



UNIVERSIDAD
COMPLUTENSE
MADRID

*MINERÍA DE
DATOS Y
MODELIZACIÓN
PREDICTIVA*

Kristian Sthefan Cortés Prieto

Introducción

En la era de la información, el análisis de datos desempeña un papel fundamental en la comprensión de los fenómenos sociales y políticos. En este contexto, el presente trabajo de minería de datos se enfoca en el estudio de las últimas elecciones en España, desglosadas demográficamente por comunidades autónomas. El análisis se llevará a cabo utilizando el lenguaje de programación Python 3.10, respaldado por las librerías de *numpy*, *pandas*, *itertools*, *pyplot*, y la librería personalizada proporcionada en la cátedra impartida denominada *FuncionesMineria*.

El objetivo principal de este estudio es explorar las relaciones y patrones ocultos en los datos electorales para determinar si es posible predecir el comportamiento de los votantes en función de factores demográficos y socioeconómicos. Esta investigación busca responder a la intrigante pregunta: ¿Pueden los datos de las elecciones pasadas proporcionar información valiosa para predecir tendencias electorales futuras en función de las comunidades autónomas y sus características demográficas?

A lo largo de este informe, se llevará a cabo un proceso sistemático de análisis de datos que incluirá la importación y limpieza de datos, el análisis descriptivo, la detección de patrones y la construcción de modelos de predicción. Cada etapa se desarrollará minuciosamente, proporcionando una visión detallada de cómo los datos se utilizan para responder a la hipótesis planteada.

Este trabajo representa un esfuerzo por aprovechar la riqueza de los datos electorales para comprender mejor las preferencias de voto en España y, posiblemente, ofrecer perspectivas que puedan contribuir a una toma de decisiones política más informada en el futuro. A medida que avanzamos en el análisis, se explorarán las relaciones complejas y las influencias que subyacen en las elecciones autonómicas, brindando así un enfoque sólido para abordar la hipótesis propuesta.

Desarrollo

Introducción al objetivo del problema y las variables implicadas.

El objetivo central de este estudio es analizar la influencia de una serie de variables demográficas sobre los distintos municipios de España y como esto influyó en los votos de izquierda y derecha en las últimas elecciones en España. Entre las variables consideradas se incluyen la edad de los votantes, los porcentajes de desempleo en diferentes regiones, las industrias predominantes en las comunidades autónomas y otras variables socioeconómicas relevantes. El análisis de estas variables proporcionará una comprensión más profunda de los factores que impulsan las preferencias de voto y, a su vez, ayudará a formular estrategias políticas más informadas.

A lo largo de este informe, se seguirá una metodología estructurada que incluirá la importación y corrección de datos, análisis descriptivo, manejo de valores atípicos y perdidos, transformaciones de variables y la construcción de modelos de regresión lineal y logística. Cada etapa se abordará de manera rigurosa para garantizar la calidad y validez de los resultados obtenidos.

Es importante basar los primeros pasos en lo que sería el *entendimiento del negocio*, el *estudio y compresión de los datos* para poder realizar una preparación de los datos lo más ideal posible, para ello primero revisamos el contenido del conjunto de datos proporcionado.

Variable	Descripción	Análisis
Name	Nombre del municipio	Es el valor utilizado como un index, sabiendo esto podemos evitarlo
CodigoProvincia	Código de la provincia (coincide con los dos primeros dígitos del código postal). Toma 52 valores distintos	Variable categórica, podría ser agrupada
CCAA	Comunidad autónoma a la que pertenece el municipio	Variable categórica, podría ser agrupada
Population	Población del municipio en 2016	Variable continua
TotalCensus	Población en edad de votar en 2016	Variable continua
<u>AbstencionAlta</u>	Variable dicotómica que toma el valor 1 si el porcentaje de abstención es superior al 30% y, 0, en otro caso.	Variable dicotómica
<u>AbstentionPtge</u>	Porcentaje de abstención	Variable continua
<u>Izda_Pct</u>	Porcentaje de votos a partidos de izquierda (PSOE y Podemos)	Variable continua
<u>Dcha_Pct</u>	Porcentaje de votos a partidos de derecha (PP y Ciudadanos)	Variable continua
<u>Otros_Pct</u>	Porcentaje de votos a partidos distintos de PP, Ciudadanos, PSOE y Podemos	Variable continua
<u>Izquierda</u>	Variable dicotómica que toma el valor 1 si la suma de los votos de izquierdas es superior a la de derechas y otros y, 0, en otro caso	Variable dicotómica
<u>Derecha</u>	Variable dicotómica que toma el valor 1 si la suma de los votos de derecha es superior a la de izquierda y otros y, 0, en otro caso	Variable dicotómica
Age_0-4_Ptge	Porcentaje de ciudadanos con menos de 5 años	Variable continua
Age_under19_Ptge	Porcentaje de ciudadanos con menos de 19 años	Variable continua
Age_19_65_pct	Porcentaje de ciudadanos entre 19 y 65 años	Variable continua
Age_over65_pct	Porcentaje de ciudadanos con más de 65 años	Variable continua
WomanPopulationPtge	Porcentaje de mujeres	Variable continua
ForeignersPtge	Porcentaje de extranjeros	Variable continua
SameComAutonPtge	Porcentaje de ciudadanos que reside en la misma provincia en la que nacieron	Variable continua
SameComAutonDiffProvPtge	Porcentaje de ciudadanos que reside en la misma CCAA en la que nacieron, pero distinta provincia	Variable continua
DifComAutonPtge	Porcentaje de ciudadanos que reside en la distinta CCAA de la que nacieron	Variable continua
UnemployLess25_Ptge	Porcentaje de parados de menos de 25 años	Variable continua
Unemploy25_40_Ptge	Porcentaje de parados entre 25 y 40 años	Variable continua
UnemployMore40_Ptge	Porcentaje de parados de más de 40 años	Variable continua

AgricultureUnemploymentPtge	Porcentaje de parados en el sector de la agricultura	Variable continua
IndustryUnemploymentPtge	Porcentaje de parados en el sector de la industria	Variable continua
ConstructionUnemploymentPtge	Porcentaje de parados en el sector de la construcción	Variable continua
ServicesUnemploymentPtge	Porcentaje de parados en el sector servicios	Variable continua
totalEmpresas	Número total de empresas en el municipio	Variable continua
Industria	Número de empresas del sector industrial en el municipio	Variable continua
Construccion	Número de empresas del sector de la construcción en el municipio	Variable continua
ComercTTEHosteleria	Número de empresas dedicadas a comercio, transporte u hostelería en el municipio	Variable continua
Servicios	Número de empresas del sector servicios en el municipio	Variable continua
ActividadPpal	Actividad principal de las actividades del municipio (Industria, Construcción, ComercTTEHosteleria,Servicios y Otros)	Variable categórica
inmuebles	Número de inmuebles en el municipio	Variable continua
Pob2010	Población en el municipio en 2010	Variable continua
SUPERFICIE	Superficie del municipio	Variable continua
densidad	Densidad de población del municipio: MuyBaja (<1 hab/ha), Baja (entre 1 y 5 hab/ha), Alta (>5 hab/ha)	Variable categórica
PobChange_pct	Porcentaje de cambio en la población (valores negativos indican que ha disminuido). Respecto a las anteriores elecciones	Variable continua
PersonasInmueble	Número medio de personas que habita un inmueble	Variable continua
Explotaciones	Número de explotaciones agrícolas en el municipio	Variable continua

Tabla 1: Variables con descripción

Esto servirá al momento de realizar la importación de los datos dentro de Python poder hacer las correcciones de los tipos de datos. Además de estas variables se resaltaron las 7 posibles variables objetivos que se deben tomar en cuenta, de las cuales hay que seleccionar únicamente 2, una continua a la que se le aplicara una regresión lineal y una dicotómica que se le aplicara la regresión logística, estas variables son:

- *AbstentionPtge*
- *Izda Pct*
- *Dcha Pct*
- *Otros Pct*
- *AbstencionAlta*
- *Izquierda*
- *Derecha*

Importación del conjunto de datos y asignación correcta de los tipos de variables.

La primera etapa de este análisis implica la importación del conjunto de datos proporcionado por el profesor y la asignación adecuada de los tipos de variables. Esta fase es esencial para garantizar la consistencia y precisión en el análisis subsiguiente.

A medida que se avanza en este informe, cada uno de los puntos mencionados anteriormente se desarrollará en detalle, proporcionando un panorama completo del proceso de minería de datos aplicado a las elecciones y su relación con los votos de izquierda y derecha en España.

The screenshot shows a Jupyter Notebook interface with two code cells. The first cell, titled 'Cargo las librerías', imports various Python libraries including os, pandas, numpy, seaborn, itertools, matplotlib.pyplot, pickle, sys, and sklearn.model_selection. It also appends the path to a local directory and changes the working directory to that path. It then imports functions from 'FuncionesMineria' related to categorical analysis, imputation, and model selection. The second cell, titled 'Cargo los datos', reads an Excel file named 'DatosEleccionesEspaña.xlsx' and displays its first few rows. The data includes columns like Name, CódigoProvincia, CCAA, Population, TotalCensus, AbstencionPtge, AbstencionAlta, Izda_Pct, Dcha_Pct, Otros_Pct, ComercTTEHosteleria, Servicios, ActividadPpal, inmuebles, Pob2010, and S. The data shows values for different regions and their respective election results and socio-economic metrics.

```
import os
import pandas as pd
import numpy as np
import seaborn as sns
import itertools
import matplotlib.pyplot as plt
import pickle
import sys
from sklearn.model_selection import train_test_split

sys.path.append('/Users/SoapyGenie/Dropbox/Documentos/Estudios/Master IA/Mineria de datos y modelización predictiva')

# Establecemos nuestro escritorio de trabajo
os.chdir('/Users/SoapyGenie/Dropbox/Documentos/Estudios/Master IA/Mineria de datos y modelización predictiva/Datos')

# Cargo las funciones que voy a utilizar
from FuncionesMineria import (analisar_variables_categoricas, cuentaDistintos, freq_variables_num, atipicosAmissing,
                                patron_perdidos, ImputacionCuartil, ImputacionCramer, mosaico_targetbinaria,
                                boxplot_targetbinaria, hist_targetbinaria, Transf_Auto, lm, Rsq, validacion_cruzada_lm,
                                modelEffectSizes, crear_data_modelo, Vcramer, lm_forward, lm_backward, lm_stepwise,
                                glm, summary_glm, validacion_cruzada_glm, pseudoR2, impVariablesLog, curva_roc, sensEspCorte)

[52] ✓ 0.0s
```

```
[6] ✓ 5.0s
```

