



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
2. DEPURACIÓN DE DATOS.....	5
a) IMPORTACIÓN DEL CONJUNTO DE DATOS Y ASIGNACIÓN CORRECTA DE LOS TIPOS DE VARIABLES	5
b) ANÁLISIS DESCRIPTIVO DEL CONJUNTO DE DATOS.....	6
Dimensiones y estructura general	7
Variables numéricas con valores no razonables.....	7
Variables cualitativas con errores de codificación.....	7
Análisis de asimetría y curtosis	8
Variables con alta dispersión y valores extremos	8
Valores atípicos previstos	8
Estructura de valores perdidos preliminar	8
c) CORRECCIÓN DE LOS ERRORES DETECTADOS.....	8
Objetivo de la limpieza.....	8
Corrección de errores numéricos	9
Corrección de errores cualitativos.....	9
Tratamiento de valores atípicos	10
Eliminación de observaciones incompletas	10
Resultado tras la limpieza	10
3. CONSTRUCCIÓN DEL MODELO DE REGRESIÓN LINEAL.....	11
a) Objetivo.....	11
b) Selección Clásica de Variables.....	11
Resultados obtenidos	12
Interpretación de los resultados.....	12
c) Selección Aleatoria de Variables.....	13
Justificación metodológica	13
Resultados obtenidos	14
Interpretación	14
d) Selección del Modelo Ganador y Validación Cruzada	15
Resultados obtenidos	16
Interpretación del gráfico	16
Conclusión.....	16
e) Interpretación de los Coeficientes del Modelo Ganador.....	17
Objetivo	17
Variable continua: Age_under19_Ptge	17
Variable categórica: CCAA_Rioja.....	17
Conclusión.....	18
4. CONSTRUCCIÓN DEL MODELO DE REGRESIÓN LOGÍSTICA.....	18
a) Objetivo.....	18
Restricciones metodológicas	18
Desarrollo	18
Código base	19
b) Selección Clásica de Variables (Regresión Logística).....	19
Justificación metodológica	19
Resultados obtenidos	20
Interpretación de los resultados.....	21
Conclusión.....	21

c)	Selección Aleatoria de Variables (Regresión Logística)	21
	Justificación metodológica	21
	Parámetros utilizados	21
	Resultados obtenidos	22
	Análisis e interpretación	22
d)	Selección del Modelo Ganador y Validación Cruzada (Regresión Logística)	23
	Objetivo	23
	Justificación metodológica	23
	Selección de modelos a comparar	24
	Resultados	24
	Análisis del gráfico de caja y bigotes	24
	Síntesis	25
e)	Determinación del Punto de Corte Óptimo	25
	Resultado obtenido	25
	Interpretación	25
f)	Interpretación de Coeficientes del Modelo Ganador	27
	1. CCAA_Cataluña	27
	2. WomanPopulationPtge	28
	Interpretación:	28
6.	CONCLUSIÓN FINAL	29

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 $\text{AbstentionPtge} > 30\%$, 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

a) 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 quedó correctamente preparado para continuar con el análisis descriptivo.

