Módulo 7: Minería de datos y modelización predictiva

UNIVERSIDAD
COMPLUTENSE
MADRID

Profesor: Rosa Espinola
Alumno: Pablo Pérez Calvo

Índice

1. Depuración de datos

1.1. Introducción al objetivo y variables implicadas.

El objetivo de esta práctica es, a partir de una base de datos de las elecciones en cada municipio de España, obtener dos modelos de regresión, lineal y logística. La variable continua que se va a utilizar para la construcción del modelo de regresión lineal es *"Izquierda_Pct"*, correspondiente al porcentaje de votos a partidos de izquierda, y la variable binaria utilizada para el de regresión logística es *"Izquierda"*, variable dicotómica que toma el valor 1 si la suma de los votos de izquierda es superior a la de derechas y 0 en caso contrario.

El modelo de regresión lineal nos permitirá estudiar la relación de todas las variabes con nuestra variable objetivo y el de regresión logística, predecir la probabilidad de pertenecer a cada clase.

El resto de variables que presenta la base de datos y que se utilizarán en la práctica son las siguientes.

Variable	Descripción
Name	Nombre del municipio
CodigoProvincia	Código de la provincia (coincide con los dos primeros dígitos del codigo postal).
CCAA	Comunidad autónoma a la que pertenece el municipio
Population	Población del municipio en 2016
TotalCensus	Población en edad de votar en 2016
Age_0-4_Ptge **Age_under19_Ptge** **Age_19_65_pct** **Age_over65_pct**	Porcentaje de ciudadanos por edad
WomanPopulationPtge	Porcentaje de mujeres
ForeignersPtge	Porcentaje de extranjeros
SameComAutonPtge **SameComAutonDiffProvPtge** **DifComAutonPtge**	Porcentaje de ciudadanos por lugar de nacimiento y residencia.
UnemployLess25_Ptge **Unemploy25_40_Ptge** **UnemployMore40_Ptge**	Porcentaje de parados por edad
AgricultureUnemploymentPtge **IndustryUnemploymentPtge** **ConstructionUnemploymentPtge** **ServicesUnemploymentPtge**	Porcentaje de parados por sector
totalEmpresas	Número total de empresas en el municipio
Industria **Construccion** **ComercTTEHosteleria** **Servicios**	Número de empresas por sector en el municipio
ActividadPpal	Actividad principal de las actividades del municipio
inmuebles	Número de inmuebles en el municipio
Pob2010	Población en el municipio en 2010
SUPERFICIE	Superficie del municipio
densidad	Densidad de población del municipio: MuyBaja (<1 hab/ha), Baja (entre 1 y 5 hab/ha), Alta (>5 hab/ha)
PobChange_pct	Porcentaje de cambio en la población respecto a las anteriores elecciones
PersonasInmueble	Número medio de personas que habita un inmueble
Explotaciones	Número de explotaciones agrícolas en el municipio

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

Para importar los datos y eliminar el resto de variables objetivo que no hemos elegido, usamos el siguiente código:

```
datos = pd.read_excel('DatosEleccionesEspana.xlsx')
datos = datos.drop(['Dcha_Pct', 'Derecha', 'Otros_Pct',
 'AbstentionPtge', 'AbstencionAlta', 'CodigoProvincia'],axis=1)
```

Se ha decidido eliminar también la variable *Codigoprovincia* porque la información de esa variable ya la tenemos en *CCAA*.

La variable *Name* se utilizará como índice porque no aporta información para el objetivo de este problema.

```
datos = datos.set_index(datos['Name']).drop('Name', axis = 1)
```

Vamos a observar los tipos de las variables para revisar que los datos se han leído correctamente.

```
datos.dtypes
```

Izquierda int64

Las variable *Izquierda* se ha asignado como numérica cuando debería ser categórica, vamos a corregirlo.

```
numericasAcategoricas = ['Izquierda']

# Las transformo en categoricas
for var in numericasAcategoricas:
    datos[var] = datos[var].astype(str)
```

1.3. Análisis descriptivo del conjunto de datos.

Realizaremos el análisis descriptivo de las variables cuantitativas con el siguiente código.

```python
descriptivos_num = datos.describe().T

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)
```

Index	count	mean	std	min	25%	50%	75%	max	Asimetria	Kurtosis	Rango
Izda_Pct	8117	34.404	16.4843	0	21.891	35.165	46.032	94.117	0.0598817	-0.493538	94.117
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
UnemployLess25_Ptge	8117	7.32024	9.4088	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	1.41209	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	14.6202	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

Figura 1: Análisis descriptivo de las variables cuantitativas

Podemos observar como la variable *Explotaciones* tiene valores codificados como 99999 y que existen algunas variables correspondientes a porcentajes, cuyos valores no pertenecen al intervalo $[0, 100]$, excepto $PobChange\_pct$ cuyos valores no tienen por qué pertenecer a ese intervalo.