	Name	CódigoProvincia	CCAA	Population	TotalCensus	AbstencionPtge	AbstencionAlta	Izda_Pct	Dcha_Pct	Otros_Pct	ComercTTEHosteleria	Servicios	ActividadPpal	inmuebles	Pob2010	S
0	Abadia	10	Extremadura	336	282	20.213	0	60.444	35.556	1.778	...	0.0	0.0	Otro	216.0	326.0
1	Abertura	10	Extremadura	429	364	25.275	0	54.779	44.118	0.368	...	0.0	0.0	Otro	382.0	459.0
2	Acebo	10	Extremadura	569	569	27.241	0	44.203	53.140	0.966	...	0.0	0.0	Otro	918.0	674.0
3	Acehúche	10	Extremadura	822	704	30.114	1	50.813	45.325	0.000	...	0.0	0.0	Otro	599.0	842.0

Imagen 1: Importación de datos

En un principio, al terminar de hacer la importación de los datos es necesario ver cual fue el tipo de dato que Python lo tomo por defecto, de esta forma se conocerá cuáles son las variables que deben ser transformadas en un paso posterior

The screenshot shows a Jupyter Notebook cell displaying a table of data types for each column in the dataset. The columns and their corresponding data types are:

Name	object
CódigoProvincia	object
CCAA	object
Population	float64
TotalCensus	float64
Izda_Pct	float64
Dcha_Pct	float64
Otros_Pct	float64
Izquierda	object
Derecha	object
Age_0-4_Ptge	float64
Age_under19_Ptge	float64
Age_19_65_pct	float64
Age_over65_pct	float64
WomanPopulationPtge	float64
ForeignersPtge	float64
SameComAutonPtge	float64
SameComAutonDiffProvPtge	float64
DifComAutonPtge	float64
UnemployLess25_Ptge	float64
Unemploy25_40_Ptge	float64
UnemployMore40_Ptge	float64
AgricultureUnemploymentPtge	float64
IndustryUnemploymentPtge	float64
ConstructionUnemploymentPtge	float64
ServicesUnemploymentPtge	float64
totalEmpresas	float64
Industria	float64
Construcción	float64
ComercTTEHosteleria	float64

Imagen 2: Tipos de datos

Se cambian los valores que son considerados categóricos, pero fueron tomados como floats:

```
for var in ['Izquierda', 'Derecha', 'AbstencionAlta', 'ActividadPpal', 'Densidad',  
'CCAA', 'CodigoProvincia']:
```

```
    datos[var] = datos[var].astype(str)
```

Análisis descriptivo del conjunto de datos. Número de observaciones, número y naturaleza de variables, datos erróneos etc.

En esta fase, se llevará a cabo un análisis descriptivo exhaustivo del conjunto de datos. Esto incluirá la determinación del número de observaciones, así como la identificación y descripción de la naturaleza de las variables presentes. Además, se prestará especial atención a la detección de datos erróneos o inconsistentes que puedan afectar la calidad de los resultados.

Analizamos la cantidad de elementos diferentes en las variables categóricas y la distribución de estos en cada elemento además de cual variable tiene pocos elementos diferentes y pueda ser convertida en categórica:

```
analizar_variables_categoricas(datos)
```

```
cuentaDistintos(datos)
```

		Columna	Distintos
0		Population	3597
1		TotalCensus	3310
2		AbstentionPct	5676
3		Izda_Pct	6569
4		Dcha_Pct	6682
5		Censo_Pct	4319
6		Age_14_19_prc	7791
7		Age_19_25_pct	5491
8		Age_25_35_pct	6216
9		Age_over65_pct	6778
10		WomenPopulationPct	4524
11		ForeignersPct	2329
12		SameComAutonPct	6151
13		SameComAutonDiffProvPct	4207
14		DHCComAutonPct	5574
15		UnemploymentPct	3432
16		Unemploy25_40_pct	2681
17		UnemployMore40_pct	2751
18		AgricultureUnemploymentPct	2626
19		IndustryUnemploymentPct	2538
20		ConstructionUnemploymentPct	2505
21		ServicesUnemploymentPct	2904
22		totalEmpresas	1226
23		Industria	308
24		Construcción	457
25		ComerCTEInstancia	803
26		Servicios	758
27		irmuebles	3088
28		Pop2010	3625
29		SUPERFICIE	8110
30		PopChange_pct	3049
31		PersonasInmueble	283
32		Explotaciones	758

Imagen 3: Datos por variable

Con esto se puede analizar la posible agrupación de diferentes elementos que estén poco representados, en este se puede observar que los elementos 51, 52 y 35 del *CodigoProvincia* están poco representados y podrían agruparse, de la misma forma las *CCAA* de *Ceuta*, *Melilla* y *Murcia*.

Otros Elementos que valen la pena resaltar serian *Industria* y *Construcción* de la variable *ActividadPpal*, debido a que tienen 13 y 14 repeticiones respectivamente son una representación muy baja al momento de compararlo con 8119 registros que son en total

Además, se puede ver si existe algún elemento que no esté considerado en el arreglo de elementos predeterminado, como es el caso de: *Densidad* que tiene un “?” que no entra en los elementos contemplados

Posteriormente, procedemos a examinar las variables continuas en busca de posibles valores que estén fuera de los límites establecidos, si estos límites existen, o que puedan presentar valores atípicos susceptibles de corrección en etapas posteriores del análisis.

	count	mean	std	min	25%	50%	75%	max	Asimetría	Kurtosis	Mediana	Rango
Population	8119.0	5741.854785	46215.203797	5.0000	166.0000	549.0000	2427.5000	3141991.000	45.996406	2816.861977	549.00000	3.141986e+06
TotalCensus	8119.0	4260.665599	34428.890744	5.0000	140.0000	447.0000	1846.5000	2363829.000	46.510817	2890.837337	447.00000	2.363824e+06
AbstentionPtge	8119.0	26.506951	7.540091	0.0000	21.6780	26.42900	31.4750	57.576	-0.049941	0.497101	26.42900	5.757600e+01
Izda_Pct	8119.0	34.403789	16.482285	0.0000	21.8925	35.16500	46.0320	94.117	0.059920	-0.492922	35.16500	9.411700e+01
Dcha_Pct	8119.0	48.915409	19.945087	0.0000	38.6905	51.58200	62.2010	100.000	-0.468014	-0.175175	51.58200	1.000000e+02
Otros_Pct	8119.0	14.666183	25.093642	0.0000	0.7595	1.88300	16.4970	100.000	1.801732	1.852872	1.88300	1.000000e+02
Age_0_4_Ptge	8119.0	3.019429	2.053726	0.0000	1.3890	2.97800	4.5330	13.245	0.344483	-0.206360	2.97800	1.324500e+01
Age_under19_Ptge	8119.0	13.567747	6.780648	0.0000	8.3340	13.88900	19.0585	33.696	-0.103561	-0.790818	13.88900	3.369600e+01
Age_19_65_pct	8119.0	57.371541	6.818072	23.4590	53.8450	58.65500	61.8180	100.002	-0.814636	2.156761	58.65500	7.654300e+01
Age_over65_pct	8119.0	29.073583	11.745849	0.0000	19.8245	27.55900	36.9080	76.471	0.598441	0.075884	27.55900	7.647100e+01
WomanPopulationPtge	8119.0	47.302755	4.361907	11.7650	45.7250	48.48500	50.0000	72.683	-1.671491	5.802718	48.48500	6.091800e+01
ForeignersPtge	8119.0	5.619553	7.348553	-8.9600	1.0600	3.59000	8.1800	71.470	2.497559	11.353066	3.59000	8.043000e+01
SameComAutonPtge	8119.0	81629141	12.289063	0.0000	75.8060	84.49300	90.4620	127.156	-1.521517	3.473499	84.49300	1.271560e+02
SameComAutonDiffProvPtge	8119.0	4.336688	6.394440	0.0000	0.6760	2.19000	5.2770	67.308	3.287183	14.563302	2.19000	6.730800e+01
DifComAutonPtge	8119.0	10.729018	8.847295	0.0000	4.9330	8.27100	13.8980	100.000	2.425228	9.660987	8.27100	1.000000e+02
UnemployLess25_Ptge	8119.0	7.322292	9.408555	0.0000	0.0000	5.88200	10.4695	100.000	4.149784	31.656421	5.88200	1.000000e+02
Unemploy25_40_Ptge	8119.0	37.003976	20.317306	0.0000	28.5710	39.93500	46.6670	100.000	0.213126	1.412434	39.93500	1.000000e+02
UnemployMore40_Ptge	8119.0	50.180442	22.803515	0.0000	41.6670	50.00000	60.0390	100.000	-0.230105	0.856235	50.00000	1.000000e+02
AgricultureUnemploymentPtge	8119.0	8.400982	12.958405	0.0000	0.0000	3.49300	11.7325	100.000	3.229281	15.576107	3.49300	1.000000e+02
IndustryUnemploymentPtge	8119.0	10.007836	12.528441	0.0000	0.0000	7.14300	14.2860	100.000	3.089803	16.050513	7.14300	1.000000e+02
ConstructionUnemploymentPtge	8119.0	10.837496	13.281177	0.0000	0.0000	8.33300	14.2860	100.000	3.094081	14.624696	8.33300	1.000000e+02
ServicesUnemploymentPtge	8119.0	58.649705	24.259562	0.0000	50.0000	62.01800	72.1230	100.000	-0.805963	0.800897	62.01800	1.000000e+02
totalEmpresas	8114.0	398.603032	4219.366083	0.0000	7.0000	30.00000	147.0000	299397.000	53.704675	3474.993974	30.00000	2.993970e+05
Industria	7931.0	23.419367	158.610811	0.0000	0.0000	0.00000	14.0000	10521.000	44.274183	2644.341401	0.00000	1.052100e+04
Construcción	7980.0	48.878321	421.863266	0.0000	0.0000	0.00000	25.0000	30343.000	52.575695	3506.586408	0.00000	3.034300e+04
ComercTTEHosteleria	8110.0	146.735265	1233.023418	0.0000	0.0000	0.00000	65.0000	80856.000	45.414126	2649.231539	0.00000	8.085600e+04
Servicios	8057.0	172.149684	2446.812300	0.0000	0.0000	0.00000	40.0000	177677.000	57.504563	3834.075292	0.00000	1.776770e+05
inmuebles	7981.0	3246.160256	24314.710959	6.0000	180.0000	486.00000	1589.0000	1615548.000	44.549632	2645.967366	486.00000	1.615542e+06
Pob2010	8112.0	5795.811637	47535.678654	5.0000	177.7500	582.00000	2483.0000	3273049.000	47.165520	2942.100273	582.00000	3.273044e+06
SUPERFICIE	8110.0	6214.695257	9218.194603	2.5784	1839.1918	3487.73745	6893.8778	175022.910	6.073564	62.340157	3487.73745	1.750203e+05
PobChange_pct	8112.0	-4.897406	10.383417	-52.2700	-10.4000	-4.96000	0.0925	138.460	1.505331	15.099319	-4.96000	1.907300e+02
PersonasInmuble	7981.0	1.296009	0.566620	0.1100	0.8500	1.25000	1.7300	3.330	0.264212	-0.633310	1.25000	3.220000e+00
Explotaciones	8119.0	2447.204582	15062.738051	1.0000	22.0000	52.00000	137.0000	99999.000	6.322101	37.987694	52.00000	9.999800e+04

