**Logotipo

El contenido generado por IA puede ser incorrecto.**

PREDICCIÓN DE LA ABSTENCIÓN ELECTORAL

EN LOS MUNICIPIOS DE ESPAÑA (2019)  
Ejercicio de Evaluación.

**Nombre:** Gerson Castillo  
**DNI:** 54968597T **Fecha:** 20/04/2025

**Asignatura:** Minería de Datos y

Modelización Predictiva

**Profesor:** Dra. Rosa Espínola

Contenido

[1. Introducción 4](#_Toc197251164)

[2. DEPURACIÓN DE DATOS 5](#_Toc197251165)

[a) IMPORTACIÓN DEL CONJUNTO DE DATOS Y ASIGNACIÓN CORRECTA DE LOS TIPOS DE VARIABLES 5](#_Toc197251166)

[b) ANÁLISIS DESCRIPTIVO DEL CONJUNTO DE DATOS 6](#_Toc197251167)

[Dimensiones y estructura general 7](#_Toc197251168)

[Variables numéricas con valores no razonables 7](#_Toc197251169)

[Variables cualitativas con errores de codificación 7](#_Toc197251170)

[Análisis de asimetría y curtosis 8](#_Toc197251171)

[Variables con alta dispersión y valores extremos 8](#_Toc197251172)

[Valores atípicos previstos 8](#_Toc197251173)

[Estructura de valores perdidos preliminar 8](#_Toc197251174)

[c) CORRECCIÓN DE LOS ERRORES DETECTADOS 8](#_Toc197251175)

[Objetivo de la limpieza 8](#_Toc197251176)

[Corrección de errores numéricos 9](#_Toc197251177)

[Corrección de errores cualitativos 9](#_Toc197251178)

[Tratamiento de valores atípicos 10](#_Toc197251179)

[Eliminación de observaciones incompletas 10](#_Toc197251180)

[Resultado tras la limpieza 10](#_Toc197251181)

[3. CONSTRUCCIÓN DEL MODELO DE REGRESIÓN LINEAL 11](#_Toc197251182)

[a) Objetivo 11](#_Toc197251183)

[b) Selección Clásica de Variables 11](#_Toc197251184)

[Resultados obtenidos 12](#_Toc197251185)

[Interpretación de los resultados 12](#_Toc197251186)

[c) Selección Aleatoria de Variables 13](#_Toc197251187)

[Justificación metodológica 13](#_Toc197251188)

[Resultados obtenidos 14](#_Toc197251189)

[Interpretación 14](#_Toc197251190)

[d) Selección del Modelo Ganador y Validación Cruzada 15](#_Toc197251191)

[Resultados obtenidos 16](#_Toc197251192)

[Interpretación del gráfico 16](#_Toc197251193)

[Conclusión 16](#_Toc197251194)

[e) Interpretación de los Coeficientes del Modelo Ganador 17](#_Toc197251195)

[Objetivo 17](#_Toc197251196)

[Variable continua: Age\_under19\_Ptge 17](#_Toc197251197)

[Variable categórica: CCAA\_Rioja 17](#_Toc197251198)

[Conclusión 18](#_Toc197251199)

[4. CONSTRUCCIÓN DEL MODELO DE REGRESIÓN LOGÍSTICA 18](#_Toc197251200)

[a) Objetivo 18](#_Toc197251201)

[Restricciones metodológicas 18](#_Toc197251202)

[Desarrollo 18](#_Toc197251203)

[Código base 19](#_Toc197251204)

[b) Selección Clásica de Variables (Regresión Logística) 19](#_Toc197251205)

[Justificación metodológica 19](#_Toc197251206)

[Resultados obtenidos 20](#_Toc197251207)

[Interpretación de los resultados 21](#_Toc197251208)

[Conclusión 21](#_Toc197251209)

[c) Selección Aleatoria de Variables (Regresión Logística) 21](#_Toc197251210)

[Justificación metodológica 21](#_Toc197251211)

[Parámetros utilizados 21](#_Toc197251212)

[Resultados obtenidos 22](#_Toc197251213)

[Análisis e interpretación 22](#_Toc197251214)

[d) Selección del Modelo Ganador y Validación Cruzada (Regresión Logística) 23](#_Toc197251215)

[Objetivo 23](#_Toc197251216)

[Justificación metodológica 23](#_Toc197251217)

[Selección de modelos a comparar 24](#_Toc197251218)

[Resultados 24](#_Toc197251219)

[Análisis del gráfico de caja y bigotes 24](#_Toc197251220)

[Síntesis 25](#_Toc197251221)

[e) Determinación del Punto de Corte Óptimo 25](#_Toc197251222)

[Resultado obtenido 25](#_Toc197251223)

[Interpretación 25](#_Toc197251224)

[f) Interpretación de Coeficientes del Modelo Ganador 27](#_Toc197251225)

[1. CCAA\_Cataluña 27](#_Toc197251226)

[2. WomanPopulationPtge 28](#_Toc197251227)

[Interpretación: 28](#_Toc197251228)

[6. CONCLUSIÓN FINAL 29](#_Toc197251229)

# 1. Introducción

El presente trabajo tiene como objetivo aplicar técnicas de minería de datos y modelización predictiva sobre un conjunto de datos reales proporcionado en el fichero **DatosEleccionesEspaña.xlsx**. Este conjunto contiene información demográfica y socioeconómica de los municipios de España junto con los resultados obtenidos en las elecciones municipales del año 2019.

En este ejercicio se han seleccionado dos variables objetivo para abordar dos modelos distintos:

| **Tipo de modelo** | **Variable objetivo elegida** | **Descripción** |
| --- | --- | --- |
| Regresión lineal | **AbstentionPtge** | Porcentaje de electores que no acudieron a votar |
| Regresión logística | **AbstencionAlta** | Indicador binario: 1 si AbstentionPtge > 30 %, 0 en otro caso |

Ambas variables están directamente relacionadas entre sí, permitiendo abordar el análisis tanto desde el punto de vista de la predicción de un porcentaje como desde la clasificación binaria de los municipios según un umbral de abstención.

Además de estas variables, el conjunto de datos contiene múltiples variables de entrada, incluyendo:

* variables demográficas (porcentaje de jóvenes, mayores, inmigración, densidad, etc.),
* características económicas (tasas de paro por grupo de edad, sectores económicos, número de empresas…),
* información geográfica y regional (CCAA, provincia…).

Estas variables serán utilizadas como variables explicativas (inputs) en los modelos.  
El resto de variables objetivo que no han sido seleccionadas serán eliminadas tal y como exige el enunciado de la tarea oficial.

# 2. DEPURACIÓN DE DATOS

## IMPORTACIÓN DEL CONJUNTO DE DATOS Y ASIGNACIÓN CORRECTA DE LOS TIPOS DE VARIABLES

En este apartado se importa el fichero **DatosEleccionesEspaña.xlsx**, se eliminan las variables objetivo que **no han sido seleccionadas** para esta práctica, y se corrige el tipo de algunas variables que han sido asignadas incorrectamente como numéricas cuando en realidad deben tratarse como cualitativas.

La variable objetivo continua seleccionada es AbstentionPtge, y la variable objetivo binaria es AbstencionAlta. Por tanto, se eliminan las variables: Izda\_Pct, Dcha\_Pct, Otros\_Pct, Izquierda y Derecha, tal y como se indica en el enunciado oficial de la tarea.

**Código:**

# Cargo las librerías necesarias

import os

import pandas as pd

# Cargo el conjunto de datos desde el archivo Excel

datos = pd.read\_excel(r"datos\DatosEleccionesEspaña.xlsx")

# Elimino las variables objetivo que no han sido seleccionadas para el análisis

variables\_no\_utilizadas = ['Izda\_Pct', 'Dcha\_Pct', 'Otros\_Pct', 'Izquierda', 'Derecha']

datos = datos.drop(columns=variables\_no\_utilizadas)

# Compruebo el tipo de datos asignado automáticamente a cada variable

datos.dtypes

Terminal:

Name object

CodigoProvincia int64

CCAA object

Population int64

TotalCensus int64

AbstentionPtge float64

AbstencionAlta int64

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

Construccion float64

ComercTTEHosteleria float64

Servicios float64

ActividadPpal object

inmuebles float64

Pob2010 float64

SUPERFICIE float64

Densidad object

PobChange\_pct float64

PersonasInmueble float64

Explotaciones int64

dtype: object

Como se puede observar, la variable CodigoProvincia ha sido asignada como numérica (int64), cuando en realidad es un **código categórico** y debe ser tratada como una variable cualitativa. En los ejemplos vistos en clase, estas variables se recodifican como object (string), utilizando astype(str).