Haremos uso de la función analizar_variables_categoricas de *FuncionesMineria* para analizar las variables cualitativas.

```
    'CCAA':        n         %
CastillaLeon     2248   0.276950
Cataluna          947   0.116669
CastillaMancha    919   0.113219
Andalucia         773   0.095232
Aragon            731   0.090058
ComValenciana     542   0.066773
Extremadura       387   0.047678
Galicia           314   0.038684
Navarra           272   0.033510
PaisVasco         251   0.030923
Madrid            179   0.022052
Rioja             174   0.021436
Cantabria         102   0.012566
Canarias           88   0.010841
Asturias           78   0.009609
Baleares           67   0.008254
Murcia             45   0.005544,
'Izquierda':        n         %
0   6308   0.777134
1   1809   0.222866,
'ActividadPpal':      n         %
Otro                 4932   0.607614
ComercTTEHosteleria  2538   0.312677
Servicios             620   0.076383
Construccion           14   0.001725
Industria              13   0.001602,
'Densidad': n      %
MuyBaja   6416   0.790440
Baja      1053   0.129728
Alta       556   0.068498
?           92   0.011334}
```

Podemos encontrar en la variable *densidad*, valores missings declarados como *"?"*. Además, las categorías *Industria* y *Construcción* de *ActividadPpal* y la mayoría de categorías de *CCAA* están poco representadas.

1.4. Corrección de los errores detectados

- Los valores 99999 que aparecen en *Explotaciones* se trataran como valores perdidos.

```
datos['Explotaciones'] = datos['Explotaciones'].replace(99999, np.nan)
```

- Las variables correspondientes a porcentajes se ajustarán al rango $[0, 100]$, los valores que superen estos límites se trataran como valores perdidos. Esto se realizará solo a las variables que se hayan detectado con valores fuera de rango en el apartado anterior.

```
datos['Age_19_65_pct'] = [x if 0 <= x <= 100 else np.nan for x in datos[
     'Age_19_65_pct']]
datos['Age_over65_pct'] = [x if 0 <= x <= 100 else np.nan for x in datos
     ['Age_over65_pct']]
datos['ForeignersPtge'] = [x if 0 <= x <= 100 else np.nan for x in datos
     ['ForeignersPtge']]
datos['SameComAutonPtge'] = [x if 0 <= x <= 100 else np.nan for x in
     datos['SameComAutonPtge']]
```

- Los valores *"?"* de la variable *Densidad* se sustituirán por *np.nan*.

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

- Las categorías *Industria* y *Construcción* serán agrupadas con la categoría *Otro* debido a su baja representación.

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

- Por último, debido a la poca representación encontrada en algunas comunidades, agruparemos las categorías de *CCAA* por su renta per cápita.

```
ccaa = {
'Madrid': 'Madrid - Pais Vasco - Navarra ',
'PaisVasco': 'Madrid - Pais Vasco - Navarra ',
'Navarra': 'Madrid - Pais Vasco - Navarra ',
'Cataluna': 'Cataluna - Aragon- Baleares ',
'Aragon': 'Cataluna - Aragon - Baleares ',
'Baleares': 'Cataluna - Aragon - Baleares ',
'Rioja': 'La Rioja - Castilla y Leon',
'CastillaLeon': 'La Rioja - Castilla y Leon',
'Cantabria': 'Cantabria - Galicia - Asturias ',
'Galicia': 'Cantabria - Galicia - Asturias ',
'Asturias': 'Cantabria - Galicia - Asturias ',
'ComValenciana': 'Comunidad Valenciana - Murcia',
'Murcia': 'Comunidad Valenciana - Murcia',
'CastillaMancha': 'Castilla -La Mancha - Canarias ',
'Canarias': 'Castilla -La Mancha - Canarias ',
'Andalucia': 'Andalucia - Extremadura ',
'Extremadura': 'Andalucia - Extremadura '
}
datos['CCAA'] = datos['CCAA'].replace(ccaa)
```

1.5. Análisis de valores atípicos.

Haciendo uso de la función *atipicosaMissing* de *FuncionesMineria*, calculamos la proporción de valores atípicos para cada columna numérica.

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

Además, estos valores atípicos se tratarán como valores perdidos.

```
for x in numericas_input: datos_input[x] = atipicosAmissing(datos_input[
    x])[0]
```

1.6. Análisis de valores perdidos.

Primero de todo, visualizaremos la matriz de correlación de valores perdidos entre las diferentes variables.

```
patron_perdidos(datos_input)
```

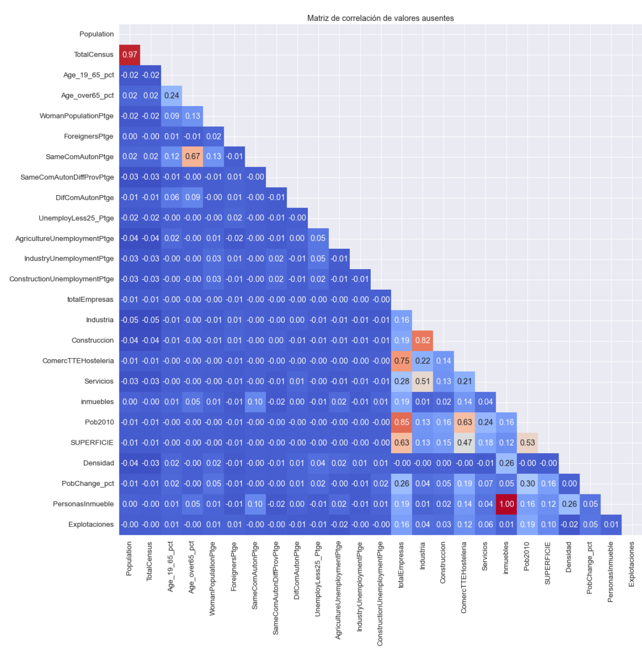


Figura 2: Matriz de correlación de valores perdidos

En segundo lugar, calcularemos el número de valores perdidos que tiene cada variable y su proporción.

```
# Muestra total de valores perdidos por cada variable
datos_input[variables_input].isna().sum()

# Muestra proporcion de valores perdidos por cada variable
prop_missingsVars = datos_input.isna().sum()/len(datos_input)
```

Podemos observar que la variable con mayor proporción de valores perdidos es *Population* con un $0,099$, luego como no hay ninguna que supere el $50\,\%$ no eliminamos ninguna variable.