Imagen 4: Tabla descriptiva para variables continuas

Donde se lograron observar variables que tenían métricas muy extrañas para lo que deberían representar, tales como: *ForeignsPtge* con porcentaje negativo, *SameComAutonPtge* con porcentaje superior a 100, además de variables como *Population*, *TotalCensus*, *TotalEmpresas*, *Industria* entre otras que poseían un rango muy exagerado y revisando los rangos de 25%, 50% y 75% se puede dar cuenta de que no debería ser tan grande.

Además con la tabla anterior y ejecutando *datos[variables].isna().sum()* lo que expondrá cual es la cantidad de nulos que se tiene en la muestra.

Corrección de los errores detectados.

Cualquier dato erróneo o inconsistente identificado durante el análisis descriptivo será corregido de manera adecuada para garantizar la integridad de los datos utilizados en las siguientes etapas del análisis.

Por ello se partido realizando las transformaciones de todas las variables con elementos extraños o fuera del rango.

for x in categoricas:

datos[x] = datos[x].replace(['nan', '?'], np.nan)

*datos['SameComAutonPtge'] = [x if 0 <= x <= 100 else np.nan for x in
datos['SameComAutonPtge']]*

*datos['ForeignersPtge'] = [x if 0 <= x <= 100 else np.nan for x in
datos['ForeignersPtge']]*

*datos['PobChange_pct'] = [x if -100 <= x <= 100 else np.nan for x in
datos['PobChange_pct']]*

*datos['PersonasInmueble'] = [x if 0 <= x else np.nan for x in
datos['PersonasInmueble']]*

Luego se procedió a agrupar las variables que eran poco representativas para que pudieran ser consideradas por si solas, pero en otro subconjunto podrían seguir aportando.

*datos['ActividadPpal'] =
datos['ActividadPpal'].replace(['Construccion', 'Industria', 'Servicios'],
'Construccion/Industria/Servicios')*

*datos['CCAA'] = datos['CCAA'].replace(['Murcia', 'Ceuta', 'Melilla'],
'Murcia/Ceuta/Melilla')*

*datos['CodigoProvincia'] =
datos['CodigoProvincia'].replace(['35', '52', '51'], '35/52/51')*

Análisis de valores atípicos. Decisiones.

Se llevó a cabo una minuciosa evaluación de los valores atípicos presentes en el conjunto de datos, tomando decisiones fundamentadas sobre la estrategia a seguir para su gestión. La presencia de valores atípicos puede tener un impacto sustancial en los resultados, por lo que se abordaron con atención y justificación. Para llevar a cabo esta tarea, se utilizó la función *atipicosAmissing*, la cual identificó, para cada variable, los registros que no se encontraban dentro de los rangos definidos y los transformó en valores nulos, asegurando así la integridad de los datos.

```
resultados = {x: atipicosAmissing(datos_input[x])[1] /  
len(datos_input) for x in numericas_input}
```

resultados

for x in numericas_input:

```
datos_input[x] = atipicosAmissing(datos_input[x])[0]
```

En un paso posterior, se introdujo una nueva variable denominada *prop_missings* que tenía como finalidad determinar si algún registro presentaba más de la mitad de sus variables con valores nulos, con el propósito de identificar y potencialmente eliminar dichos registros.

```
datos_input['prop_missings'] = datos_input.isna().mean(axis = 1)
```

patron_perdidos(datos_input)

Matriz de correlación de valores ausentes	
Population	1.0
Otros_Pct	0.0501
WomanPopulationPtge	0.0202 (0.0)
SameComAutonPtge	0.0202 (0.0)
DifComAutonPtge	0.0202 (0.0)
AgricultureUnemploymentPtge	0.0202 (0.0)
ConstructionUnemploymentPtge	0.0202 (0.0)
Industria	0.0505 (0.0)
ComercTTEHosteleria	0.0505 (0.0)
inmuebles	0.0000 (0.0)
SUPERFICIE	0.0000 (0.0)
PobChange_pct	0.0000 (0.0)
Ejplotaciones	0.0000 (0.0)
Population	1.0
TotalCensus	0.0505 (0.0)
Otros_Pct	0.1965 (-0.0)
Age_19_65_pct	0.0202 (0.0)
ForeignersPtge	0.0202 (0.0)
SameComAutonPtge	0.0202 (0.0)
DifComAutonDiffProvPtge	0.0202 (0.0)
UnemployLess25_Ptge	0.0202 (0.0)
IndustryUnemploymentPtge	0.0202 (0.0)
ConstructionUnemploymentPtge	0.0202 (0.0)
Industria	0.0505 (0.0)
ComercTTEHosteleria	0.0505 (0.0)
Servicios	0.0505 (0.0)
inmuebles	0.0000 (0.0)
Pob2010	0.0505 (0.0)
SUPERFICIE	0.0000 (0.0)
Densidad	0.0000 (0.0)
PobChange_pct	0.0000 (0.0)
PersonasInmueble	0.0000 (0.0)
Ejplotaciones	0.0000 (0.0)

Imagen 5: Matriz correlación de valores ausentes.

Después de esta fase, se realizó una revisión de la matriz de correlación de valores faltantes con el fin de identificar la variable que presentó la mayor cantidad de datos faltantes entre todas las variables consideradas.

Análisis de valores perdidos. Imputaciones.

La presencia de valores perdidos se abordó mediante técnicas de imputación apropiadas para evitar la pérdida de información valiosa y garantizar la integridad del conjunto de datos, primero se buscó por variable cuantos elementos en nulo tenía.

# Muestra total de valores perdidos por cada variable	
✓ 0.0s	
CodigoProvincia	0
CCAA	0
Population	806
TotalCensus	781
Izda_Pct	0
Dcha_Pct	0
Otros_Pct	845
Izquierda	0
Derecha	0
Age_0-4_Ptge	0
Age_under19_Ptge	0
Age_19_65_pct	24
Age_over65_pct	0
WomanPopulationPtge	21
ForeignersPtge	653
SameComAutonPtge	3
SameComAutonDiffProvPtge	165
DifComAutonPtge	40
UnemployLess25_Ptge	26
Unemploy25_40_Ptge	0
UnemployMore40_Ptge	0
AgricultureUnemploymentPtge	162
IndustryUnemploymentPtge	48
ConstructionUnemploymentPtge	53
ServicesUnemploymentPtge	0
...	
Densidad	92
PobChange_pct	75
PersonasInmueble	138
Ejplotaciones	559
dtype:	int64

Imagen 6: Valores en null por cada variable

Con la variable *prop_missing* se buscó si existía algún registro con más del 50% de sus variables en nulo, los datos no presentaron ningún registro así por lo que no se borró ningún valor, el código utilizado fue el siguiente:

```
eliminar = datos_input['prop_missings'] > 0.5

datos_input = datos_input[~eliminar]

varObjBin = varObjBin[~eliminar]

varObjCont = varObjCont[~eliminar]

eliminar = [prop_missingsVars.index[x] for x in
range(len(prop_missingsVars)) if prop_missingsVars[x] > 0.5]

datos_input = datos_input.drop(eliminar, axis = 1)
```

Luego de esto se procedió a la asignación de forma aleatoria de los valores siguiendo un patrón, esto se realizó con los siguientes comandos:

```
for x in numericas_input:

    datos_input[x] = ImputacionCuant(datos_input[x], 'aleatorio')

for x in categoricas_input:

    datos_input[x] = ImputacionCuali(datos_input[x], 'aleatorio')
```

Al finalizar esto se volvió a revisar por variable cuantos elementos en nulo tenía para constatar que ya no existiera ninguno.

# Reviso que no queden datos missings	
datos_input.isna().sum()	
✓ 0.0s	
CodigoProvincia	0
CCAA	0
Population	0
TotalCensus	0
Izda_Pct	0
Dcha_Pct	0
Otros_Pct	0
Izquierda	0
Derecha	0
Age_0-4_Ptge	0
Age_under19_Ptge	0
Age_19-65_pct	0
Age_over65_pct	0
WomanPopulationPtge	0
ForeignersPtge	0
SameComAutonPtge	0
SameComAutonDiffProvPtge	0
DifComAutonPtge	0
UnemployLess25_Ptge	0
Unemploy25_40_Ptge	0
UnemployMore40_Ptge	0
AgricultureUnemploymentPtge	0
IndustryUnemploymentPtge	0
ConstructionUnemploymentPtge	0
ServicesUnemploymentPtge	0
...	
PobChange_pct	0
PersonasInmueble	0
Explotaciones	0
prop_missings	0
dtype: int64	

Imagen 7: Valores en null por cada variable

1. Transformaciones de variables y relaciones con las variables objetivo.

Se realizaron transformaciones de las variables con la función *Trans_Auto*, lo cual arrojo una serie nueva de variables que pasaron por todos los procesos posteriores a este, como serian la matriz de correlación, eliminación de variables muy correlacionadas, y con estas implementar los modelos de regresión lineal y logística.

