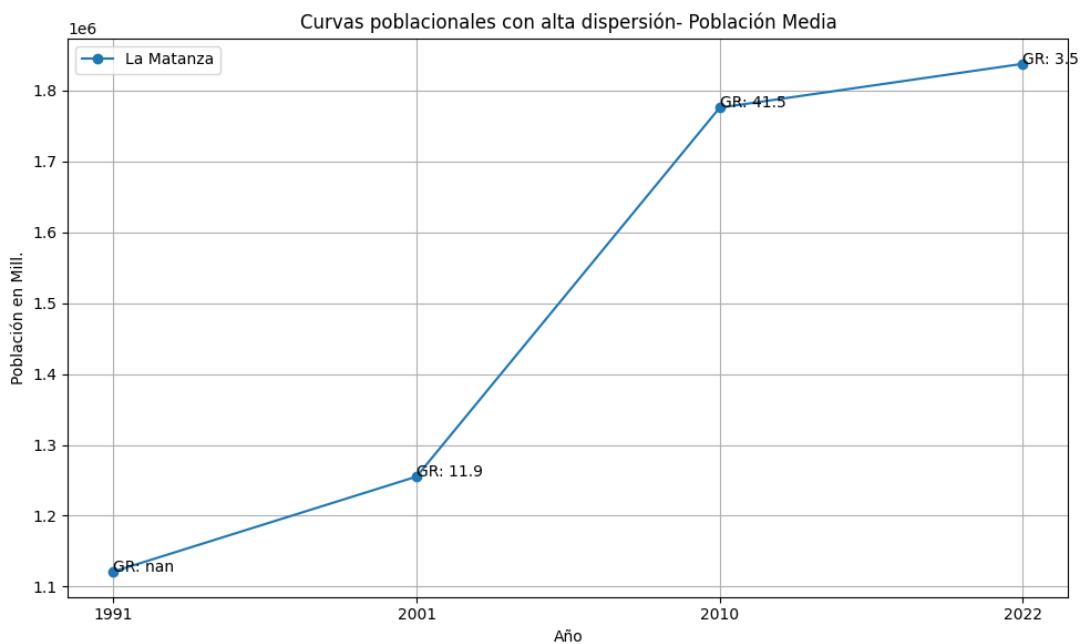


Figura 9: Curvas poblacionales de alta dispersión - Población Alta

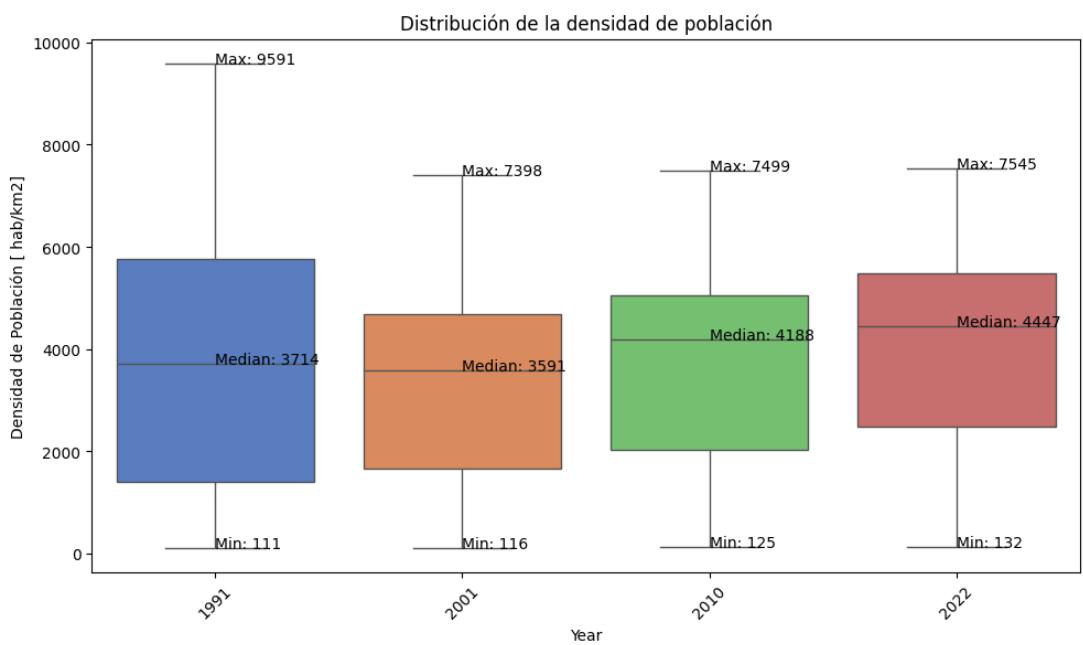


Figura 10: Densidad de Población. Análisis Univariado

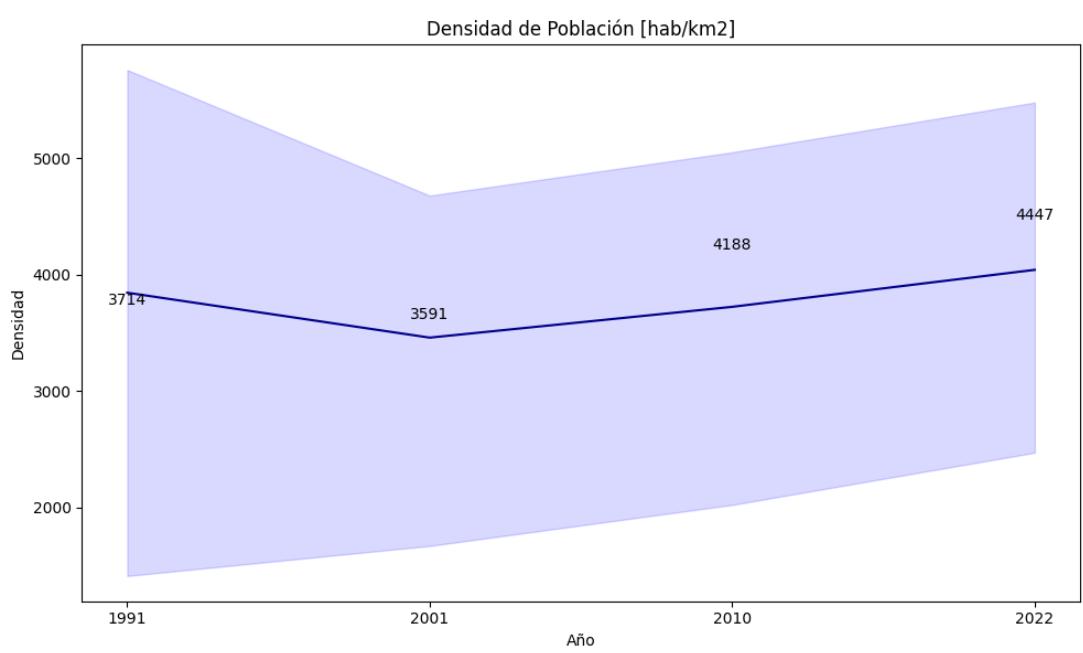


Figura 11: Densidad de Población. Análisis Univariado

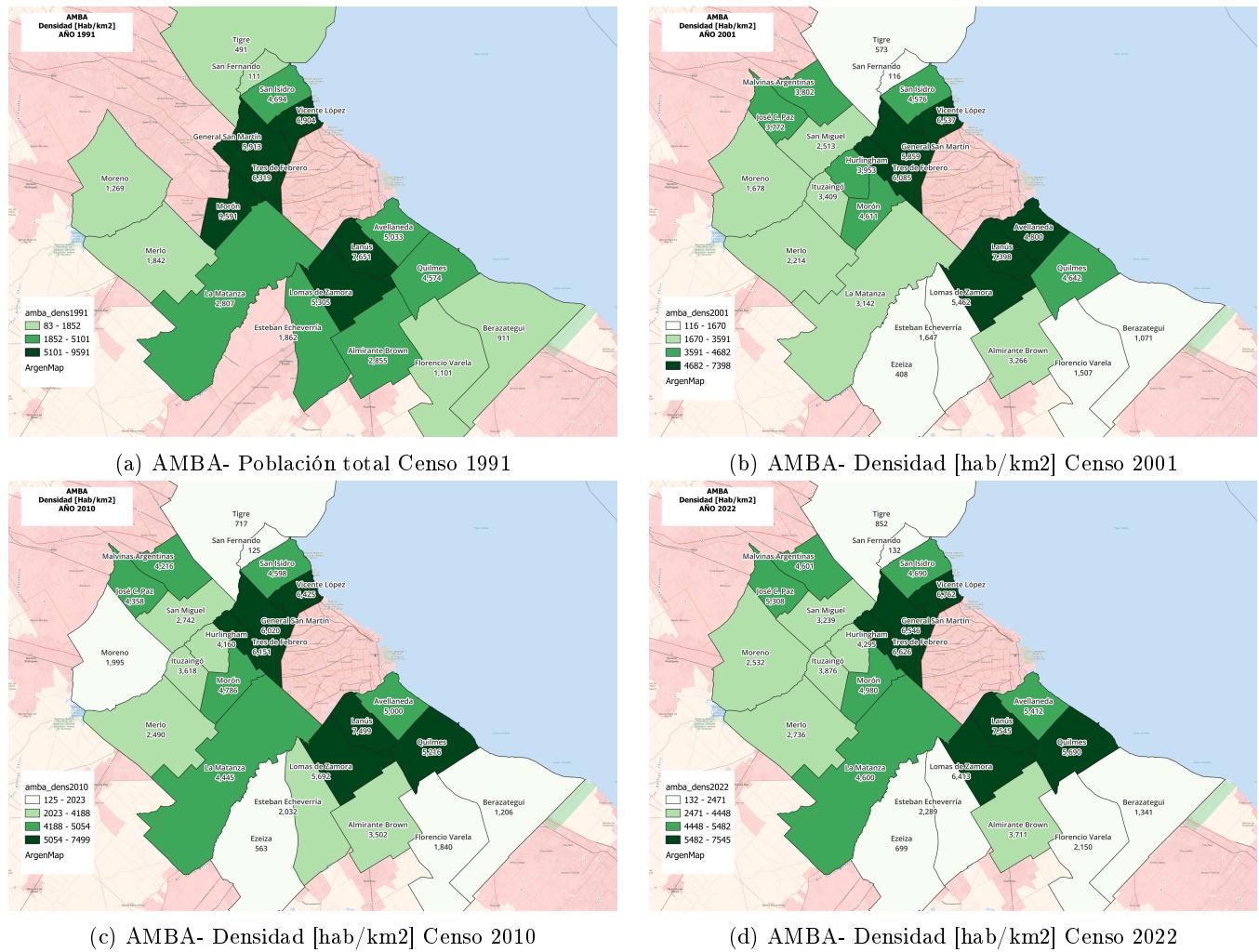


Figura 12: AMBA- Densidad [hab/km²] Censos 1991 2022

9.6. Tipo de Vivienda

A partir del censo 2001 se incorpora como atributo descriptivo en cada censo la composición o tipo de la vivienda, ya sea de tipo particular o colectiva. Se detalla en cada caso la cantidad de viviendas particulares ('vivpart') en determinado departamento, así como la cantidad de viviendas colectivas presentes. Se presenta entonces la posibilidad de realizar un análisis univariado del porcentaje de viviendas particulares respecto al total de viviendas para un determinado departamento y año censal. Para ello, se agrega al dataset una nueva variable definida como:

$$\text{VivPart \%} = \frac{\text{VivPart}}{\text{VivPart} + \text{VivColTot}} \times 100 \quad (3)$$

Surge de inmediato que la mayoría de la población vive en viviendas particulares (mayor a 99.9 %). Al analizar el comportamiento de este indicador se observa un leve incremento sostenido en el tiempo desde 1991 hasta 2022. Es decir, se observan cada vez más peso de las viviendas particulares. El análisis univariado de este indicador puede observarse en las Figuras 13 y 14.

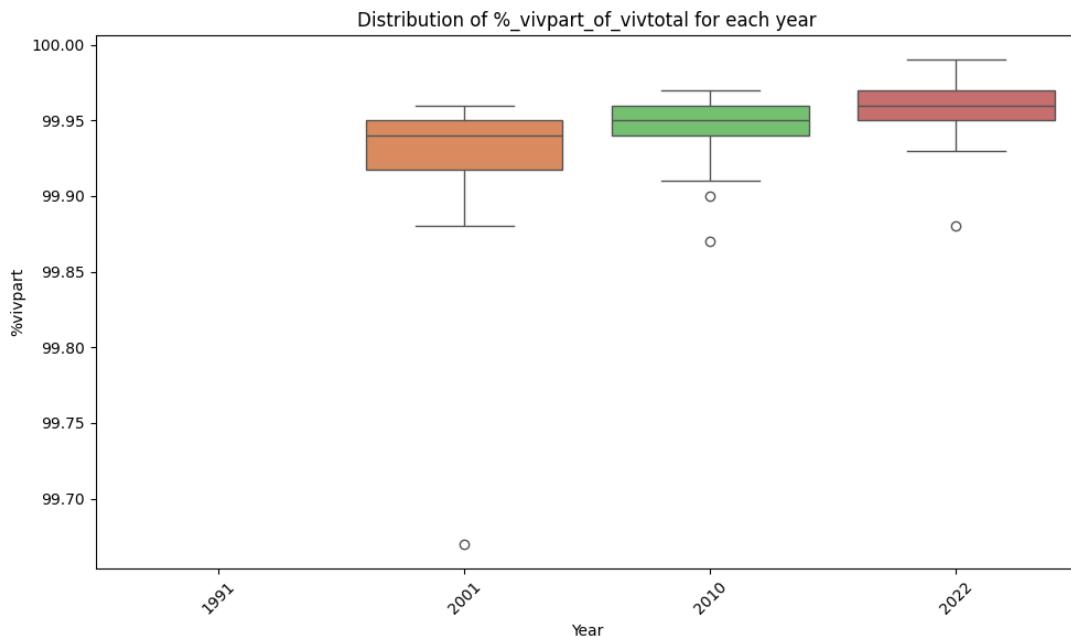


Figura 13: Composición de viviendas. Análisis Univariado

9.7. Índice de Masculinidad

Un indicador habitual de las muestras poblacionales es el índice de masculinidad, que resultada de dividir el número de hombres entre el número de mujeres de una unidad geográfica o administrativa, expresado como porcentaje:

$$\text{IndMasc} = \frac{\text{Varones}}{\text{Mujeres}} \times 100 \quad (4)$$

Al analizar el comportamiento de este índice a lo largo del tiempo, se observa un descenso sostenido del mismo desde 1991 hasta 2022. Particularmente el máximo de la muestra presenta un descenso de 5 puntos porcentuales para el año 2022, así como 2 puntos porcentuales en la mediana. Este resultado puede observarse en las Figuras 15 y 16.