Por último, calculamos el número de valores perdidos por observación y se realiza un estudio descriptivo.

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

Según los resultados, como ninguna variable supera el $50\,\%$ no eliminaremos ninguna de las observaciones.

Ahora vamos a sustituir estos valores perdidos por valores válidos, esto es lo que se conoce como **imputación**. Utilizaremos, tanto para las variables cualitativas como para las cuantitativas, la imputación aleatoria, es decir, los valores pérdidos se sustituirán por valores aleatorios manteniendo la distribución de la variable.

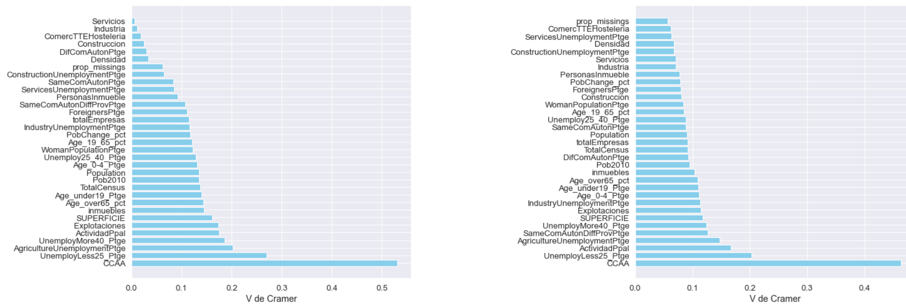
```
for x in numericas_input: datos_input[x] = ImputacionCuant(datos_input[x
    ], 'aleatorio')

for x in categoricas_input: datos_input[x] = ImputacionCuali(datos_input
    [x], 'aleatorio)
```

1.7. Detección de las relaciones entre las variables input continuas, así como las relaciones entre todas las variables input y cada una de las variables objetivo

En primer lugar, existen algunas variables que pueden ser calculadas a partir de otras, son linealmente dependientes por lo que no aportan nueva información. Por tanto, eliminaremos estas variables para reducir la complejidad del modelo.

```
datos_input = datos_input.drop(['Age_over65_pct', 'DifComAutonPtge', '
    Unemploy25_40_Ptge', 'AgricultureUnemploymentPtge'] , axis = 1)
```

En segundo lugar, vamos a ver la relación que tienen las demás variables con las dos variables objetivos haciendo uso de la función *graficoVCramer*.



(a) Variable binaria *Izquierda* (b) Variable continua *Izquierda_Pct*

Figura 3: V de Cramer

Se observa claramente que la variable *CCAA* es la que más relación tiene con ambas variables objetivos, *Servicios* es la que menos relación tiene con *Izquierda* y *prop_missings* la que menos relación tiene con *Izquierda_Pct*.

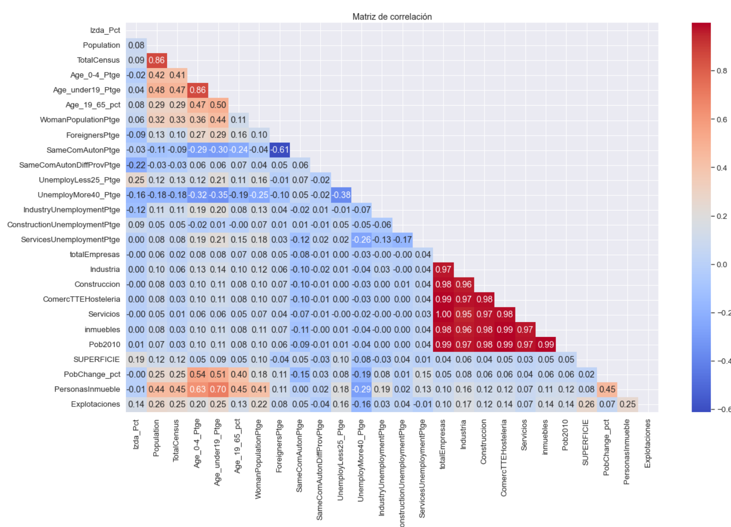


Figura 4: Matriz de correlación de las variables numéricas

Si observamos la matriz de correlación de las variables numéricas, vemos como *totalEmpresas* tiene gran correlación con 6 variables por lo que las podemos eliminar.

```
datos_input =datos_input.drop(['Pob2010','Industria','Construccion',
'ComercTTEHosteleria', 'Servicios', 'inmuebles'] , axis = 1)
```

2. Construcción del modelo de regresión lineal

2.1. Selección de variables mediante métodos clásicos

Una vez que hemos depurado la base de datos inicial, eliminando variables redundantes para el objetivo de nuestro problema, vamos a proceder a la construcción de un modelo de regresión lineal mediante los métodos de selección de variablels clásicos.