Pero debido a que la transformación de estas variables no arrojo un cambio muy significativo al momento de las regresiones lineales se procedió a dejar las variables sin transformar. Los comandos que se utilizaron para la transformación de las variables fueron:

```
input_cont = pd.concat([datos_input,
Transf_Auto(datos_input[numericas], varObjCont)], axis = 1)
```

```
input_bin = pd.concat([datos_input,
Transf_Auto(datos_input[numericas], varObjBin)], axis = 1)
```

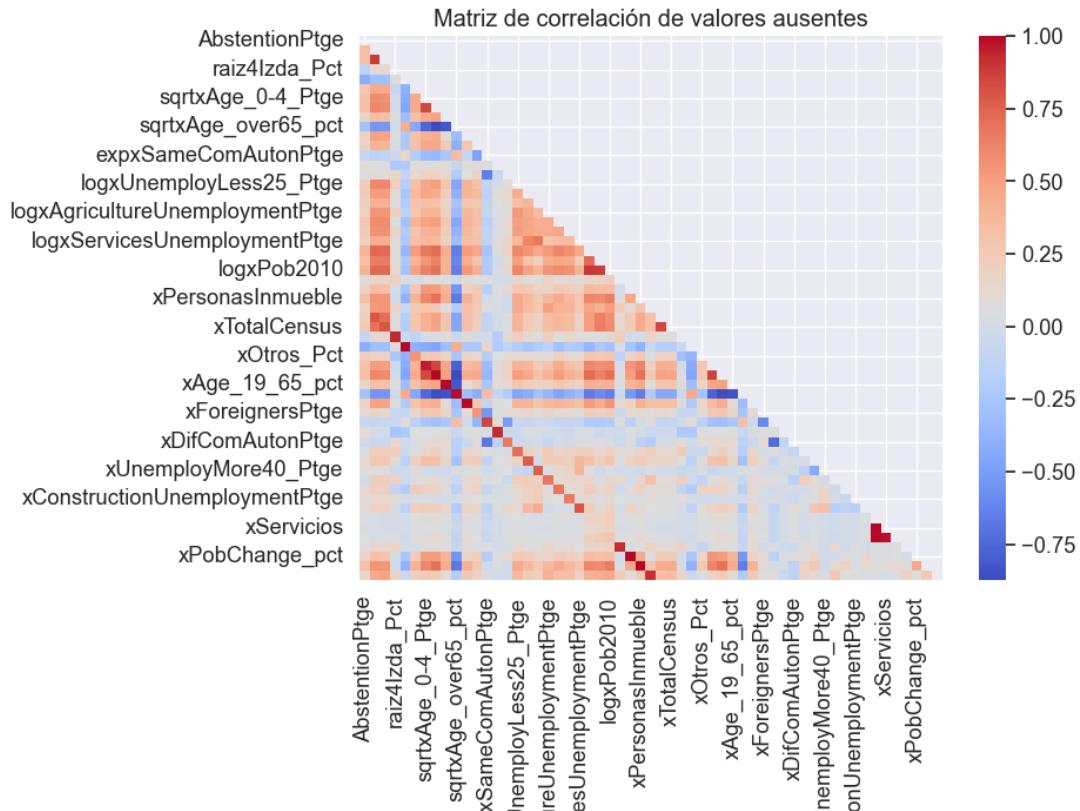


Imagen 8: Matriz de correlación de variables transformadas

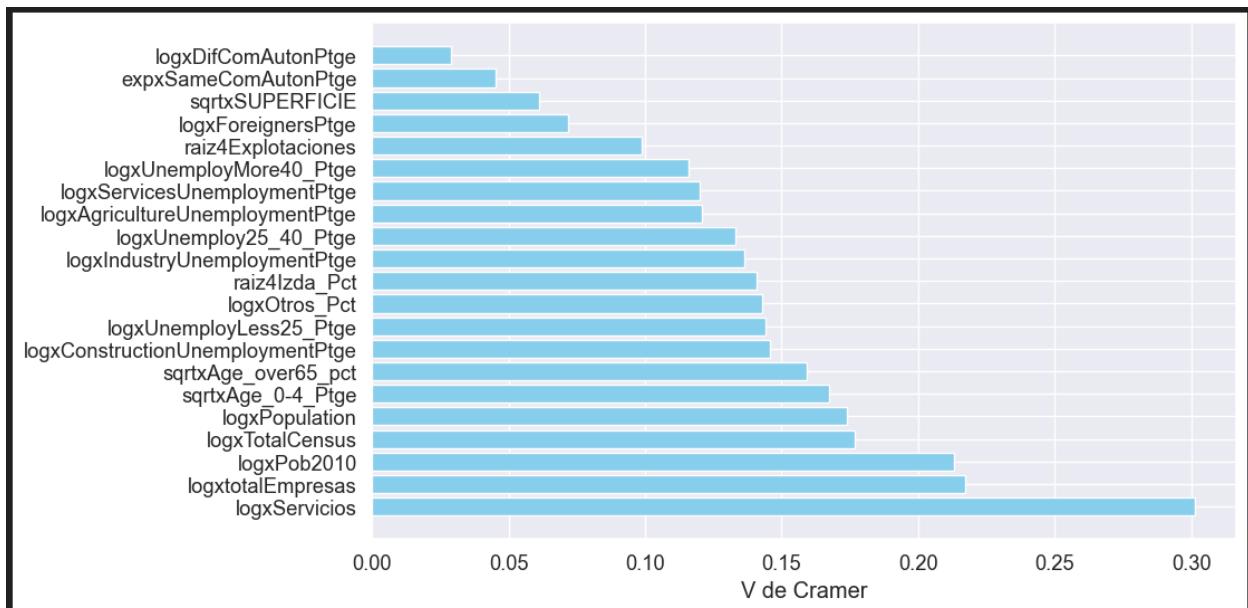


Imagen 9: V de Cramer de variables transformadas contra la variable continua

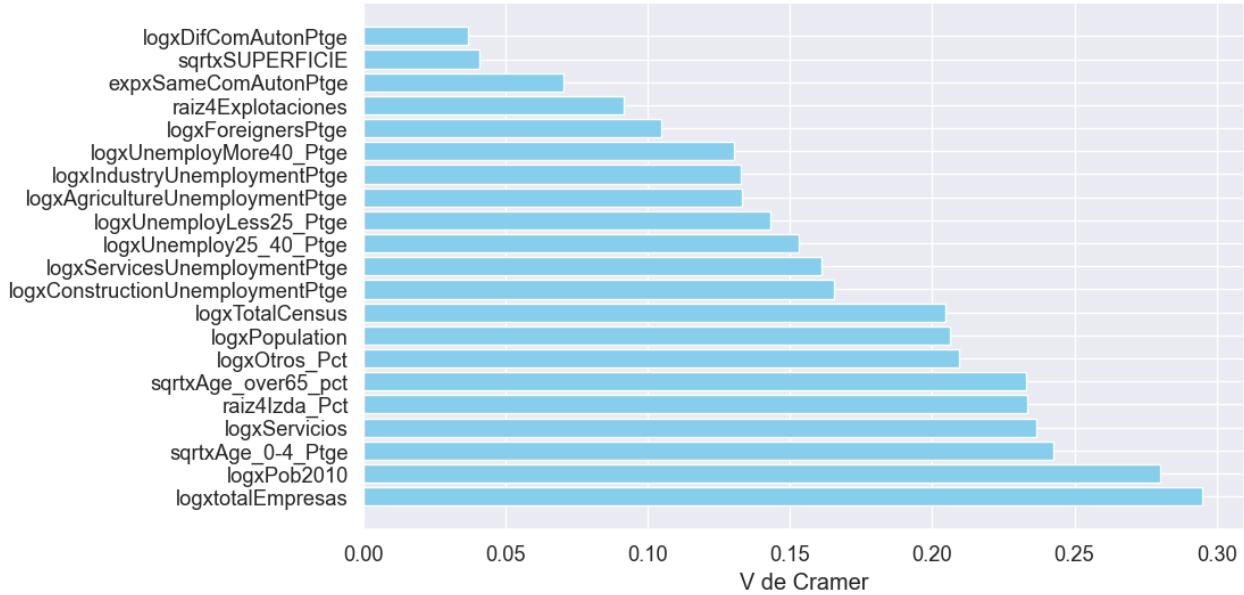


Imagen 9: *V de Cramer de variables transformadas contra la variable dicotómica*

8. Detección de las relaciones entre las variables input y objetivo.

Esta etapa se centró en detectar las relaciones significativas entre las variables de entrada. Se utilizaron técnicas estadísticas para identificar patrones y tendencias como es *V de Cramer*. Para empezar, se seleccionaron las variables objetivo que se iban a tomar, en este caso *AbstentionPtge* y *AbstencionAlta* y se eliminaron del conjunto de datos, además de la variable que actúa como índice, en este caso *Name*.

```

varObjCont = datos['AbstentionPtge']

varObjBin = datos['AbstencionAlta']

datos_input = datos.drop(['AbstentionPtge', 'AbstencionAlta','Name'],
axis = 1)

```

A continuación, se procedió a evaluar el nivel de correlación entre las variables mediante una matriz de correlaciones. En esta matriz, se identificaron las relaciones más fuertes mediante un mayor porcentaje de correlación. Es importante destacar que, en ocasiones, la fuerte correlación entre dos variables dentro del mismo conjunto puede no ser tan relevante para el análisis.