10. Compartiva de los Modelos de Predicción

En base a los censos 1991, 2001 y 2010, sumado a las variables sintomáticas antes descriptas, se pretende ajustar y entrenar los modelos para luego predecir la población total de cada departamento del AMBA para el Año 2022. Al comparar con los resultados publicados por INDEC para el censo 2022 (INDEC, 2022)[10] se determinó la precisión de cada metodología aplicada.

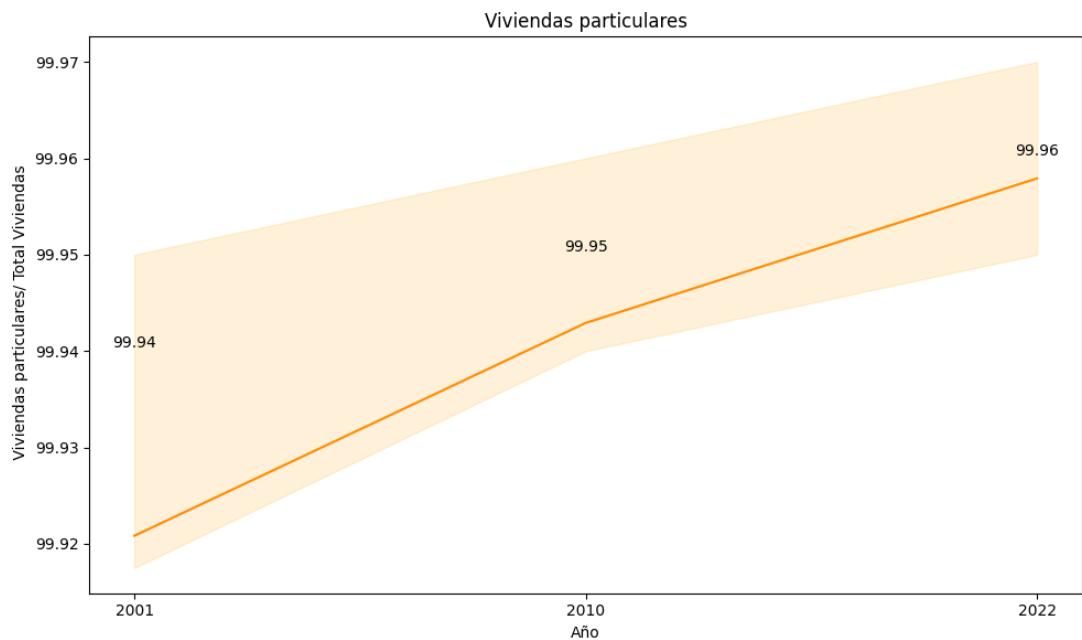


Figura 14: Composición de viviendas. Análisis Univariado

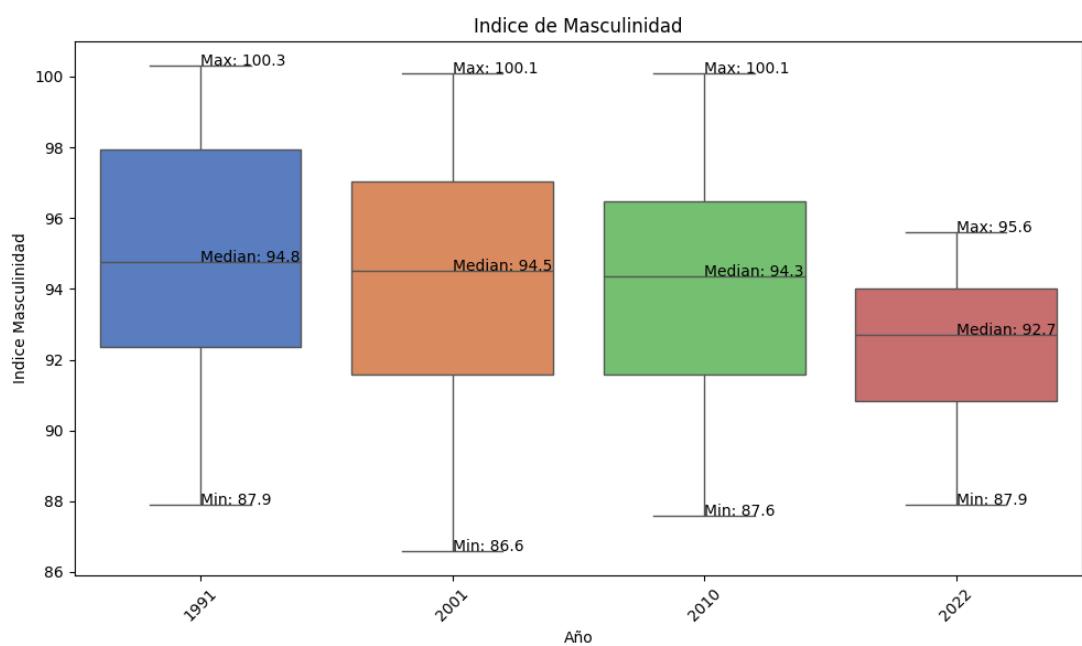


Figura 15: Índice de Masculinidad. Análisis Univariado

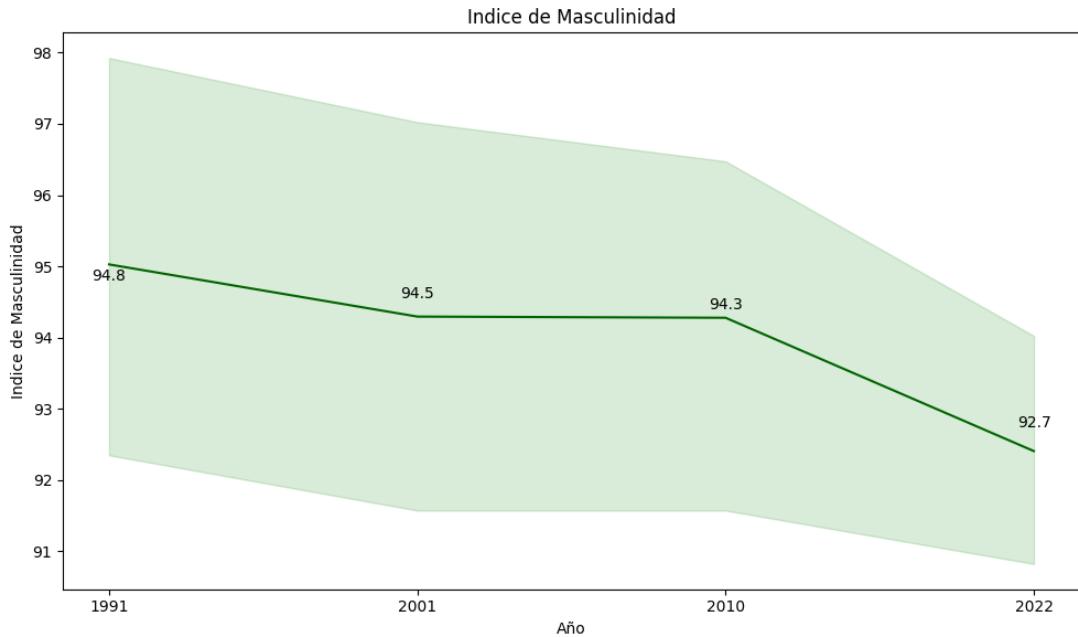


Figura 16: Índice de Masculinidad. Análisis Univariado

Muchas de estas herramientas tiene problemas con los valores nulos o faltantes. Es por esto que los departamentos que no existían en 1991, demandaron un tratamiento especial.

10.1. Variables Censales

Respecto a las variables censales podemos decir que presentan una correlación lineal directa muy importante y no aportan variabilidad. Por este motivo no son atributos significativos para los modelos propuestos. Para su análisis se utilizó la matriz de correlación, que puede verse en la Figura 17.

10.2. Errores típicos

Para determinar la precisión y realizar la comparativa entre los modelos aplicados se recurrió a los errores típicos: Error Cuadrático Medio (Mean Squared Error, MSE), Raíz del Error Cuadrático Medio (Root Mean Squared Error, RMSE) y Error Porcentual Absoluto Medio (Mean Absolute Percentage Error, MAPE).

MSE es una medida de la calidad de un estimador. Se calcula promediando el cuadrado de los errores (diferencias entre los valores predichos y los valores reales):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

RMSE es la raíz cuadrada del MSE, y proporciona una medida de la magnitud promedio del error en las mismas unidades que los valores predichos:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

MAPE es una medida de precisión que expresa el error como un porcentaje, y se calcula promediando el valor absoluto de los errores porcentuales:

$$MAPE = \frac{100 \%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

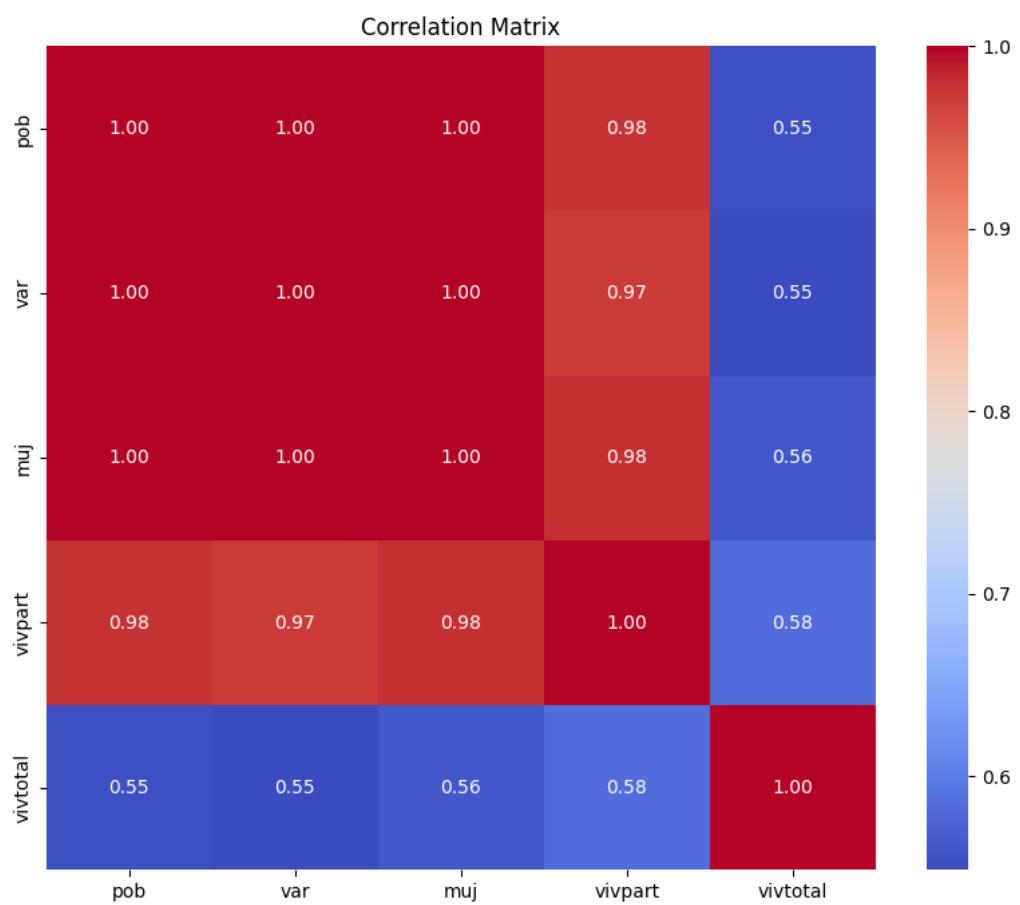


Figura 17: Matriz de correlación . Variables Censales

10.3. Dataset para Entrenamiento

Como se mencionó en las secciones anteriores, se trabajó con los censos de los años 1991 , 2001 y 2010, conjunto al que se le agregó el valor de las variables sintomáticas para la Jurisdicción provincial- Provincia de Buenos Aires para dichos años. Se tomó el enfoque planteado por (Álvarez et al., 2001)[1], entendiendo que el comportamiento de estas variables puede ayudar a explicar fenómenos a nivel departamental. Un fragmento del dataset utilizado como input de los modelos puede verse en el Cuadro 14.

10.4. Predicciones Población año 2022

En base al dataset de datos predefinido, se entrenaron los distintos modelos para luego predecir el valor de población total en cada departamento para el año 2022. Las predicciones realizadas corresponden a Regresión Lineal, CART, Random Forest , LightGMB, junto a las estimaciones realizadas por el INDEC en el año 2010 (INDEC, 2015)[12]. Para cada estimación se obtuvieron los errores típicos mencionados (MSE, RMSE, MAPE). En la sección 13.1 del presente trabajo se presentan los resultados para cada metodología a nivel de departamento. A modo de resumen en el siguiente cuadro 12 se puede observar el valor del censo 2022, las proyecciones de cada metodología y el error medio absoluto porcentual (MAPE) por departamento.