- **Método BackWard**: Este método comienza con todas las variables en el modelo y elimina gradualmente las menos relevantes. En cada iteración, se prueba eliminar una de ellas y se observa el rendimiento del modelo con y sin ella. si al eliminarla el modelo no es significativamente peor se descarta. Una vez descartada una variable no puede volver a entrar al modelo.
- **Método Forward**: Este método comienza con el modelo vacío y agrega variables progresivamente. Se inicia con la variable más relevante y en cada iteración se añade la que más mejora al rendimiento del modelo hasta alcanzar un conjunto óptimo de variables. Una vez añadida una variable al modelo, no puede salir.
- **Método StepWise**: Este método combina los métodos anteriores, agrega variables que mejoran el rendimiento y elimina las menos significativas en cada iteración. Es recomendable incluir un número máximo de iteraciones para evitar bucles.

Vamos a construir 6 modelos, aplicando los tres métodos con los criterios AIC y BIC, sin usar transformaciones y utilizando interacciones solo entre las variables continuas.

```
x_train, x_test, y_train, y_test = train_test_split(datos_input,
    varObjCont, test_size = 0.2, random_state = 1234567)

modeloStepAIC = lm_stepwise(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'AIC')
modeloBackAIC = lm_backward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'AIC')
modeloForwAIC = lm_forward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'AIC')
modeloStepBIC = lm_stepwise(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'BIC')
modeloBackBIC = lm_backward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'BIC')
modeloForwBIC = lm_forward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'BIC')
```

Los resultados obtenidos tras aplicar los distintos métodos de selección se recogen en la siguiente tabla.

Método	**Métrica**	R^2**Train**	R^2**Test**	**Nº Parametros**
BackWard	AIC	0.619789	0.582542	89
Forward	AIC	0.616781	0.590118	69
Stepwise	AIC	0.616431	0.591853	63
BackWard	BIC	0.608961	0.580071	36
Forward	BIC	0.604563	0.585485	29
Stepwise	BIC	0.603670	0.588953	28

Podemos observar en la tabla que los valores de R^2 son similares para todos los métodos aplicados, sin embargo, el número de parámetros que utiliza cada modelo es significativamente diferente. Por ello,seleccionamos el método *Stepwise* con el criterio **BIC** como el modelo más adecuado, siguiendo el principio de parsimonia.

2.2. Selección de variables aleatoria

La selección de variables aleatoria consiste en generar varias submuestras aleatorias, realizar una selección de variables "clásica" y generar una tabla resumen con los modelos generados en cada una de las submuestras.

Se realizará la selección aleatoria con el método *Stepwise* y criterio **BIC** porque en el apartado anterior vimos que era el más adecuado, el código que se ha utilizado es el siguiente.

```
variables_seleccionadas = {
    'Formula': [],
    'Variables': []}

# Realizar 30 iteraciones de seleccion aleatoria.
for x in range(30):
    print('--------------------------- iter: ' + str(x))

    # Dividir los datos de entrenamiento en conjuntos de entrenamiento y prueba.
    x_train2, x_test2, y_train2, y_test2 = train_test_split(x_train, y_train, test_size = 0.3, random_state = 1234567 + x)

    # Realizar la seleccion stepwise utilizando el criterio BIC en la submuestra.
    modelo = lm_stepwise(y_train2.astype(int), x_train2, var_cont, var_categ, interacciones_unicas, 'BIC')

    # Almacenar las variables seleccionadas y la formula correspondiente.
    variables_seleccionadas['Variables'].append(modelo['Variables'])
    variables_seleccionadas['Formula'].append(sorted(modelo['Modelo'].model.exog_names))

# Unir las variables en las formulas seleccionadas en una sola cadena.
variables_seleccionadas['Formula'] = list(map(lambda x: '+'.join(x), variables_seleccionadas['Formula']))

# Calcular la frecuencia de cada formula y ordenarlas por frecuencia.
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()

# Identificar las dos modelos mas frecuentes y las variables correspondientes.
var_1 = variables_seleccionadas['Variables'][variables_seleccionadas['Formula'].index(
    frec_ordenada['Formula'][0])]
var_2 = variables_seleccionadas['Variables'][variables_seleccionadas['Formula'].index(
    frec_ordenada['Formula'][1])]
```

2.3. Selección del modelo ganador

Para seleccionar el modelo ganador, utilizaremos la validación cruzada entre el método clásico Stepwise con criterio BIC y los dos obtenidos en la selección de variables aleatoria.

Modelo	Media R^2	Desviación Típica R^2	Nº Parametros
Modelo 1	0.598993	0.017682	28
Modelo 2	0.594516	0.019126	859
Modelo 3	0.595925	0.020907	23

El primer modelo es el que tiene mayor valor medio de R^2 y el que menor variabilidad presenta; sin embargo, utiliza 28 parámetros. Dado que la diferencia de

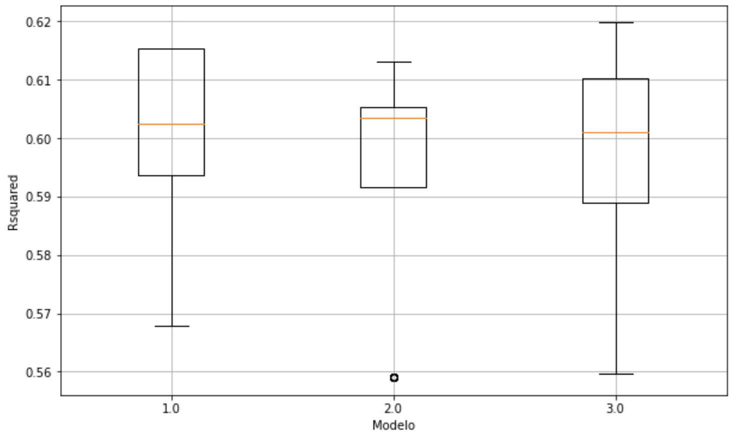


Figura 5: Validación cruzada modelos regresión lineal

valores con el tercer modelo es insignificante y este último emplea solo 23 parámetros, seleccionamos el **tercer modelo** como el modelo ganador siguiendo el principio de parsimonia.

```
ModeloGanador = lm(y_train, x_train, var_2 ['cont'], var_2['categ'],
    var_2 ['inter'])
```

