Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

TALLER DE MODELADO MATEMÁTICO II
PARTE I / 25P

PROYECTO FINAL

ALAN BADILLO SALAS
BANDON EDUARDO ANTONIO GÓMEZ

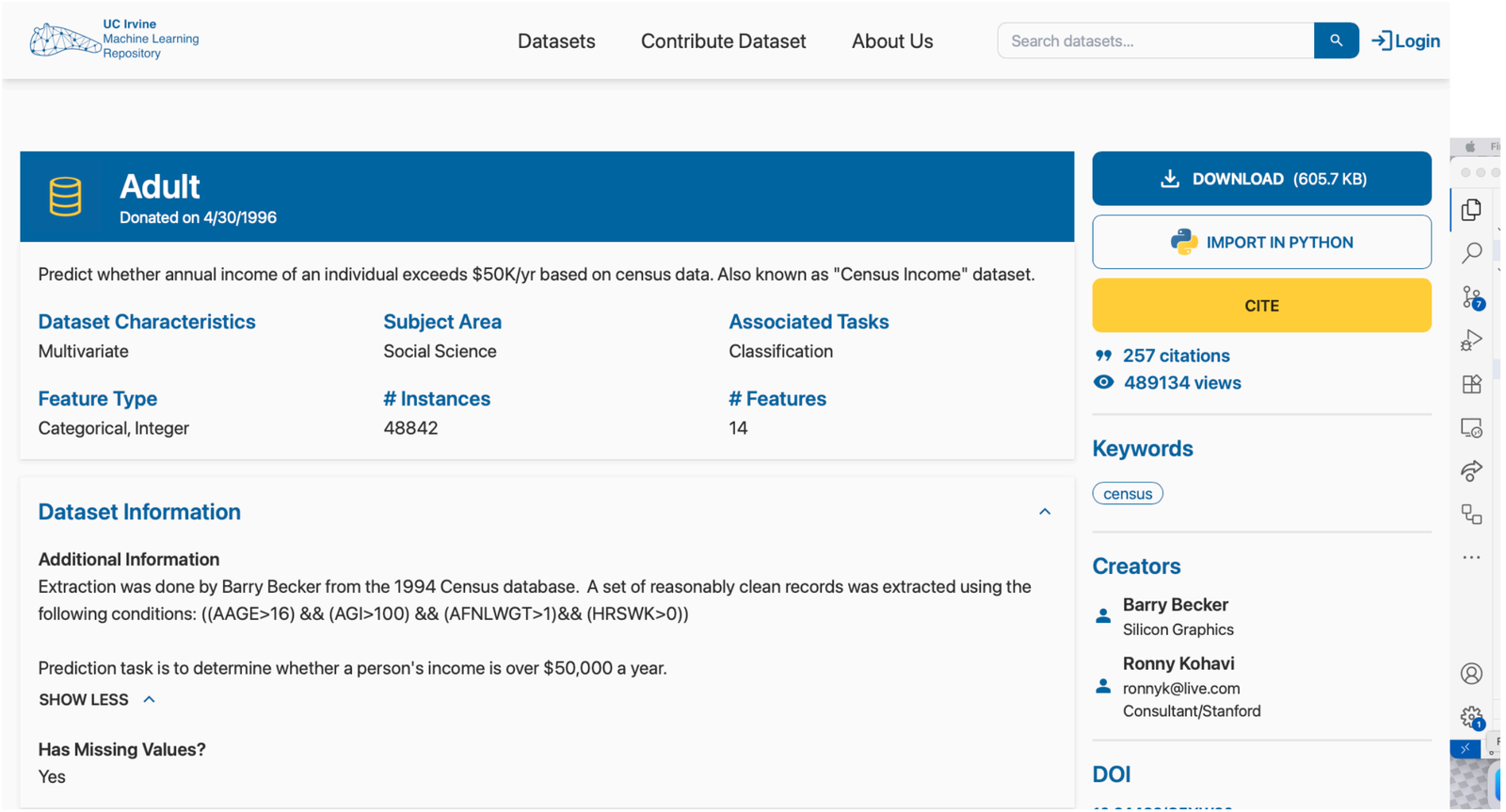
Dr. Alejandro Román Vázquez

INTRODUCCIÓN

Conjunto de datos Adult

El conjunto de datos contine **información laboral de adultos** y si su ingreso es mayor o menor a $50,000 dólares anuales.

Posee **14 características** y 32,561 registros, es un problema de clasificación, para predecir **si un adulto ganará más de $50k al año.**

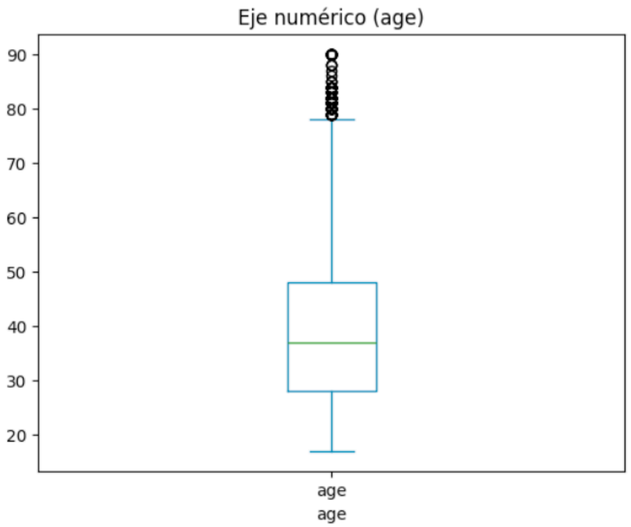


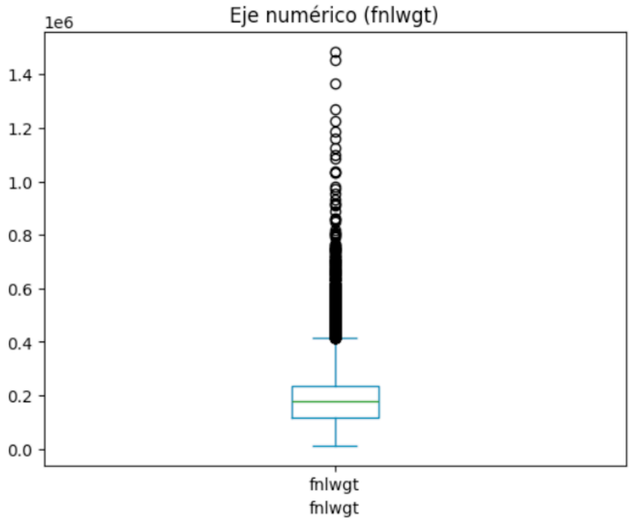
Conjunto de datos Adult

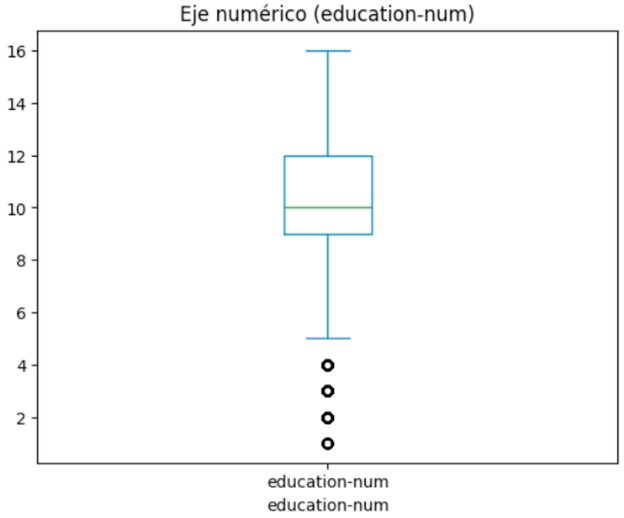
Se tienen 15 columnas, donde Income será la columna de respuesta para el problema de clasificación y las 14 restantes serán características para las covariables

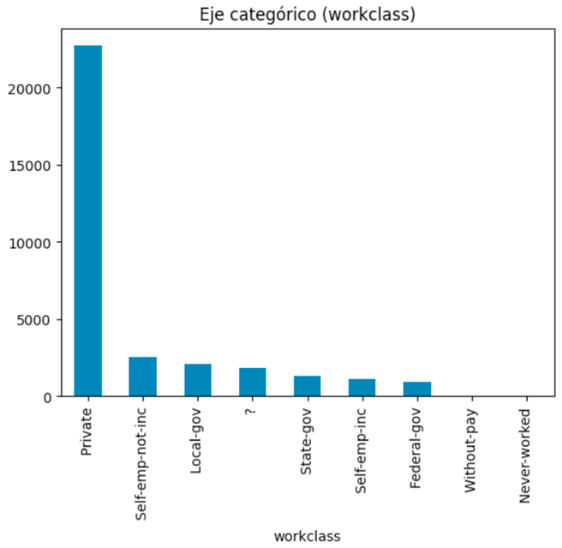
1. age: edad (numérico).
2. Worlclass, clase de trabajo: Privado,(categórico) Autónomo-no-inc, Autónomo-inc, Federal-gov, Local-gov, Estatal-without-pay, gov, (Sin sueldo, Nunca-trabajó.) (categórico)
3. fnlwgt: Final Weight. Es un peso de muestra (sampling weight) asignado por la Oficina del Censo de EE.UU. Se usa para extrapolar los datos de la muestra a toda la población estadounidense (numérico)
4. education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. (categórico)
5. education-num: (numérico).
6. marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.(categórico)
7. occupation: Apoyo técnico, Reparación artesanal, Otros servicios, Ventas, Directivo, Profesional especializado, Manipulador-limpiador, Operador de maquinaria, Administrativo, Agricultor-pescador, Transporte, Servicio doméstico privado, Servicio de protección, Fuerzas armadas.(categórico)
8. relationship (relación): Esposa, Hijo propio, Esposo, No familiar, Otro pariente, Soltero.(categórico)
9. race: Blanco, Asiático-Pacífico-Islandés, Amerindio-Esquimal, Otro, Negro.(categórico)
10. sex: Mujer, Hombre.(categórico)
11. capital-gain (plusvalía): (numérico)
12. capital-loss (minusvalía): (numérico)
13. Hours-per-week (horas-semana): (numérico).
14. native-country (país-nativo): Estados Unidos, Camboya, Inglaterra, Puerto Rico, Canadá, Alemania, EE.UU. periférico (Guam-USVI-etc.), India, Japón, Grecia, Sur, China, Cuba, Irán, Honduras, Filipinas, Italia, Polonia, Jamaica, Vietnam, México, Portugal, Irlanda, Francia, República Dominicana, Laos, Ecuador, Taiwán, Haití, Colombia, Hungría, Guatemala, Nicaragua, Escocia, Tailandia, Yugoslavia, El Salvador, Trinad & Tobago, Perú, Hong, Holanda. (categórico)
15. Income : ingreso (será la variable respuesta Y) (categórico)