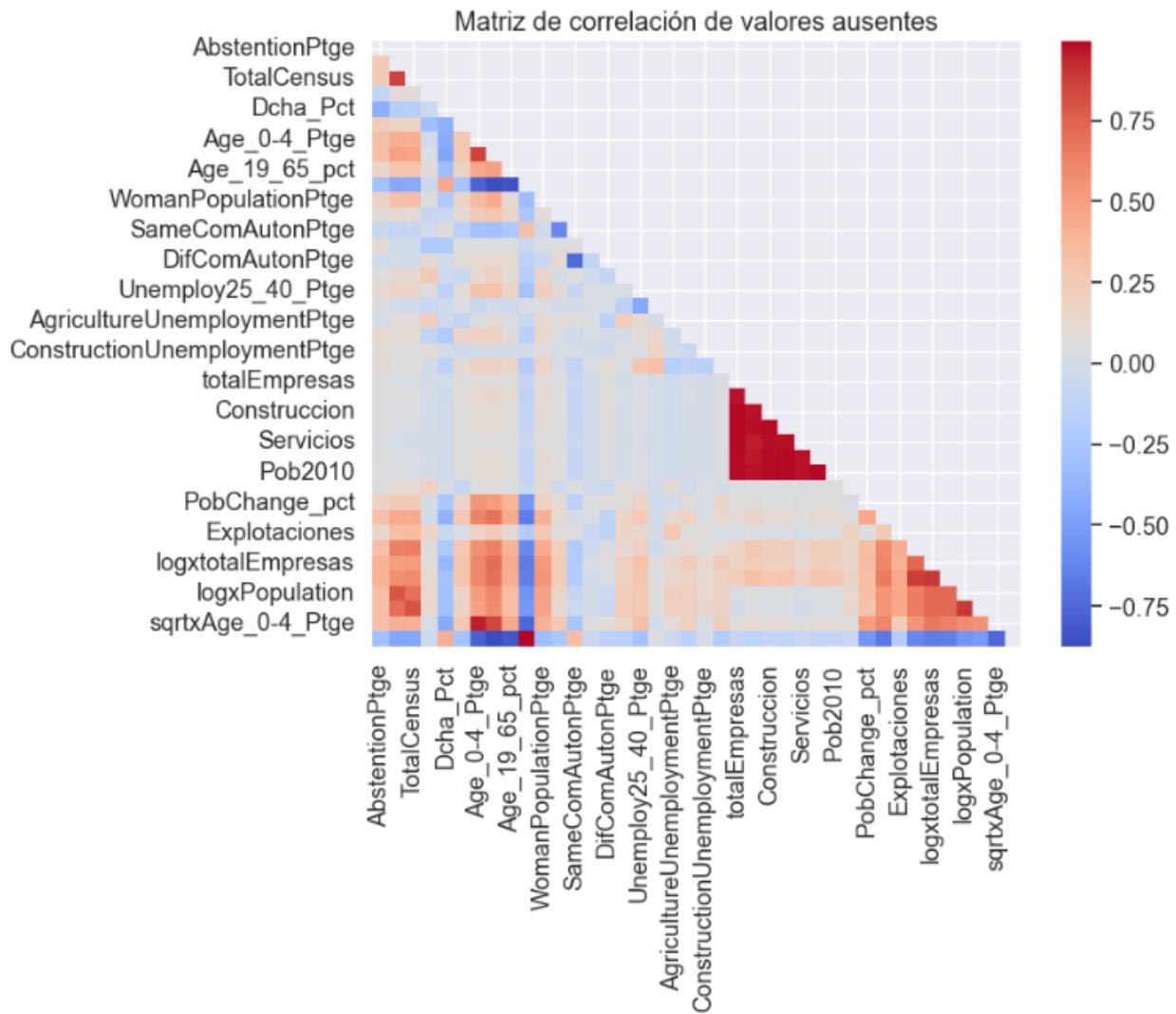


Imagen 10: Matriz de correlación de variables y variables transformadas

Luego se procedió a realizar un *V de Cramer* del conjunto de datos con las variables objetivo para poder revisar el nivel de importancia de cada una de estas.

graficoVcramer(datos_input, varObjBin)

graficoVcramer(datos_input, varObjCont)

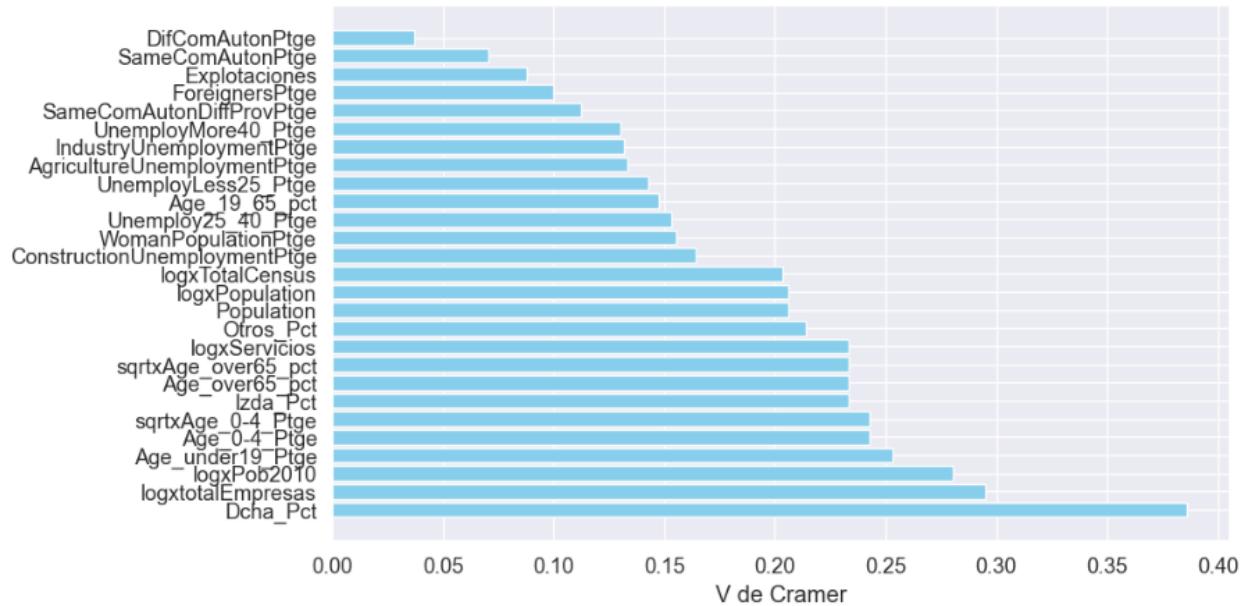


Imagen 11: V de Cramer de variables input contra variable objetivo dicotómica

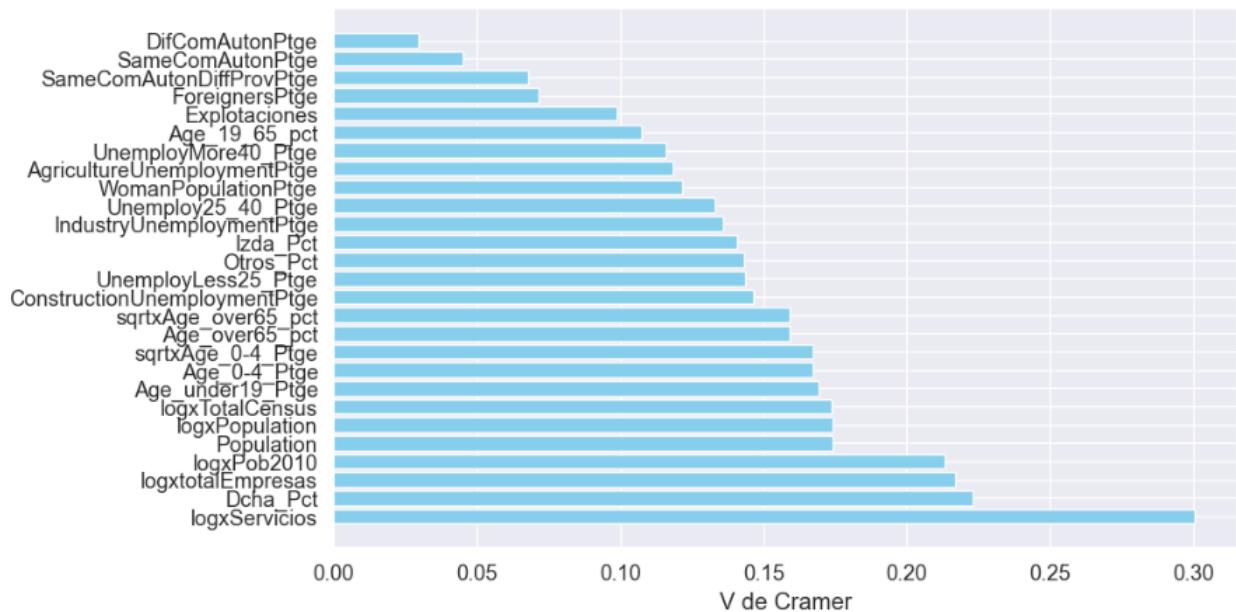


Imagen 12: V de Cramer de variables input contra variable objetivo continua

Tambien se analizaron las variables mas influyentes y menos influyentes, con el fin de ver graficamente que importancia tienen estas con la variable objetivo

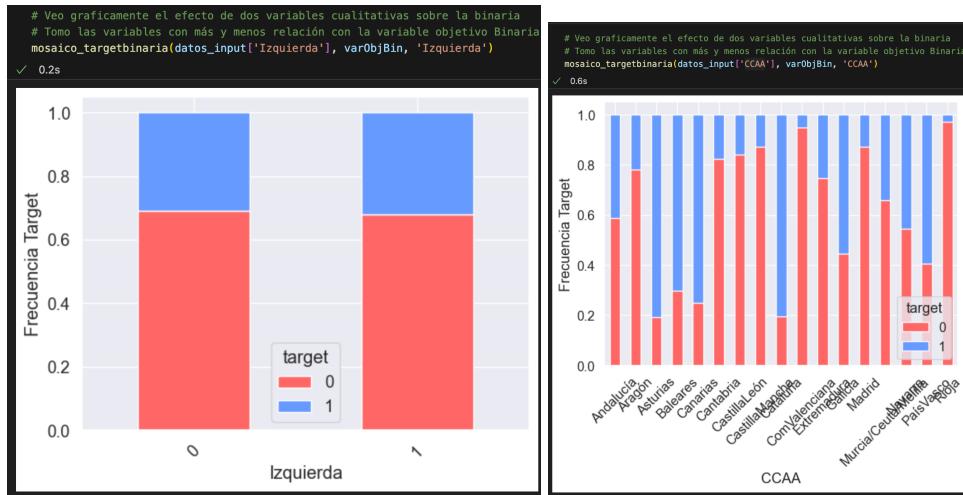


Imagen 13: Graficas de barra de varianza de valores de variable input contra variable dicotómica

En este paso se realizo por cada variable significativa una representacion grafica de los valores distribuidos segun el valor de la variable dicotomica, de esta forma se asegura que los datos esten bien.