b) 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, "Asimetría"] = datos[num].skew()
    descriptivos_num.loc[num, "Kurtosis"] = datos[num].kurtosis()
    descriptivos_num.loc[num, "Rango"] = np.ptp(datos[num].dropna().values)
```

descriptivos_num - DataFrame

índice	count	mean	std	min	25%	50%	75%	max	Asimetría	Kurtosis	Rango
Population	8117	5722.34	46204.2	5	166	548	2427	3.14199e+06	46.0407	2820.33	3.14199e+06
TotalCensus	8117	4247.86	34423.4	5	140	447	1843	2.36383e+06	46.5446	2893.45	2.36382e+06
AbstencionPtge	8117	26.5016	7.53344	0	21.678	26.424	31.471	57.576	-0.0537359	0.493671	57.576
AbstencionAlta	8117	0.311199	0.463012	0	0	0	1	1	0.815735	-1.3349	1
Age_0-4_Ptge	8117	3.01827	2.05263	0	1.389	2.975	4.533	13.245	0.343639	-0.206688	13.245
Age_under19_Ptge	8117	13.5641	6.77745	0	8.334	13.881	19.055	33.696	-0.104763	-0.79225	33.696
Age_19_65_pct	8117	57.3706	6.81864	23.459	53.845	58.655	61.818	100.002	-0.814264	2.15584	76.543
Age_over65_pct	8117	29.0653	11.767	-18.052	19.827	27.559	36.911	76.472	0.584708	0.102323	94.524
WomanPopulationPtge	8117	47.3023	4.36235	11.765	45.725	48.485	50	72.683	-1.6711	5.80063	60.918
ForeignersPtge	8117	5.61832	7.3487	-8.96	1.06	3.59	8.18	71.47	2.49826	11.3568	80.43
SameComAutonPtge	8117	81.6335	12.2873	0	75.806	84.493	90.462	127.156	-1.52276	3.47954	127.156
SameComAutonDiffProvPtge	8117	4.33764	6.39494	0	0.676	2.19	5.277	67.308	3.28683	14.5601	67.308
DifComAutonPtge	8117	10.7273	8.84763	0	4.933	8.269	13.891	100	2.42599	9.66397	100
UnemploymentLess25_Ptge	8117	7.32024	9.40888	0	0	5.882	10.467	100	4.15096	31.6648	100
Unemploy25_40_Ptge	8117	37.0013	20.3191	0	28.571	39.927	46.667	100	0.213481	1.41289	100
UnemployMore40_Ptge	8117	55.6785	22.0877	0	44.171	52	64.583	100	0.259701	0.705886	100
AgricultureUnemploymentPtge	8117	8.40287	12.9594	0	0	3.497	11.741	100	3.22892	15.5728	100
IndustryUnemploymentPtge	8117	10.0096	12.5295	0	0	7.143	14.286	100	3.08944	16.0472	100
ConstructionUnemploymentPtge	8117	10.8384	13.2827	0	0	8.333	14.286	100	3.0936	14.6282	100
ServicesUnemploymentPtge	8117	58.6468	24.2619	0	50	62	72.131	100	-0.805605	0.800001	100
totalEmpresas	8112	397.701	4219.49	0	7	30	147	299397	53.7136	3475.48	299397
Industria	7929	23.4053	158.628	0	0	0	14	10521	44.2709	2643.85	10521
Construccion	7978	48.8115	421.895	0	0	0	25	30343	52.5774	3506.44	30343
ComercTTEHosteleria	8188	146.209	1232.71	0	0	0	65	80856	45.4596	2652.63	80856
Servicios	8055	171.85	2447.04	0	0	0	40	177677	57.503	3833.62	177677
inmuebles	7979	3240.04	24314.7	6	180	485	1586.5	1.61555e+06	44.5615	2646.69	1.61554e+06
Pob2010	8110	5777.93	47527.9	5	177.25	581.5	2482.75	3.27305e+06	47.2006	2944.83	3.27304e+06
SUPERFICIE	8109	6215.3	9218.6	2.5784	1839.24	3488.55	6894.53	175023	6.07333	62.335	175020
PobChange_pct	8110	-4.90073	10.3824	-52.27	-10.4	-4.965	0.09	138.46	1.50644	15.1121	190.73
PersonasInmueble	7979	1.29556	0.565993	0.11	0.85	1.25	1.73	3.33	0.259748	-0.645897	3.22
Explotaciones	8117	2447.81	15064.5	1	22	52	137	99999	6.32127	37.9771	99998

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.
- **PersonasInmuble**: 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 **Construcción 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, Construcción, 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.

c) 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 'Construcción' e 'Industria' como 'Otras'
datos['ActividadPpal'] = datos['ActividadPpal'].replace({'Construcción': '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:

- 'Construcción'
- 'Industria'

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

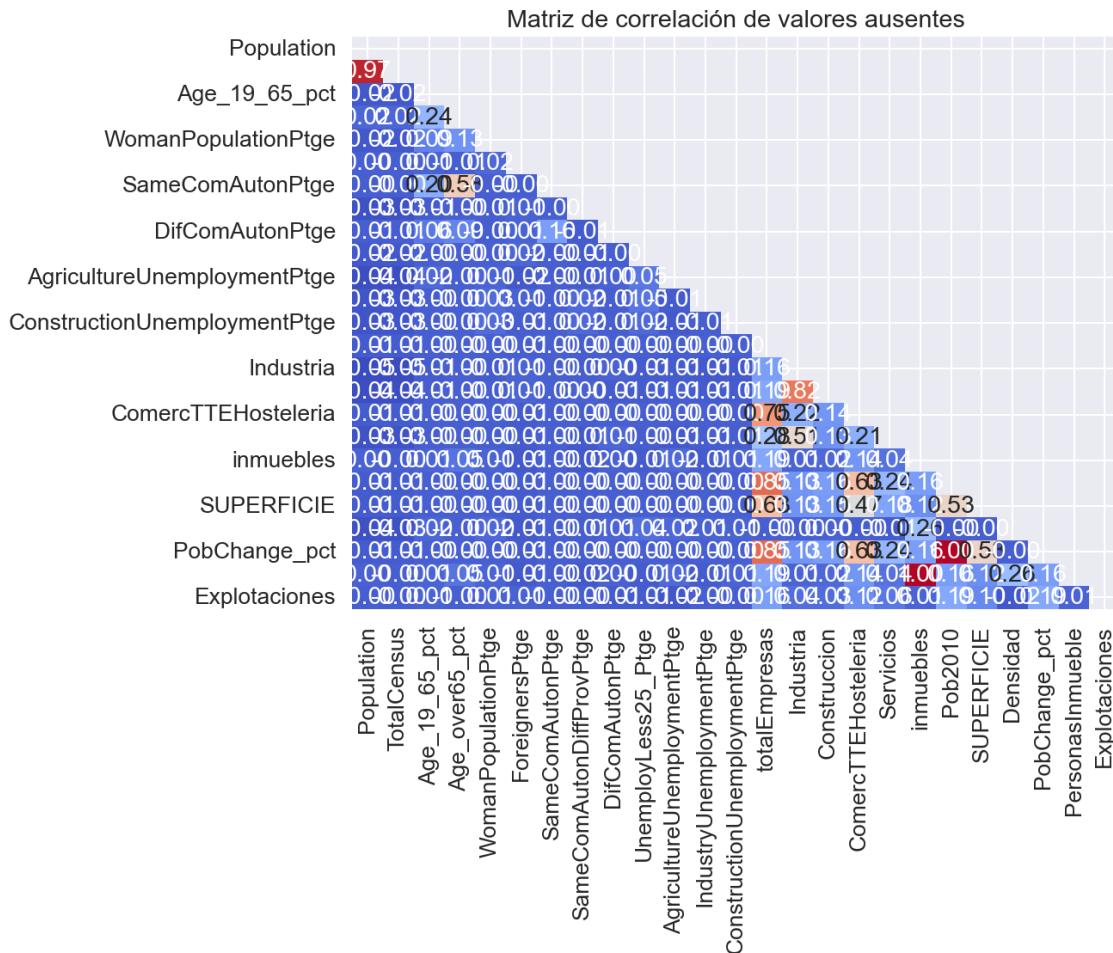
Tratamiento de valores atípicos

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.



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%

Variable	% de missing
Age_over65_pct	0.4%

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

3. CONSTRUCCIÓN DEL MODELO DE REGRESIÓN LINEAL

a) 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.

b) 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),
}