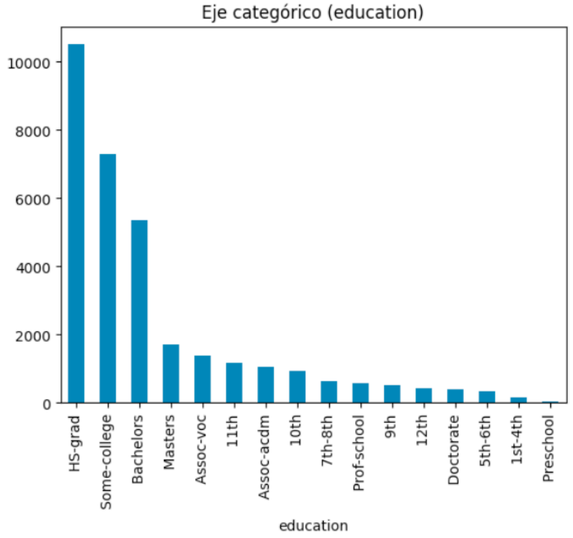
ANÁLISIS EXPLORATORIO

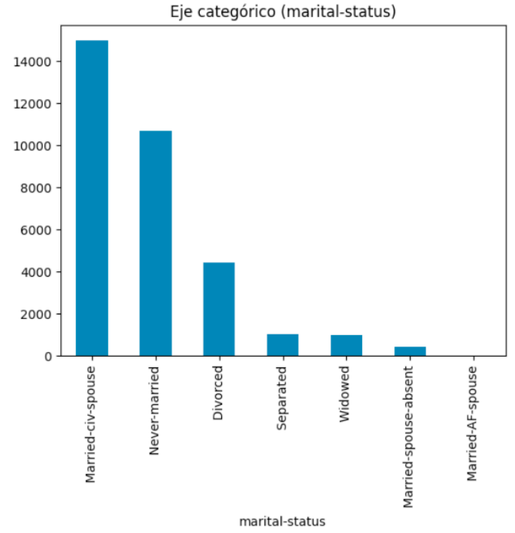




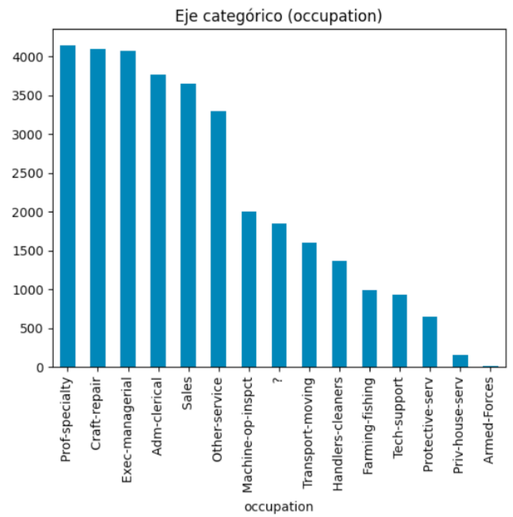


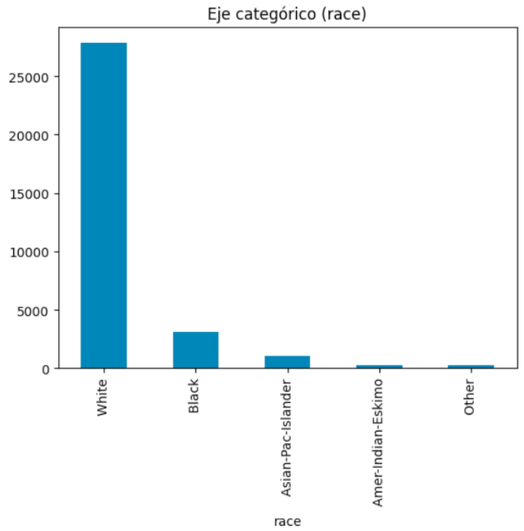


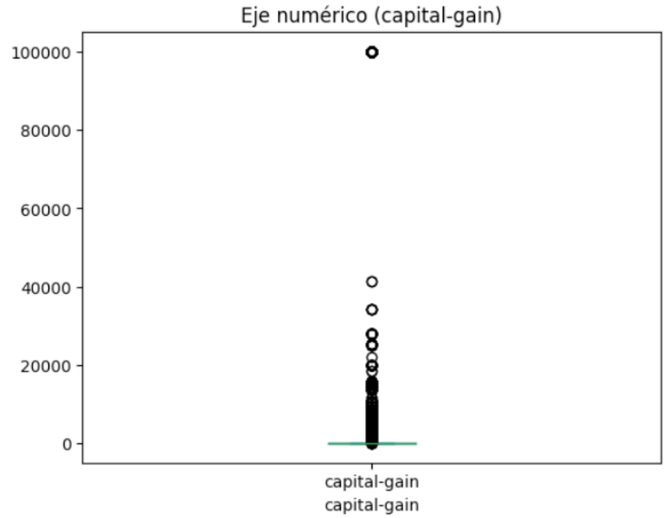


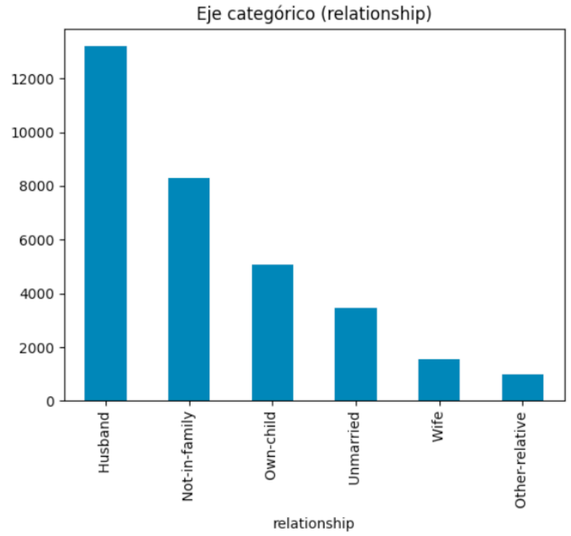


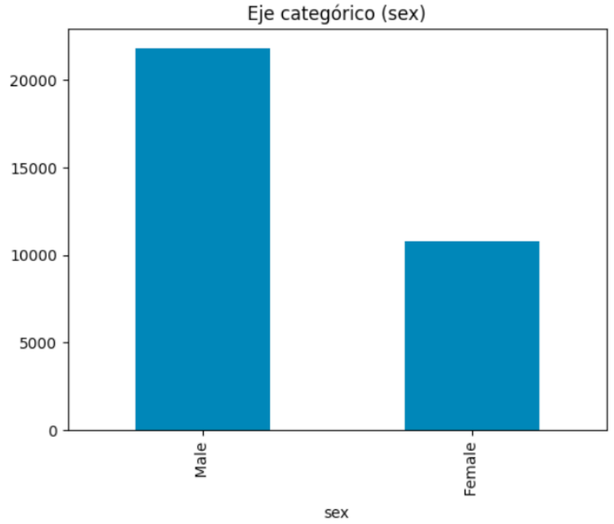
ANÁLISIS EXPLORATORIO

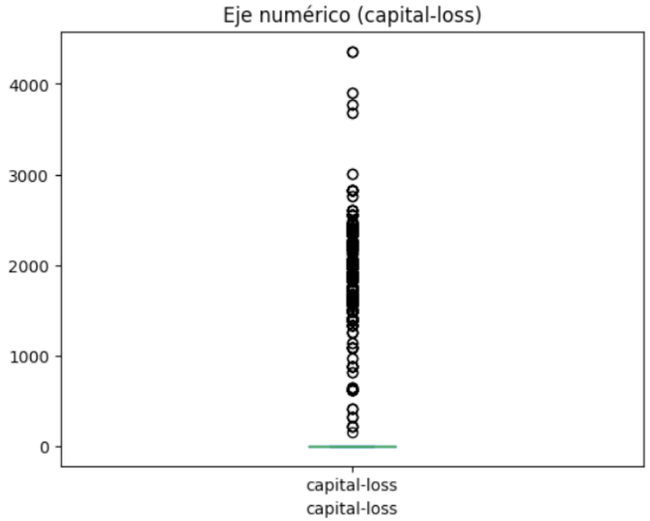


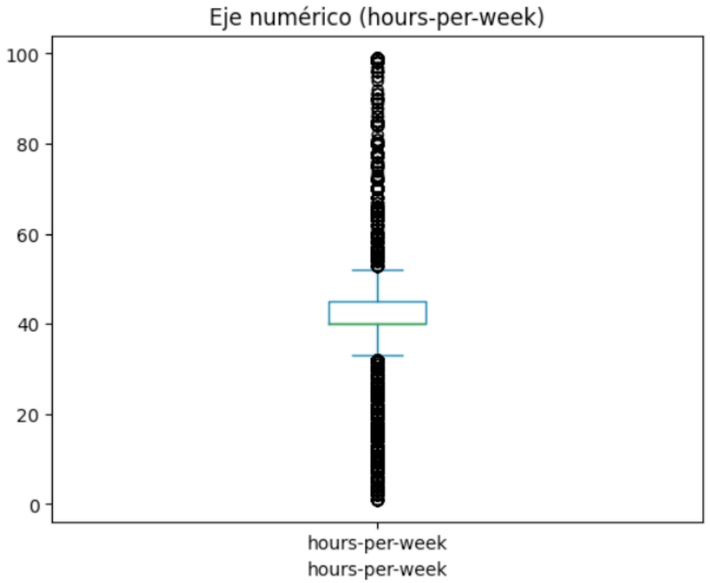


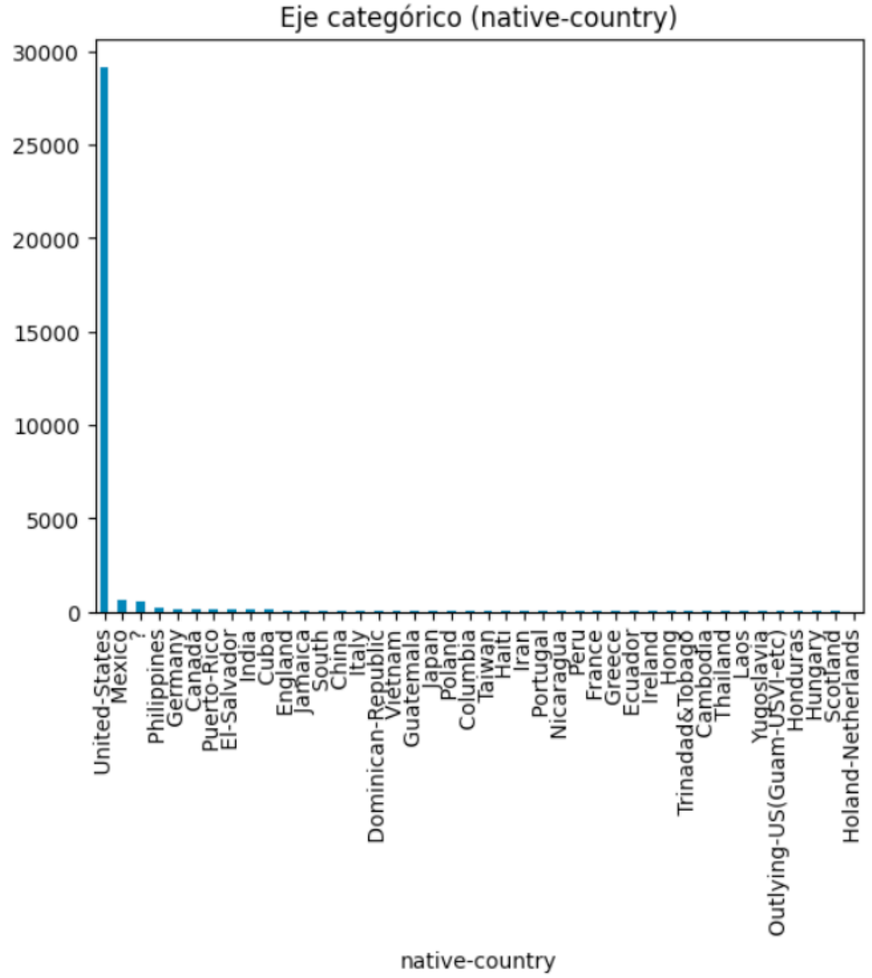


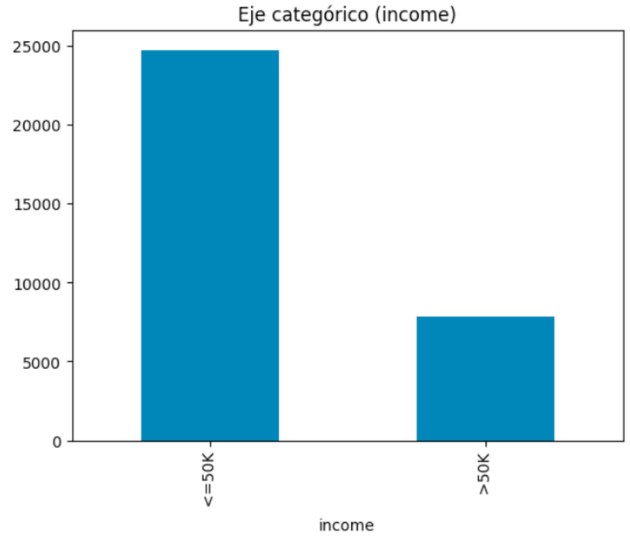







Eje numérico (hours-per-week)
100
80
60
40
20
0
hours-per-week
hours-per-week


Eje categórico (native-country)
30000
25000
20000
15000
10000
5000
0
United-States
Mexico
?
Philippines
Germany
Canada
Puerto-Rico
El-Salvador
India
Cuba
England
Jamaica
South
China
Italy
Dominican-Republic