Departamento	Censo2022	PredLR	PredRT	PredRF	PredLGB	PredINDEC
Almirante Brown	585,852	602,696 (2.9 %)	552,902 (5.6 %)	521,336 (11.0 %)	506,385 (13.6 %)	605,271 (3.3 %)
Avellaneda	370,939	360,939 (2.7 %)	342,677 (7.6 %)	337,916 (8.9 %)	338,883 (8.6 %)	358,512 (3.4 %)
Berazategui	360,582	372,685 (3.4 %)	287,913 (20.2 %)	295,665 (18.0 %)	285,695 (20.8 %)	372,889 (3.4 %)
Esteban Echeverría	339,030	376,939 (11.2 %)	243,974 (28.0 %)	278,778 (17.8 %)	273,575 (19.3 %)	383,538 (13.1 %)
Ezeiza	203,283	223,608 (10.0 %)	118,807 (41.6 %)	nan	nan	229,276 (12.8 %)
Florencio Varela	497,818	528,718 (6.2 %)	426,005 (14.4 %)	367,660 (26.1 %)	343,324 (31.0 %)	533,446 (7.2 %)
General San Martín	450,335	428,981 (4.7 %)	403,107 (10.5 %)	409,244 (9.1 %)	408,037 (9.4 %)	426,556 (5.3 %)
Hurlingham	187,122	193,235 (3.3 %)	181,241 (3.1 %)	nan	nan	195,596 (4.5 %)
Ituzaingó	179,788	180,761 (0.5 %)	167,824 (6.7 %)	nan	nan	182,993 (1.8 %)
José C. Paz	323,918	313,678 (3.2 %)	265,981 (17.9 %)	nan	nan	314,878 (2.8 %)
La Matanza	1,837,774	2,469,853 (34.4 %)	1,775,816 (3.4 %)	1,437,887 (21.8 %)	1,384,134 (24.7 %)	2,374,149 (29.2 %)
Lanús	462,051	467,504 (1.2 %)	453,082 (1.9 %)	459,486 (0.6 %)	460,302 (0.4 %)	462,693 (0.1 %)
Lomas de Zamora	694,330	649,524 (6.5 %)	591,345 (14.8 %)	599,412 (13.7 %)	593,985 (14.4 %)	652,937 (6.0 %)
Malvinas Argentinas	351,788	364,620 (3.6 %)	290,691 (17.4 %)	nan	nan	366,479 (4.2 %)
Merlo	580,806	606,506 (4.4 %)	528,494 (9.0 %)	475,189 (18.2 %)	463,112 (20.3 %)	620,307 (6.8 %)
Moreno	574,374	548,507 (4.5 %)	452,505 (21.2 %)	389,610 (32.2 %)	373,574 (35.0 %)	558,068 (2.8 %)
Morón	334,178	336,747 (0.8 %)	321,109 (3.9 %)	381,258 (14.1 %)	424,681 (27.1 %)	317,584 (5.0 %)
Quilmes	636,026	668,483 (5.1 %)	582,943 (8.3 %)	544,713 (14.4 %)	537,655 (15.5 %)	679,375 (6.8 %)
San Fernando	172,524	179,385 (4.0 %)	151,131 (12.4 %)	155,854 (9.7 %)	153,045 (11.3 %)	176,795 (2.5 %)
San Isidro	298,777	294,708 (1.4 %)	292,878 (2.0 %)	293,647 (1.7 %)	294,469 (1.4 %)	291,704 (2.4 %)
San Miguel	326,215	306,995 (5.9 %)	276,190 (15.3 %)	nan	nan	308,784 (5.3 %)
Tigre	447,785	476,591 (6.4 %)	301,223 (32.7 %)	327,135 (26.9 %)	311,842 (30.4 %)	nan
Tres de Febrero	366,377	344,876 (5.9 %)	336,467 (8.2 %)	341,817 (6.7 %)	341,971 (6.7 %)	344,172 (6.1 %)
Vicente López	283,510	263,204 (7.2 %)	269,420 (5.0 %)	277,881 (2.0 %)	277,669 (2.1 %)	266,880 (5.9 %)

Cuadro 12: Predicciones de población según metodología (MAPE %) por departamento.

LR : Regresión Lineal - RT: Árboles de Regresión (CART) - RF: Random Forest - LGB: LightGBM

Para la comparativa entre los modelos se evaluó el desempeño de cada modelo sobre los 24 departamentos del AMBA. En el cuadro 13 se pueden observer el valor medio de los errores típicos para cada metodología, se considera el MAPE como principal indicador de desempeño.

Method	MSE	RMSE	MAPE
Linear Regression	1.7e+10	43720.0	5.8
Regression Trees	4.1e+09	52144.0	13.0
Random Forest	1.5e+10	82930.0	14.0
LightGBM	1.9e+10	94541.0	16.2
INDEC	6.5e+09	80792.0	6.1

Cuadro 13: Errores típicos por metodología. (Promedio)

Departamento	cod _{depto}	ano	pob	var	muj	vivpart	vivtotal	sup	ind _{masc}	dens _{pob}	TMI	TGF	TBN	TBM	TCV	Mat1ria
Almirante Brown	6028	1991	450698.0	222042.0	228656.0	nan	nan	157.87	97.1	2854.87	24.2	2.6	18.4	7.9	10.5	1752994.0
Almirante Brown	6028	2001	515556.0	252454.0	263102.0	143543.0	88.0	157.87	96.0	3265.7	15.0	2.3	16.9	8.2	8.7	1658221.0
Almirante Brown	6028	2010	552902.0	270247.0	282655.0	156218.0	78.0	157.87	95.6	3502.26	12.0	2.5	18.9	8.4	10.5	1667278.0
Avellaneda	6035	1991	344991.0	164243.0	180748.0	nan	nan	68.54	90.9	5033.43	24.2	2.6	18.4	7.9	10.5	1752994.0
Avellaneda	6035	2001	328980.0	155450.0	173530.0	117200.0	59.0	68.54	89.6	4799.82	15.0	2.3	16.9	8.2	8.7	1658221.0
Avellaneda	6035	2010	342677.0	162264.0	180413.0	121307.0	68.0	68.54	89.9	4999.66	12.0	2.5	18.9	8.4	10.5	1667278.0
Berazategui	6091	1991	244929.0	120870.0	124059.0	nan	nan	268.91	97.4	910.82	24.2	2.6	18.4	7.9	10.5	1752994.0
Berazategui	6091	2001	287913.0	141163.0	146750.0	81511.0	38.0	268.91	96.2	1070.67	15.0	2.3	16.9	8.2	8.7	1658221.0
Berazategui	6091	2010	324244.0	158608.0	165636.0	96029.0	37.0	268.91	95.8	1205.77	12.0	2.5	18.9	8.4	10.5	1667278.0
Esteban Echeverría	6260	1991	275793.0	136784.0	139009.0	nan	nan	148.12	98.4	1861.96	24.2	2.6	18.4	7.9	10.5	1752994.0
Esteban Echeverría	6260	2001	243974.0	120110.0	123864.0	70535.0	26.0	148.12	97.0	1647.14	15.0	2.3	16.9	8.2	8.7	1658221.0
Esteban Echeverría	6260	2010	300959.0	147980.0	152979.0	88164.0	26.0	148.12	96.7	2031.86	12.0	2.5	18.9	8.4	10.5	1667278.0

Cuadro 14: Subset de datos input. Primeras 15 filas. Censos 1991, 2001, 2010 enriquecidos con las variables sintomáticas

Los resultados obtenidos implican que las metodologías tradicionales aproximan mejor este tipo de datos poblacionales dispersos ("sparse"). Tanto la regresión Lineal, como las proyecciones realizadas por el INDEC presentan mejor precisión- menor error- y una menor desviación estándar. Este comportamiento se puede observar al graficar el Error Porcentual Absoluto Medio (MAPE) para cada metodología para las 24 proyecciones realizadas, figura 18.

Las algoritmos de data mining presentan dificultades debido a las características particulares de la información censal, así como el hecho de estar trabajando sólo con la información de tres Censos Nacionales. Sumado a ésto, la granularidad analizada en este caso (nivel departamental) hace difícil enriquecer el dataset con variables sintomáticas a este nivel y se debe recurrir a información agregada a nivel Provincial. Estos modelos presentan valores de error notablemente mayores y una amplia dispersión de resultados.

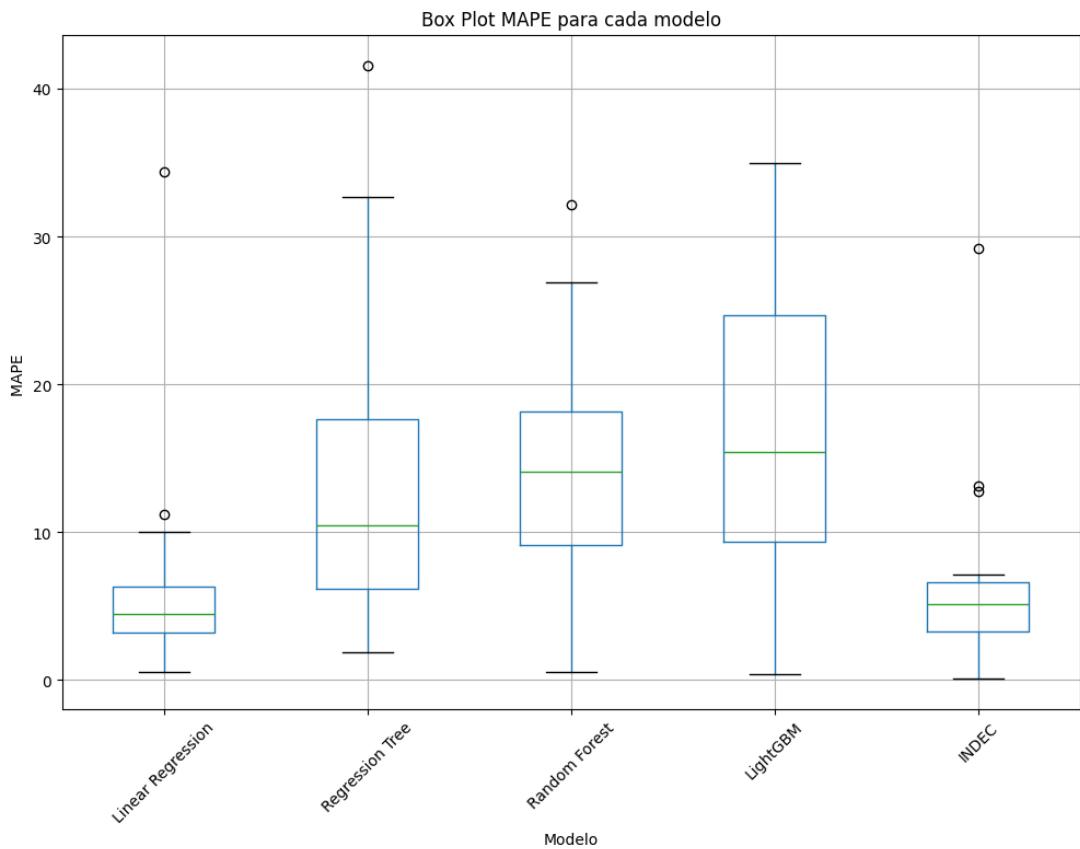


Figura 18: Box plot del Error Porcentual Absoluto Medio para cada modelo, sobre la predicción población año 2022. Todos los departamentos.

En cada modelo los outliers del boxplot son aquellos departamentos donde dicho modelo tuvo una muy mala performance relativa. Si se analizan los modelos con mejor precisión, en Regresión Lineal tanto Esteban Echeverría (MAPE:11.2 %) como la Matanza (MAPE:41.6 %) son outliers, donde la predicción estuvo muy alejada del valor poblacional del censo 2022. Para el caso de las proyecciones según INDEC, aparecen también Esteban Echeverría (MAPE:13.1 %), Ezeiza (MAPE:12.8 %) y la Matanza (MAPE:29.2 %) como outliers. El caso de Esteban Echeverría tiene que ver con la cesión de territorio(cambio administrativo) desde el año 1991 al 2001. Esto implicó que su población descienda entre 1991 y 2001 un -11,5 % mientras que entre 2001 y 2010 creció un 23.3 %. Por ende Esteban Echeverría presenta una curva poblacional con un comportamiento particular, no producida por el fenómeno demográfico. La Matanza no ha sufrido modificaciones territoriales en este periodo y por tanto presenta una curva poblacional con singularidades, con un comportamiento difenciado del resto de los departamentos.

Al analizar la curva poblacional de La Matanza (ver Figura 19), se observan cambios significativos en el ratio de crecimiento intercensal, a lo largo de los distintos censos. Si se observa el año 2001 , el ratio intercensal es de 11.9 %, lo que se ubica un 75 % por encima del crecimiento promedio (6.8 %) para los municipios del AMBA. En el

año 2010 se observa un ratio de 41.5 %, que representa un 250 % por encima del crecimiento promedio (11.8 %). Esto representa un salto importante, siendo el municipio con mayor crecimiento en este periodo. Mientras que para el año 2022 se observa un ratio mucho menor, del orden de 3.5 %, un 70 % por debajo del crecimiento promedio (10.6 %).

Esto implica que La Matanza es uno de los departamentos con mayor desviación estandar y coeficiente de variación en tasas de crecimiento, tal como se describió en apartados anteriores. El crecimiento promedio de los departamentos del AMBA para el periodo 1991 a 2010 es de 9,9 % , mientras que La Matanza en el mismo periodo creció un 26,7 % siendo uno de los municipios con mayor crecimiento poblacional. Obviamente el promedio cae al incorporar el año 2022, dando lugar a que La Matanza en toda la serie se acerque al crecimiento promedio de otros municipios. Seguramente estas singularidades hayan dificultado la predicción del valor poblacional para el departamento con todas la metodologías aplicadas, incluyendo aquellas que pudieron resultar más efectivas en la mayoría de los casos.