```

Resultados obtenidos

Índice	Método	R ² Train	R ² Test	Nº Parámetros
0	Backward AIC	0.64915	0.641702	74
1	Backward BIC	0.62638	0.629579	30
2	Forward AIC	0.646248	0.637385	66
3	Forward BIC	0.623681	0.628918	28
4	Stepwise AIC	0.646248	0.637385	66
5	Stepwise BIC	0.623537	0.628912	27

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.

c) 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:

Índice	Formula	Frecuencia
0	AbstencionAlta+Age_under19_Ptge+AgricultureUnemploymentPtge+CCAA_Aragón+CCAA_Asturias+CCAA_Baleares+CCAA_Canarias+CCAA_Cantabria+CCAA_CastillaLeón+CCAA_CastillaMancha+CCAA_Cataluña+CCAA_España+CCAA_Extranjero+CCAA_Galicia+CCAA_Huesca+CCAA_La Rioja+CCAA_Murcia+CCAA_Provincia+CCAA_Soria+CCAA_Teruel+CCAA_Vizcaya+CCAA_Zamora	4
1	AbstencionAlta+Age_under19_Ptge+AgricultureUnemploymentPtge+CCAA_Aragón+CCAA_Asturias+CCAA_Baleares+CCAA_Canarias+CCAA_Cantabria+CCAA_CastillaLeón+CCAA_CastillaMancha+CCAA_Cataluña+CCAA_España+CCAA_Extranjero+CCAA_Galicia+CCAA_Huesca+CCAA_La Rioja+CCAA_Murcia+CCAA_Provincia+CCAA_Soria+CCAA_Teruel+CCAA_Vizcaya+CCAA_Zamora	3
2	AbstencionAlta+Age_0-4_Ptge+AgricultureUnemploymentPtge+CCAA_Aragón+CCAA_Asturias+CCAA_Baleares+CCAA_Canarias+CCAA_Cantabria+CCAA_CastillaLeón+CCAA_CastillaMancha+CCAA_Cataluña+CCAA_España+CCAA_Extranjero+CCAA_Galicia+CCAA_Huesca+CCAA_La Rioja+CCAA_Murcia+CCAA_Provincia+CCAA_Soria+CCAA_Teruel+CCAA_Vizcaya+CCAA_Zamora	3
3	AbstencionAlta+Age_0-4_Ptge+AgricultureUnemploymentPtge+CCAA_Aragón+CCAA_Asturias+CCAA_Baleares+CCAA_Canarias+CCAA_Cantabria+CCAA_CastillaLeón+CCAA_CastillaMancha+CCAA_Cataluña+CCAA_España+CCAA_Extranjero+CCAA_Galicia+CCAA_Huesca+CCAA_La Rioja+CCAA_Murcia+CCAA_Provincia+CCAA_Soria+CCAA_Teruel+CCAA_Vizcaya+CCAA_Zamora	1