Vietnam
Guatemala
Japan
Poland
Columbia
Taiwan
Haiti
Iran
Portugal
Nicaragua
Peru
France
Greece
Ecuador
Ireland
Hong
Trinadad&Tobago
Cambodia
Thailand
Laos
Yugoslavia
Outlying-US(Guam-USVI-etc)
Honduras
Hungary
Scotland
Holand-Netherlands
native-country

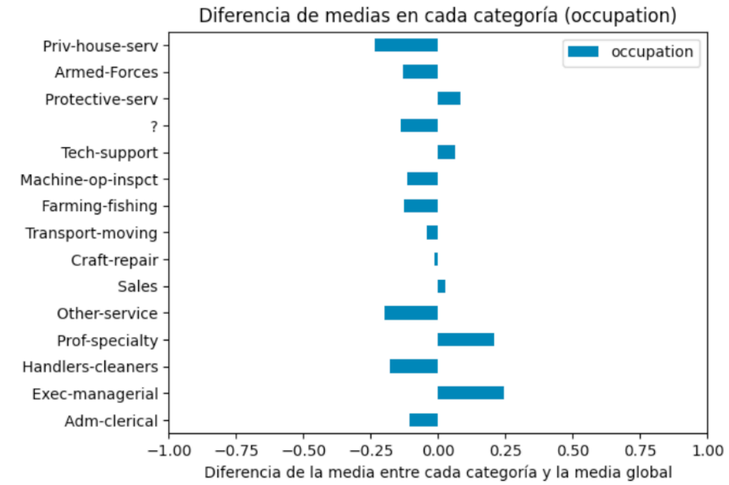

Eje categórico (income)
25000
20000
15000
10000
5000
0
<=50K
>50K
income

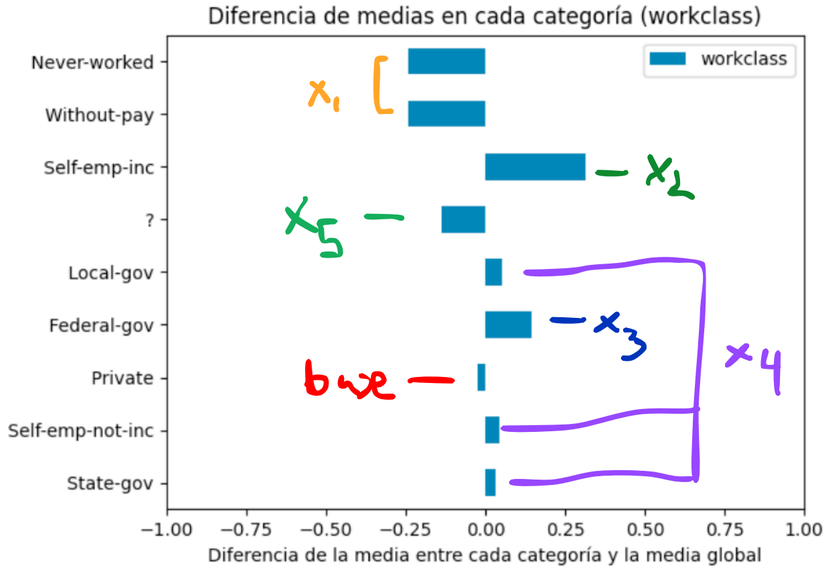
2.1. Mean encoding (centrado)

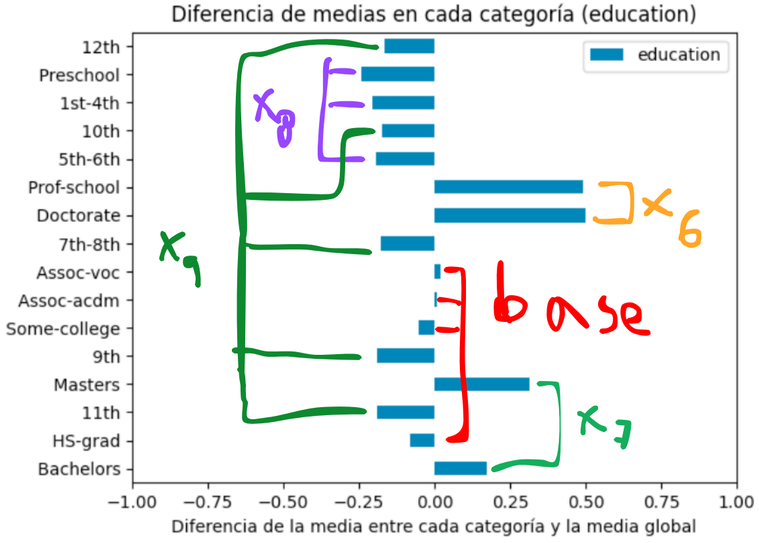
Procedemos a realizar la codificación para las variables categóricas, en este caso usaremos mean encoding centrado ya que esta codificación realiza un contraste directamente con la base, es decir la base cero para cada categoría es aquella categoría con mayor frecuencia, a continuación explicamos en que consiste:

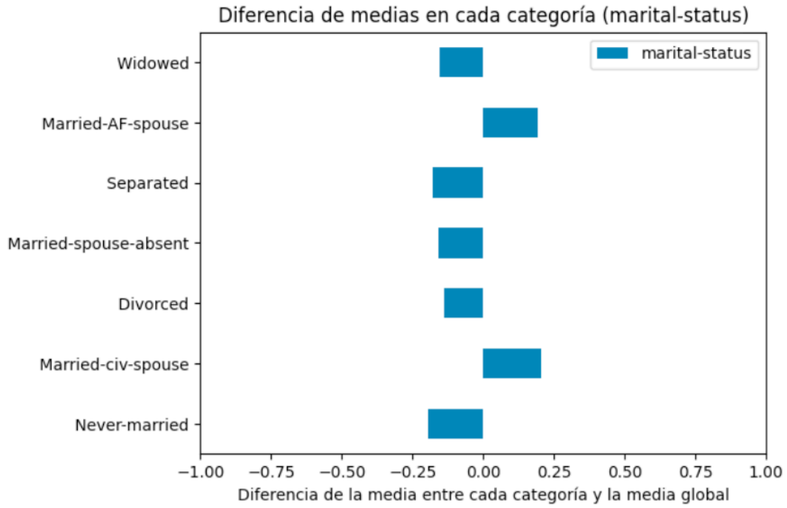
- Se elige una categoría base (por ejemplo, la más frecuente o la de mayor peso).
- Se calcula la media global μ y la media de cada categoría μ_j
- En vez de usar μ_j directamente, se usa $\mu_j - \mu_{\text{base}}$
- Así, la base queda con (efecto cero), y el resto refleja el desvío respecto a ella.

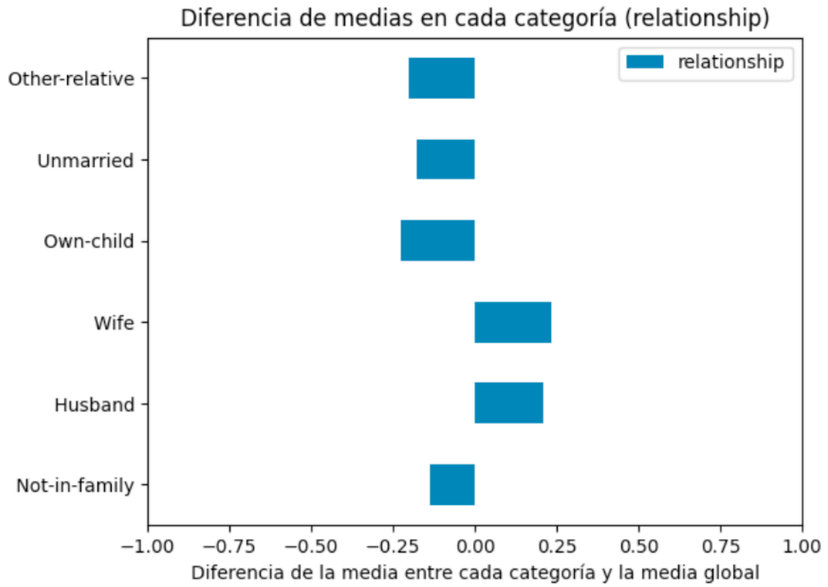
Generamos las gráficas respecto a la diferencia de medias para cada categoría

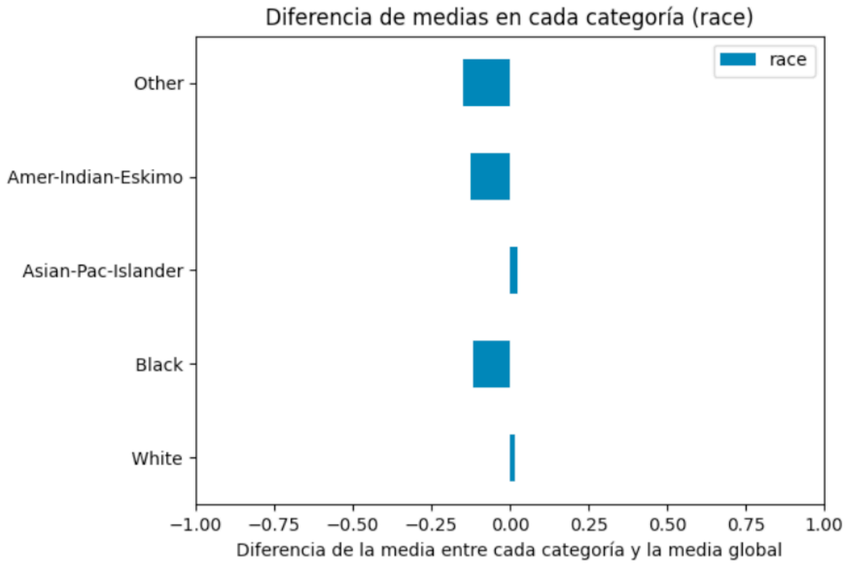


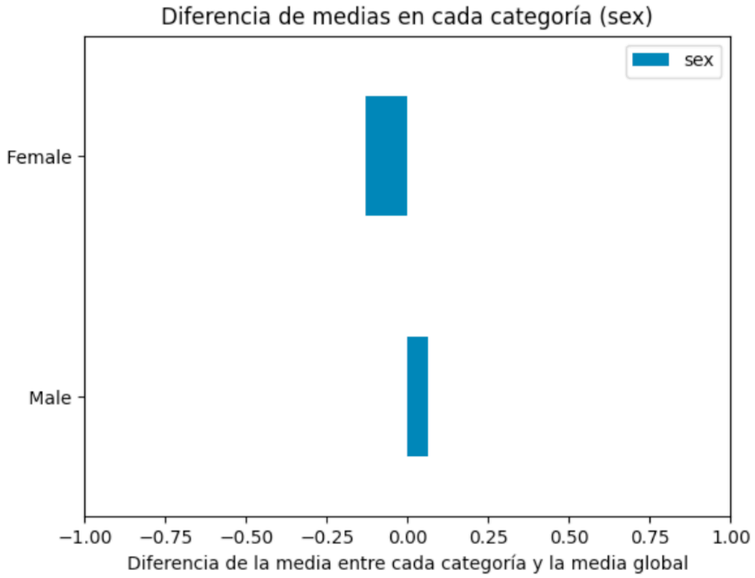







Diferencia de medias en cada categoría (relationship)
relationship
Other-relative
Unmarried
Own-child
Wife
Husband
Not-in-family
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
Diferencia de la media entre cada categoría y la media global


Diferencia de medias en cada categoría (race)
race
Other
Amer-Indian-Eskimo
Asian-Pac-Islander
Black
White
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
Diferencia de la media entre cada categoría y la media global


Diferencia de medias en cada categoría (sex)
sex
Female
Male
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
Diferencia de la media entre cada categoría y la media global

INGENIERÍA DE VARIABLES

Casa abierta al tiempo

	Categoría
0	State-gov
1	Self-emp-not-inc
2	Private
3	Federal-gov
4	Local-gov
5	?
6	Self-emp-inc
7	Without-pay
8	Never-worked

```
x1 = test_categorias(adult["workclass"], [8, 7])
x2 = test_categorias(adult["workclass"], [6])
x3 = test_categorias(adult["workclass"], [3])
x4 = test_categorias(adult["workclass"], [4, 0, 1])
x5 = test_categorias(adult["workclass"], [5])
```

	Categoría
0	Bachelors
1	HS-grad
2	11th
3	Masters
4	9th
5	Some-college
6	Assoc-acdm
7	Assoc-voc
8	7th-8th
9	Doctorate
10	Prof-school
11	5th-6th
12	10th
13	1st-4th
14	Preschool
15	12th

```
x6 = test_categorias(adult["education"], [9, 10])
x7 = test_categorias(adult["education"], [3, 0])
x8 = test_categorias(adult["education"], [14, 13, 11])
x9 = test_categorias(adult["education"], [8, 4, 12, 2, 15])
```

	Categoría
0	Never-married
1	Married-civ-spouse
2	Divorced
3	Married-spouse-absent
4	Separated
5	Married-AF-spouse
6	Widowed

```
x10 = test_categorias(adult["marital-status"], [5, 1])
```

	Categoría
0	Adm-clerical
1	Exec-managerial
2	Handlers-cleaners
3	Prof-specialty
4	Other-service
5	Sales
6	Craft-repair
7	Transport-moving
8	Farming-fishing
9	Machine-op-inspct
10	Tech-support
11	?
12	Protective-serv
13	Armed-Forces
14	Priv-house-serv

```
x11 = test_categorias(adult["occupation"], [14, 4, 2])
x12 = test_categorias(adult["occupation"], [3, 1])
x13 = test_categorias(adult["occupation"], [12, 10, 5])
```

En las variables categóricas se agruparon por las diferencias de medias

```
def test_categorias(x, indices=[]):
    categorias = x.unique()
    s = numpy.zeros_like(x)
    for j in indices:
        cat_j = categorias[j]
        s = s + (x == cat_j).astype(int)
    return s
```

INGENIERÍA DE VARIABLES

Casa abierta al tiempo

	Categoría
0	Not-in-family
1	Husband
2	Wife
3	Own-child
4	Unmarried
5	Other-relative

```
x14 = test_categorias(adult["relationship"], [2, 1])
```

	Categoría
0	White
1	Black
2	Asian-Pac-Islander
3	Amer-Indian-Eskimo
4	Other

```
x15 = test_categorias(adult["race"], [4, 3, 1])
```