2.4. Interpretación de los coeficientes de dos variables incluidas en el modelo ganador, una binaria y otra continua

El resumen del modelo ganador es el siguiente:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Izda_Pct   R-squared:                       0.600
Model:                            OLS   Adj. R-squared:                  0.598
Method:                 Least Squares   F-statistic:                     440.8
Date:                Tue, 04 Feb 2025   Prob (F-statistic):               0.00
Time:                        17:23:08   Log-Likelihood:                -24429.
No. Observations:                6493   AIC:                         4.890e+04
Df Residuals:                    6470   BIC:                         4.906e+04
Df Model:                          22
Covariance Type:            nonrobust
==============================================================================================================
                                                  coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
const                                          52.8550      2.042     25.884      0.000      48.852      56.858
SameComAutonPtge                               -0.1880      0.015    -12.513      0.000      -0.217      -0.159
ServicesUnemploymentPtge                        0.1233      0.032      3.812      0.000       0.060       0.187
UnemployLess25_Ptge                            -0.7504      0.180     -4.172      0.000      -1.103      -0.398
CCAA_Cantabria - Galicia - Asturias           -20.7987      0.699    -29.736      0.000     -22.170     -19.428
CCAA_Castilla -La Mancha - Canarias           -10.5797      0.577    -18.338      0.000     -11.711      -9.449
CCAA_Cataluña - Aragón - Baleares              -8.2766      0.603    -13.715      0.000      -9.460      -7.094
CCAA_Cataluña - Aragón- Baleares              -40.2902      0.604    -66.694      0.000     -41.474     -39.106
CCAA_Comunidad Valenciana - Murcia            -28.1669      0.638    -44.178      0.000     -29.417     -26.917
CCAA_La Rioja - Castilla y León               -16.8069      0.496    -33.873      0.000     -17.780     -15.834
CCAA_Madrid - Pais Vasco - Navarra            -16.1655      0.614    -26.316      0.000     -17.370     -14.961
ActividadPpal_Otro                             -1.5556      0.413     -3.768      0.000      -2.365      -0.746
ActividadPpal_Servicios                        -2.3543      0.562     -4.192      0.000      -3.455      -1.253
Age_19_65_pct_WomanPopulationPtge               0.0049      0.000     10.472      0.000       0.004       0.006
WomanPopulationPtge_UnemployLess25_Ptge         0.0180      0.004      4.550      0.000       0.010       0.026
Population_UnemployLess25_Ptge                  0.0001   1.53e-05      9.213      0.000       0.000       0.000
Age_19_65_pct_ForeignersPtge                   -0.0035      0.000     -8.845      0.000      -0.004      -0.003
WomanPopulationPtge_ServicesUnemploymentPtge   -0.0021      0.001     -2.926      0.003      -0.003      -0.001
Population_ServicesUnemploymentPtge          -1.48e-05   2.24e-06     -6.601      0.000   -1.92e-05   -1.04e-05
SameComAutonDiffProvPtge_PersonasInmueble      -0.1173      0.024     -4.966      0.000      -0.164      -0.071
Age_0-4_Ptge_PobChange_pct                     -0.0169      0.005     -3.646      0.000      -0.026      -0.008
Age_19_65_pct_ConstructionUnemploymentPtge      0.0011      0.000      4.844      0.000       0.001       0.001
TotalCensus_Explotaciones                    -9.49e-07   2.62e-07     -3.622      0.000   -1.46e-06   -4.35e-07
==============================================================================
Omnibus:                      188.876   Durbin-Watson:                   2.019
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              325.602
Skew:                           0.251   Prob(JB):                     1.98e-71
Kurtosis:                       3.976   Cond. No.                     1.05e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.05e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

- **Variable categórica: CCAA_Madrid - País Vasco - Navarra**: El coeficiente asociado a esta variable es $-16{,}1655$, esto indica que la zona de Madrid, País Vasco y Navarra tienen $16{,}17$ puntos porcentuales menos en el porcentaje de votos a la izquierda que la categoría de referencia, Andalucía - Extremadura, manteniendo las demás variables del modelo constantes. Al ser el p-valor asociado $0 < 0{,}05$ es un coeficiente significativo.

- **Variable continua: UnemployLess25_Ptge**: El coeficiente asociado a esta variable es $-0{,}7504$, esto indica que por cada unidad de aumento en esta variable, el porcentaje de votos a la izquierda disminuye $0{,}7504$ en promedio, manteniendo las demás variables constantes. El p-valor también es 0 por lo que es estadísticamente significativo.

2.5. Justificar por qué es el modelo ganador y medir la calidad del mismo.

- El modelo presenta un $R^2 = 0{,}6$, explica el $60\,\%$ de la variabilidad en el porcentaje de votos a la izquierda. El R^2 ajustado es $0{,}598$, lo que respalda el buen ajuste del modelo.