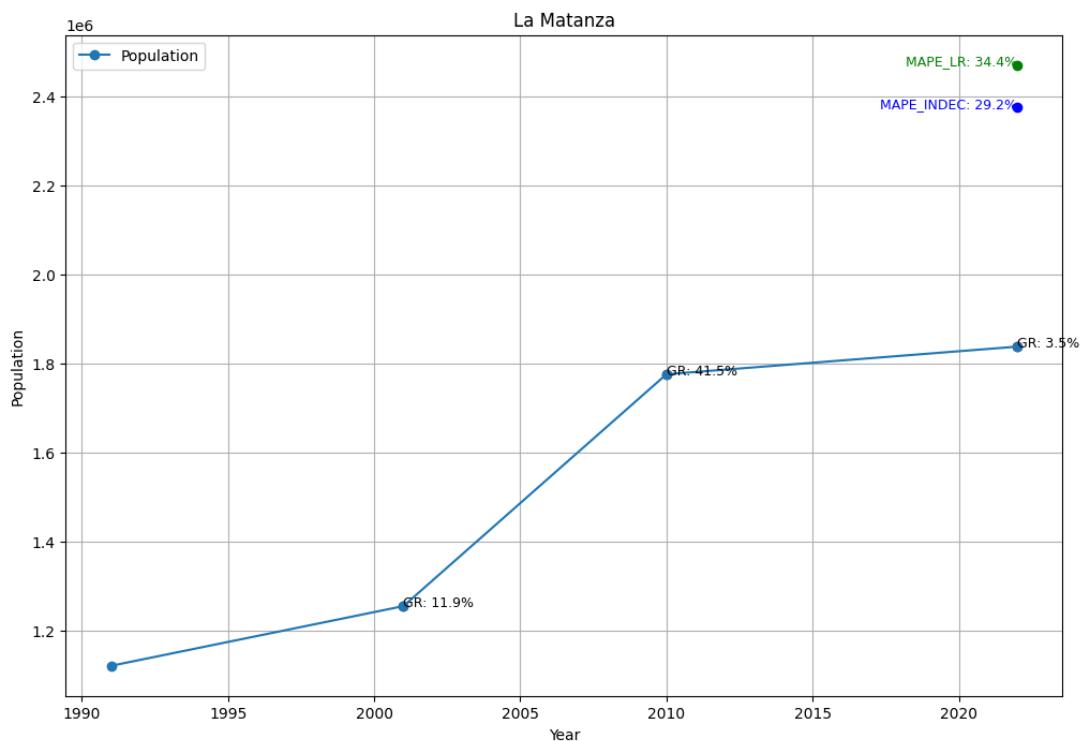


Figura 19: Curva Poblacional La Matanza. GR:Ratio de crecimiento intercensal.

11. Conclusiones y Trabajo Futuro

La información estadística que brindan las proyecciones de población representa un insumo vital en la planificación de políticas públicas de corto, mediano y largo plazo. Contar con esta información le permite al Estado determinar los recursos presupuestarios necesarios para satisfacer la demanda potencial de bienes y servicios en distintas áreas. De hecho, en la provincia de Buenos Aires, ciertos aspectos del presupuesto son asignados en base a la población de cada municipio. Por este motivo, aunque en muchos casos dichas proyecciones se encuentran a nivel País o Provincia, es necesario contar con la información con un mayor nivel de desagregación (Municipios).

La elaboración de proyecciones de población es una tarea compleja que debe ser realizada a través de un análisis exhaustivo que permita considerar los censos anteriores como también registros vitales y estimaciones de migración. Aunque, en general, se ha utilizado el Método de las Componentes para elaborar dichas proyecciones, para el caso de las desagregación a nivel municipio, no resulta simple su aplicación, especialmente por la inestabilidad de la migración interna. En Argentina, el Instituto Nacional de Estadísticas y Censos (INDEC) provee proyecciones de población a nivel nacional, desagregados a nivel departamental. Dichas estimaciones se apoyan en el crecimiento intercensal, que comprende la evolución en conjunto de las tres variables básicas del análisis demográfico: la fecundidad, la mortalidad

y la migración del período entre censos.

El objetivo del presente trabajo consistió en demostrar que la tasa de crecimiento y curva poblacional en el período 1991-2022 de un municipio particular de la Provincia de Buenos Aires, la Matanza, presenta una singularidad en su curva de crecimiento poblacional, tanto en número de habitantes como en tasas intercensales, respecto a los municipios aledaños.

Para llevar adelante este objetivo, se tomó la información censal del INDEC desde 1991 hasta 2022 para distintos niveles de granularidad, ya que en algunos casos no se presentan todas las variables. Sobre la misma, se realizó un trabajo de limpieza y pre-procesamiento importante, con enriquecimiento de datos geográficos obtenidos desde el Instituto Geográfico Nacional. Para resolver el problema de los cambios de departamento en el tiempo, que ceden parte de su terreno o se fusionan, se utilizó el enfoque Slowly Changing Dimension Tipo 3, conservando en el registro el valor anterior, de qué otro departamento proviene y la superficie correspondiente asociada. Finalmente, sobre el dataset obtenido, se realizó un análisis exploratorio de cada variable., para determinar outliers.

En la fase experimental, tomando como base los tres censos anteriores (1991-2001-2010), se realizaron proyecciones para la población censo 2022 con metodologías tradicionales modernas, y con técnicas data mining, en este caso, CART, Random Forest y LightGBM.

Sobre los resultados obtenidos a través de los diferentes métodos, podemos concluir que, para la predicción de valores de población a nivel departamental, se destacan las metodologías tradicionales, como regresión lineal o bien metodologías que se apoyan en el crecimiento intercensal que comprende la evolución en conjunto de las tres variables básicas del análisis demográfico: la fecundidad, la mortalidad y la migración. Sin embargo, debe considerarse que la elaboración de proyecciones de población de áreas menores resulta compleja, debido a la imposibilidad de aplicar un método estrictamente demográfico, que requiere la estimación y proyección independiente de cada una de las variables del crecimiento de la población. A este nivel de desagregación, los hechos vitales presentan fluctuaciones anuales más acentuadas cuanto menor es el número de población y, consecuentemente, de nacimientos y defunciones, que pueden afectar las estimaciones de la fecundidad y la mortalidad. Asimismo, se hace casi imposible la determinación de la migración interna, que suele ser un elemento muy importante del crecimiento de dichas áreas. Esto se debe a la dificultad de obtener estimaciones de saldos migratorios consistentes a nivel departamental y a la complejidad para su proyección futura, por tratarse de un factor estrechamente asociado a las condiciones económicas y sociales del momento.

Por otro lado, puede concluirse que las técnicas de data mining no han producido, en este caso, resultados precisos. Presentan dificultades debido a las características particulares de la información censal, el hecho de estar trabajando sólo con la información de tres Censos Nacionales. Esto, sumado al hecho de la granularidad analizada en este caso (nivel departamental) hace difícil enriquecer el dataset con variables sintomáticas a este nivel y se debe recurrir a información agregada a nivel Provincial. Estos modelos presentan valores de error notablemente mayores y una amplia dispersión de resultados.

Ciertamente si bien en la generalidad de los casos es posible estimar un valor población futuro con un grado de confianza aceptable, algunos departamentos presentan comportamiento singulares en sus curvas poblacionales que dificultan su predicción. Las razones de estas singularidades escapan al alcance del presente trabajo, pero bien pueden deberse a causales del complejo fenómeno demográfico así como también posibles errores en el relevamiento censal. En esta línea, y como trabajo a futuro, se pueden incorporar datos desde otras fuentes, como tasa de empleo y condiciones de vida en cada municipio/provincia, para incorporarlo a la metodología, en pos de mejorar las proyecciones censales.

12. Bibliografía

Referencias

- [1] Gustavo. Álvarez et al. *Estimación de población en áreas menores mediante variables sintomáticas : una aplicación para los departamentos de la República Argentina (1991 y 1996)*. Naciones Unidas, CEPAL/ECLAC, 2001, pág. 36. ISBN: 9213217749.

- [2] UN-CEPAL. *CENSO 2001 Procesamiento del Censo Nacional de Población, Hogares y Viviendas 2001*. 2023. URL: <https://redatam.indec.gob.ar/argbin/RpWebEngine.exe/PortalAction?BASE=CPV2001ARG> (visitado 30-10-2023).
- [3] UN- CEPAL. *CENSO 2010 Procesamiento del Censo Nacional de Población, Hogares y Viviendas 2010 - Cuestionario Ampliado*. 2023. URL: <https://redatam.indec.gob.ar/binarg/RpWebEngine.exe/Portal?BASE=CPV2010A&lang=ESP> (visitado 30-10-2023).
- [4] Manan Chawda, Rutuja Rane y Skrinanth Giri. «Demographic Progress Analysis of Census Data Using Data Mining». En: 2018, págs. 1894-1897. ISBN: 978-1-5386-1974-2.
- [5] Ministerio de Educación. <https://data.educacion.gob.ar/reporte-matricula.php>. 2022.
- [6] Arindam Gupta, Sabyasachi Bhattacharya y Asis Kumar Chattyopadhyay. «Exploring New Models for Population Prediction in Detecting Demographic Phase Change for Sparse Census Data». En: (2012).
- [7] Nazrul Hoque. «Evaluation of small area population estimates produced by Housing Unit, Ratio-correlation, and Component Method II compared to 2000 Census counts». En: *Canadian Studies in Population* 39, No. 1–2 (2012), págs. 91-108.
- [8] IGM. *Dataset GEO IGM–Instituto Geográfico Nacional, Argentina Capas SIG - Departamento*. 2024. URL: <https://www.ign.gob.ar/NuestrasActividades/InformacionGeoespacial/CapasSIG> (visitado 30-03-2024).
- [9] INDEC. *CENSO 2001 Censo Nacional de Población, Hogares y Viviendas del año 2001*. 2023. URL: https://www.indec.gob.ar/micro_sitios/webcenso/index.asp (visitado 30-11-2023).
- [10] INDEC. *CENSO 2022 Resultados Previsionales*. 2022. URL: https://www.censo.gob.ar/index.php/datos_provisionales (visitado 30-10-2023).
- [11] INDEC. *DEMOGRAFICO Programa de Análisis Demográfico de la Dirección de Estadísticas Poblacionales*. 2022. URL: <https://www.indec.gob.ar/indec/web/Institucional-Indec-IndicadoresDemograficos> (visitado 10-02-2024).
- [12] INDEC. *Estimaciones de población por sexo, departamento y año calendario 2010-2025*. 1 ed. 2015.
- [13] INDEC. *Proyecciones provinciales de población por sexo y grupo de edad 2010-2014*. - Instituto Nacional de Estadística y Censos - INDEC, 2013. E-Book. Vol. 1a ed. 2013. ISBN: 978-950-896-433-5.
- [14] Ralph Kimball. «The Data Warehouse Toolkit: Practical Techniques For Building Dimensional Data Warehouses-Bom». En: (1996).

13. ANEXO

13.1. Resultados Metodologías. Prediciones poblacionales año 2022

Se presentan a continuación los cuadros con resultados de la aplicación de las metodologías analizadas, así como los errores típicos definidos al comparar la estimación poblacional para el año 2022 con el resultado del Censo 2022 (INDEC, 2022)[10].

Departamento	Censo 2022	Pred LR	MAPE LR	MSE LR	RMSE LR
Almirante Brown	585,852	602,696	2.88	2.8e+08	16,845
Avellaneda	370,939	360,939	2.70	1.0e+08	9,999.30
Berazategui	360,582	372,685	3.36	1.5e+08	12,103
Esteban Echeverría	339,030	376,939	11.18	1.4e+09	37,909
Ezeiza	203,283	223,608	10.00	4.1e+08	20,326
Florencio Varela	497,818	528,718	6.21	9.6e+08	30,900
General San Martín	450,335	428,981	4.74	4.6e+08	21,354
Hurlingham	187,122	193,235	3.27	3.7e+07	6,113.70
Ituzaingó	179,788	180,761	0.54	947,000	973.30
José C. Paz	323,918	313,678	3.16	1.0e+08	10,240
La Matanza	1,837,774	2,469,853	34.39	4.0e+11	632,079
Lanús	462,051	467,504	1.18	3.0e+07	5,453.30
Lomas de Zamora	694,330	649,524	6.45	2.0e+09	44,806
Malvinas Argentinas	351,788	364,620	3.65	1.6e+08	12,832
Merlo	580,806	606,506	4.42	6.6e+08	25,700
Moreno	574,374	548,507	4.50	6.7e+08	25,866
Morón	334,178	336,747	0.77	6,600,000	2,569.70
Quilmes	636,026	668,483	5.10	1.0e+09	32,457
San Fernando	172,524	179,385	3.98	4.7e+07	6,861.30
San Isidro	298,777	294,708	1.36	1.7e+07	4,068.30
San Miguel	326,215	306,995	5.89	3.7e+08	19,220
Tigre	447,785	476,591	6.43	8.3e+08	28,807
Tres de Febrero	366,377	344,876	5.87	4.6e+08	21,501
Vicente López	283,510	263,204	7.16	4.1e+08	20,306

Cuadro 15: Regresión Lineal. Predicciones valor poblacional por departamentos y errores típicos