	Categoría
0	Male
1	Female

```
x16 = test_categorias(adult["sex"], [1])
```

	Categoría
0	United-States
1	Cuba
2	Jamaica
3	India
4	?
5	Mexico
6	South
7	Puerto-Rico
8	Honduras
9	England
10	Canada
11	Germany
12	Iran
13	Philippines
14	Italy
15	Poland
16	Columbia
17	Cambodia
18	Thailand
19	Ecuador
20	Laos
21	Taiwan
22	Haiti
23	Portugal
24	Dominican-Republic
25	El-Salvador
26	France
27	Guatemala
28	China
29	Japan
30	Yugoslavia
31	Peru
32	Outlying-US(Guam-USVI-etc)
33	Scotland
34	Trinadad&Tobago
35	Greece
36	Nicaragua
37	Vietnam
38	Hong
39	Ireland
40	Hungary
41	Holand-Netherlands

```
x17 = test_categorias(adult["native-country"], [12, 26, 3, 21, 29, 30, 17])
x18 = test_categorias(adult["native-country"], [14, 9, 10, 11, 13, 38])
x19 = test_categorias(adult["native-country"], [18, 19, 2, 20, 23, 34, 7, 22])
x20 = test_categorias(adult["native-country"], [25, 8, 37, 31, 36, 5, 27])
x21 = test_categorias(adult["native-country"], [16, 24, 32, 41])
```

```python
def winzorizado(x):
    Q1 = x.quantile(0.25)
    Q3 = x.quantile(0.75)
    IQR = Q3 - Q1

    xmin = Q1 - 1.5 * IQR # -8
    xmax = Q3 + 1.5 * IQR # 8

    xp = (x < xmin) * xmin + ((x >= xmin) & (x <= xmax)) * x + (x > xmax) * xmax

    return xp
```

Para las variables numéricas se realizó un proceso de winzorizado para limitar los puntos atípicos

```python
x22 = winzorizado(adult["age"])
x23 = winzorizado(adult["fnlwgt"])
# x25 = winzorizado(adult["capital-gain"])
# x26 = winzorizado(adult["capital-loss"])
x24 = winzorizado(adult["education-num"])
x27 = winzorizado(adult["hours-per-week"])
```

```python
x25 = (adult["capital-gain"] > 0).astype(int)
x26 = (adult["capital-loss"] > 0).astype(int)
```

Variables de análisis

```
X = pandas.DataFrame([
    x1, x2, x3, x4, x5, x6, x7, x8, x9, x10,
    x11, x12, x13, x14, x15, x16, x17, x18, x19, x20,
    x21, x22, x23, x24, x25, x26, x27
], index=[
    "x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10",
    "x11", "x12", "x13", "x14", "x15", "x16", "x17", "x18", "x19", "x20",
    "x21", "x22", "x23", "x24", "x25", "x26", "x27"
]).T

X.head()
```

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	...	x18	x19	x20	x21	x22	x23	x24	x25	x26	x27
0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	39.0	77516.0	13.0	1.0	0.0	40.0
1	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	50.0	83311.0	13.0	0.0	0.0	32.5
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	38.0	215646.0	9.0	0.0	0.0	40.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	...	0.0	0.0	0.0	0.0	53.0	234721.0	7.0	0.0	0.0	40.0
4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	28.0	338409.0	13.0	0.0	0.0	40.0

5 rows × 27 columns

En total se construyeron **27 variables** derivadas de las 14 columnas de características, para poder predecir la **variable de respuesta Income**

Las **variables categóricas fueron codificadas como dummies**, agrupadas por las diferencias de medias y las **variables numéricas fueron winzorizadas** para que los puntos atípicos no afectaran demasiado

Reporte

✅ Exactitud Proporción de predicciones correctas sobre el total de casos.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

✅ Precisión (Precision) Qué proporción de las predicciones positivas fueron correctas.

$$\text{Precision} = \frac{TP}{TP + FP}$$

✅ Sensibilidad (Recall o TPR) Qué proporción de los positivos reales fueron correctamente identificados.

$$\text{Recall} = \frac{TP}{TP + FN}$$

✅ Especificidad (TNR) Qué proporción de los negativos reales fueron correctamente identificados.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

✅ F1-score Media armónica entre precisión y recall.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

La matriz de confusión esta dada por:

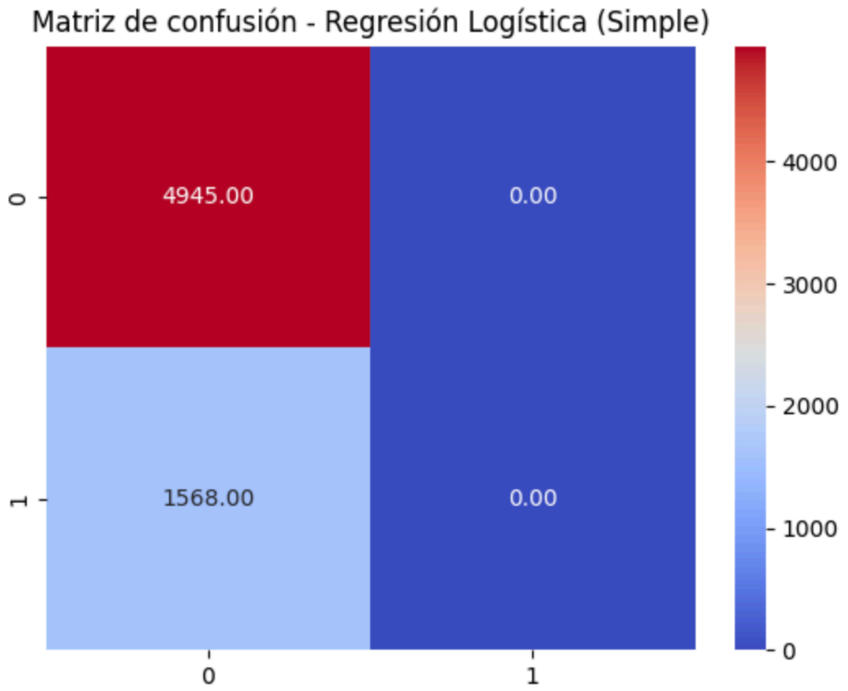
$$\text{Matriz de Confusión} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

donde:

- TN: True Negatives (verdaderos negativos)
- FP: False Positives (falsos positivos)
- FN: False Negatives (falsos negativos)
- TP: True Positives (verdaderos positivos)

Regresión Logística (Simple)

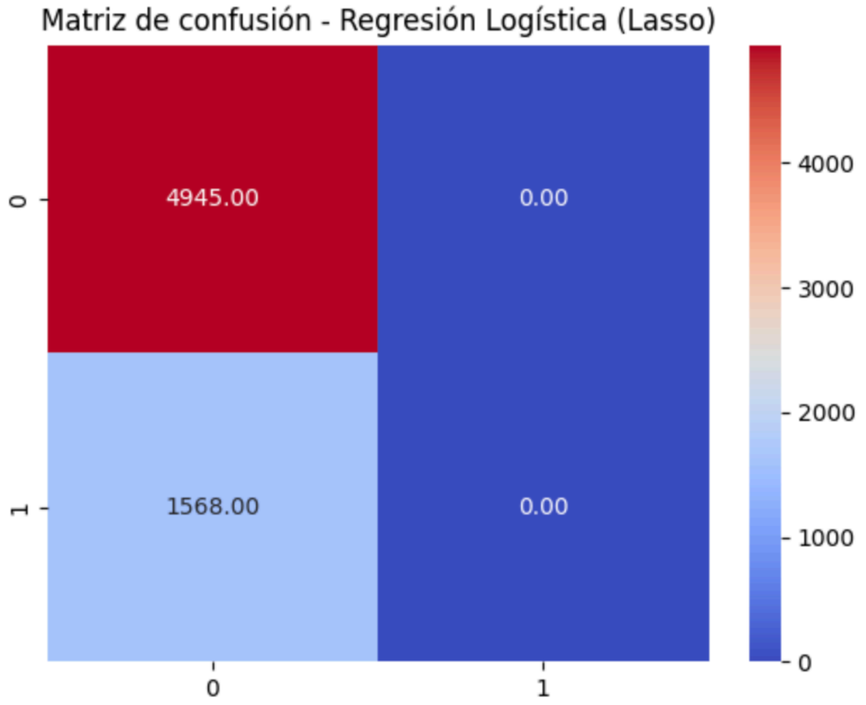
	Valor
Exactitud	0.759251
Presición	NaN
Sensibilidad	0.000000
Especificidad	1.000000
F1	NaN

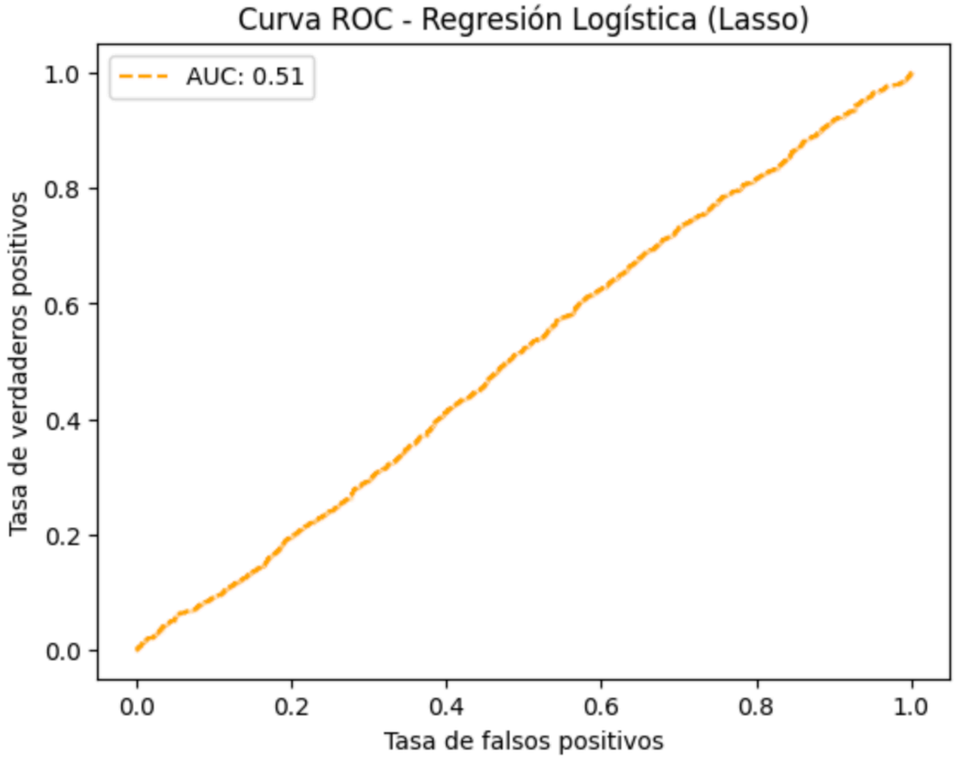


Casa abierta al tiempo

Regresión Logística (Lasso)