- Todas las variables son significativas, debido a que todas presentan un $p-valor < 0{,}05$.

- Los valores del *AIC* y *BIC* son relativamente bajos, logran un equilibrio entre complejidad y ajuste.

- $F-statistic = 440{,}8$ y $p-valor < 0{,}0001$, esto indica que al menos una de las variables explicativas tiene un efecto significativo sobre la variable dependiente, lo que valida la utilidad del modelo.

- El valor de *Durbin-Watson* es $2{,}019$, al ser un valor cercano a 2 nos indica que los residuos están incorrelados.

- La prueba de *Jarque-Bera* tiene un *p-valor* asociado cercano a 0, lo que nos sugiere que los residuos no siguen una distribución normal. En muestras grandes esto no siempre supone un problema.

El modelo presenta un gran equilibrio entre capacidad explicativa y simplicidad. Aunque la normalidad de los residuos puede ser un problema, el valor del R^2 y la significancia de las variables, respaldan al modelo como una herramienta útil con buena capacidad predictiva.

Vamos a analizar la importancia de las variables en el modelo ganador con el siguiente código.

```
modelEffectSizes(ModeloGanador, y_train, x_train, ModeloGanador['
    Variables']['cont'], ModeloGanador['Variables']['categ'],
    ModeloGanador['Variables']['inter'])
```

Esta función muestra cuánto varía el valor de R^2 al eliminar cada una de las variables. De esta forma podemos determinar cuales son las variables que más aportan al modelo.

Como podemos observar en los resultados, la variable *CCAA* es la más determinante para predecir el porcentaje de votos a los partidos de izquierda.

```
Variables        R2
 TotalCensus_Explotaciones  0.000811
 Age_0-4_Ptge_PobChange_pct  0.000822
 ServicesUnemploymentPtge  0.000899
UnemployLess25_Ptge  0.001077
Age_19_65_pct_ConstructionUnemploymentPtge  0.001451
SameComAutonDiffProvPtge_PersonasInmueble  0.001526
ActividadPpal  0.001572
Population_ServicesUnemploymentPtge  0.003622
Age_19_65_pct_ForeignersPtge  0.004839
Population_UnemployLess25_Ptge  0.008050
Age_19_65_pct_WomanPopulationPtge  0.009328
SameComAutonPtge  0.009685
CCAA  0.356476
```

3. Construcción del modelo de regresión logística

3.1. Selección de variables mediante métodos clásicos

De forma análoga al modelo de regresión lineal, vamos a construir un modelo de regresión logística considerando la variable categórica *Izquierda* como objetivo, aplicando los 3 métodos de selección de variables clásica con los criterios *AIC* y *BIC*, sin usar transformaciones.

Debido a las limitaciones computacionales y al tiempo de ejecución prolongado al incluir interacciones entre todas las variables continuas, se ha decidido seleccionar 4 variables continuas para aplicar interacciones.

```
interacciones = ['Population', 'totalEmpresas','Age_under19_Ptge', '
    UnemployLess25_Ptge']
interacciones_unicas = list(itertools.combinations(interacciones, 2))
```

```
x_train, x_test, y_train, y_test = train_test_split(datos_input,
    varObjBin, test_size = 0.2, random_state = 1234567)
y_train, y_test = y_train.astype(int), y_test.astype(int)

modeloStepAIC = glm_stepwise(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'AIC')
modeloBackAIC = glm_backward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'AIC')
modeloForwAIC = glm_forward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'AIC')
modeloStepBIC = glm_stepwise(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'BIC')
modeloBackBIC = glm_backward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'BIC')
modeloForwBIC = glm_forward(y_train, x_train, var_cont, var_categ,
    interacciones_unicas, 'BIC')
```

Los resultados obtenidos tras aplicar los distintos métodos de selección se recogen en la siguiente tabla.

Método	**Métrica**	*pseudo* R^2 **Train**	*pseudo* R^2 **Test**	**Nº Parametros**
BackWard	AIC	0.292739	0.305018	17
Forward	AIC	0.295209	0.303365	23
Stepwise	AIC	0.294559	0.301383	21
BackWard	BIC	0.292739	0.305018	17
Forward	BIC	0.295209	0.303365	23
Stepwise	BIC	0.294559	0.301383	21

Podemos observar en la tabla que los valores de $pseudo\ R2$ son similares para todos los métodos aplicados. Por ello, seleccionamos el método que utiliza menos parámetros, **Backward** con el criterio **BIC**, como el modelo más adecuado, siguiendo el principio de parsimonia.

3.2. Selección de variables aleatoria

Se realizará la selección aleatoria con el método **Backward** con el criterio **BIC** y se utilizará un código similar al utilizado en la selección de variables aleatoria de la regresión lineal. En este caso, se realizarán 20 iteraciones e identificaremos las 3 fórmulas más frecuentes.

```
var_1 = variables_seleccionadas['Variables'][variables_seleccionadas['
     Formula'].index(
    frec_ordenada['Formula'][0])]
var_2 = variables_seleccionadas['Variables'][variables_seleccionadas['
     Formula'].index(
    frec_ordenada['Formula'][1])]
var_3 = variables_seleccionadas['Variables'][variables_seleccionadas['
     Formula'].index(
    frec_ordenada['Formula'][2])]
```

3.3. Selección del modelo ganador

Para determinar el modelo ganador, realizaremos validación cruzada entre el modelo Backward con criterio BIC y los 3 obtenidos en la selección de variables aleatoria. En este caso, la medida usada para comparar los modelos será el *AUC*, área bajo la curva ROC.