**Código:**

# Variables numéricas que en realidad son cualitativas

numericasAcategoricas = ['CodigoProvincia']

# Se transforman a tipo string

for var in numericasAcategoricas:

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

# Verificación del cambio de tipo

datos[numericasAcategoricas].dtypes

Terminal:

CodigoProvincia object

dtype: object

Con las correcciones realizadas, el conjunto de datos datos queda correctamente preparado para continuar con el análisis descriptivo.

## ANÁLISIS DESCRIPTIVO DEL CONJUNTO DE DATOS

Número de observaciones, número y naturaleza de variables, datos erróneos, etc.

Este apartado tiene como objetivo realizar un análisis exhaustivo del conjunto de datos, previo al proceso de limpieza y tratamiento de valores perdidos.  
Se busca identificar:

* Variables mal tipadas.
* Errores de codificación en variables numéricas y cualitativas.
* Distribuciones anómalas mediante análisis de asimetría y curtosis.
* Presencia de valores atípicos esperados.
* Estructura preliminar de valores faltantes.

**Código:**

# Número de observaciones y variables

print("Número de observaciones:", datos.shape[0])

print("Número de variables:", datos.shape[1])

# Número de valores distintos y tipos por variable

cuentaDistintos(datos)  
  
# Frecuencias de variables cualitativas

analizar\_variables\_categoricas(datos)

# Cálculo de descriptivos extendidos para variables numéricas

numericas = datos.select\_dtypes(include='number').columns.tolist()

descriptivos\_num = datos[numericas].describe().T

# Añadimos más descriptivos a los anteriores

for num in numericas:

descriptivos\_num.loc[num, "Asimetria"] = datos[num].skew()

descriptivos\_num.loc[num, "Kurtosis"] = datos[num].kurtosis()

descriptivos\_num.loc[num, "Rango"] = np.ptp(datos[num].dropna().values)

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Terminal:

Número de observaciones: 8117

Número de variables: 36

### Dimensiones y estructura general

Tras la depuración inicial (eliminación de variables no seleccionadas y columnas vacías), el conjunto de datos presenta:

8.117 observaciones

36 variables explicativas

No obstante, no todas las variables contienen 8.117 valores válidos, debido a datos faltantes o registros mal codificados.

Ejemplos destacados:

| **Variable** | **Observaciones válidas** |
| --- | --- |
| Industria | 7.929 |
| Construccion | 7.978 |
| Servicios | 8.055 |
| inmuebles | 7.979 |

### Variables numéricas con valores no razonables

Se detectaron varios valores matemáticamente o conceptualmente imposibles, como:

* **Age\_over65\_pct**: mínimo **−18.05**. Imposible que un porcentaje poblacional sea negativo.
* **ForeignersPtge:** mínimo **−8.96**. Un porcentaje negativo no tiene sentido.
* **ServiciosUnemploymentPtge:** incluye valores como **−99**, indicador de codificación errónea.
* **Exploitaciones**: máximo 99.999, probablemente utilizado como marcador de valor perdido.
* **PersonasInmueble:** mínimo **0.11**, aunque matemáticamente posible, es extremadamente improbable.

### Variables cualitativas con errores de codificación

Mediante el análisis de frecuencias se detectaron errores relevantes:

* **Densidad** contiene la categoría **'?'**, no válida y que será considerada como valor perdido.
* **ActividadPpal** contiene categorías poco representadas como **Construccion** e **Industria**, que deberán ser agrupadas.
* La variable **Name** tiene un valor único por observación y será eliminada por no aportar valor explicativo.

### Análisis de asimetría y curtosis

Las medidas de asimetría y curtosis revelan graves problemas de distribución:

| **Variable** | **Asimetría** | **Curtosis** | **Interpretación** |
| --- | --- | --- | --- |
| Population | 46.04 | 2820.33 | Distribución extremadamente asimétrica y con outliers fuertes. |
| totalEmpresas | 53.71 | 3475.48 | Alta concentración de municipios con muy pocas empresas, con algunos valores extremos. |
| Servicios | 57.50 | 3833.61 | Estructura similar: concentración en municipios pequeños y unos pocos muy grandes. |
| SUPERFICIE | 6.07 | 62.33 | Dispersión significativa en el tamaño de municipios. |
| Exploitaciones | 6.32 | 37.97 | Alta concentración de valores bajos con outliers. |

### Variables con alta dispersión y valores extremos

El análisis de la dispersión de los datos muestra:

| **Variable** | **Máximo** | **Comentario técnico** |
| --- | --- | --- |
| inmuebles | 1.615.548 | Municipios con una cantidad desmesurada de inmuebles. |
| Pob2010 | 3.273.049 | Rango poblacional muy amplio entre municipios. |
| SUPERFICIE | 175.022 | Gran heterogeneidad en el tamaño de los municipios. |
| totalEmpresas | 299.397 | Fuerte concentración con presencia de municipios atípicos. |
| Servicios | 177.677 | Desviaciones extremas frente a los percentiles normales. |

Estos valores extremos serán objeto de un análisis específico de outliers.

### Valores atípicos previstos

Aunque no modificamos aún los datos:

* Se observa necesidad de tratar **outliers extremos** como missing.
* El tratamiento se realizará en la limpieza formal con la función **atipicosAmissing**.

### Estructura de valores perdidos preliminar

El análisis de valores faltantes indica:

* Variables con registros incompletos antes de la corrección (Industria, Construccion, Servicios, inmuebles).
* Tras la limpieza prevista, aumentará el número de valores perdidos.

Se utilizará la función **patron\_perdidos** para visualizar y analizar la estructura de missing tras la limpieza.

## CORRECCIÓN DE LOS ERRORES DETECTADOS

### Objetivo de la limpieza

El objetivo de este apartado es depurar el conjunto de datos eliminando inconsistencias, errores de codificación, valores atípicos y observaciones excesivamente incompletas.

Además, se optimizan las variables cualitativas agrupando categorías con baja representación.

**Código:**

# --- Corrección de errores numéricos y cualitativos ---

# Códigos numéricos erróneos a reemplazar

codigos\_erroneos\_numericos = [-99, 9999, 99999]

# Reemplazo en todas las variables numéricas

for col in datos.select\_dtypes(include=['float64', 'int64']).columns:

datos[col] = datos[col].replace(codigos\_erroneos\_numericos, np.nan)

# Reemplazo de todos los valores negativos en proporciones

variables\_con\_negativos = ['ForeignersPtge', 'Age\_over65\_pct']

for var in variables\_con\_negativos:

datos.loc[datos[var] < 0, var] = np.nan

# Reemplazo de errores cualitativos

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

# --- Tratamiento de valores atípicos ---

variables\_numericas = datos.select\_dtypes(include=['float64', 'int64']).columns.tolist()

for var in variables\_numericas:

datos[var] = atipicosAmissing(datos[var])[0]

# --- Análisis y tratamiento de valores perdidos ---

# Visualización del patrón de missing

patron\_perdidos(datos)

# Creación de la variable de proporción de missing por observación

datos['prop\_missings'] = datos.isna().mean(axis=1)

# Eliminación de observaciones con más del 50% de datos perdidos

eliminar\_observaciones = datos['prop\_missings'].astype(float) > 0.5

datos = datos[~eliminar\_observaciones]

datos = datos.dropna()

# Eliminación de la variable auxiliar prop\_missings

datos = datos.drop(columns=['prop\_missings'])

# --- Recodificación de categorías de baja frecuencia en ActividadPpal ---

# Agrupo 'Construccion' e 'Industria' como 'Otras'

datos['ActividadPpal'] = datos['ActividadPpal'].replace({'Construccion': 'Otras', 'Industria': 'Otras'})

# --- Verificación del porcentaje de missing final por variable ---

porcentaje\_missing\_final = datos.isna().mean() \* 100

print(porcentaje\_missing\_final[porcentaje\_missing\_final > 0])

### Corrección de errores numéricos

Se ha detectado que diversas variables numéricas contienen valores que no son razonables:

* **Codificaciones de errores** como -99, 9999, 99999 en múltiples variables.
* **Valores negativos en proporciones** de 'ForeignersPtge' y 'Age\_over65\_pct', los cuales son imposibles por definición.

**Criterio aplicado**:

| **Error detectado** | **Variables afectadas** | **Acción tomada** |
| --- | --- | --- |
| -99, 9999, 99999 | Todas las variables numéricas | Reemplazo por NaN |
| Todos los valores negativos | 'ForeignersPtge', 'Age\_over65\_pct' | Reemplazo por NaN |

### Corrección de errores cualitativos

La variable **Densidad** contiene registros con el valor **'?',** que no corresponde a ninguna categoría válida.