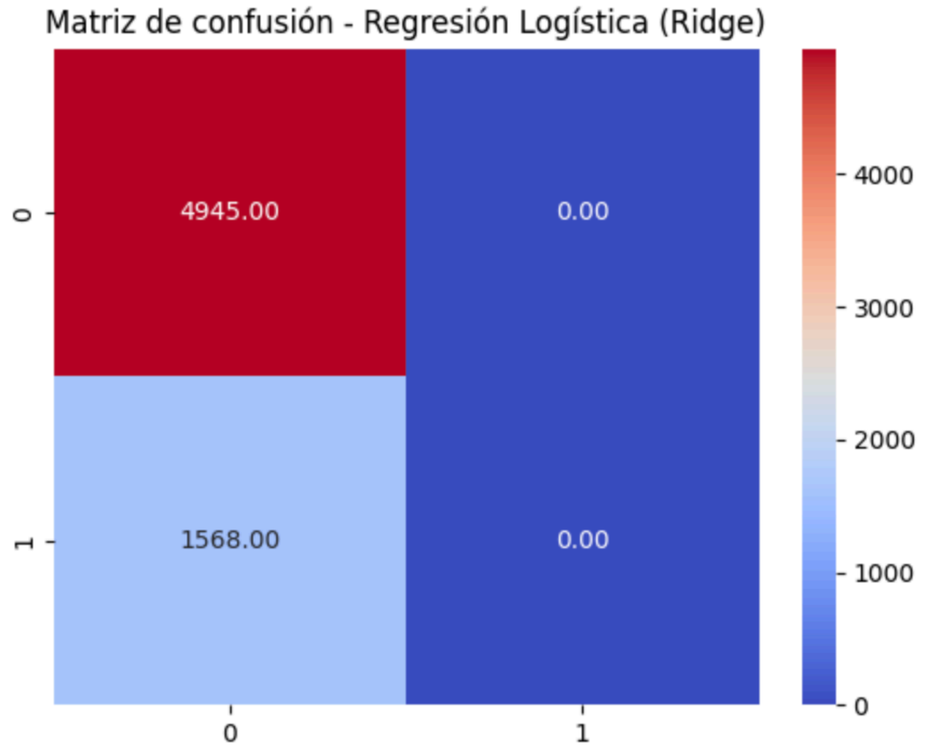
	Valor
Exactitud	0.759251
Presición	NaN
Sensibilidad	0.000000
Especificidad	1.000000
F1	NaN

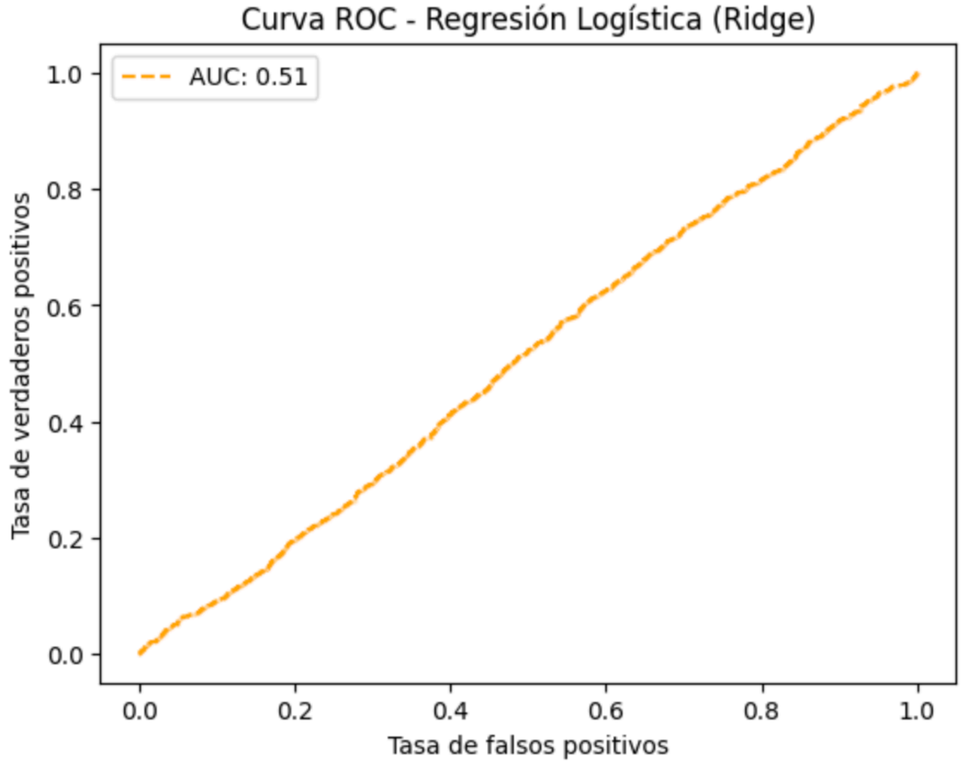




Regresión Logística (Ridge)

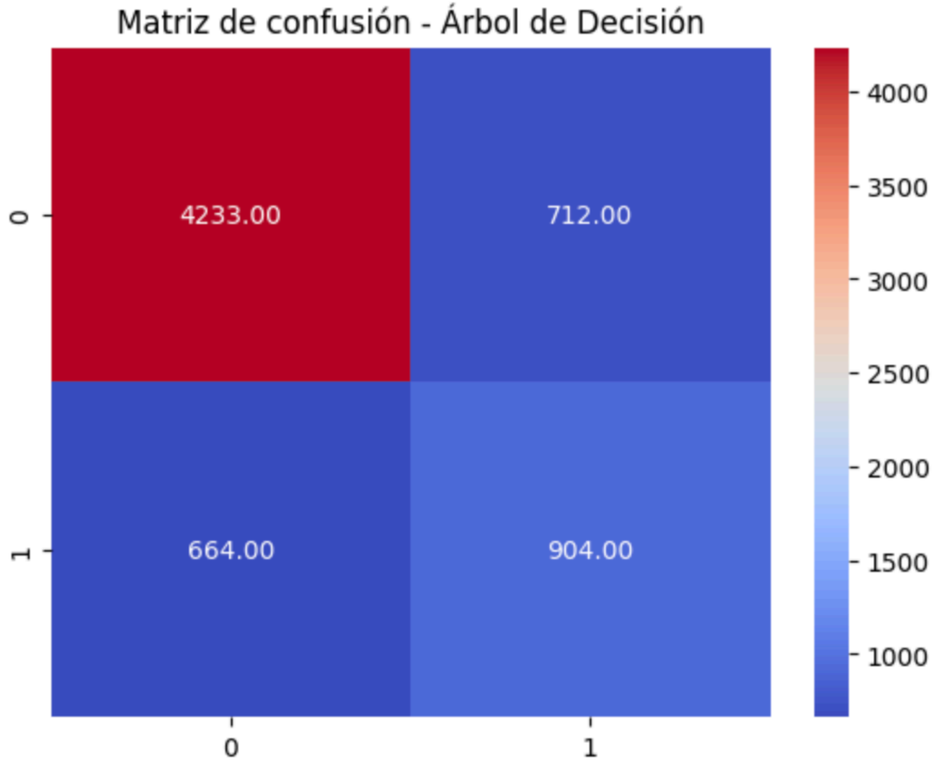
	Valor
Exactitud	0.759251
Presición	NaN
Sensibilidad	0.000000
Especificidad	1.000000
F1	NaN