Modelo	**Media AUC**	**Desviacion Tipica AUC**	**Nº Parametros**
Modelo 1	0.8444	0.01311	17
Modelo 2	0.8413	0.01279	13
Modelo 3	0.8439	0.01344	16
Modelo 4	0.8439	0.01217	16

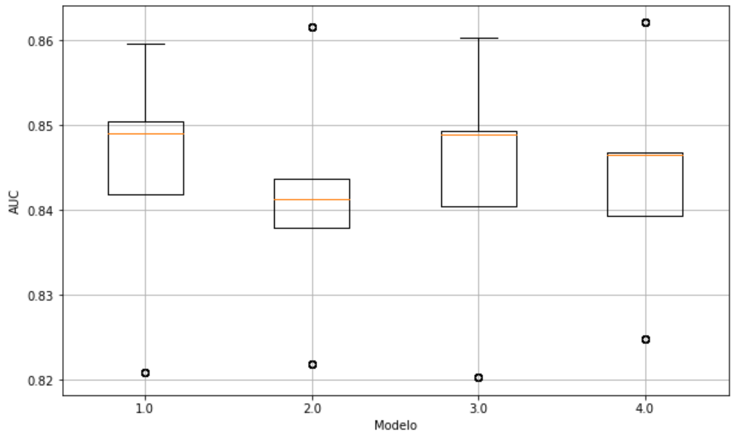


Figura 6: Validación cruzada regresión logística

Los modelos 1, 3 y 4 tienen valor medio de *AUC* muy similares, el modelo 4 es el que menor desviación típica tiene lo que indica que es el más consistente. Además, solo utiliza 16 parámetros por lo que elegiremos como modelo ganador al **cuarto modelo**. Es cierto que el segundo modelo es el que menos parámetros usa pero también es el que menos media de *AUC* presenta.

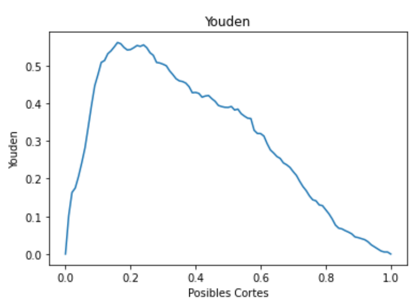
```
ModeloGanador = glm(y_train, x_train, var_3['cont'], var_3['categ'],
    var_3['inter'])
```

3.4. Determinar el punto de corte óptimo

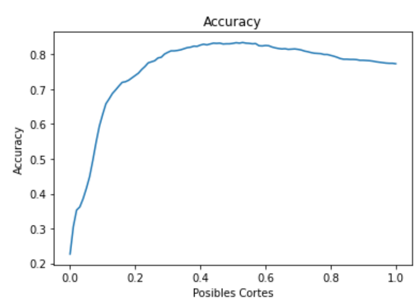
A diferencia de los modelos de regresión lineal es que para regresión logística es necesario determinar la probabilidad a partir de la cual se consideraría una observación como evento, $Izquierda = 1$.

Para ello, vamos a obtener, para una rejilla de posibles puntos de corte y el conjunto de datos de prueba, el valor de la tasa de acierto, la sensibilidad, la especificidad y el índice de Youden con el siguiente código.

```python
# Generamos una rejilla de puntos de corte
posiblesCortes = np.arange(0, 1.01, 0.01).tolist()  # Generamos puntos de corte de 0 a 1 con intervalo de 0.01
rejilla = pd.DataFrame({
    'PtoCorte': [],
    'Accuracy': [],
    'Sensitivity': [],
    'Specificity': [],
    'PosPredValue': [],
    'NegPredValue': []
})
for pto_corte in posiblesCortes:
    rejilla = pd.concat(
        [rejilla, sensEspCorte(ModeloGanador['Modelo'], x_test, y_test, pto_corte, ModeloGanador['Variables']['cont'] ,
    ModeloGanador['Variables'][ 'categ'] , ModeloGanador['Variables'][ 'inter'])],
        axis=0
    )
rejilla['Youden'] = rejilla['Sensitivity'] + rejilla['Specificity'] - 1  # Calculamos el indice de Youden

# Graficamos los posibles puntos de corte
plt.plot(rejilla['PtoCorte'], rejilla['Youden'])
plt.xlabel('Posibles Cortes')
plt.ylabel('Youden')
plt.title('Youden')
plt.show()

plt.plot(rejilla['PtoCorte'], rejilla['Accuracy'])
plt.xlabel('Posibles Cortes')
plt.ylabel('Accuracy')
plt.title('Accuracy')
plt.show()
```



(a) Índice de Youden



(b) Accuracy

Figura 7: Posibles puntos de corte

Existen diferentes estrategias para determinar el punto de corte óptimo, entre ellas se encuentran la de optimizar la tasa de acierto y el índice de Youden.

```
rejilla['PtoCorte'][rejilla['Youden'].idxmax()]
rejilla['PtoCorte'][rejilla['Accuracy'].idxmax()]
```

El resultado obtenido es $0{,}16$ para el índice de Youden y $0{,}53$ para Accuracy. Vamos a compararlos con la función *sensEspCorte*.

```
PtoCorte   Accuracy   Sensitivity   Specificity   PosPredValue  NegPredValue
0.16   0.719828      0.891304      0.669586       0.441454       0.954597

PtoCorte   Accuracy   Sensitivity   Specificity   PosPredValue  NegPredValue
0.53    0.83436      0.432065      0.952229       0.726027       0.851246
```

Como nuestro objetivo es predecir de manera correcta los valores de la variable *Izquierda*, priorizamos la tasa de acierto. Por ello, nos quedaremos con el punto de corte $0{,}53$ porque tiene mayor *Accuracy*.