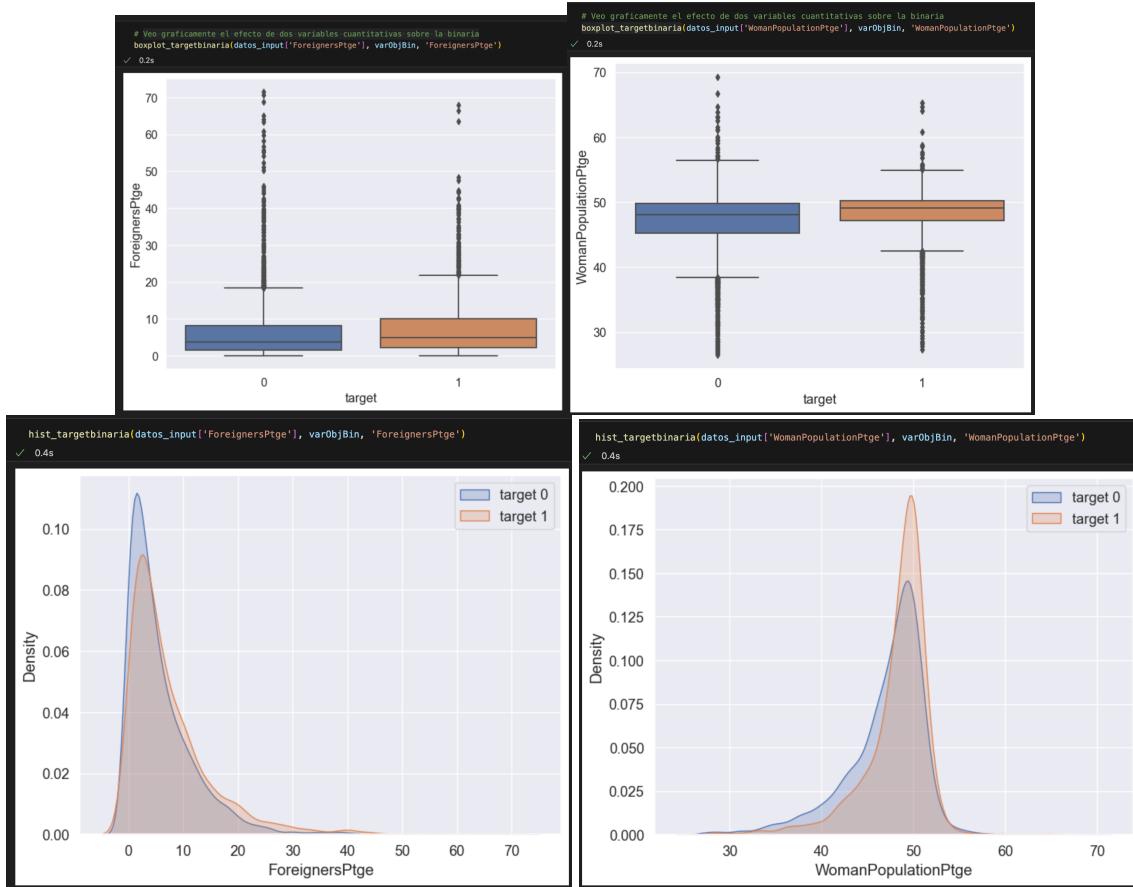


Imagen 14: Graficas de varianza de valores de variable input contra variable dicotómica

Para finalizar se borran las variables con más correlación, dejando una de ellas que sea la representante del grupo eliminado.

```
datos_input =  
datos_input.drop(['Industria','ComercTTEHosteleria','Construccion'], axis = 1)
```

Construcción del modelo de regresión lineal.

Se construye un modelo de regresión lineal que permitió analizar la influencia de las variables de entrada en los resultados electorales. Se llevó a cabo la selección de variables, tanto de forma clásica como aleatoria, y se determinará el mejor modelo. Los coeficientes de las variables serán interpretados y se justificará por qué se considera el mejor modelo, además de medir su calidad.

Selección de variables clásica

Se procedió a realizar las diferentes generaciones de modelos a través de la selección de variables de forma clásica con *Stepwise* y *Backward* y *AIC* y *BIC*, agregando también interacciones únicas dando como resultado los siguientes resultados

Método	R ²	Cantidad de variables
modeloStepAIC	0.4471232918243764	84
modeloBackAIC	0.4479882838219581	90
modeloStepBIC	0.4458510926892616	59
modeloBackBIC	0.4458510926892616	59
modeloStepAIC_int	0.479	147
modeloStepBIC_int	0.4451136329252585	65

Tabla 2: Modelos clásicos con sus R² y cantidad de variables

Selección de variables aleatoria

Se realizó la selección de manera aleatoria de las variables, dando como resultado diferentes modelos de los cuales se resaltaron los 3 mejores, teniendo unos resultados:

Método	R ²	Cantidad de variables
Aleatoria 1	0.36025640430829675	124
Aleatoria 2	0.3603530006054655	126
Aleatoria 3	0.36064294292734755	127

Tabla 3: Modelos aleatorio con sus R² y cantidad de variables

Selección del modelo ganador

Se procedió a introducir todos los modelos con sus validaciones cruzadas en un *dataframe* donde se observaron los valores de estos:

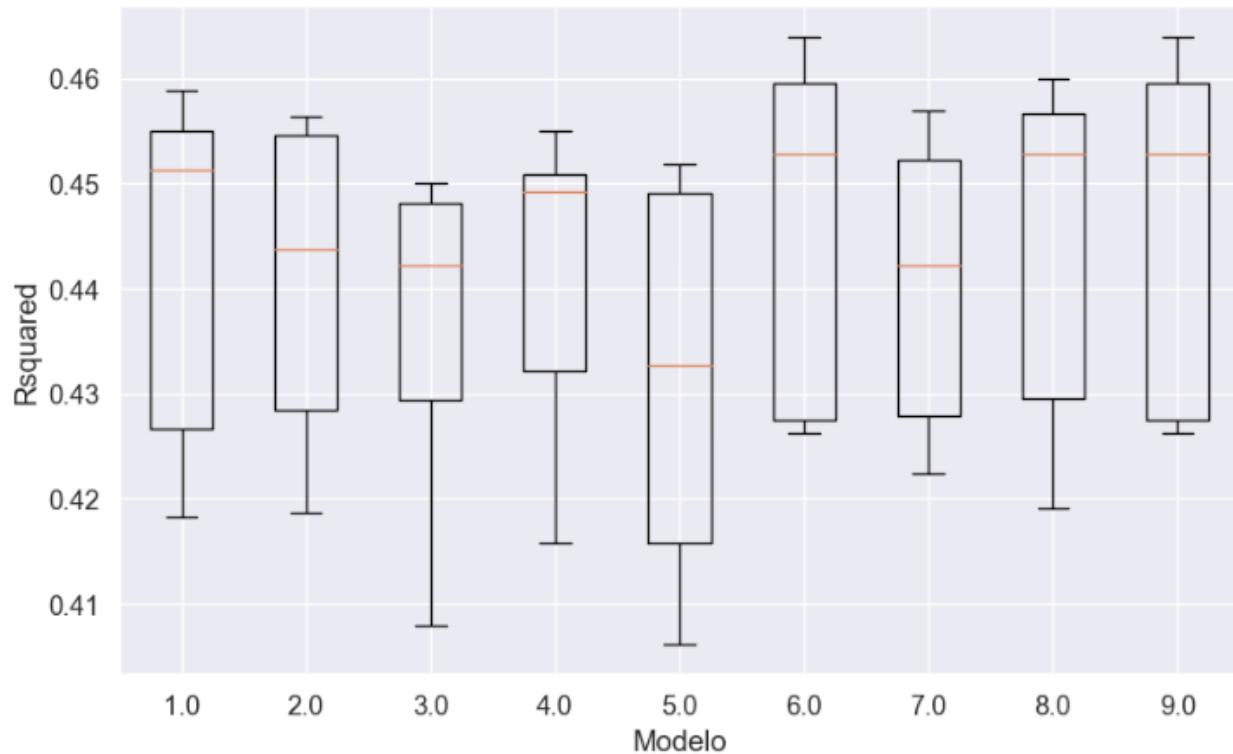


Imagen 15: R^2 de las validaciones cruzadas de cada modelo

Es evidente que el Modelo 6 destaca como el mejor intérprete en comparación con los demás, y es relevante destacar que solo utiliza 65 parámetros, lo que lo posiciona como uno de los modelos más eficientes en términos de requisitos de parámetros y R^2 .

Interpretación de los coeficientes de dos variables incluidas en el modelo

	coef	std err	t	P> t	[0.025	0.975]
const	51.3431	1.944	26.415	0.000	47.533	55.153
logxPob2010	2.1125	0.105	20.032	0.000	1.906	2.319
SameComAutonPtge	-0.0786	0.008	-9.829	0.000	-0.094	-0.063
Age_over65_pct	0.3920	0.051	7.637	0.000	0.291	0.493
sqrtxAge_over65_pct	-11.4224	1.885	-6.058	0.000	-15.118	-7.726
logxServicios	-0.8676	0.114	-7.612	0.000	-1.091	-0.644
SameComAutonDlffProvPtge	-0.0869	0.020	-4.415	0.000	-0.125	-0.048
Dcha_Pct	-0.0883	0.011	-7.994	0.000	-0.110	-0.067
CodigoProvincia_10	2.2931	1.220	1.879	0.060	-0.099	4.685

Imagen 16: Resumen del modelo de regresión lineal seleccionado

En la imagen 16 podemos observar las diferentes variables del modelo y como estas afectan a este con sus respectivos coeficientes, para esta interpretación tomaremos dos variables, una continua y otra binaria:

- Binaria: La variable objetivo que se observó fue *CodigoProvincia_10*, donde hay que destacar que la variable de referencia fue *CodigoProvincia_0* ya que era una variable categórica y al momento de realizar el modelo se transformó en una variable dummy, y esto indica que cada cambio en la variable objetivo va a afectar la variable referente en un 2.2931 frente a la variable *CodigoProvincia_0*
- Continua: se puede observar la variable *logxPob2010* tiene un coeficiente de 2.1125 en donde por cada punto de cambio en la variable objetivo *AbstentionPtge* esta aumentará un 2.1125

Justificar porqué es el mejor modelo y medir la calidad del mismo

El Modelo 6, derivado del proceso de selección de características utilizando el método *Stepwise* con el criterio de información bayesiano (BIC) y múltiples iteraciones, se destaca como la elección preferida. Esto se debe a que presenta un porcentaje de R² ligeramente superior en comparación con los demás modelos, con un 0.445, superando en hasta 3 puntos la mediana de los resultados. Además, es notable que este modelo requiere la menor cantidad de variables, con un total de solo 65, lo que lo convierte en una opción especialmente eficiente (Ver imagen 15).