Departamento	Censo 2022	Pred RT	MAPE RT	MSE RT	RMSE RT
Almirante Brown	585,852	552,902	5.60	1.1e+09	32,950
Avellaneda	370,939	342,677	7.60	8.0e+08	28,262
Berazategui	360,582	287,913	20.20	5.3e+09	72,669
Esteban Echeverría	339,030	243,974	28.00	9.0e+09	95,056
Ezeiza	203,283	118,807	41.60	7.1e+09	84,476
Florencio Varela	497,818	426,005	14.40	5.2e+09	71,813
General San Martín	450,335	403,107	10.50	2.2e+09	47,228
Hurlingham	187,122	181,241	3.10	3.5e+07	5,881.00
Ituzaingó	179,788	167,824	6.70	1.4e+08	11,964
José C. Paz	323,918	265,981	17.90	3.4e+09	57,937
La Matanza	1,837,774	1,775,816	3.40	3.8e+09	61,958
Lanús	462,051	453,082	1.90	8.0e+07	8,969.00
Lomas de Zamora	694,330	591,345	14.80	1.1e+10	102,985
Malvinas Argentinas	351,788	290,691	17.40	3.7e+09	61,097
Merlo	580,806	528,494	9.00	2.7e+09	52,312
Moreno	574,374	452,505	21.20	1.5e+10	121,869
Morón	334,178	321,109	3.90	1.7e+08	13,069
Quilmes	636,026	582,943	8.30	2.8e+09	53,083
San Fernando	172,524	151,131	12.40	4.6e+08	21,393
San Isidro	298,777	292,878	2.00	3.5e+07	5,899.00
San Miguel	326,215	276,190	15.30	2.5e+09	50,025
Tigre	447,785	301,223	32.70	2.1e+10	146,562
Tres de Febrero	366,377	336,467	8.20	8.9e+08	29,910
Vicente López	283,510	269,420	5.00	2.0e+08	14,090

Cuadro 16: CART. Árboles de Regresión. Predicciones valor poblacional por departamentos y errores típicos

Departamento	Censo 2022	Pred RF	MAPE RF	MSE RF	RMSE RF
Almirante Brown	585,852	521,336	11.01	4.2e+09	64,516
Avellaneda	370,939	337,916	8.90	1.1e+09	33,023
Berazategui	360,582	295,665	18.00	4.2e+09	64,917
Esteban Echeverría	339,030	278,778	17.77	3.6e+09	60,252
Ezeiza	203,283	nan	nan	nan	nan
Florencio Varela	497,818	367,660	26.15	1.7e+10	130,158
General San Martín	450,335	409,244	9.12	1.7e+09	41,091
Hurlingham	187,122	nan	nan	nan	nan
Ituzaingó	179,788	nan	nan	nan	nan
José C. Paz	323,918	nan	nan	nan	nan
La Matanza	1,837,774	1,437,887	21.76	1.6e+11	399,887
Lanús	462,051	459,486	0.56	6,579,430	2,565.04
Lomas de Zamora	694,330	599,412	13.67	9.0e+09	94,918
Malvinas Argentinas	351,788	nan	nan	nan	nan
Merlo	580,806	475,189	18.18	1.1e+10	105,617
Moreno	574,374	389,610	32.17	3.4e+10	184,764
Morón	334,178	381,258	14.09	2.2e+09	47,080
Quilmes	636,026	544,713	14.36	8.3e+09	91,313
San Fernando	172,524	155,854	9.66	2.8e+08	16,670
San Isidro	298,777	293,647	1.72	2.6e+07	5,130.29
San Miguel	326,215	nan	nan	nan	nan
Tigre	447,785	327,135	26.94	1.5e+10	120,650
Tres de Febrero	366,377	341,817	6.70	6.0e+08	24,560
Vicente López	283,510	277,881	1.99	3.2e+07	5,629.40

Cuadro 17: Random Forest. Predicciones valor poblacional por departamentos y errores típicos

Departamento	Censo 2022	Pred LGB	MAPE LGB	MSE LGB	RMSE LGB
Almirante Brown	585,852	506,385	13.56	6.3e+09	79,467
Avellaneda	370,939	338,883	8.64	1.0e+09	32,056
Berazategui	360,582	285,695	20.77	5.6e+09	74,887
Esteban Echeverría	339,030	273,575	19.31	4.3e+09	65,455
Ezeiza	203,283	nan	nan	nan	nan
Florencio Varela	497,818	343,324	31.03	2.4e+10	154,494
General San Martín	450,335	408,037	9.39	1.8e+09	42,298
Hurlingham	187,122	nan	nan	nan	nan
Ituzaingó	179,788	nan	nan	nan	nan
José C. Paz	323,918	nan	nan	nan	nan
La Matanza	1,837,774	1,384,134	24.68	2.1e+11	453,640
Lanús	462,051	460,302	0.38	3,059,001	1,749.00
Lomas de Zamora	694,330	593,985	14.45	1.0e+10	100,345
Malvinas Argentinas	351,788	nan	nan	nan	nan
Merlo	580,806	463,112	20.26	1.4e+10	117,694
Moreno	574,374	373,574	34.96	4.0e+10	200,800
Morón	334,178	424,681	27.08	8.2e+09	90,503
Quilmes	636,026	537,655	15.47	9.7e+09	98,371
San Fernando	172,524	153,045	11.29	3.8e+08	19,479
San Isidro	298,777	294,469	1.44	1.9e+07	4,308.33
San Miguel	326,215	nan	nan	nan	nan
Tigre	447,785	311,842	30.36	1.8e+10	135,943
Tres de Febrero	366,377	341,971	6.66	6.0e+08	24,406
Vicente López	283,510	277,669	2.06	3.4e+07	5,841.00

Cuadro 18: Light GMB. Predicciones valor poblacional por departamentos y errores típicos

Departamento	Censo 2022	Pred INDEC	MAPE INDEC	MSE INDEC	RMSE INDEC
Almirante Brown	585,852	605,271	3.31	6.5e+09	80,792
Avellaneda	370,939	358,512	3.35	6.5e+09	80,792
Berazategui	360,582	372,889	3.41	6.5e+09	80,792
Esteban Echeverría	339,030	383,538	13.13	6.5e+09	80,792
Ezeiza	203,283	229,276	12.79	6.5e+09	80,792
Florencio Varela	497,818	533,446	7.16	6.5e+09	80,792
General San Martín	450,335	426,556	5.28	6.5e+09	80,792
Hurlingham	187,122	195,596	4.53	6.5e+09	80,792
Ituzaingó	179,788	182,993	1.78	6.5e+09	80,792
José C. Paz	323,918	314,878	2.79	6.5e+09	80,792
La Matanza	1,837,774	2,374,149	29.19	6.5e+09	80,792
Lanús	462,051	462,693	0.14	6.5e+09	80,792
Lomas de Zamora	694,330	652,937	5.96	6.5e+09	80,792
Malvinas Argentinas	351,788	366,479	4.18	6.5e+09	80,792
Merlo	580,806	620,307	6.80	6.5e+09	80,792
Moreno	574,374	558,068	2.84	6.5e+09	80,792
Morón	334,178	317,584	4.97	6.5e+09	80,792
Quilmes	636,026	679,375	6.82	6.5e+09	80,792
San Fernando	172,524	176,795	2.48	6.5e+09	80,792
San Isidro	298,777	291,704	2.37	6.5e+09	80,792
San Miguel	326,215	308,784	5.34	6.5e+09	80,792
Tigre	447,785	nan	nan	nan	nan
Tres de Febrero	366,377	344,172	6.06	6.5e+09	80,792
Vicente López	283,510	266,880	5.87	6.5e+09	80,792

Cuadro 19: INDEC. Predicciones valor poblacional por departamentos y errores típicos

13.2. Diccionario de Departamentos

Debido a las inconsistencias y falta de normalización de los archivos CSV que provee el INDEC, fue necesaria la creación de un diccionario de departamento asociando las distintas acepciones del nombre de departamento con su correspondiente código para que el proceso de ETL pudiese utilizar los códigos de departamento como clave foránea común a todos los Censos Nacionales. Se encontraron casos donde el mismo departamento figura nombrado con o sin tilde, con espacios o abreviaturas en el campo.

De esta forma en la ingesta (ETL) se buscaba el código de departamento correspondiente, que funciona como clave única. El formato del diccionario puede verse en el Cuadro 20.

CodigoDpto	Departamento
6005	General Sarmiento
6005	General Sarmiento (4)
6260	Esteban Echeverría (1)
6260	Esteban Echeverria
6260	Esteban Echeverría
6270	Ezeiza
6270	Ezeiza (2)
6274	Florencio Varela (3)
6274	Florencio Varela
6274	Florencio Varela (3)

Cuadro 20: Diccionario. Primeras 10 filas

13.3. Diagrama de entidad relación

A modo descriptivo se indica el diagrama de entidad relación de la base de datos confeccionada para este trabajo. Figura 20.

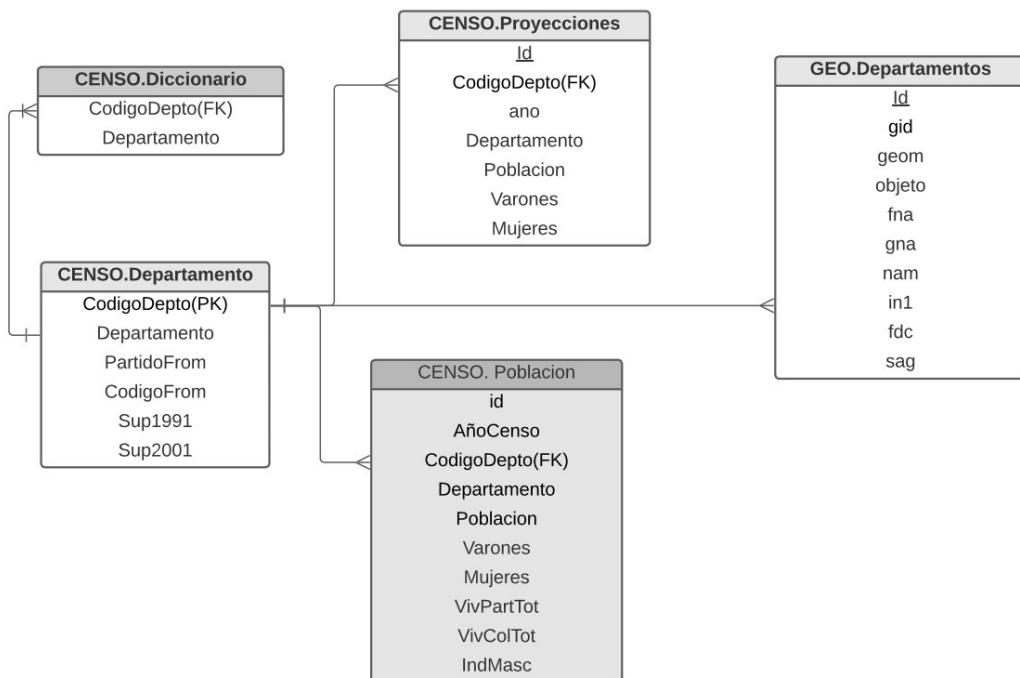


Figura 20: Diagrama de Entidad Relación

13.4. Scripts SQL y Python - Código

13.4.1. Script Creación Tablas.sql

```
1      -- ## POBLACION
2      CREATE TABLE IF NOT EXISTS public.poblacion (
3          Id SERIAL PRIMARY KEY,
4          "AnoCenso" VARCHAR(50),
5          "CodigoDpto" VARCHAR(50),
6          "Departamento" VARCHAR(150),
7          "Poblacion" INT,
8          "Varones" INT,
9          "Mujeres" INT,
10         "VivPartTot" INT,
11         "VivColectTot" INT,
12         "IndMasc" FLOAT,
13         "Superficie" INT,
14         "DensPob" FLOAT
15     );
16
17     TRUNCATE TABLE public.poblacion;
18
19     -- ## DIM Departamento
20     CREATE TABLE IF NOT EXISTS public.DimDeptos (
21         Id SERIAL PRIMARY KEY,
22         "CodigoDpto" VARCHAR(50),
23         "Departamento" VARCHAR(150),
24         "PartidoFrom" VARCHAR(150),
25         "CodigoFrom" VARCHAR(50),
26         "Sup1991" INT,
27         "Sup2001" INT,
28         "IsAMBA" BOOLEAN,
29         "Comentarios" VARCHAR(550)
30     );
31
32     TRUNCATE TABLE public.DimDeptos;
33
34     -- ## DICTIONARIO DATOS PARTIDO CODIGO
35     CREATE TABLE IF NOT EXISTS public.diccionario (
36         "CodigoDpto" VARCHAR(50),
37         "Departamento" VARCHAR(150)
38     );
39
40     TRUNCATE TABLE public.diccionario;
41
42     -- ## PROYECCIONES
43     CREATE TABLE IF NOT EXISTS public.proyecciones (
44         Id SERIAL PRIMARY KEY,
45         "CodigoDpto" VARCHAR(50),
46         "ano" INT,
47         "Departamento" VARCHAR(150),
48         "Poblacion" INT,
49         "Varones" INT,
50         "Mujeres" INT
51     );
52
```

```

53
54     TRUNCATE TABLE public.proyecciones;
55
56     CREATE TABLE IF NOT EXISTS geo.amba_pob AS
57     SELECT *
58     FROM geo."vAMBOgeom"
59     ALTER TABLE geo.amba_pob
60     ADD COLUMN pob1991 INT,
61     ADD COLUMN pob2001 INT,
62     ADD COLUMN pob2010 INT,
63     ADD COLUMN pob2022 INT;
64 ######
65     ALTER TABLE geo.amba_pob
66     ADD COLUMN dens1991 INT,
67     ADD COLUMN dens2001 INT,
68     ADD COLUMN dens2010 INT,
69     ADD COLUMN dens2022 INT;
70
71     UPDATE geo.amba_pob AS am
72     SET pob1991 = c.pob
73     FROM public.v_censos_amba c
74     WHERE am.cod_depto = c.cod_depto AND anio='1991';
75     --- Update Poblacion de los censos
76     UPDATE geo.amba_pob AS am
77     SET pob2001 = c.pob
78     FROM public.v_censos_amba c
79     WHERE am.cod_depto = c.cod_depto AND anio='2001';
80
81     UPDATE geo.amba_pob AS am
82     SET pob2010 = c.pob
83     FROM public.v_censos_amba c
84     WHERE am.cod_depto = c.cod_depto AND anio='2010';
85     UPDATE geo.amba_pob AS am
86     SET pob2022 = c.pob
87     FROM public.v_censos_amba c
88     WHERE am.cod_depto = c.cod_depto AND anio='2022';
89
90     --- Update DENSIDAD de los censos
91     UPDATE geo.amba_pob AS am
92     SET dens1991 = c.dens_pob
93     FROM public.v_censos_amba c
94     WHERE am.cod_depto = c.cod_depto AND anio='1991';
95     UPDATE geo.amba_pob AS am
96     SET dens2001 = c.dens_pob
97     FROM public.v_censos_amba c
98     WHERE am.cod_depto = c.cod_depto AND anio='2001';
99
100    UPDATE geo.amba_pob AS am
101    SET dens2010 = c.dens_pob
102    FROM public.v_censos_amba c
103    WHERE am.cod_depto = c.cod_depto AND anio='2010';
104    UPDATE geo.amba_pob AS am
105    SET dens2022 = c.dens_pob
106    FROM public.v_censos_amba c
107    WHERE am.cod_depto = c.cod_depto AND anio='2022';