3.5. Interpretación de los coeficientes de dos variables incluidas en el modelo ganador, una binaria y otra continua.

El resumen del modelo ganador es el siguiente:

```
{'Contrastes':                                Variable  Estimate  ...   p value  signif
 0                            (Intercept)  0.316100  ...  0.578727
 1                           Age_19_65_pct  0.030494  ...  0.000002     ***
 2                          ForeignersPtge -0.030847  ...  0.000003     ***
 3                        SameComAutonPtge -0.015092  ...  0.000049     ***
 4                           PobChange_pct -0.015417  ...  0.000595     ***
 5                           Explotaciones -0.000652  ...  0.000884     ***
 6    CCAA_Cantabria - Galicia - Asturias  -2.412211  ...  0.000000     ***
 7    CCAA_Castilla -La Mancha - Canarias  -1.620289  ...  0.000000     ***
 8      CCAA_Cataluña - Aragón - Baleares  -1.168597  ...  0.000000     ***
 9       CCAA_Cataluña - Aragón- Baleares  -4.904623  ...  0.000000     ***
 10    CCAA_Comunidad Valenciana - Murcia -3.562694  ...  0.000000     ***
 11       CCAA_La Rioja - Castilla y León -2.826746  ...  0.000000     ***
 12   CCAA_Madrid - Pais Vasco - Navarra  -1.401935  ...  0.000000     ***
 13                    ActividadPpal_Otro -0.284384  ...  0.009535      **
 14               ActividadPpal_Servicios -0.682840  ...  0.000203     ***
 15           Population_Age_under19_Ptge -0.000022  ...  0.000000     ***
 16        Population_UnemployLess25_Ptge  0.000049  ...  0.000000     ***

 [17 rows x 6 columns],
 'BondadAjuste':            LLK          AIC          BIC
 0 -2437.108857  4878.217714  4891.774674}
```

- **Variable categórica: CCAA_Comunidad Valenciana - Murcia.** El coeficiente asociado a esta variable es $\beta = -3{,}5627$, luego $e^{-3{,}5627} = 0{,}0285 < 1$. Esto significa que la ODD de *Izquierda* $= 1$ se reduce un $1 - 0{,}0285 = 0{,}9715 = 97{,}15\,\%$ si pertenece a la Comunidad Valenciana o a Murcia en comparación con Andalucía o Extremadura.

- **Variable continua: PobChange_pct.** El coeficiente asociado a esta variable es $\beta = -0{,}015417$, luego $e^-0{,}0154 = 0{,}9847$. Esto significa que el aumento de un punto porcentual en la variable *PobChange_pct*, disminuye en un $1{,}53\,\%$ la ODD de $Izquierda = 1$.

3.6. Justificar por qué es el modelo ganador y medir la calidad del mismo

La curva ROC y el área bajo esta curva se usa como medida de bondad, esta representa la tasa de verdaderos positivos frente a la tasa de falsos positivos para distintos puntos de corte. Cuánto mas cóncava sea la curva, mejor será el modelo.

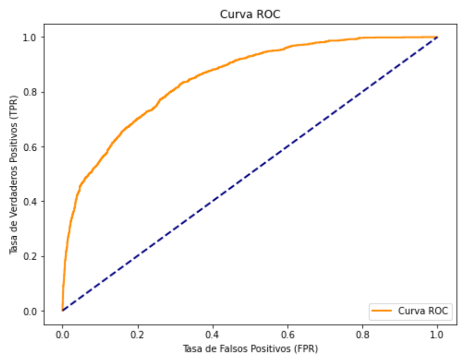


Figura 8: Curva ROC del modelo ganador

- Todas las variables presentan p-valores muy bajos, lo que indican que contribuyen significativamente. Además, todas las variables presentan dos o tres asteriscos en la columna *signif*.
- El valor del Area Bajo la Curva ROC es $0{,}8602$ lo que indica un gran poder predictivo del modelo.
- El modelo tiene $LLK = -2437{,}11$, $AIC = 4878{,}22$ y $BIC = 4891{,}77$, esto indica un buen ajuste puesto que los valores AIC y BIC no son excesivamente altos.
- El $pseudo - R^2$ de este modelo es $0{,}2909$, un valor que respalda el buen ajuste del modelo y su capacidad predictiva.

Para finalizar, estudiaremos la importancia de cada variable en el modelo ganador.

```
         Variables        R2
PobChange_pct  0.001848
Explotaciones  0.002002
SameComAutonPtge  0.002658
ActividadPpal  0.002731
Age_19_65_pct  0.003075
ForeignersPtge  0.003340
Population_Age_under19_Ptge  0.011535
Population_UnemployLess25_Ptge  0.015257
CCAA  0.174061
```

Las variables más influyentes en el modelo son *CCAA*, *Population_UnemployLess25_Ptg* y *Population_Age_under19_Ptge*