Cuando se evalúa un modelo de regresión, hay que comprender la relevancia de las variables que lo componen. Una estrategia útil para lograr esto implica evaluar cómo el modelo se ve afectado al eliminar cada variable de manera individual. De este modo, las variables más influyentes provocarán una degradación significativa en el rendimiento del modelo si se eliminan, mientras que las menos relevantes apenas tendrán un impacto en la calidad global del mismo. Para obtener estos insights, se emplea la función *modelEffectSizes*. Esta función calcula la disminución en el coeficiente de determinación (R²) al suprimir variables específicas y presenta esta información de manera visual en un gráfico informativo.

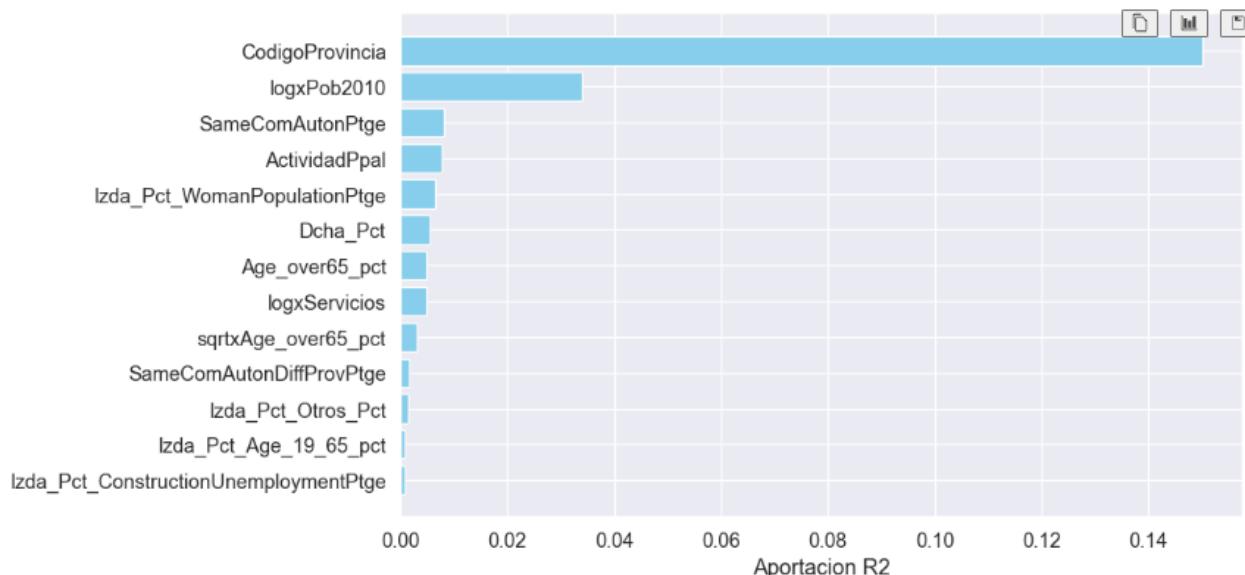


Imagen 17: Aporte de cada variable al R²

Construcción del modelo de regresión logística.

En esta etapa, se construirá un modelo de regresión logística para analizar las relaciones entre las variables de entrada y las preferencias de voto de izquierda y derecha. Se llevará a cabo la selección de variables, se determinará el modelo ganador y se establecerá el punto de corte óptimo. Los coeficientes de las variables serán interpretados y se justificará por qué se considera el mejor modelo, además de medir su calidad.

A lo largo de este informe, se proporcionarán detalles específicos sobre cada una de estas etapas, incluyendo resultados, gráficos y análisis detallados. El objetivo final es proporcionar una comprensión sólida de cómo los factores socioeconómicos influyen en los votos de izquierda y derecha en las elecciones españolas y, a partir de ello, contribuir a una toma de decisiones más informada en el ámbito político.

Selección de variables clásica

Se procedió a realizar las diferentes generaciones de modelos a través de la selección de variables de forma clásica con *Stepwise* y *Backward* y *AIC* y *BIC*, agregando también interacciones únicas dando como resultado los siguientes resultados

Método	R ²	Cantidad de variables
modeloStepAIC	0.3061886353550778	74
modeloBackAIC	0.3050675164984846	58
modeloStepBIC	0.3061886353550778	74
modeloBackBIC	0.3050675164984846	58
modeloStepAIC_int	0.3138560145913579	123
modeloStepBIC_int	0.3138560145913579	123

Tabla 4: Modelos clásicos con sus R² y cantidad de variables

Selección de variables aleatoria

Se realizó la selección de manera aleatoria de las variables, dando como resultado diferentes modelos de los cuales se resaltaron los 3 mejores, teniendo unos resultados:

Método	R ²	Cantidad de variables
Aleatoria 1	0.3145855958043553	123
Aleatoria 2	0.31492154071921297	125
Aleatoria 3	0.31550125182359234	126

Tabla 5: Modelos aleatorio con sus R² y cantidad de variables

Selección del modelo ganador

Se procedió a introducir todos los modelos con sus validaciones cruzadas en un *dataframe* donde se observaron los valores de estos:

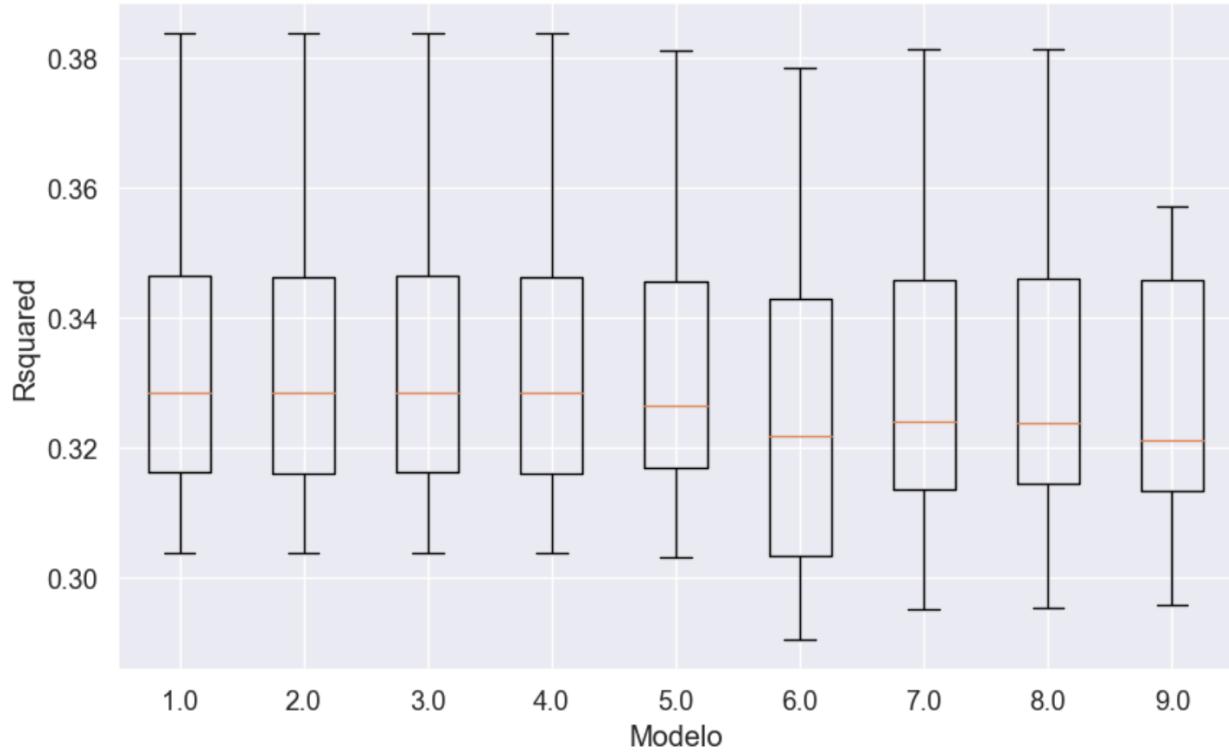


Imagen 18: R^2 de las validaciones cruzadas de cada modelo

No se observan diferencias muy marcadas en cuanto a los R^2 , por lo que se procedió a realizar un filtro de los modelos por la cantidad de parámetros que utiliza, es por eso que el modelo 2 y 4 son los mejores con tan solo 58 parámetros, pero se seleccionó el segundo, el obtenido con *Backward*.

Determinar el punto de corte óptimo

Para buscar el punto de corte óptimo del modelo, se graficó la curva ROC, la cual exhibe un valor sólido de 0.8483405137936549 bajo la curva, lo que indica un rendimiento favorable en la capacidad de discriminación del modelo. Este valor cercano a 1 sugiere que el modelo tiene una buena capacidad para distinguir entre clases positivas y negativas, lo que es un indicador positivo de su precisión en la clasificación. En términos simples, la curva ROC confirma que el modelo presenta un desempeño robusto y confiable en su capacidad para realizar predicciones acertadas.