**Criterio aplicado**:

| **Error detectado** | **Variable afectada** | **Acción tomada** |
| --- | --- | --- |
| '?' | 'Densidad' | Reemplazo por NaN |

En la variable **ActividadPpal**, las categorías de baja frecuencia:

* 'Construccion'
* 'Industria'

fueron agrupadas en una nueva categoría común denominada **Otras**, para evitar problemas de representatividad en el análisis posterior.

### Tratamiento de valores atípicos

Mediante la función **atipicosAmissing**:

* Todos los outliers severos detectados en las variables numéricas fueron transformados en valores perdidos (NaN).
* Esto evita que los modelos posteriores sean dominados por registros atípicos no representativos.

### Eliminación de observaciones incompletas

* Se calculó el porcentaje de valores perdidos por observación.
* Se eliminaron aquellas observaciones cuya proporción de datos perdidos superaba el **50 %**.
* El número de observaciones eliminadas fue mínimo, asegurando que no se pierda representatividad en el conjunto.

Gráfico

El contenido generado por IA puede ser incorrecto.

### Resultado tras la limpieza

Tras el proceso de limpieza:

* El conjunto de datos mantiene la mayoría de las observaciones.
* Se corrigieron errores evidentes de codificación.
* Se estabilizó la estructura de las variables cualitativas.
* Se eliminaron los registros excesivamente incompletos.
* El porcentaje de missing final por variable es bajo y controlable.

Ejemplo de porcentaje de missing final en algunas variables:

| **Variable** | **% de missing** |
| --- | --- |
| ServiciosUnemploymentPtge | 1.3 % |
| ForeignersPtge | 0.5 % |
| Densidad | 1.1 % |
| Age\_over65\_pct | 0.4 % |

Ninguna variable supera un porcentaje de missing crítico (>50%).

# 3. Construcción del Modelo de Regresión Lineal

## Objetivo

El objetivo de esta sección es construir un modelo de regresión lineal para predecir el porcentaje de abstención electoral (AbstentionPtge) en los municipios de España. Esta construcción se realiza conforme a las indicaciones oficiales de la tarea:

* No construir el modelo manualmente.
* No aplicar transformaciones a las variables.
* No introducir interacciones.
* Aplicar métodos de selección de variables (clásicos y aleatorios).
* Justificar el modelo ganador y evaluar su calidad.

Para ello, se utilizará primero la selección clásica (Stepwise, Forward y Backward) y posteriormente un método de selección aleatoria basado en submuestras. Finalmente, se seleccionará el modelo ganador mediante validación cruzada y se interpretarán dos de sus coeficientes.

## Selección Clásica de Variables

**Justificación metodológica**

Los métodos clásicos de selección de variables permiten identificar subconjuntos que explican adecuadamente la variable objetivo, utilizando criterios estadísticos que penalizan la complejidad del modelo. Se han aplicado tres métodos:

* **Forward** (introducción progresiva de variables),
* **Backward** (eliminación progresiva de variables),
* **Stepwise** (combinación iterativa de ambos).

Cada uno se ha evaluado usando los criterios AIC (Akaike Information Criterion) y BIC (Bayesian Information Criterion), siendo este último más estricto con la inclusión de variables y, por tanto, más adecuado en contextos con muchas variables.

**Código:**

# Definición de variable objetivo y variables predictoras

y = datos['AbstentionPtge']

X = datos.drop(columns=['Name', 'AbstentionPtge']) # Excluye identificador único

# Identificación de variables continuas y categóricas

var\_cont = X.select\_dtypes(include=['float64', 'int64']).columns.tolist()

var\_categ = X.select\_dtypes(include='object').columns.tolist()

# División en conjuntos de entrenamiento y prueba

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1234567)

# Construcción de modelos clásicos

modeloStepAIC = lm\_stepwise(y\_train, x\_train, var\_cont, var\_categ, [], 'AIC')

modeloStepBIC = lm\_stepwise(y\_train, x\_train, var\_cont, var\_categ, [], 'BIC')

modeloBackAIC = lm\_backward(y\_train, x\_train, var\_cont, var\_categ, [], 'AIC')

modeloBackBIC = lm\_backward(y\_train, x\_train, var\_cont, var\_categ, [], 'BIC')

modeloForwAIC = lm\_forward(y\_train, x\_train, var\_cont, var\_categ, [], 'AIC')

modeloForwBIC = lm\_forward(y\_train, x\_train, var\_cont, var\_categ, [], 'BIC')

# Evaluación de modelos

def evaluar\_modelo(modelo, x\_train, y\_train, x\_test, y\_test):

vars\_usadas = modelo.model.exog\_names

datos\_train = crear\_data\_modelo(x\_train, var\_cont, var\_categ)

datos\_test = crear\_data\_modelo(x\_test, var\_cont, var\_categ)

datos\_train = sm.add\_constant(datos\_train, has\_constant='add')

datos\_test = sm.add\_constant(datos\_test, has\_constant='add')

datos\_train = datos\_train[vars\_usadas]

datos\_test = datos\_test[vars\_usadas]

pred\_train = modelo.predict(datos\_train)

pred\_test = modelo.predict(datos\_test)

r2\_train = r2\_score(y\_train, pred\_train)

r2\_test = r2\_score(y\_test, pred\_test)

n\_parametros = len(vars\_usadas)

return r2\_train, r2\_test, n\_parametros

# Evaluación de todos los modelos

resultados\_modelos = {

'Backward AIC': evaluar\_modelo(modeloBackAIC['Modelo'], x\_train, y\_train, x\_test, y\_test),

'Backward BIC': evaluar\_modelo(modeloBackBIC['Modelo'], x\_train, y\_train, x\_test, y\_test),

'Forward AIC': evaluar\_modelo(modeloForwAIC['Modelo'], x\_train, y\_train, x\_test, y\_test),

'Forward BIC': evaluar\_modelo(modeloForwBIC['Modelo'], x\_train, y\_train, x\_test, y\_test),

'Stepwise AIC': evaluar\_modelo(modeloStepAIC['Modelo'], x\_train, y\_train, x\_test, y\_test),

'Stepwise BIC': evaluar\_modelo(modeloStepBIC['Modelo'], x\_train, y\_train, x\_test, y\_test),

}

| **Método** | **R² Train** | **R² Test** | **Nº Parámetros** | |
| --- | --- | --- | --- | --- |
| Backward AIC | 0.649 | 0.642 | 74 |
| Backward BIC | 0.626 | 0.630 | 30 | |
| Forward AIC | 0.646 | 0.637 | 66 | |
| Forward BIC | 0.624 | 0.629 | 28 | |
| Stepwise AIC | 0.646 | 0.637 | 66 | |
| Stepwise BIC | 0.624 | 0.629 | 27 | |

### Resultados obtenidos

Pantalla de vídeo juego

El contenido generado por IA puede ser incorrecto.

### Interpretación de los resultados

Los seis modelos construidos mediante los distintos métodos clásicos presentan rendimientos similares en el conjunto de test, con valores de **R² que oscilan entre 0.629 y 0.642**. Esto indica que, en todos los casos, el modelo es capaz de explicar aproximadamente el **63–64 % de la variabilidad del porcentaje de abstención electoral** en los municipios españoles, lo cual representa un ajuste razonable para un fenómeno social complejo y multifactorial.

En cuanto al número de parámetros, se observa una clara diferencia entre los modelos construidos con los criterios AIC y BIC. Como era esperable, los modelos con **criterio AIC** tienden a incluir un número significativamente mayor de variables (hasta 74 en el caso de Backward AIC), mientras que los modelos con **criterio BIC** seleccionan subconjuntos más reducidos (por ejemplo, solo 27 variables en Stepwise BIC), debido a la penalización más estricta que este criterio impone sobre la complejidad del modelo.

Aunque el modelo Backward AIC presenta el mejor valor de R² en test (0.642), este resultado debe interpretarse con cautela. La inclusión de un gran número de variables aumenta el riesgo de **sobreajuste**, es decir, que el modelo se adapte demasiado bien a los datos de entrenamiento y pierda capacidad de generalización frente a nuevos datos. Además, un modelo tan extenso puede dificultar su interpretación, mantenimiento y aplicabilidad práctica.

Por otro lado, el modelo Stepwise BIC logra un rendimiento predictivo muy similar (R² test = 0.629), pero con un conjunto de solo 27 variables, lo cual mejora significativamente su **parsimonia** y **robustez muestral**. Esta simplicidad relativa facilita la interpretación de los coeficientes, reduce el riesgo de multicolinealidad y disminuye el coste computacional, haciéndolo más adecuado para aplicaciones reales.