```

Listing 1: CreateTables.sql

13.4.2. Script PopulateTables.sql

```
1      -- Step 1: Insert data from CSV file without the primary key column
2      --- POBLACION START ---
3      COPY public.poblacion ("AnoCenso", "CodigoDpto", "Departamento", "Poblacion"
4          , "Varones", "Mujeres", "VivPartTot", "VivColectTot", "IndMasc", "
5              Superficie","DensPob")
6      FROM 'C:/Temp/1991_A~1.CSV'
7      WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
8          ENCODING 'UTF8');
9
10
11
12
13
14      COPY public.poblacion ("AnoCenso", "CodigoDpto", "Departamento", "Poblacion"
15          , "Varones", "Mujeres", "VivPartTot", "VivColectTot", "IndMasc", "
16              Superficie","DensPob")
17      FROM 'C:/Temp/1991_Resto.CSV'
18      WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
19          ENCODING 'UTF8');
20
21
22
23
24      COPY public.poblacion ("AnoCenso", "CodigoDpto", "Departamento", "Poblacion"
25          , "Varones", "Mujeres", "VivPartTot", "VivColectTot", "IndMasc", "
26              Superficie","DensPob")
27      FROM 'C:/Temp/2001.CSV'
28      WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
29          ENCODING 'UTF8');
30
31
32
33
34      COPY public.poblacion ("AnoCenso", "CodigoDpto", "Departamento", "Poblacion"
35          , "Varones", "Mujeres", "VivPartTot", "VivColectTot", "IndMasc", "
36              Superficie","DensPob")
37      FROM 'C:/Temp/2010.CSV'
38      WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
39          ENCODING 'UTF8');
40
41
42
43
44      COPY public.poblacion ("AnoCenso", "CodigoDpto", "Departamento", "Poblacion"
45          , "Varones", "Mujeres", "VivPartTot", "VivColectTot", "IndMasc", "
46              Superficie","DensPob")
47      FROM 'C:/Temp/2022.CSV'
48      WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
49          ENCODING 'UTF8');
50
51
52      -- POBLACION END -----
53      ---- DICCIONARIO ----
54      COPY public.diccionario ("CodigoDpto", "Departamento")
55      FROM 'C:/Temp/DiccionarioPartidosCodigo.CSV'
56      WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
57          ENCODING 'UTF8');
```

```

35      ---      DIMdepto -----
36 COPY public.dimdepto (
37   "CodigoDpto",
38   "Departamento",
39   "PartidoFrom",
40   "CodigoFrom",
41   "Sup1991",
42   "Sup2001",
43   "IsAMBA",
44   "Comentarios")
45 FROM 'C:/Temp/DIM Departamento.CSV'
46 WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
47       ENCODING 'UTF8');

48
49      ---      Proyeccion 2025 -----
50 COPY public.proyecciones (
51   "CodigoDpto",
52   "ano",
53   "Departamento",
54   "Poblacion",
55   "Varones",
56   "Mujeres")
57 FROM 'C:/Temp/proy_1025.CSV'
58 WITH (FORMAT csv, HEADER true, DELIMITER ';;', QUOTE ''', ESCAPE '''',
      ENCODING 'UTF8');

```

Listing 2: PopulateTables.sql

13.4.3. Regresión Lineal. Jupyter Notebook

```

1  ## Imports
2
3  import json
4  import pandas as pd
5  import matplotlib.pyplot as plt
6  import seaborn as sns
7  import psycopg2
8  import numpy as np
9  from sklearn.cluster import KMeans
10 from sklearn.preprocessing import StandardScaler
11 from sklearn.ensemble import RandomForestClassifier
12 from sklearn.model_selection import train_test_split
13 from sklearn.metrics import accuracy_score
14 from sklearn.linear_model import LinearRegression
15 from sklearn.metrics import mean_squared_error, r2_score
16
17 from statsmodels.tsa.arima.model import ARIMA
18 from statsmodels.tsa.stattools import adfuller
19 from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
20
21
22
23 from utils import read_table_into_dataframe
24 from utils import create_table_pdf
25 from utils import dataframe_to_latex

```

```

26     from utils import dataframe_to_image
27     ##### CONNECT TO POSTGRES DATABASE
28     ## AMBA
29
30     import psycopg2
31
32     # Establish connection parameters
33     dbname = 'AMBA',
34     user = 'postgres',
35     password = 'Fermi1987',
36     host = 'localhost' # By default, localhost
37     port = '5432' # By default, 5432
38
39     # Connect to the PostgreSQL database
40     try:
41         conn = psycopg2.connect(
42             dbname=dbname,
43             user=user,
44             password=password,
45             host=host,
46             port=port
47         )
48
49         # Create a cursor object
50         cursor = conn.cursor()
51
52         # Execute a query
53         cursor.execute("SELECT version();")
54         db_version = cursor.fetchone()
55         print("Connected to:", db_version)
56
57         # Commit the transaction
58         conn.commit()
59
60     except psycopg2.Error as e:
61         print("Error connecting to PostgreSQL:", e)
62
63     finally:
64         # Close the cursor and connection
65         if 'cursor' in locals() and cursor is not None:
66             cursor.close()
67         # if 'conn' in locals() and conn is not None:
68         #     conn.close()
69         # Read vCensoAMBA
70         df = read_table_into_dataframe('public.proyecciones')
71         df=df.sort_values(by=['Departamento', 'ano'])
72         if df is not None:
73             print(df)
74             dataframe_to_latex(df.head(10), 'proyecciones2025.tex')
75
76             # Example usage:
77             ## Assuming 'df' is your DataFrame
78             # dataframe_to_image(df.head(), 'output', 'svg') # Save as SVG image
79             # dataframe_to_image(df.head(), 'output', 'jpeg') # Save as JPEG image
80             # READ TASAS .csv

```

```

81 dfTasas = pd.read_csv('C:/Users/Fer/ITBA_TFI/datasets/tasasBA.csv', sep='; ', header='infer')
82
83 # Display the first few rows of the DataFrame
84 print(dfTasas.head())
85
86 dataframe_to_latex(dfTasas, 'tasas.tex')
87 # Read vCensosAmBa
88 df = read_table_into_dataframe('public.v_censos_amba')
89 df=df.sort_values('nam')
90 df.drop('Superficie', axis=1, inplace=True)
91 if df is not None:
92     print(df)
93     # Convert 'anio' in censos_amba to match the type in dfTasas
94     df['anio'] = df['anio'].astype(int)
95
96 # Select the relevant columns from dfTasas to add to censos_amba
97 columns_to_add = ['TMI', 'TGF', 'TBN', 'TBM', 'TCV', 'Matiria']
98 dfTasas_selected = dfTasas[['Ano']] + columns_to_add].copy()
99
100 # Rename the 'Ano' column to match the 'anio' column in censos_amba
101 dfTasas_selected.rename(columns={'Ano': 'anio'}, inplace=True)
102
103 # Merge the two dataframes on the common column 'anio'
104 merged_df = pd.merge(df, dfTasas_selected, how='left', on='anio')
105
106 # Fill NaN values with 0 if necessary
107 for col in columns_to_add:
108     merged_df[col] = merged_df[col].fillna(0)
109
110 # Display the first few rows of the merged dataframe
111 print(merged_df.head())
112
113 # Assuming df is your DataFrame containing the data
114 # Loop through each unique value in the 'nam' column
115 features=['anio']
116
117 def perform_linear_regression(df, department, features):
118     # Filter DataFrame for the specified department
119     df_dept = df[df['nam'] == department]
120
121     # Drop rows with missing values in any of the selected features
122     df_dept = df_dept.dropna(subset=features + ['pob'])
123
124     # Extract the features and target variable
125     X = df_dept[features].values.reshape(-1, len(features)) # Reshape X to be 2D
126         array
127     y = df_dept['pob'].values
128
129     # Perform linear regression
130     model = LinearRegression()
131     model.fit(X, y)
132
133     # Predict population values using the fitted model
134     y_pred = model.predict(X)

```

```

135 # Plot the results
136 plt.figure(figsize=(10, 6))
137 plt.scatter(df_dept['anio'], df_dept['pob'], color='blue', label='Actual
138 Population')
139 plt.plot(df_dept['anio'], y_pred, color='red', label='Predicted Population')
140 plt.title(f'Population Projection for {department}')
141 plt.xlabel('Year')
142 plt.ylabel('Population')
143 plt.legend()
144 plt.grid(True)
145 plt.show()

146 # Get unique department names
147 departments = df['nam'].unique()

148 # Iterate over each department and perform linear regression
149 for department in departments:
150     perform_linear_regression(df, department, features)
151
152 def linear_regression_forecast(df_group):
153     # Drop rows with NaN values in the target variable 'pob'
154     df_group = df_group.dropna(subset=['pob'])

155
156     if df_group.empty:
157         print("No data available for forecasting in this group.")
158         return
159
160     # Filter data for years 1990, 2001, and 2010
161     X_train = df_group[df_group['anio'].isin([1991, 2001, 2010])]['anio'].values.
162         reshape(-1, 1)
163     y_train = df_group[df_group['anio'].isin([1991, 2001, 2010])]['pob'].values

164     # Create and fit the Linear Regression Model
165     model = LinearRegression()
166     model.fit(X_train, y_train)

167
168     # Predict population for the year 2022
169     X_test = [[2022]]
170     y_pred = model.predict(X_test)

171
172     # Visualize the Results
173     plt.scatter(X_train, y_train, color='blue', label='Training Data')
174     plt.plot(X_test, y_pred, color='red', marker='o', markersize=10, label='
175         Predicted Population for 2022')
176     plt.xlabel('Year')
177     plt.ylabel('Population')
178     plt.title(f'Linear Regression: Population Forecast for {df_group["nam"].iloc
179         [0]}')
180     plt.legend()
181     plt.show()

182 # Apply the linear_regression_forecast function to each 'nam' group
183 for name, group in df.groupby('nam'):
184     print("Forecast for", name)
185     linear_regression_forecast(group)

186
187     def linear_regression_forecast(df_group):

```

```

187 # Drop rows with NaN values in the target variable 'pob'
188 df_group = df_group.dropna(subset=['pob'])
189
190 if df_group.empty:
191     print("No data available for forecasting in this group.")
192     return
193
194 # Filter data for years 1990, 2001, and 2010
195 X_train = df_group[df_group['anio'].isin([1990, 2001, 2010])]['anio'].values.reshape(-1, 1)
196 y_train = df_group[df_group['anio'].isin([1990, 2001, 2010])]['pob'].values
197
198 # Create and fit the Linear Regression Model
199 model = LinearRegression()
200 model.fit(X_train, y_train)
201
202 # Predict population for the year 2022
203 X_test = [[2022]]
204 y_pred = model.predict(X_test)
205
206 # Calculate metrics
207 actual_population_2022 = df_group[df_group['anio'] == 2022]['pob'].values
208 mse = mean_squared_error(actual_population_2022, y_pred)
209 rmse = np.sqrt(mse)
210 MAPE = np.mean(np.abs((actual_population_2022 - y_pred) /
211     actual_population_2022)) * 100
212
213 # Print metrics
214 print(f"Mean Squared Error (MSE): {mse}")
215 print(f"Root Mean Squared Error (RMSE): {rmse}")
216 print(f"Mean Absolute Percentage Error (MAPE): {MAPE}%")
217
218 # Visualize the Results
219 plt.scatter(X_train, y_train, color='blue', label='Training Data')
220 plt.plot(X_test, y_pred, color='red', marker='o', markersize=10, label='Predicted Population for 2022')
221 plt.xlabel('Year')
222 plt.ylabel('Population')
223 plt.title(f'Linear Regression: Population Forecast for {df_group["nam"].iloc[0]}')
224 plt.legend()
225 plt.show()
226
227 # Apply the linear_regression_forecast function to each 'nam' group
228 for name, group in df.groupby('nam'):
229     print("Forecast for", name)
230     linear_regression_forecast(group)
231
232     from cgi import print_exception
233
234 def linear_regression_forecast(df_group):
235     # Drop rows with NaN values in the target variable 'pob'
236     df_group = df_group.dropna(subset=['pob'])
237
238     if df_group.empty:

```

```

239     print("No data available for forecasting in this group.")
240     return
241
242     # Filter data for years 1990, 2001, and 2010
243     X_train = df_group[df_group['anio'].isin([1990, 2001, 2010])]['anio'].values.reshape(-1, 1)
244     y_train = df_group[df_group['anio'].isin([1990, 2001, 2010])]['pob'].values
245
246     # Create and fit the Linear Regression Model
247     model = LinearRegression()
248     model.fit(X_train, y_train)
249
250     # Predict population for the year 2022
251     X_test = [[2022]]
252     y_pred_2022 = model.predict(X_test)
253
254
255     # Calculate error for the year 2022
256     actual_population_2022 = df_group[df_group['anio'] == 2022]['pob'].values
257     error_2022 = actual_population_2022 - y_pred_2022
258     mse = mean_squared_error(actual_population_2022, y_pred_2022).round(1)
259     rmse = np.sqrt(mse).round(1)
260     mape = np.mean(np.abs((actual_population_2022 - y_pred_2022) /
261                         actual_population_2022)).round(4) * 100
262
263
264     # Create a DataFrame with the forecasted data for the year 2022 and error
265     forecast_data = pd.DataFrame({
266         'nam': [df_group['nam'].iloc[0]],
267         'real_2022': int(actual_population_2022),
268         'prediction_2022': int(y_pred_2022),
269         'AbsError': error_2022.round(1),
270         'MSE': "{:.2e}".format(mse),
271         'RMSE': rmse,
272         'MAPE': mape
273     })
274
275     return forecast_data
276     # Create an empty list to store the forecast datasets for each 'nam' group
277     forecast_datasets = []
278
279     # Apply the linear_regression_forecast function to each 'nam' group
280     for name, group in df.groupby('nam'):
281         print("Forecast for", name)
282         forecast_data = linear_regression_forecast(group)
283         forecast_datasets.append(forecast_data)
284
285     # Concatenate datasets into a single DataFrame
286     result_dataset = pd.concat(forecast_datasets, ignore_index=True)
287
288     # Display the resulting dataset
289     print(result_dataset)
290
291     # Plotting