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.

d) 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 = freq_ordenada['Formula'][0]
formula2 = freq_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()
```

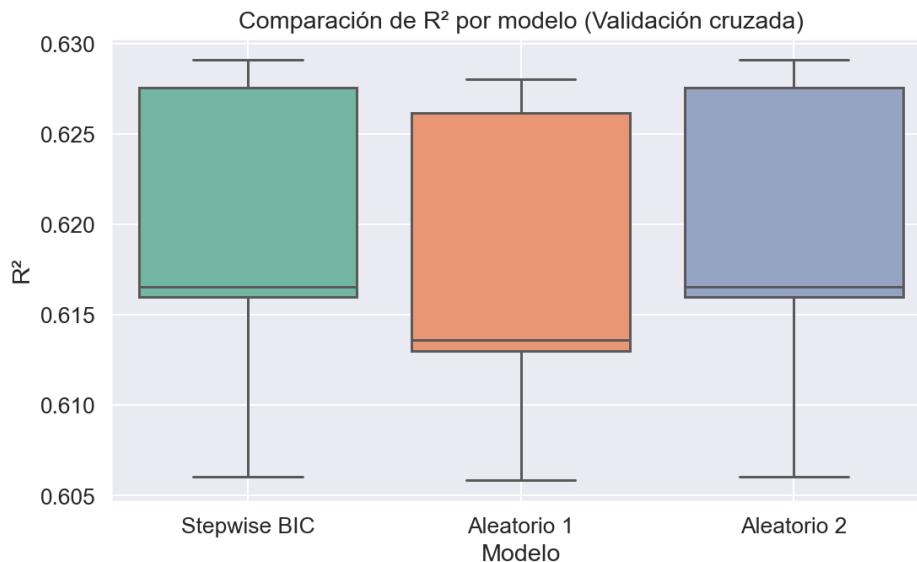
Resultados obtenidos

Los valores medios de R^2 en validación cruzada fueron los siguientes:

Modelo Evaluado	R^2 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 R^2 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 **buen estabilidad**.

Conclusión

Dado que el **modelo clásico (Stepwise BIC)** y el **modelo aleatorio 2** obtienen exactamente el mismo R^2 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.

e) 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

a) 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, preprocessamiento 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.

b) 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)
```

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.