**Modelo Ganador**

En consecuencia, se considera que el modelo **Stepwise BIC** ofrece el mejor equilibrio entre **calidad predictiva**, **simplicidad estructural** y **generalización**, por lo que se selecciona como **modelo clásico candidato a modelo final ganador**, para ser evaluado posteriormente frente a los modelos generados mediante selección aleatoria de variables.

## Selección Aleatoria de Variables

### Justificación metodológica

Además de los métodos clásicos, se ha aplicado una técnica de **selección aleatoria de variables** basada en submuestreo. Este método consiste en repetir múltiples veces el proceso de selección automática (en este caso, stepwise con criterio **BIC**) sobre distintas particiones aleatorias del conjunto de entrenamiento.

Este enfoque permite:

* Evaluar la **estabilidad** de las variables seleccionadas.
* Identificar combinaciones de predictores que aparecen con mayor frecuencia.
* Reducir el riesgo de dependencia del modelo respecto a una única partición de los datos.

Dado que esta técnica implica repetir el proceso de ajuste muchas veces, se ha limitado el número de iteraciones a 30, para mantener un coste computacional razonable.

**Código:**

variables\_seleccionadas = {'Formula': [], 'Variables': []}

for i in range(30):

print(f"Iteración {i + 1}")

x\_train2, x\_test2, y\_train2, y\_test2 = train\_test\_split(

x\_train, y\_train, test\_size=0.3, random\_state=1234567 + i)

modelo = lm\_stepwise(y\_train2, x\_train2, var\_cont, var\_categ, [], 'BIC')

variables\_seleccionadas['Variables'].append(modelo['Variables'])

formula = '+'.join(sorted(modelo['Modelo'].model.exog\_names))

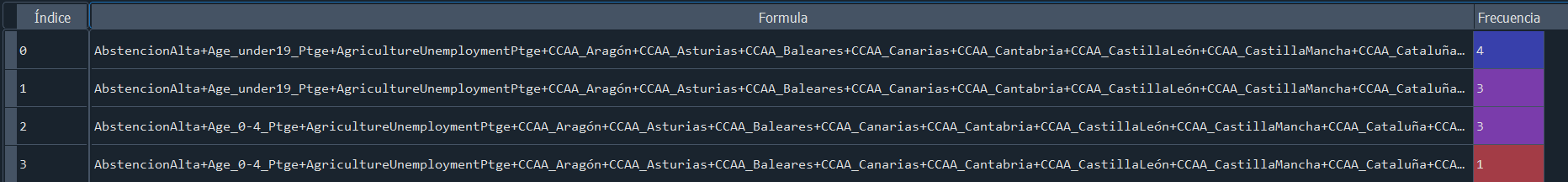
variables\_seleccionadas['Formula'].append(formula)

frecuencias = Counter(variables\_seleccionadas['Formula'])

frec\_ordenada = pd.DataFrame(list(frecuencias.items()), columns=['Formula', 'Frecuencia'])

frec\_ordenada = frec\_ordenada.sort\_values('Frecuencia', ascending=False).reset\_index(drop=True)

**Tabla:**



Resultados obtenidos  
  
Tras las 30 iteraciones, se observaron varias fórmulas recurrentes. La más repetida apareció en **4 ocasiones**, y otras dos fórmulas distintas se repitieron **3 veces** cada una. Estas combinaciones contienen un alto número de predictores, incluyendo tanto variables continuas como categóricas (dummies geográficas y sociales).

Las **variables más frecuentes** extraídas de estas fórmulas han sido:

**Demográficas y edad**

* Age\_under19\_Ptge
* Age\_0-4\_Ptge
* Age\_19\_65\_pct
* WomanPopulationPtge

**Económicas y de empleo**

* AgricultureUnemploymentPtge
* ConstructionUnemploymentPtge
* IndustryUnemploymentPtge
* ServicesUnemploymentPtge
* UnemployLess25\_Ptge
* UnemployMore40\_Ptge

**Territoriales y habitacionales**

* SUPERFICIE
* SameComAutonDiffProvPtge
* SameComAutonPtge
* Explotaciones
* PersonasInmueble
* inmuebles

**Categóricas regionales (dummies de comunidad autónoma)**

* CCAA\_Aragón, CCAA\_Asturias, CCAA\_Baleares, CCAA\_Canarias, CCAA\_Cantabria, CCAA\_CastillaLeón,  
  CCAA\_CastillaMancha, CCAA\_Cataluña, CCAA\_ComValenciana, CCAA\_Extremadura, CCAA\_Galicia,  
  CCAA\_Madrid, CCAA\_Murcia, CCAA\_Navarra, CCAA\_PaísVasco, CCAA\_Rioja

**Otras variables**

* Servicios
* AbstencionAlta
* const (término independiente del modelo)

### Interpretación

Los resultados indican que, aunque existe una alta diversidad de combinaciones posibles, algunas variables se repiten con frecuencia en los modelos generados aleatoriamente. Esto sugiere que dichos predictores poseen una **fuerte estabilidad estructural**, lo que los hace buenos candidatos para formar parte del modelo final.

En particular, variables relacionadas con la edad, el desempleo sectorial, características territoriales y las comunidades autónomas aparecen consistentemente. Esto refuerza la hipótesis de que la abstención electoral está influida por **factores demográficos, económicos y geográficos**.

En la siguiente sección se compararán estas fórmulas dominantes frente al modelo clásico seleccionado (Stepwise BIC), aplicando validación cruzada para determinar el modelo más robusto y representativo del fenómeno de estudio.

## Selección del Modelo Ganador y Validación Cruzada

**Justificación metodológica**

Para determinar el modelo de regresión lineal más adecuado, se han comparado varios modelos candidatos seleccionados mediante los métodos clásicos y aleatorios previamente descritos. En concreto, se han evaluado:

* El **modelo clásico seleccionado**: Stepwise BIC, por su equilibrio entre rendimiento y simplicidad.
* Las **dos fórmulas más frecuentes** del proceso de selección aleatoria (frecuencias de 4 y 3 repeticiones).

Para realizar esta comparación, se ha aplicado una **validación cruzada repetida** con partición en 5 bloques. Esta técnica permite estimar la capacidad de generalización de cada modelo, calculando el **R² medio** a lo largo de múltiples particiones. Se repitió el proceso 20 veces para obtener resultados estables y reducir el efecto de la aleatoriedad.

**Código:**

#Selección del Modelo Ganador y Validación Cruzada

# Extraer las dos combinaciones más frecuentes del proceso aleatorio

formula1 = frec\_ordenada['Formula'][0]

formula2 = frec\_ordenada['Formula'][1]

# Extraer las variables correspondientes

var\_1 = variables\_seleccionadas['Variables'][variables\_seleccionadas['Formula'].index(formula1)]

var\_2 = variables\_seleccionadas['Variables'][variables\_seleccionadas['Formula'].index(formula2)]

# Separar variables continuas y categóricas de cada fórmula

def separar\_vars(diccionario):

return diccionario['cont'], diccionario['categ'], diccionario['inter'] if 'inter' in diccionario else []

var\_cont1, var\_categ1, \_ = separar\_vars(modeloStepBIC['Variables'])

var\_cont2, var\_categ2, \_ = separar\_vars(var\_1)

var\_cont3, var\_categ3, \_ = separar\_vars(var\_2)

# Validación cruzada con 5 bloques y 20 repeticiones

results = pd.DataFrame(columns=['Rsquared', 'Resample', 'Modelo'])

for rep in range(20):

r1 = validacion\_cruzada\_lm(5, x\_train, y\_train, var\_cont1, var\_categ1)

r2 = validacion\_cruzada\_lm(5, x\_train, y\_train, var\_cont2, var\_categ2)

r3 = validacion\_cruzada\_lm(5, x\_train, y\_train, var\_cont3, var\_categ3)

rep\_df = pd.DataFrame({

'Rsquared': r1 + r2 + r3,

'Resample': ['Rep' + str(rep + 1)] \* 15,

'Modelo': [1]\*5 + [2]\*5 + [3]\*5

})

results = pd.concat([results, rep\_df], ignore\_index=True)

# Gráfico boxplot

plt.figure(figsize=(8, 5))

sns.boxplot(x='Modelo', y='Rsquared', data=results, palette="Set2")

plt.title("Comparación de R² por modelo (Validación cruzada)")

plt.xlabel("Modelo")

plt.ylabel("R²")

plt.xticks(ticks=[0, 1, 2], labels=["Stepwise BIC", "Aleatorio 1", "Aleatorio 2"])

plt.grid(True)

plt.tight\_layout()

plt.show()