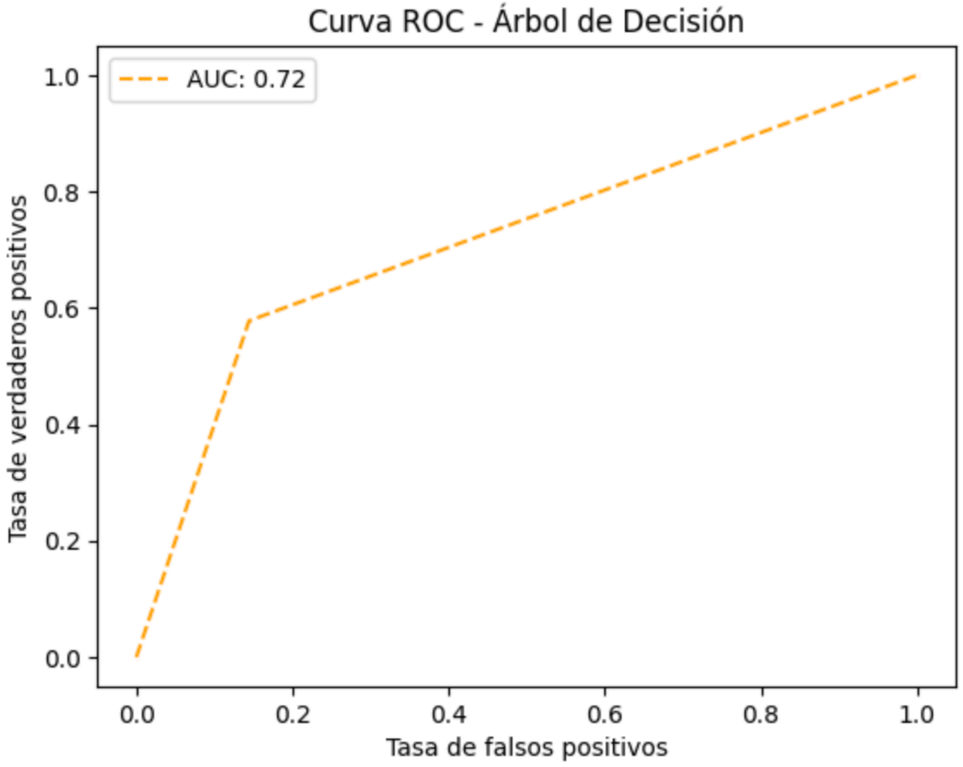




Árbol de Decisión

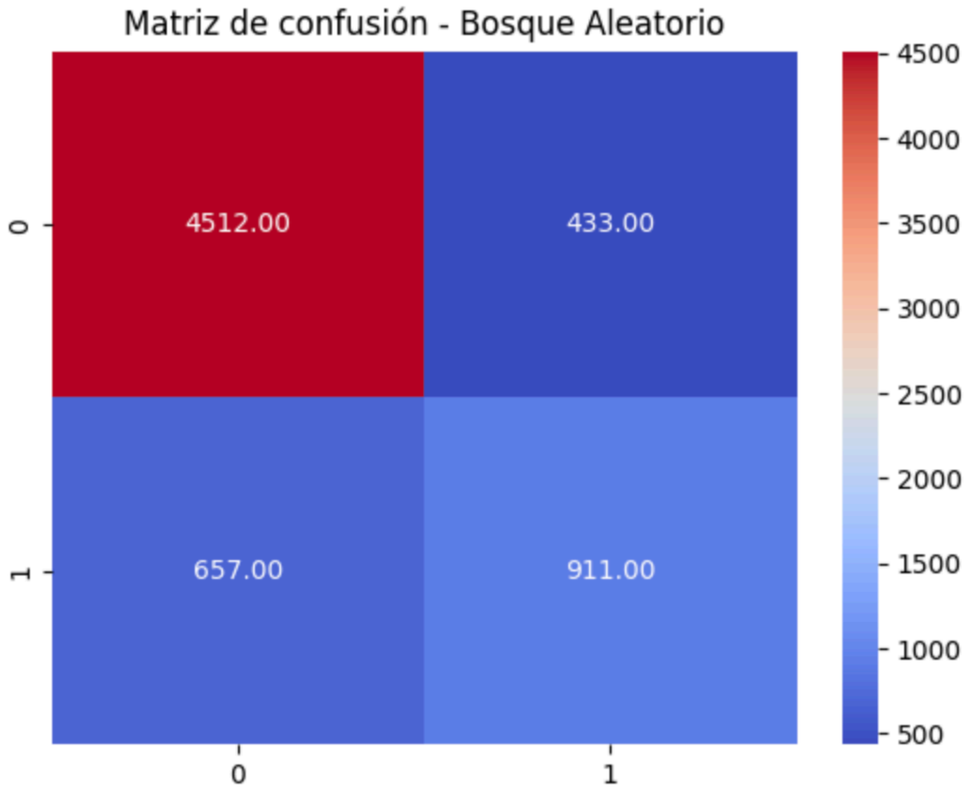
	Valor
Exactitud	0.788730
Presición	0.559406
Sensibilidad	0.576531
Especificidad	0.856016
F1	0.567839

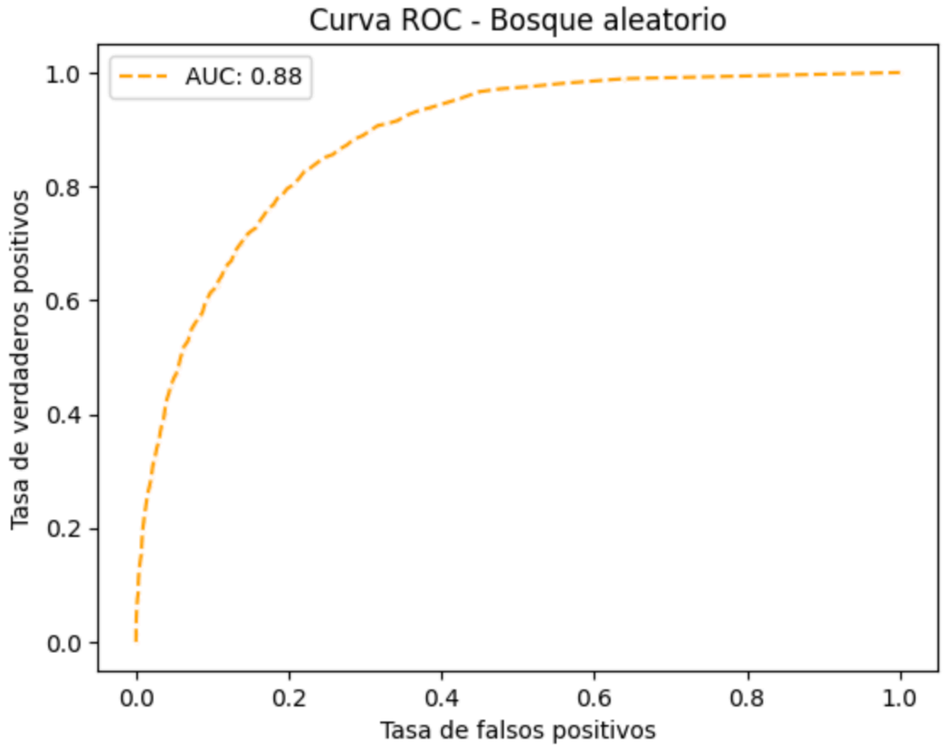




Bosque Aleatorio

	Valor
Exactitud	0.832642
Presición	0.677827
Sensibilidad	0.580995
Especificidad	0.912437
F1	0.625687

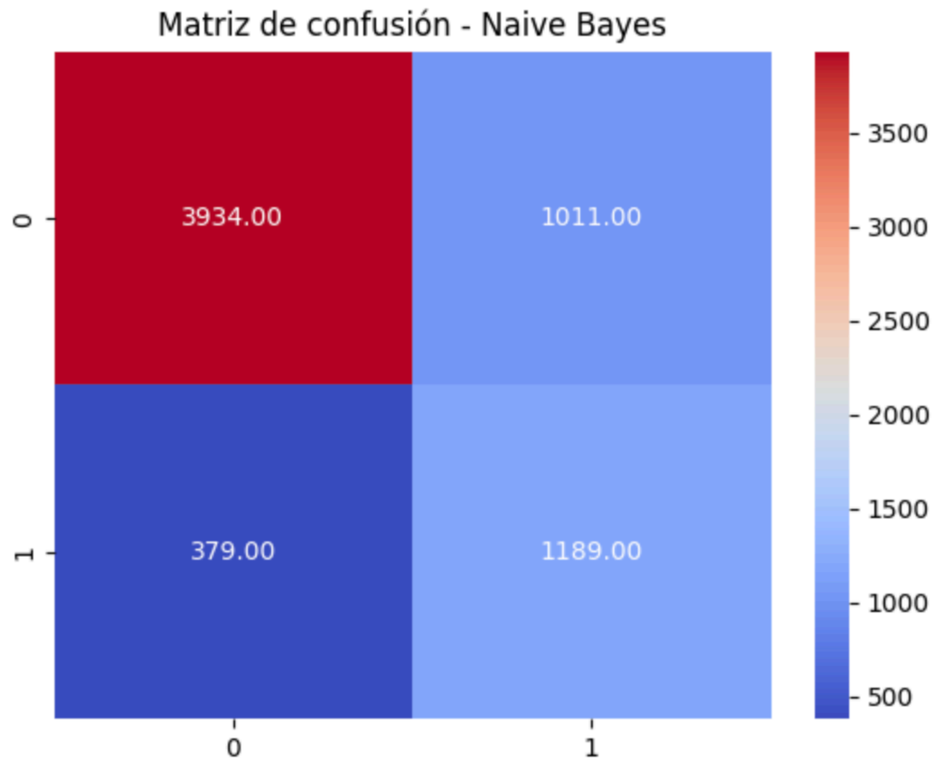


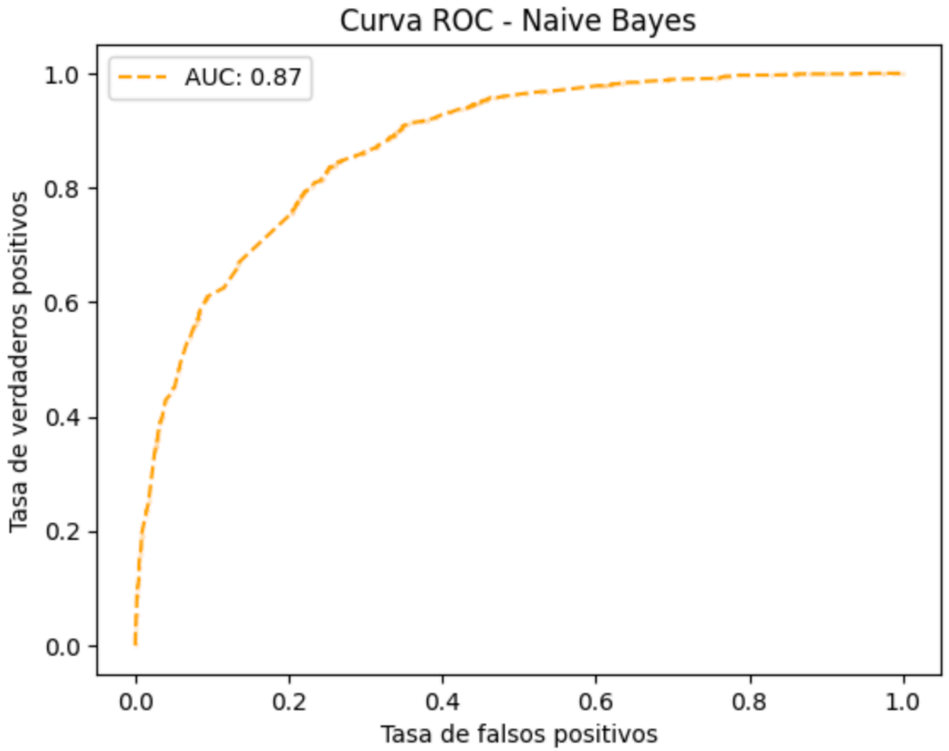


Casa abierta al tiempo

Naive Bayes

	Valor
Exactitud	0.786581
Presición	0.540455
Sensibilidad	0.758291
Especificidad	0.795551
F1	0.631104