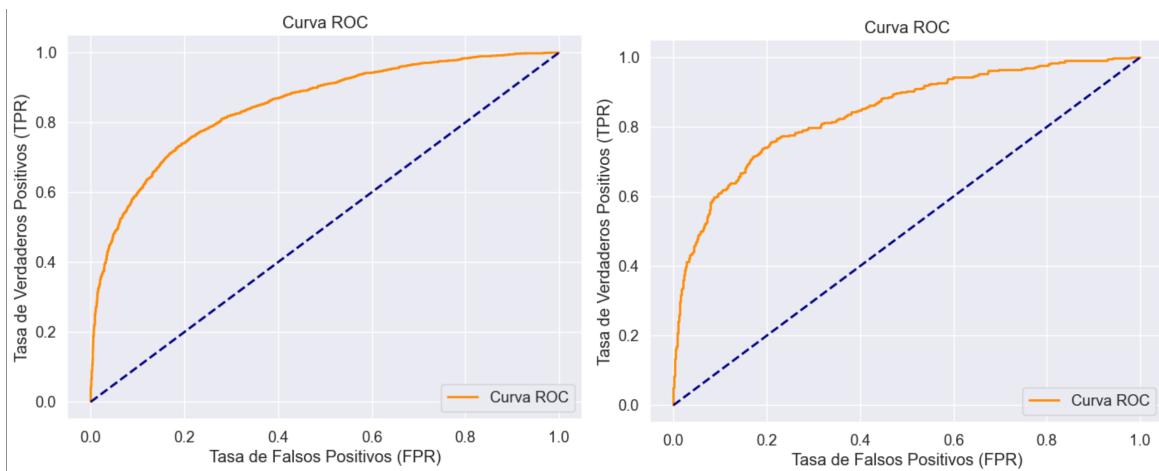


Imagen 19: Gráfica de la curva ROC

Adicionalmente se utilizaron dos métricas: la gráfica de Youden y la gráfica accuracy. La gráfica de Youden permitió encontrar el punto en la curva ROC donde se maximiza la suma de la sensibilidad y la especificidad menos uno. Este punto representa un equilibrio óptimo entre la capacidad del modelo para identificar verdaderos positivos y minimizar los falsos positivos. Por otro lado, la gráfica de accuracy mostró una medida general de la exactitud del modelo en la clasificación.

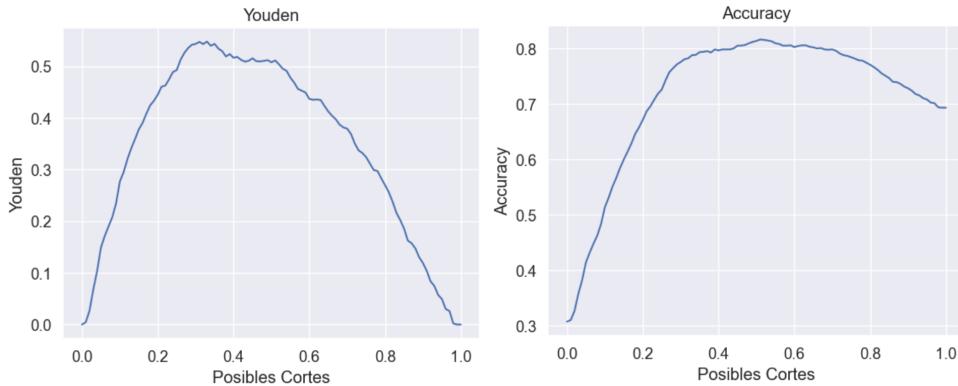


Imagen 20: Gráficas de Youden y Accuracy

```

# Calculamos la diferencia de las medidas de calidad entre train y test
[sensEspCorte(modeloStepBIC['Modelo'], x_train, y_train, 0.51, modeloStepBIC['Variables']['cont'], modeloStepBIC['Variables']['categ'])]

```

0.0s

PtoCorte	Accuracy	Sensitivity	Specificity	PosPredValue	NegPredValue
0 0.51	0.806313	0.565025	0.916013	0.753614	0.822441,
PtoCorte	Accuracy	Sensitivity	Specificity	PosPredValue	NegPredValue
0 0.51	0.816502	0.598394	0.912966	0.752525	0.837134]


```

t, y_test, 0.33, modeloStepBIC['Variables']['cont'], modeloStepBIC['Variables']['categ'], modeloStepBIC['Variables']['inter'])

```

0.3s

PtoCorte	Accuracy	Sensitivity	Specificity	PosPredValue	NegPredValue
0 0.33	0.792764	0.71133	0.829787	0.655172	0.863435,
PtoCorte	Accuracy	Sensitivity	Specificity	PosPredValue	NegPredValue
0 0.33	0.788177	0.736948	0.810835	0.632759	0.874521]

Imagen 21: Puntos de corte seleccionados en 0.33 y 0.51

Se eligió el punto de corte con la mayor sensibilidad, que corresponde al valor de 0.33, ya que alcanza una sensibilidad del 0.71, destacando así su capacidad para detectar de manera efectiva los casos positivos.

Interpretación de los coeficientes de dos variables incluidas en el modelo

...	'Contrastes':	Variable	Estimate	Std. Error	z value	p value \
0	(Intercept)	6.695414	3.500836e+06	1.912519e-06	0.999998	
1	logxPob2010	0.555942	5.754971e-02	9.660204e+00	0.000000	
2	SameComAutonPtge	-0.029817	3.728004e-03	-7.998018e+00	0.000000	
3	Age_over65_pct	0.089013	2.353340e-02	3.782421e+00	0.000157	
4	WomanPopulationPtge	-0.052212	1.107527e-02	-4.714254e+00	0.000002	
..
70	CCAA_Madrid	-0.909942	1.637292e+07	-5.557602e-08	1.000000	

Imagen 22: Resumen del modelo de regresión logística seleccionado

En la imagen 22 podemos observar las diferentes variables del modelo y como estas afectan a este con sus respectivos coeficientes, para esta interpretación tomaremos dos variables, una continua y otra binaria:

- Binaria: La variable objetivo que se observó fue *CCAA_Madrid*, donde hay que destacar que la variable de referencia fue *CCAA_CastillaLeon* ya que era una variable categórica y al momento de realizar el modelo se transformó en una variable dummy, y esto indica que cada cambio en la variable objetivo va a afectar la variable referente en un *CCAA_CastillaLeon* frente a la variable *CCAA_Madrid*.
- Continua: Se puede observar la variable *logxPob2010* tiene un coeficiente de 0.555942 en donde por cada punto de cambio en la variable objetivo *AbstencionAlta* esta aumentara un 0.555942.

Justificar porqué es el mejor modelo y medir la calidad del mismo

El Modelo 2, resultado del proceso de selección de características mediante el método *Backward*, se destaca por mostrar un coeficiente de determinación (R^2) significativamente más elevado en comparación con otros modelos, alcanzando un valor de 0.2952. Este R^2 superior demuestra su capacidad para explicar la variabilidad en los datos de manera efectiva. Además, es relevante destacar que este modelo requiere un número relativamente reducido de parámetros, con un total de tan solo 58, lo que lo posiciona como una opción eficiente para el análisis de datos (Consultar Figura 15).

De igual forma que se justificó en la regresión lineal, se procedió a ejecutar el *modelEffectSizes*, que mostró la disminución del R^2 al ir suprimiendo variables, de esta forma se analizó como estas afectaban al modelo.

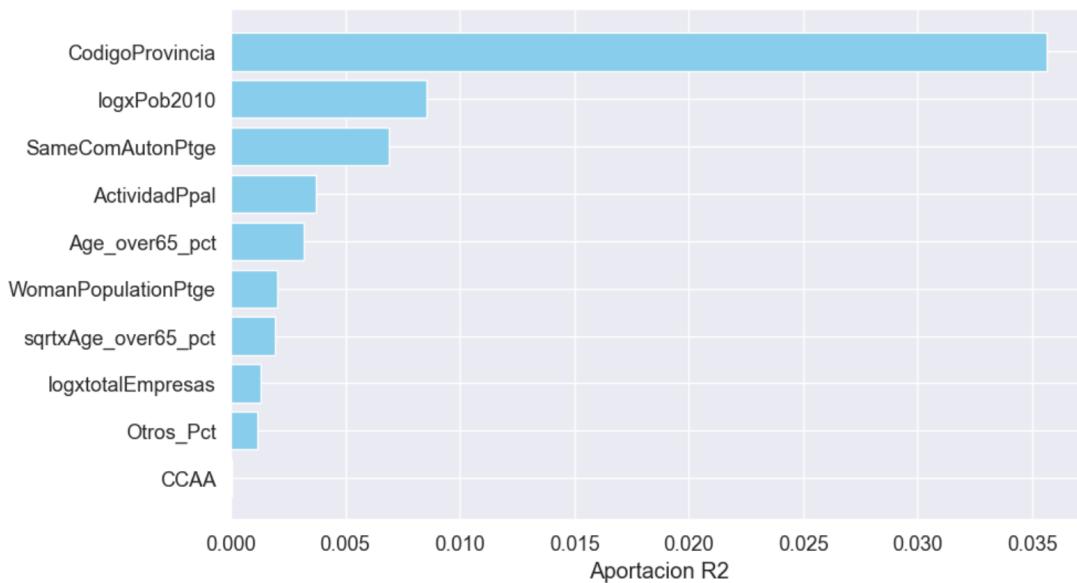


Imagen 23: Aporte de cada variable al R^2

Conclusión

En conclusión, al realizar un análisis comparativo entre los modelos de regresión logística y regresión lineal, se obtuvieron resultados destacados que influyeron en la elección de los modelos ganadores.

En el caso de la regresión logística, el modelo *modeloBackAIC* se destacó como el ganador. Este modelo se destacó por ofrecer uno de los mejores R^2 y por utilizar una cantidad reducida de variables. Esta combinación de un alto rendimiento predictivo y una eficiencia en la selección de características lo posiciona como la elección preferida en este enfoque.

En cuanto a la regresión lineal, el modelo *modeloStepBIC_int* se destacó como el ganador debido a su capacidad para utilizar un conjunto de solo 65 variables, lo que lo convierte en una opción eficiente para el análisis. Además, en el análisis de la curva ROC, se observó que el área bajo la curva alcanzó un impresionante valor de 0.8483405137936549, lo que sugiere que el modelo tiene una sólida capacidad de discriminación. Los puntos de corte identificados, especialmente el punto de corte en 0.51, demostraron

una mejor sensibilidad, lo que significa que el modelo es capaz de identificar de manera efectiva los casos positivos.

En resumen, los modelos *modeloBackAIC* y *modeloStepBIC_int* emergieron como los ganadores en sus respectivos enfoques, destacándose por su rendimiento predictivo, eficiencia en la selección de variables y capacidad de discriminación en el caso de la regresión logística y la regresión lineal, respectivamente. Estos resultados subrayan la importancia de seleccionar modelos adecuados para abordar preguntas específicas en el análisis de datos.