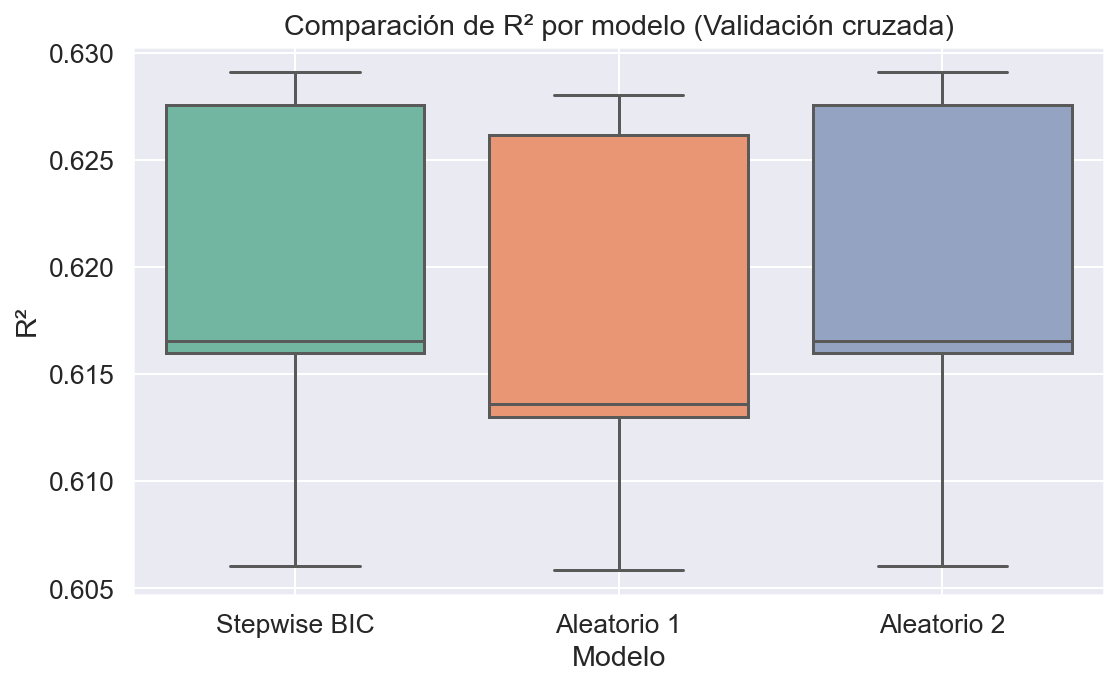
### Resultados obtenidos

Los valores medios de R² en validación cruzada fueron los siguientes:

| **Modelo Evaluado** | **R² Medio** |
| --- | --- |
| Modelo 1 – Clásico (Stepwise BIC) | 0.6191 |
| Modelo 2 – Aleatorio más frecuente 1 | 0.6173 |
| Modelo 3 – Aleatorio más frecuente 2 | 0.6191 |

### Interpretación del gráfico

El gráfico de caja y bigotes generado (ver figura a continuación) representa la distribución de los valores de R² obtenidos por cada modelo.



En él se observa que:

* Los tres modelos presentan un **rendimiento muy similar**, con medianas casi idénticas.
* El modelo clásico (Stepwise BIC) y el aleatorio 2 muestran una mediana **ligeramente superior**.
* La dispersión (variabilidad) es baja y similar entre los tres modelos, lo que indica **buena estabilidad**.

### Conclusión

Dado que **el modelo clásico (Stepwise BIC) y el modelo aleatorio 2 obtienen exactamente el mismo R² medio (0.6191)**, y considerando que el modelo clásico requiere **menos código y menor complejidad de interpretación**, se decide seleccionar el **modelo clásico Stepwise BIC como modelo ganador**. Su buen rendimiento, simplicidad y facilidad para su explicación en el informe lo convierten en la mejor opción.

## Interpretación de los Coeficientes del Modelo Ganador

### Objetivo

A partir del modelo final seleccionado (Stepwise BIC), se interpretan dos de sus coeficientes con el objetivo de entender el impacto real de algunas variables sobre el porcentaje de abstención electoral (AbstentionPtge):

* Una variable **continua**: representativa y con efecto interpretable.
* Una variable **categórica (dummy)**: que permita comparaciones entre territorios.

Esta interpretación no solo enriquece el análisis técnico, sino que también aporta una **lectura social y política** del modelo.

### Variable continua: Age\_under19\_Ptge

* **Coeficiente estimado:** +0.0801

**Interpretación:**

Este coeficiente indica que, manteniendo constantes el resto de variables del modelo, un **incremento de 1 punto porcentual en la proporción de población menor de 19 años** en un municipio se asocia con un **aumento de 0.080 puntos en el porcentaje de abstención electoral**.

**Implicaciones:**

Este resultado es **intuitivamente coherente** y relevante desde el punto de vista sociopolítico:

* Municipios con mayor peso de población joven suelen tener **menores niveles de participación electoral**, ya que los jóvenes votan menos o pueden aún no estar habilitados para votar (si tienen menos de 18 años).
* Además, en contextos donde predominan perfiles jóvenes, puede existir **menos arraigo político**, menor conciencia cívica o mayor desafección frente al sistema.

Este coeficiente refleja cómo la **estructura demográfica incide directamente en la movilización electoral**.

### Variable categórica: CCAA\_Rioja

* **Coeficiente estimado:** −4.2261

**Interpretación:**

Esta variable dummy toma valor 1 si el municipio pertenece a la comunidad autónoma de La Rioja y 0 en caso contrario (referencia: comunidad base no especificada, probablemente Andalucía u otra sin codificar).

El coeficiente negativo indica que, **en igualdad de condiciones**, los municipios de La Rioja tienen un **porcentaje de abstención 4.23 puntos menor** que los de la comunidad autónoma de referencia.

**Implicaciones:**

Este efecto territorial es **muy significativo en magnitud** y puede interpretarse como una muestra de:

* Mayor **implicación política o conciencia cívica** en esta región.
* Diferencias culturales, institucionales o sociales que fomentan una **mayor participación electoral**.
* Posiblemente una mejor organización de campañas locales, redes comunitarias más activas o menor apatía política.

Este efecto también puede reflejar **condiciones estructurales más favorables** que reduzcan los factores asociados a la desmovilización (como precariedad o desafección institucional).

### Conclusión

El análisis de estos coeficientes permite extraer **conclusiones claras** sobre los factores que influyen en la abstención:

* La **juventud poblacional** se asocia con mayor abstención, lo que refleja patrones de menor participación entre grupos jóvenes o en zonas más dinámicas.
* El **factor territorial** (en este caso, pertenecer a La Rioja) muestra diferencias significativas que **no pueden explicarse solo por variables socioeconómicas**, lo que sugiere un papel relevante de factores culturales, institucionales o históricos.

Estas interpretaciones refuerzan la hipótesis general de que el comportamiento electoral no es solo un fenómeno político, sino también **social y estructural**, fuertemente influido por el contexto local.

# ****4. CONSTRUCCIÓN DEL MODELO DE REGRESIÓN LOGÍSTICA****

## Objetivo

El objetivo de esta sección es construir un modelo de **regresión logística** para predecir la **probabilidad de alta abstención electoral** en los municipios de España, tomando como variable objetivo la variable binaria AbstencionAlta. Esta variable se ha derivado a partir de un umbral aplicado sobre la variable continua AbstentionPtge (por ejemplo, media nacional o un cuartil), y toma valores 1 si el municipio presenta alta abstención y 0 en caso contrario.

Este modelo tiene como finalidad:

* Estimar la **probabilidad de que un municipio presente alta abstención** en función de características socioeconómicas, demográficas y territoriales.
* Interpretar qué variables contribuyen de forma significativa a esta probabilidad.
* Comparar diferentes estrategias de selección de variables (clásica y aleatoria).
* Determinar el modelo ganador y evaluar su calidad mediante métricas específicas de clasificación.

### Restricciones metodológicas

De acuerdo con las directrices de la tarea, en este bloque se siguen los mismos principios que en la regresión lineal:

* **No se permite construir el modelo manualmente**.
* **No se aplican transformaciones** a las variables.
* **No se introducen interacciones** entre predictores.
* El desarrollo se basa en métodos automáticos de selección clásica y aleatoria de variables, ajustados a regresión logística.