```

```

291     plt.figure(figsize=(8, 6))
292     sns.boxplot(y='MAPE', data=result_dataset)
293     plt.ylabel('MAPE')
294     plt.title('Distribution of Mean Absolute Percentage Error (MAPE) by name')
295     plt.xticks(rotation=90) # Rotate x-axis labels for better visibility
296     plt.tight_layout()
297     plt.show()
298
299     mape_stats = result_dataset['MAPE'].describe()
300 print(mape_stats)

```

Listing 3: Regresion Lineal.ipynb

13.4.4. CART.Árboles de Regresión. Jupyter Notebook

```

1      ##### CONNECT TO POSTGRES DATABASE
2      ## AMBA
3
4      import psycopg2
5
6      # Establish connection parameters
7      dbname = 'AMBA'
8      user = 'postgres'
9      password = 'Ferm1987'
10     host = 'localhost' # By default, localhost
11     port = '5432' # By default, 5432
12
13     # Connect to the PostgreSQL database
14     try:
15         conn = psycopg2.connect(
16             dbname=dbname,
17             user=user,
18             password=password,
19             host=host,
20             port=port
21         )
22
23         # Create a cursor object
24         cursor = conn.cursor()
25
26         # Execute a query
27         cursor.execute("SELECT version();")
28         db_version = cursor.fetchone()
29         print("Connected to:", db_version)
30
31         # Commit the transaction
32         conn.commit()
33
34     except psycopg2.Error as e:
35         print("Error connecting to PostgreSQL:", e)
36
37     finally:
38         # Close the cursor and connection
39         if 'cursor' in locals() and cursor is not None:
40             cursor.close()
41         # if 'conn' in locals() and conn is not None:

```

```

42     #     conn.close()
43 # Read vCensosAmbar
44 df = read_table_into_dataframe('public.proyecciones')
45 df=df.sort_values(by=['Departamento', 'ano'])
46 if df is not None:
47     print(df)
48 dataframe_to_latex(df.head(10), 'proyecciones2025.tex')
49
50     # Example usage:
51 # # Assuming 'df' is your DataFrame
52 # dataframe_to_image(df.head(), 'output', 'svg') # Save as SVG image
53 # dataframe_to_image(df.head(), 'output', 'jpeg') # Save as JPEG image
54
55 # Read vCensosAmbar
56 df = read_table_into_dataframe('public.v_censos_amba')
57 df=df.sort_values('nam')
58 df.drop('Superficie', axis=1, inplace=True)
59 if df is not None:
60     print(df)
61     # Convert 'anio' in censos_amba to match the type in dfTasas
62     df['anio'] = df['anio'].astype(int)
63
64     # Select the relevant columns from dfTasas to add to censos_amba
65     columns_to_add = ['TMI', 'TGF', 'TBN', 'TBM', 'TCV', 'Matiria']
66     dfTasas_selected = dfTasas[['Ano']] + columns_to_add].copy()
67
68     # Rename the 'Ano' column to match the 'anio' column in censos_amba
69     dfTasas_selected.rename(columns={'Ano': 'anio'}, inplace=True)
70
71     # Merge the two dataframes on the common column 'anio'
72     merged_df = pd.merge(df, dfTasas_selected, how='left', on='anio')
73
74     # Fill NaN values with 0 if necessary
75     for col in columns_to_add:
76         merged_df[col] = merged_df[col].fillna(0)
77     # Display the first few rows of the merged dataframe
78     print(merged_df.head())
79
80     df= merged_df
81     # Drop the 'vivpart' column from the DataFrame df
82     df = df.drop(columns=['vivpart', 'vivtotal'])
83     df=df.dropna()
84     df.describe()
85     df.shape
86     df.columns
87     df.dtypes
88     df.info()
89     df.isnull().sum()
90
91
92 # Step 1: Filter the DataFrame
93 years_of_interest = [1991, 2001, 2010, 2022]
94 df_filtered = df[df['anio'].isin(years_of_interest)]
95
96 # Step 2: Group the data by 'nam'
97 grouped = df_filtered.groupby('nam')

```

```

98
99 # Step 3 and 4: Train a decision tree regression model for each group and make
100 # predictions
100 predictions = {}
101 for name, group_data in grouped:
102     # Split data into features (X) and target variable (y)
103     X = group_data[group_data['anio'] != 2022][['anio', 'pob']]
104     y = group_data[group_data['anio'] != 2022]['pob']
105
106     # Train-test split for validation
107     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
108             random_state=42)
109
110     # Initialize and train the decision tree regression model
111     model = DecisionTreeRegressor(random_state=42)
112     model.fit(X_train, y_train)
113
114     # Make predictions for the year 2022
115     X_2022 = group_data[group_data['anio'] == 2022][['anio', 'pob']]
116     y_pred = model.predict(X_2022)
117
118     # Store predictions
119     predictions[name] = y_pred
120
121     # Display predictions
122     for name, prediction in predictions.items():
123         print(f"Predicted population for {name} in 2022: {prediction}")
124         import numpy as np
125
126         # Initialize lists to store evaluation metrics
127         mse_list = []
128         rmse_list = []
129         mape_list = []
130
131         # Iterate through predictions
132         for name, prediction in predictions.items():
133             # Actual population for 2022
134             actual_population = df[(df['nam'] == name) & (df['anio'] == 2022)]['pob'].values[0]
135
136             # Calculate squared error
137             squared_error = (actual_population - prediction) ** 2
138
139             # Calculate MSE
140             mse = np.mean(squared_error)
141             mse_list.append(mse)
142
143             # Calculate RMSE
144             rmse = np.sqrt(mse)
145             rmse_list.append(rmse)
146
147             # Calculate MAPE
148             mape = np.mean(np.abs((actual_population - prediction) /
149             actual_population)) * 100
150             mape_list.append(mape)

```

```

150     # Calculate average metrics
151     avg_mse = np.mean(mse_list)
152     avg_rmse = np.mean(rmse_list)
153     avg_mape = np.mean(mape_list)
154
155     print(f"Average Mean Squared Error (MSE): {avg_mse}")
156     print(f"Average Root Mean Squared Error (RMSE): {avg_rmse}")
157     print(f"Average Mean Absolute Percentage Error (MAPE): {avg_mape}%")
158     # Initialize lists to store data
159     data = {'nam': [], 'Actual_Population_2022': [], 'Predicted_Population_2022': []
160         , 'Error': [], 'MSE': [], 'RMSE': [], 'MAPE': []}
161
162     # Iterate through predictions
163     for name, prediction in predictions.items():
164         # Actual population for 2022
165         actual_population = df[(df['nam'] == name) & (df['anio'] == 2022)]['pob'
166             ].values
167
168         # Calculate error
169         error = actual_population - prediction
170
171         # Calculate evaluation metrics
172         mse = int(np.mean(error ** 2))
173         rmse = np.sqrt(mse)
174         mape = np.mean(np.abs(error / actual_population)).round(3) * 100
175
176         # Append data to lists
177         data['nam'].extend([name] * len(prediction))
178         data['Actual_Population_2022'].extend(actual_population)
179         data['Predicted_Population_2022'].extend(prediction)
180         data['Error'].extend(error)
181         data['MSE'].extend([mse] * len(prediction))
182         data['RMSE'].extend([rmse] * len(prediction))
183         data['MAPE'].extend([mape] * len(prediction))
184
185     # Create DataFrame
186     result_df = pd.DataFrame(data)
187
188     # Display the DataFrame
189     print(result_df)
190
191     # Plot MAPE for each 'nam' in a bar chart
192     plt.figure(figsize=(10, 6))
193     result_df.groupby('nam')['MAPE'].mean().plot(kind='bar', color='skyblue')
194     plt.title('Mean Absolute Percentage Error (MAPE) for each nam')
195     plt.xlabel('nam')
196     plt.ylabel('MAPE')
197     plt.xticks(rotation=45)
198     plt.grid(axis='y')
199     plt.tight_layout()
200     plt.show()
201
202     # Create a box plot of the MAPE series
203     plt.figure(figsize=(8, 6))

```

```

204 plt.boxplot(result_df['MAPE'])
205 plt.title('Box plot of Mean Absolute Percentage Error (MAPE)')
206 plt.ylabel('MAPE')
207 plt.grid(axis='y')
208 plt.tight_layout()
209 plt.show()

```

Listing 4: RegTrees.ipynb

13.4.5. Random Forest. Jupyter Notebook

```

1   # Read vCensosAmba
2   df = read_table_into_dataframe('public.v_censos_amba')
3   df=df.sort_values('nam')
4   df.drop('Superficie', axis=1, inplace=True)
5   if df is not None:
6       print(df)
7   # Convert 'anio' in censos_amba to match the type in dfTasas
8   df['anio'] = df['anio'].astype(int)
9
10  # Select the relevant columns from dfTasas to add to censos_amba
11  columns_to_add = ['TMI', 'TGF', 'TBN', 'TBM', 'TCV', 'Matiria']
12  dfTasas_selected = dfTasas[['Ano'] + columns_to_add].copy()
13
14  # Rename the 'Ano' column to match the 'anio' column in censos_amba
15  dfTasas_selected.rename(columns={'Ano': 'anio'}, inplace=True)
16
17  # Merge the two dataframes on the common column 'anio'
18  merged_df = pd.merge(df, dfTasas_selected, how='left', on='anio')
19
20  # Fill NaN values with 0 if necessary
21  for col in columns_to_add:
22      merged_df[col] = merged_df[col].fillna(0)
23  # Display the first few rows of the merged dataframe
24  print(merged_df.head())
25
26  df= merged_df
27  df=df.drop(['vivpart', 'vivtotal'], axis=1)
28  df=df.dropna()
29  df.describe()
30  df.info()
31
32  # Step 1: Filter the DataFrame
33  years_of_interest = [1991, 2001, 2010, 2022]
34  df_filtered = df[df['anio'].isin(years_of_interest)]
35
36  # Step 2: Group the data by 'nam'
37  grouped = df_filtered.groupby('nam')
38
39  # Step 3 and 4: Train a decision tree regression model for each group and make
40  # predictions
41  predictions = {}
42  for name, group_data in grouped:
43      # Split data into features (X) and target variable (y)
44      X = group_data[group_data['anio'] != 2022][['anio', 'pob']]
      y = group_data[group_data['anio'] != 2022]['pob']

```

```

45
46     # Train-test split for validation
47     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
48             random_state=42)
49
50     # Initialize and train the decision tree regression model
51     model = DecisionTreeRegressor(random_state=42)
52     model.fit(X_train, y_train)
53
54     # Make predictions for the year 2022
55     X_2022 = group_data[group_data['anio'] == 2022][['anio', 'pob']]
56     y_pred = model.predict(X_2022)
57
58     # Store predictions
59     predictions[name] = y_pred
60
61     # Display predictions
62     for name, prediction in predictions.items():
63         print(f"Predicted population for {name} in 2022: {prediction}")
64         import numpy as np
65
66     # Initialize lists to store evaluation metrics
67     mse_list = []
68     rmse_list = []
69     mape_list = []
70
71     # Iterate through predictions
72     for name, prediction in predictions.items():
73         # Actual population for 2022
74         actual_population = df[(df['nam'] == name) & (df['anio'] == 2022)]['pob'].values[0]
75
76         # Calculate squared error
77         squared_error = (actual_population - prediction) ** 2
78
79         # Calculate MSE
80         mse = np.mean(squared_error)
81         mse_list.append(mse)
82
83         # Calculate RMSE
84         rmse = np.sqrt(mse)
85         rmse_list.append(rmse)
86
87         # Calculate MAPE
88         mape = np.mean(np.abs((actual_population - prediction) / actual_population))
89         * 100
90         mape_list.append(mape)
91
92     # Calculate average metrics
93     avg_mse = np.mean(mse_list)
94     avg_rmse = np.mean(rmse_list)
95     avg_mape = np.mean(mape_list)
96
97     print(f"Average Mean Squared Error (MSE): {avg_mse}")
98     print(f"Average Root Mean Squared Error (RMSE): {avg_rmse}")
99     print(f"Average Mean Absolute Percentage Error (MAPE): {avg_mape}%")