c) 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)

```

Resultados obtenidos

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

Índice	Formula	Frecuencia
0	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+SameComAutonPtge+WomanPopulationPtge+const+inmuebles	4
1	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+DifComAutonPtge+Pob2010+Servicios+WomanPopulationPtge+const	3
2	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+DifComAutonPtge+WomanPopulationPtge+const+inmuebles	3
3	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+DifComAutonPtge+Population+Servicios+WomanPopulationPtge+const+inmuebles	2
4	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+Pob2010+SameComAutonPtge+Servicios+WomanPopulationPtge+const	2
5	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+Population+SameComAutonPtge+Servicios+WomanPopulationPtge+const+inmuebles	2
6	ActividadPpal+Age_19_65_pct+CCAA+CódigoProvincia+ComercioHosteleria+DifComAutonPtge+Pob2010+SameComAutonDiffProvPtge+WomanPopulationPtge+const	1

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, CódigoProvincia, 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.

d) 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.

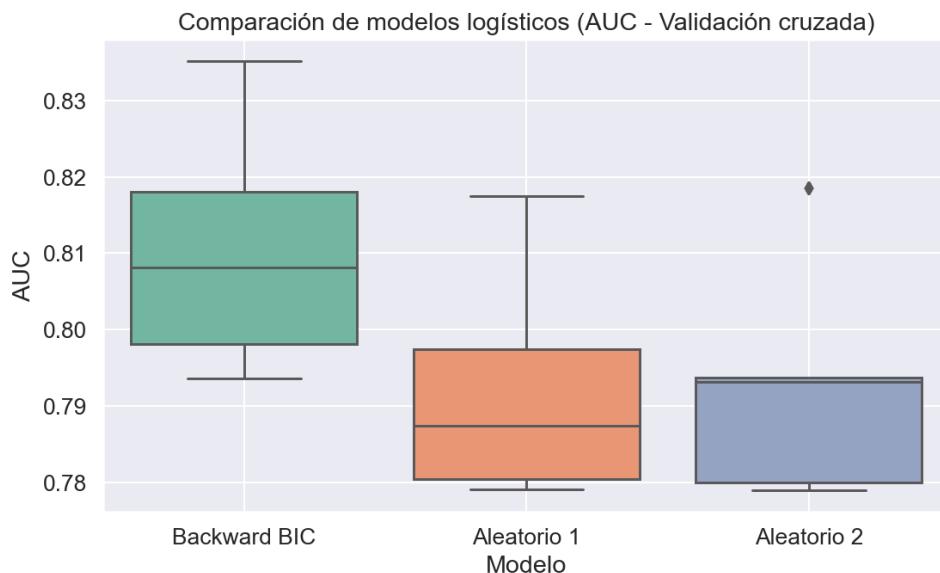
Resultados

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



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.

e) 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."

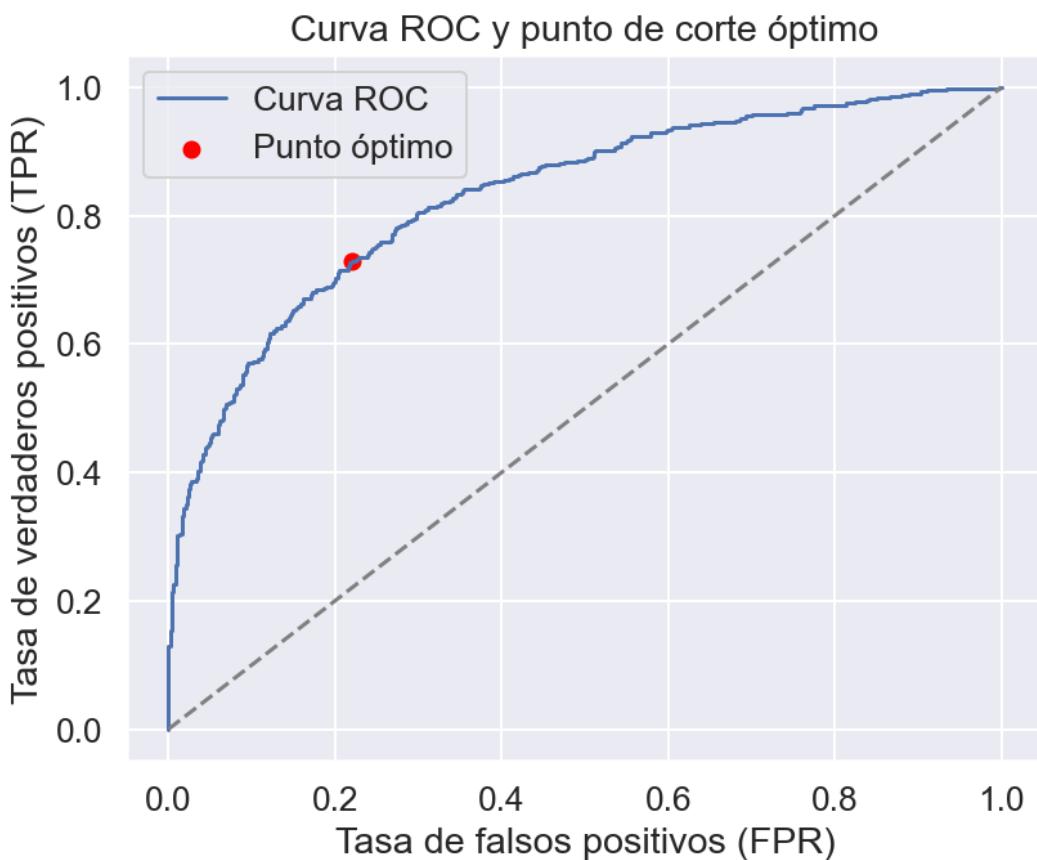
# Predecir probabilidades de clase positiva
probabilidades = modelo.predict_proba(X_test_log)[:, 1]

# 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)
```



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.

f) 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: $e^{1.876} \approx 6.53$

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: $e^{-0.042} \approx 0.959$

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 (R^2 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.