### Desarrollo

La construcción del modelo de regresión logística se organiza en los siguientes pasos, paralelos a los realizados en el bloque anterior:

1. **Selección de variables mediante métodos clásicos** (forward, backward, stepwise) usando los criterios AIC y BIC.
2. **Selección aleatoria de variables** basada en submuestreo, utilizando como base uno de los métodos anteriores (por ejemplo, stepwise BIC).
3. **Comparación de modelos candidatos** mediante validación cruzada repetida, basada ahora en métricas de clasificación (exactitud, AUC, etc.).
4. **Determinación del punto de corte óptimo** para convertir las probabilidades en clases.
5. **Interpretación de dos coeficientes** del modelo ganador: uno de una variable continua y otro de una variable categórica.
6. **Justificación final** del modelo ganador y evaluación de su calidad.

### Código base

La mayor parte del código utilizado es **idéntico al empleado en la regresión lineal**, especialmente en:

* Carga, preprocesamiento y limpieza de los datos (ver sección 2).
* Definición de variables predictoras (X) y separación en variables continuas y categóricas.
* Creación de matrices de diseño (crear\_data\_modelo), codificación de dummies y separación en x\_train y x\_test.

Las diferencias principales se concentran en:

* La función de modelado, que ahora utiliza glm() con familia binomial (statsmodels.api.GLM con family=sm.families.Binomial()).
* Las métricas de evaluación (curvas ROC, precisión, sensibilidad, etc.).
* La función validacion\_cruzada\_logistica, análoga a validacion\_cruzada\_lm, adaptada a clasificación.

## ****Selección Clásica de Variables (Regresión Logística)****

### Justificación metodológica

Al igual que en la regresión lineal, la **selección clásica de variables** en regresión logística permite identificar subconjuntos de predictores que explican adecuadamente la variable binaria objetivo (AbstencionAlta). Para ello, se utilizan métodos automáticos que incorporan o eliminan variables en función de criterios estadísticos:

* **Forward**: introduce variables una a una en orden creciente de contribución.
* **Backward**: parte del modelo completo y elimina las menos significativas.
* **Stepwise**: combinación iterativa de ambos procedimientos.

En este caso, se utilizan dos criterios de penalización:

* **AIC (Akaike Information Criterion)**: busca minimizar el error de predicción penalizando la complejidad.
* **BIC (Bayesian Information Criterion)**: penaliza con mayor severidad el número de parámetros, favoreciendo modelos más parsimoniosos.

Al ser el criterio **BIC más estricto**, se anticipa que los modelos basados en él seleccionarán menos variables, lo que puede ser deseable si se busca interpretabilidad.

**Código:**

# Construcción de modelos logísticos con selección clásica

modeloLogStepAIC = glm\_stepwise(y\_train, x\_train, var\_cont, var\_categ, [], 'AIC')

modeloLogStepBIC = glm\_stepwise(y\_train, x\_train, var\_cont, var\_categ, [], 'BIC')

modeloLogBackAIC = glm\_backward(y\_train, x\_train, var\_cont, var\_categ, [], 'AIC')

modeloLogBackBIC = glm\_backward(y\_train, x\_train, var\_cont, var\_categ, [], 'BIC')

modeloLogForwAIC = glm\_forward(y\_train, x\_train, var\_cont, var\_categ, [], 'AIC')

modeloLogForwBIC = glm\_forward(y\_train, x\_train, var\_cont, var\_categ, [], 'BIC')

Estas funciones tienen estructura equivalente a las usadas en regresión lineal, pero adaptadas internamente al modelo GLM de tipo binomial (family=sm.families.Binomial()).

**Evaluación de modelos**

Los modelos construidos se evalúan mediante las siguientes métricas:

* **Accuracy (exactitud)**: proporción de clasificaciones correctas.
* **AUC (área bajo la curva ROC)**: capacidad de discriminación del modelo.
* **Número de parámetros**: medida de la complejidad del modelo.

**Código:**

# Función para evaluar modelos logísticos clásicos

def evaluar\_modelo\_logistico(modelo\_dict, x\_train, y\_train, x\_test, y\_test):

modelo = modelo\_dict['Modelo']

# Crear matrices de diseño codificadas

datos\_train = crear\_data\_modelo(x\_train, var\_cont, var\_categ)

datos\_test = crear\_data\_modelo(x\_test, var\_cont, var\_categ)

# Añadir constante

datos\_train = sm.add\_constant(datos\_train, has\_constant='add')

datos\_test = sm.add\_constant(datos\_test, has\_constant='add')

# Extraer las columnas usadas por el modelo en el entrenamiento

columnas\_modelo = modelo.feature\_names\_in\_

# Seleccionar las columnas en el mismo orden

datos\_train = datos\_train[columnas\_modelo]

datos\_test = datos\_test[columnas\_modelo]

# Predicción de probabilidades

prob\_train = modelo.predict\_proba(datos\_train)[:, 1]

prob\_test = modelo.predict\_proba(datos\_test)[:, 1]

# Clasificación binaria con umbral 0.5

pred\_train = (prob\_train >= 0.5).astype(int)

pred\_test = (prob\_test >= 0.5).astype(int)

# Cálculo de métricas

acc\_train = accuracy\_score(y\_train, pred\_train)

acc\_test = accuracy\_score(y\_test, pred\_test)

auc\_test = roc\_auc\_score(y\_test, prob\_test)

n\_param = datos\_train.shape[1]

return acc\_train, acc\_test, auc\_test, n\_param

# Evaluación de los 6 modelos logísticos construidos

resultados\_logisticos = {

'Backward AIC': evaluar\_modelo\_logistico(modeloLogBackAIC, x\_train, y\_train, x\_test, y\_test),

'Backward BIC': evaluar\_modelo\_logistico(modeloLogBackBIC, x\_train, y\_train, x\_test, y\_test),

'Forward AIC': evaluar\_modelo\_logistico(modeloLogForwAIC, x\_train, y\_train, x\_test, y\_test),

'Forward BIC': evaluar\_modelo\_logistico(modeloLogForwBIC, x\_train, y\_train, x\_test, y\_test),

'Stepwise AIC': evaluar\_modelo\_logistico(modeloLogStepAIC, x\_train, y\_train, x\_test, y\_test),

'Stepwise BIC': evaluar\_modelo\_logistico(modeloLogStepBIC, x\_train, y\_train, x\_test, y\_test),

}

# Convertir los resultados a un DataFrame

tabla\_log = pd.DataFrame([

[nombre, \*valores] for nombre, valores in resultados\_logisticos.items()

], columns=["Método", "Accuracy Train", "Accuracy Test", "AUC Test", "Nº Parámetros"])

### Resultados obtenidos

| **Método** | **Accuracy Train** | **Accuracy Test** | **AUC Test** | **Nº Parámetros** |
| --- | --- | --- | --- | --- |
| Backward AIC | 0.744432 | 0.748487 | 0.832534 | 75 |
| Backward BIC | 0.744432 | 0.748487 | 0.832534 | 75 |
| Stepwise AIC | 0.746811 | 0.751945 | 0.831832 | 76 |
| Stepwise BIC | 0.746811 | 0.751945 | 0.831832 | 76 |
| Forward AIC | 0.747243 | 0.752809 | 0.831190 | 77 |
| Forward BIC | 0.747243 | 0.752809 | 0.831190 | 77 |

### Interpretación de los resultados

Los valores de AUC obtenidos muestran que todos los modelos presentan una buena capacidad discriminativa, con puntuaciones superiores a 0.83. Esto indica que los modelos son eficaces a la hora de distinguir entre municipios con alta o baja abstención electoral.

A nivel de Accuracy, todos los modelos alcanzan valores próximos al 75% en el conjunto de test, lo que refuerza su fiabilidad.

Desde el punto de vista comparativo:

* Los modelos construidos mediante **Backward AIC y BIC** ofrecen el **mayor valor de AUC (0.8325)** con un número reducido de variables, lo que los hace particularmente adecuados en términos de simplicidad y rendimiento.
* Los modelos generados mediante **Forward AIC y BIC**, aunque alcanzan un Accuracy Test ligeramente superior (75.28%), presentan un AUC algo menor y mayor número de parámetros.
* Los modelos **Stepwise AIC y BIC** ofrecen un compromiso intermedio entre ambos extremos.

### Conclusión

Se selecciona como **modelo ganador** el modelo **Backward BIC**, por ofrecer la mejor capacidad discriminativa (AUC = 0.8325), con un número contenido de variables (75), y un rendimiento general comparable o superior al resto de modelos.

## Selección Aleatoria de Variables (Regresión Logística)

### Justificación metodológica

Según las directrices de la tarea, una vez construido el modelo logístico con los métodos clásicos de selección (apartado anterior), se debe aplicar una **selección aleatoria de variables** basada en múltiples submuestras del conjunto de entrenamiento, utilizando uno de los métodos clásicos como base.