```

```

98 # Initialize lists to store data
99 data = {'nam': [], 'Actual_Population_2022': [], 'Predicted_Population_2022': [],
100   'Error': [], 'MSE': [], 'RMSE': [], 'MAPE': []}
101
102 # Iterate through predictions
103 for name, prediction in predictions.items():
104     # Actual population for 2022
105     actual_population = df[(df['nam'] == name) & (df['anio'] == 2022)]['pob'].values
106
107     # Calculate error
108     error = actual_population - prediction
109
110     # Calculate evaluation metrics
111     mse = int(np.mean(error ** 2))
112     rmse = np.sqrt(mse)
113     mape = np.mean(np.abs(error / actual_population)).round(3) * 100
114
115     # Append data to lists
116     data['nam'].extend([name] * len(prediction))
117     data['Actual_Population_2022'].extend(actual_population)
118     data['Predicted_Population_2022'].extend(prediction)
119     data['Error'].extend(error)
120     data['MSE'].extend([mse] * len(prediction))
121     data['RMSE'].extend([rmse] * len(prediction))
122     data['MAPE'].extend([mape] * len(prediction))
123
124 # Create DataFrame
125 result_df = pd.DataFrame(data)
126
127 # Display the DataFrame
128 print(result_df)
129
130
131 # Plot MAPE for each 'nam' in a bar chart
132 plt.figure(figsize=(10, 6))
133 result_df.groupby('nam')['MAPE'].mean().plot(kind='bar', color='skyblue')
134 plt.title('Mean Absolute Percentage Error (MAPE) for each nam')
135 plt.xlabel('nam')
136 plt.ylabel('MAPE')
137 plt.xticks(rotation=45)
138 plt.grid(axis='y')
139 plt.tight_layout()
140 plt.show()
141
142 # Create a box plot of the MAPE series
143 plt.figure(figsize=(8, 6))
144 plt.boxplot(result_df['MAPE'])
145 plt.title('Box plot of Mean Absolute Percentage Error (MAPE)')
146 plt.ylabel('MAPE')
147 plt.grid(axis='y')
148 plt.tight_layout()
149 plt.show()
150 def evaluate_model_for_nam(nam):
151     # Subset data for the given 'nam'

```

```

152 nam_data = df[df['nam'] == nam]
153
154 # Define features and target variable
155 X = nam_data.drop(['pob', 'nam'], axis=1) # Features
156 y = nam_data['pob'] # Target variable
157
158 # Initialize models
159 random_forest = RandomForestRegressor(random_state=42)
160 gradient_boosting = GradientBoostingRegressor(random_state=42)
161
162 # Perform cross-validation
163 rf_scores = cross_val_score(random_forest, X, y, cv=2, scoring='
    neg_mean_squared_error')
164 gb_scores = cross_val_score(gradient_boosting, X, y, cv=2, scoring='
    neg_mean_squared_error')
165
166 # Convert scores to positive values and calculate RMSE
167 rf_rmse_scores = np.sqrt(-rf_scores)
168 gb_rmse_scores = np.sqrt(-gb_scores)
169
170 # Print the mean RMSE scores
171 print(f"Random Forest RMSE for {nam}:", np.mean(rf_rmse_scores))
172 print(f"Gradient Boosting RMSE for {nam}:", np.mean(gb_rmse_scores))
173
174 return np.mean(rf_rmse_scores), np.mean(gb_rmse_scores)
175
176 # Apply the function to each 'nam'
177 for nam in df['nam'].unique():
178     print(f"Evaluating models for {nam}:")
179     evaluate_model_for_nam(nam)
180     print()
181     def predict_population_for_2022(nam):
182         # Subset data for the given 'nam' and the specified years
183         subset_data = df[(df['nam'] == nam) & df['anio'].isin([1991, 2001, 2010])]
184
185         # Check if there are enough samples available for training
186         if len(subset_data) < 3:
187             print(f"Not enough data available for {nam}. Skipping...")
188             return None
189
190         # Define features and target variable for training
191         X_train = subset_data.drop(['pob'], axis=1) # Features
192         y_train = subset_data['pob'] # Target variable
193
194         # Convert categorical variables to one-hot encoding
195         X_train_encoded = pd.get_dummies(X_train)
196
197         # Initialize model (Random Forest as an example)
198         model = RandomForestRegressor(random_state=42)
199
200         # Train the model
201         model.fit(X_train_encoded, y_train)
202
203         # Subset data for the year 2022
204         data_2022 = df[(df['nam'] == nam) & (df['anio'] == 2022)]
205

```

```

206     # Check if there are samples available for prediction
207     if len(data_2022) == 0:
208         print(f"No data available for prediction in 2022 for {nam}. Skipping...")
209         return None
210
211     # Define features for prediction
212     X_2022 = data_2022.drop(['pob'], axis=1)    # Features for 2022
213     X_2022_encoded = pd.get_dummies(X_2022)      # Convert categorical variables
214
215     # Ensure consistent feature names
216     # Add missing columns with 0s
217     missing_cols = set(X_train_encoded.columns) - set(X_2022_encoded.columns)
218     for col in missing_cols:
219         X_2022_encoded[col] = 0
220
221     # Reorder columns to match training data
222     X_2022_encoded = X_2022_encoded[X_train_encoded.columns]
223
224     # Predict the population for 2022
225     population_2022 = model.predict(X_2022_encoded)[0]    # Taking the first
226     # prediction
227
228     # Create a dataset with the results for the given 'nam'
229     result_data = pd.DataFrame({
230         'nam': [nam],
231         'Predicted_Population_2022': [population_2022]
232     })
233
234     return result_data
235
236     # Apply the function to each 'nam'
237     results = pd.concat([predict_population_for_2022(nam) for nam in df['nam'].unique()
238                         () if predict_population_for_2022(nam) is not None], ignore_index=True)
239
240     # Display the results dataset
241     print(results)
242     from sklearn.metrics import mean_squared_error
243     import numpy as np
244
245     # Create an empty list to store the results
246     evaluation_results = []
247
248     # Iterate over each 'nam'
249     for nam in df['nam'].unique():
250         # Get the predicted population for 2022
251         predicted_data = predict_population_for_2022(nam)
252
253         # Skip if prediction is not available
254         if predicted_data is None:
255             continue
256
257         # Get the real population for 2022
258         real_data = df[(df['nam'] == nam) & (df['anio'] == 2022)]
259
260         # Skip if real data is not available
261         if len(real_data) == 0:

```

```

260     print(f"No real data available for {nam} in 2022. Skipping...")  

261     continue  

262  

263     # Calculate MSE  

264     mse = mean_squared_error(real_data['pop'], predicted_data['  

265                               Predicted_Population_2022'])  

266  

267     # Calculate RMSE  

268     rmse = np.sqrt(mse)  

269  

270     # Calculate MAPE  

271     real_population = real_data['pop'].values[0]  

272     predicted_population = predicted_data['Predicted_Population_2022'].values[0]  

273  

274     if real_population == 0:  

275         mape = np.nan # Set MAPE to NaN if real population is zero  

276     else:  

277         mape = np.mean(np.abs((real_population - predicted_population) /  

278                           real_population)) * 100  

279  

280     # Round the predicted population to the nearest integer  

281     predicted_population = int(round(predicted_population))  

282  

283     # Append the results to the list  

284     evaluation_results.append({  

285         'nam': nam,  

286         'Real_Population_2022': real_population,  

287         'Predicted_Population_2022': predicted_population,  

288         'MSE': mse,  

289         'RMSE': rmse,  

290         'MAPE': mape.round(3)
291     })  

292  

293     # Convert the list to a DataFrame  

294     evaluation_results_df = pd.DataFrame(evaluation_results)  

295  

296     # Display the evaluation results  

297     print(evaluation_results_df)  

298     result_RandForest_df=evaluation_results_df  

299     # Calculate the average MAPE  

300     average_mape = evaluation_results_df['MAPE'].mean()  

301     print("Average MAPE:", average_mape)  

302  

303     # Plot bar chart of MAPE for each 'nam'  

304     plt.figure(figsize=(12, 6))  

305     plt.bar(evaluation_results_df['nam'], evaluation_results_df['MAPE'])  

306     plt.title('MAPE for Each nam')  

307     plt.xlabel('nam')  

308     plt.ylabel('MAPE')  

309     plt.xticks(rotation=90)  

310     plt.grid(axis='y')  

311     plt.show()  

312  

313     # Create a boxplot for the MAPE series  

314     plt.figure(figsize=(8, 6))  

315     plt.boxplot(evaluation_results_df['MAPE'], vert=False)

```

```

314 plt.title('Boxplot of MAPE')
315 plt.xlabel('MAPE')
316 plt.grid(axis='x')
317 plt.show()

```

Listing 5: Random Forest.ipynb

13.4.6. LightGBM. Jupyter Notebook

```

1 import lightgbm as lgb
2
3 # Define the function to train a LightGBM model and make predictions for 2022
4 def predict_population_for_2022_lgb(nam):
5     # Subset data for the given 'nam' and the specified years
6     subset_data = df[(df['nam'] == nam) & df['anio'].isin([1991, 2001, 2010])]
7
8     # Check if there are enough samples available for training
9     if len(subset_data) < 3:
10         print(f"Not enough data available for {nam}. Skipping...")
11         return None
12
13     # Define features and target variable for training
14     X_train = subset_data.drop(['pob'], axis=1)    # Features
15     y_train = subset_data['pob']      # Target variable
16
17     # Convert categorical variables to one-hot encoding
18     X_train_encoded = pd.get_dummies(X_train)
19
20     # Initialize LightGBM model
21     lgb_model = lgb.LGBMRegressor(random_state=42)
22
23     # Train the model
24     lgb_model.fit(X_train_encoded, y_train)
25
26     # Subset data for the year 2022
27     data_2022 = df[(df['nam'] == nam) & (df['anio'] == 2022)]
28
29     # Check if there are samples available for prediction
30     if len(data_2022) == 0:
31         print(f"No data available for prediction in 2022 for {nam}. Skipping...")
32         return None
33
34     # Define features for prediction
35     X_2022 = data_2022.drop(['pob'], axis=1)    # Features for 2022
36     X_2022_encoded = pd.get_dummies(X_2022)      # Convert categorical variables
37
38     # Ensure consistent feature names
39     # Add missing columns with 0s
40     missing_cols = set(X_train_encoded.columns) - set(X_2022_encoded.columns)
41     for col in missing_cols:
42         X_2022_encoded[col] = 0
43
44     # Reorder columns to match training data
45     X_2022_encoded = X_2022_encoded[X_train_encoded.columns]
46

```

```

47     # Predict the population for 2022
48     population_2022 = lgb_model.predict(X_2022_encoded)[0]    # Taking the first
49     # prediction
50
51     # Create a dataset with the results for the given 'nam'
52     result_data = pd.DataFrame({
53         'nam': [nam],
54         'Predicted_Population_2022': [population_2022]
55     })
56
57     return result_data
58
59     # Create an empty list to store the results
60     evaluation_results_lgb = []
61
62     # Iterate over each 'nam'
63     for nam in df['nam'].unique():
64         # Get the predicted population for 2022 using LightGBM
65         predicted_data_lgb = predict_population_for_2022_lgb(nam)
66
67         # Skip if prediction is not available
68         if predicted_data_lgb is None:
69             continue
70
71         # Get the real population for 2022
72         real_data_lgb = df[(df['nam'] == nam) & (df['anio'] == 2022)]
73
74         # Skip if real data is not available
75         if len(real_data_lgb) == 0:
76             print(f"No real data available for {nam} in 2022. Skipping...")
77             continue
78
79         # Calculate MSE
80         mse_lgb = mean_squared_error(real_data_lgb['pob'], predicted_data_lgb['Predicted_Population_2022'])
81
82         # Calculate RMSE
83         rmse_lgb = np.sqrt(mse_lgb)
84
85         # Calculate MAPE
86         real_population_lgb = real_data_lgb['pob'].values[0]
87         predicted_population_lgb = predicted_data_lgb['Predicted_Population_2022'].values[0]
88
89         if real_population_lgb == 0:
90             mape_lgb = np.nan    # Set MAPE to NaN if real population is zero
91         else:
92             mape_lgb = np.mean(np.abs((real_population_lgb -
93                                         predicted_population_lgb) / real_population_lgb)) * 100
94
95         # Round the predicted population to the nearest integer
96         predicted_population_lgb = int(round(predicted_population_lgb))
97
98         # Append the results to the list
99         evaluation_results_lgb.append({
100             'nam': nam,

```

```

99     'Real_Population_2022': real_population_lgb,
100    'Predicted_Population_2022': predicted_population_lgb,
101    'MSE': mse_lgb,
102    'RMSE': rmse_lgb,
103    'MAPE': mape_lgb
104  })
105
106 # Convert the list to a DataFrame
107 evaluation_results_lgb_df = pd.DataFrame(evaluation_results_lgb)
108
109 # Display the evaluation results for LightGBM
110 print(evaluation_results_lgb_df)
111 # Calculate the mean MAPE
112 mean_mape_lgb = evaluation_results_lgb_df['MAPE'].mean()
113 print("Mean MAPE for LightGBM:", mean_mape_lgb)
114
115 # Plot bar chart of MAPE for each 'nam' for LightGBM
116 plt.figure(figsize=(12, 6))
117 plt.bar(evaluation_results_lgb_df['nam'], evaluation_results_lgb_df['MAPE'])
118 plt.title('MAPE for Each nam (LightGBM)')
119 plt.xlabel('nam')
120 plt.ylabel('MAPE')
121 plt.xticks(rotation=90)
122 plt.grid(axis='y')
123 plt.show()
124
125 # Create a boxplot for the MAPE series for LightGBM
126 plt.figure(figsize=(8, 6))
127 plt.boxplot(evaluation_results_lgb_df['MAPE'], vert=False)
128 plt.title('Boxplot of MAPE (LightGBM)')
129 plt.xlabel('MAPE')
130 plt.grid(axis='x')
131 plt.show()
132
133 evaluation_results_lgb_df['anio']=2022
134 df1=evaluation_results_lgb_df
135 # Define the file path
136 file_path = 'ext_V3.csv'
137
138 # Read the CSV file into a DataFrame
139 ext = pd.read_csv(file_path)
140
141 # Perform the merge based on the columns 'anio' and 'nam'
142 merged_df = ext.merge(df1, on=['anio', 'nam'], how='left')
143
144 df_final=merged_df.drop(columns=['MSE', 'RMSE', 'Real_Population_2022'])
145 # Rename columns in the DataFrame
146 df_final = df_final.rename(columns={'Predicted_Population_2022': 'Pred_LGB',
147                                 'MAPE': 'MAPE_LGB'})
148
149 df_final.head()

```

Listing 6: LightGMB.ipynb