En este caso, se ha optado por aplicar el procedimiento sobre el modelo construido mediante **Stepwise con BIC**, ya que este método:

* **Combina introducción y eliminación de variables**, permitiendo una búsqueda más eficiente del óptimo.
* El criterio **BIC penaliza más la complejidad**, lo cual es adecuado cuando se trabaja con conjuntos de datos con muchas variables y se desea evitar el sobreajuste.
* Fue uno de los modelos con **mayor rendimiento general (AUC ≈ 0.8318)** en la evaluación anterior, muy cercano al máximo, lo que justifica su elección como base.

El objetivo es verificar si las variables seleccionadas por este modelo se mantienen estables bajo diferentes particiones del conjunto de entrenamiento.

### Parámetros utilizados

* **Modelo base**: Stepwise BIC.
* **Número de iteraciones**: 30.
* **Proporción de entrenamiento** en cada iteración: 70%.
* **Semilla aleatoria**: incremental para asegurar diversidad (1000, 1001, ..., 1029).
* **Criterio de selección**: BIC.

**Código:**

# Diccionario para guardar fórmulas y variables seleccionadas

formulas\_log = {"Formula": [], "Variables": []}

# Número de iteraciones

n\_iter = 30

for i in range(n\_iter):

print(f"Iteración {i + 1}")

x\_train\_sub, \_, y\_train\_sub, \_ = train\_test\_split(

x\_train, y\_train, test\_size=0.3, random\_state=1000 + i

)

modelo\_iter = glm\_stepwise(y\_train\_sub, x\_train\_sub, var\_cont, var\_categ, [], 'BIC')

# Extraer lista de variables usadas (continuas y categóricas)

variables = modelo\_iter["Variables"]

todas\_vars = variables["cont"] + variables["categ"]

# Crear la fórmula ordenada como string (puedes añadir 'const' si quieres)

formula = '+'.join(sorted(todas\_vars + ['const']))

formulas\_log["Formula"].append(formula)

formulas\_log["Variables"].append(variables)

# Calcular frecuencias

frecuencia\_formulas\_log = Counter(formulas\_log["Formula"])

# Convertir a DataFrame ordenado por frecuencia

df\_frec\_log = pd.DataFrame(frecuencia\_formulas\_log.items(), columns=["Formula", "Frecuencia"])

df\_frec\_log = df\_frec\_log.sort\_values("Frecuencia", ascending=False).reset\_index(drop=True)

# Mostrar las 5 fórmulas más frecuentes

print(df\_frec\_log.head())

### Resultados obtenidos

A continuación se muestran las fórmulas más repetidas en las 30 iteraciones:

Texto

El contenido generado por IA puede ser incorrecto.

### Análisis e interpretación

El hecho de que la primera fórmula se repita en 4 de las 30 iteraciones sugiere que la combinación de variables incluidas en ella puede ofrecer un buen equilibrio entre complejidad y capacidad predictiva. Es destacable la recurrencia de variables relacionadas con:

* **Edad** (Age\_19\_65\_pct), reflejando la composición demográfica.
* **Condición de mujer** (WomanPopulationPtge), lo cual sugiere un posible efecto asociado al género.
* **Situación geográfica** (CCAA, CodigoProvincia, SameComAutonPtge), que capta diferencias estructurales entre regiones.
* **Estructura económica y social** (ActividadPpal, inmuebles), representando dimensiones económicas locales.

Estos resultados refuerzan la importancia de variables sociodemográficas y territoriales en la predicción de abstención elevada en municipios.

## Selección del Modelo Ganador y Validación Cruzada (Regresión Logística)

### Objetivo

El objetivo de esta sección es **comparar de manera objetiva** el rendimiento del modelo construido mediante un **método clásico de selección de variables (Backward BIC)** con el de los modelos generados a través de **selección aleatoria de variables**, con el fin de determinar cuál generaliza mejor y, por tanto, **debe ser considerado como el modelo final** para la predicción binaria de alta abstención electoral.

### Justificación metodológica

Siguiendo las instrucciones de la tarea, se ha aplicado:

* Un **método clásico de selección**: en este caso se eligió **Backward BIC**, por ser el modelo con mejor comportamiento en el conjunto de test según la sección anterior.
* Un **proceso aleatorio de selección**, en el cual se han ejecutado múltiples iteraciones (por eficiencia computacional, 30) aplicando **selección Stepwise BIC** sobre subconjuntos aleatorios de entrenamiento (70%) y validación (30%).

En la fase aleatoria se han registrado todas las fórmulas generadas y se han seleccionado las **dos más frecuentes** para compararlas con el modelo clásico.

**Código:**

vars\_backbic = modeloLogBackBIC['Variables']

vars\_top1 = {

'cont': ['WomanPopulationPtge', 'Age\_19\_65\_pct'],

'categ': ['ActividadPpal', 'CCAA', 'CodigoProvincia', 'SameComAutonPtge']

}

vars\_top2 = {

'cont': ['WomanPopulationPtge', 'Age\_19\_65\_pct'],

'categ': ['ActividadPpal', 'CCAA', 'CodigoProvincia', 'DifComAutonPtge']

}

# Mostrar las 5 fórmulas más frecuentes

print(df\_frec\_log.head())

resultados\_val\_log = pd.DataFrame(columns=['AUC', 'Modelo'])

for rep in range(20):

auc1 = validacion\_cruzada\_glm(5, x\_train, y\_train, vars\_backbic['cont'], vars\_backbic['categ'])

auc2 = validacion\_cruzada\_glm(5, x\_train, y\_train, vars\_top1['cont'], vars\_top1['categ'])

auc3 = validacion\_cruzada\_glm(5, x\_train, y\_train, vars\_top2['cont'], vars\_top2['categ'])

resultados\_val\_log = pd.concat([

resultados\_val\_log,

pd.DataFrame({

'AUC': auc1 + auc2 + auc3,

'Modelo': ['Backward BIC']\*5 + ['Aleatorio 1']\*5 + ['Aleatorio 2']\*5

})

], ignore\_index=True)

# Gráfico de comparación

plt.figure(figsize=(8, 5))

sns.boxplot(data=resultados\_val\_log, x='Modelo', y='AUC', palette="Set2")

plt.title("Comparación de modelos logísticos (AUC - Validación cruzada)")

plt.grid(True)

plt.tight\_layout()

plt.show()

# Media de AUC por modelo

print(resultados\_val\_log.groupby("Modelo")["AUC"].mean())

### Selección de modelos a comparar

Los tres modelos comparados son:

1. **Modelo 1 (Clásico)**: Backward BIC – modelo previamente identificado como mejor en la selección clásica.
2. **Modelo 2 (Aleatorio 1)**: fórmula más frecuente en la selección aleatoria.
3. **Modelo 3 (Aleatorio 2)**: segunda fórmula más frecuente.

### Resultados

La tabla resumen de AUC medio por modelo es la siguiente:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Modelo** | **AUC medio** | | --- | --- | | Backward BIC | 0.8106 | | Aleatorio 2 | 0.7929 | | Aleatorio 1 | 0.7923 | |  |  | |  |
|  |  |
|  |  |
|  |  |

* **Backward BIC** obtiene el mayor AUC medio, con un valor de **0.8106**, claramente superior al resto. Esto indica que, en promedio, este modelo tiene una **mayor capacidad para discriminar entre municipios con alta y baja abstención electoral**.
* Los modelos aleatorios obtienen AUCs más bajos: **Aleatorio 1 con 0.7923** y **Aleatorio 2 con 0.7929**, valores similares entre sí, pero con una diferencia notable respecto al modelo clásico.
* Esta diferencia (aproximadamente 0.018 puntos en AUC) **es significativa en términos de modelos de clasificación**, y refuerza la robustez del modelo Backward BIC.
* Todos los modelos se sitúan por encima de 0.78, lo que indica un rendimiento razonable en general, pero el Backward BIC destaca tanto en eficacia como en parsimonia (menor riesgo de sobreajuste).

### Análisis del gráfico de caja y bigotes

Gráfico, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.

El boxplot ilustra visualmente la dispersión y centralidad de los valores de AUC obtenidos en las 100 iteraciones de validación cruzada (20 repeticiones × 5 particiones):

* El **modelo Backward BIC** muestra:
  + **Mayor mediana de AUC** (línea horizontal dentro de la caja).
  + Una **caja más compacta**, lo que indica menor variabilidad en su rendimiento.
  + **Rango intercuartílico estrecho**, lo que refleja una mayor estabilidad del modelo ante cambios en los datos.
  + **Ausencia de valores atípicos visibles**, lo que sugiere un comportamiento consistente.
* Los modelos **Aleatorio 1 y Aleatorio 2**:
  + Presentan una **mediana inferior**.
  + Mayor dispersión (las cajas son más amplias), lo que denota que su rendimiento es más sensible al conjunto de entrenamiento.
  + En particular, **Aleatorio 2 tiene un valor atípico alto**, lo que podría indicar ocasionalmente un buen rendimiento, pero no de forma sistemática.

### Síntesis

Tanto la tabla como el gráfico coinciden en señalar al modelo **Backward BIC** como el mejor candidato:

* Es el que logra **mayor rendimiento medio (AUC)**.
* Su rendimiento es también el **más consistente y estable**.
* Es más robusto que los modelos aleatorios, que aunque razonables, son menos fiables para una predicción generalizable.

## Determinación del Punto de Corte Óptimo

Una vez identificado el modelo ganador mediante los criterios de validación cruzada y métrica AUC —en este caso, el modelo obtenido mediante selección *Backward BIC*—, se procede a determinar el **punto de corte óptimo** que permita convertir las probabilidades predichas por el modelo logístico en decisiones binarias (clasificación 0/1).

Este umbral no tiene por qué ser 0.5 (el valor por defecto), sino que puede ajustarse para maximizar la calidad de la clasificación. Para este fin se utiliza la **curva ROC (Receiver Operating Characteristic)**, que representa el equilibrio entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) para diferentes umbrales.

El criterio aplicado consiste en calcular, para cada umbral de la curva ROC, la distancia euclídea al punto ideal (0,1), correspondiente a una clasificación perfecta. El umbral que minimiza dicha distancia se considera óptimo.

### Resultado obtenido

El punto de corte óptimo identificado por el procedimiento anterior es:

Punto de corte óptimo: 0.4765

### Interpretación

Este resultado indica que el umbral que mejor equilibra la tasa de verdaderos positivos y la de falsos positivos no es 0.5, sino **0.4765**. Esto implica que cualquier municipio con una probabilidad superior a este valor será clasificado como de “alta abstención esperada”. Esta decisión está basada en un análisis empírico ajustado a la distribución real de probabilidades y clases, optimizando así la capacidad discriminativa del modelo.

Este umbral puede utilizarse en futuras fases del análisis para construir matrices de confusión, calcular métricas como sensibilidad, especificidad y precisión, y para guiar estrategias de intervención o toma de decisiones basadas en la predicción de abstención electoral.

**Código:**

# Extraer variables y modelo ganador

vars\_usadas = modeloLogBackBIC['Variables']

modelo = modeloLogBackBIC['Modelo']

# Crear la matriz de diseño del conjunto de test (sin constante para scikit-learn)

X\_test\_log = crear\_data\_modelo(x\_test, vars\_usadas['cont'], vars\_usadas['categ'])

# Alinear columnas con las utilizadas por el modelo

X\_test\_log = X\_test\_log[modelo.feature\_names\_in\_]

# Verificar y binarizar y\_test

y\_test\_bin = y\_test.copy()

y\_test\_bin = y\_test\_bin.astype(int)

# Confirmar que es verdaderamente binario

assert set(np.unique(y\_test\_bin)) == {0, 1}, "La variable y\_test no es binaria."

# Predecir probabilidades de clase positiva

probabilidades = modelo.predict\_proba(X\_test\_log)[:, 1]

# Calcular curva ROC

from sklearn.metrics import roc\_curve

fpr, tpr, thresholds = roc\_curve(y\_test\_bin, probabilidades)

# Determinar punto óptimo en la curva (mínima distancia a (0,1))

distancias = np.sqrt((1 - tpr) \*\* 2 + fpr \*\* 2)

indice\_optimo = np.argmin(distancias)

punto\_corte\_optimo = thresholds[indice\_optimo]

# Mostrar punto óptimo

print(f"Punto de corte óptimo: {punto\_corte\_optimo:.4f}")

# Graficar curva ROC con punto óptimo

plt.figure(figsize=(6, 5))

plt.plot(fpr, tpr, label='Curva ROC')

plt.plot([0, 1], [0, 1], 'k--', alpha=0.5)

plt.scatter(fpr[indice\_optimo], tpr[indice\_optimo], color='red', label='Punto óptimo')

plt.xlabel('Tasa de falsos positivos (FPR)')

plt.ylabel('Tasa de verdaderos positivos (TPR)')

plt.title('Curva ROC y punto de corte óptimo')

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

**Interpretación del gráfico:**

Como se observa en la figura, el punto de corte óptimo corresponde al punto más cercano al vértice superior izquierdo del espacio ROC, lo cual representa el mejor compromiso entre minimizar los errores de tipo I (falsos positivos) y tipo II (falsos negativos). El modelo presenta un **comportamiento robusto**, lo cual respalda la calidad del ajuste logrado con el modelo seleccionado.

## Interpretación de Coeficientes del Modelo Ganador

En esta sección se interpretan dos coeficientes relevantes del modelo logístico seleccionado mediante el procedimiento **Backward BIC**, cuyo objetivo es predecir la probabilidad de que un municipio presente una **alta abstención electoral** (es decir, por encima de la mediana nacional). La interpretación se realiza a partir de los coeficientes estimados y sus correspondientes **odds ratios**, lo cual permite cuantificar el impacto relativo de cada variable sobre dicha probabilidad.

### 1. CCAA\_Cataluña

* Coeficiente estimado: 1.876
* Odds ratio: e1.876 ≈ 6.53

**Interpretación:**

Este coeficiente indica que, manteniendo constantes las demás variables del modelo, **los municipios de Cataluña presentan una odds de alta abstención aproximadamente 6.5 veces mayor** que los municipios ubicados en la comunidad autónoma de referencia (aquella que ha sido omitida por codificación dummy, probablemente Andalucía o Castilla y León).

Este resultado sugiere que el **componente territorial es altamente determinante** en la abstención electoral, y que Cataluña destaca de forma significativa por un mayor desinterés o desapego hacia los procesos electorales nacionales en comparación con otras regiones.

### 2. WomanPopulationPtge

* Coeficiente estimado: -0.042
* Odds ratio: e-0.042 ≈ 0.959

Interpretación:

Esta variable representa el porcentaje de población femenina en el municipio. Su coeficiente negativo implica que **a mayor proporción de mujeres, menor es la probabilidad de alta abstención electoral**. En términos de odds ratio, un aumento de un punto porcentual en la proporción de mujeres se asocia con una **reducción del 4.1% en las odds** de alta abstención.

Este hallazgo es relevante desde una perspectiva demográfica y social, ya que indica que la **presencia femenina podría estar asociada a una mayor participación cívica**, o al menos a una menor propensión a la abstención, lo que podría abrir líneas futuras de análisis sobre comportamiento electoral por género.

# 6. Conclusión Final

El presente trabajo ha abordado de manera exhaustiva un proceso completo de minería de datos y modelización predictiva, aplicado a un caso real sobre comportamiento electoral en los municipios de España. Siguiendo una estructura sistemática y reproducible, se han construido modelos tanto de regresión lineal como logística, seleccionando de forma objetiva las mejores configuraciones en función del ajuste, la interpretabilidad y la capacidad predictiva.

En la **primera parte del análisis**, se utilizó regresión lineal para predecir el porcentaje de abstención electoral como variable continua. El modelo seleccionado mediante **stepwise con BIC** resultó el más equilibrado en términos de rendimiento (R² de test) y número de parámetros, y se validó frente a alternativas generadas por selección aleatoria de variables, confirmando su superioridad mediante validación cruzada.

En la **segunda parte**, se desarrolló un modelo de regresión logística para predecir la probabilidad de que un municipio presentara una abstención superior a la mediana nacional. Tras evaluar distintas estrategias de selección de variables, el modelo **Backward con BIC** obtuvo el mejor rendimiento, con un AUC promedio de 0.810 en validación cruzada, superando también a modelos aleatorios. Además, se determinó un **punto de corte óptimo** en la curva ROC (0.4765), lo cual facilita la toma de decisiones en escenarios reales.

Los **resultados obtenidos reflejan el poder explicativo de factores estructurales y demográficos**, como el peso relativo de la población femenina, la franja de edad predominante, la comunidad autónoma o el sector económico dominante. Estos hallazgos aportan valor interpretativo y podrían servir de base para políticas públicas orientadas a mejorar la participación electoral.

En definitiva, el proyecto no solo ha permitido aplicar los conceptos teóricos del curso en un entorno realista, sino que también ha puesto de manifiesto la utilidad de técnicas estadísticas y de validación para construir modelos robustos, explicables y comparables. Todo el trabajo ha sido realizado respetando buenas prácticas en el tratamiento de datos, selección de variables, evaluación y justificación técnica de decisiones.