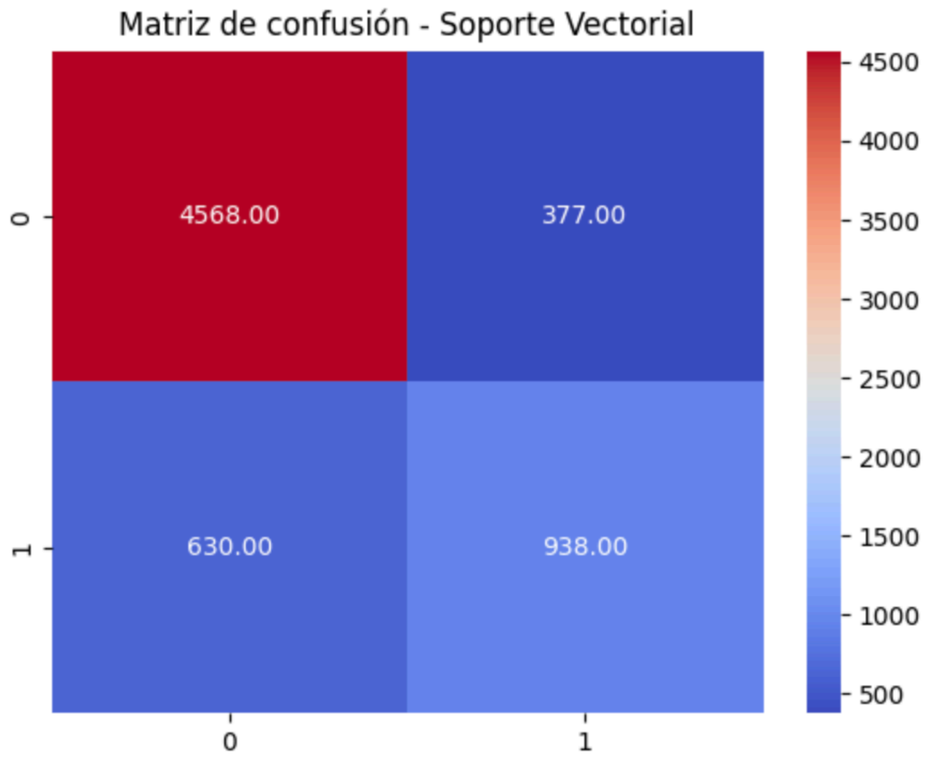


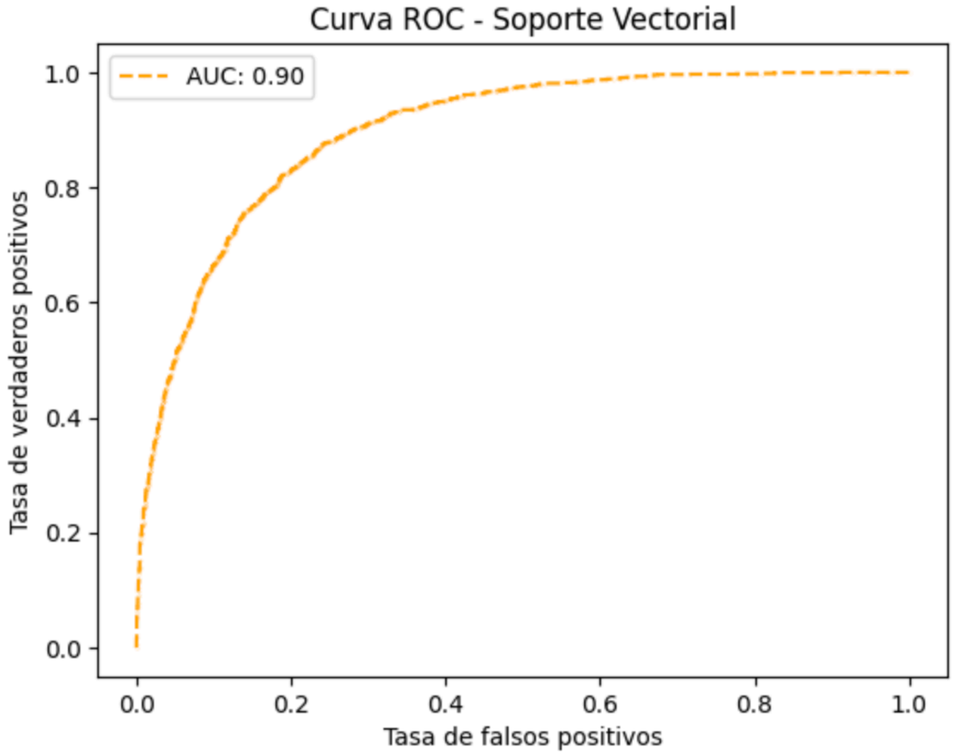


Casa abierta al tiempo

Soporte Vectorial

	Valor
Exactitud	0.845386
Presición	0.713308
Sensibilidad	0.598214
Especificidad	0.923761
F1	0.650711

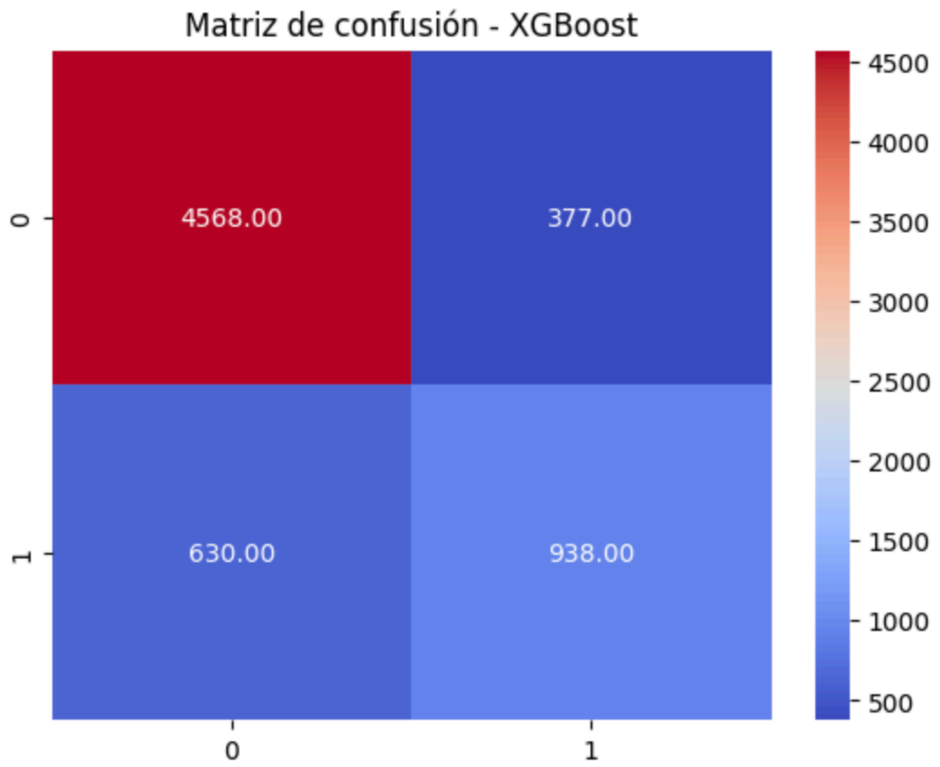


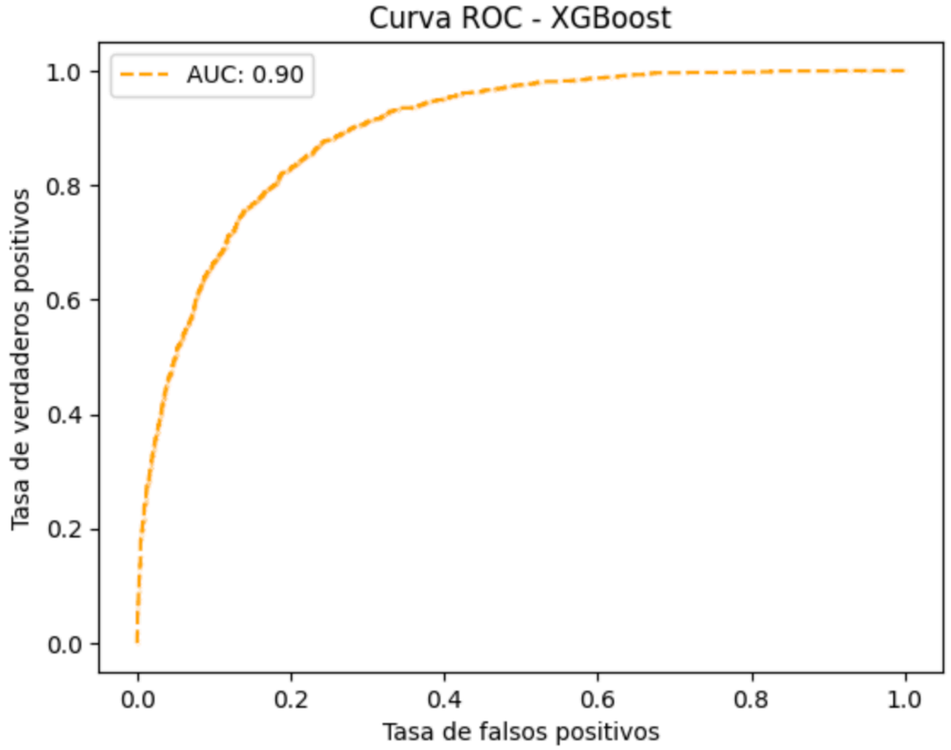


Casa abierta al tiempo

XGBoost

	Valor
Exactitud	0.845386
Presición	0.713308
Sensibilidad	0.598214
Especificidad	0.923761
F1	0.650711





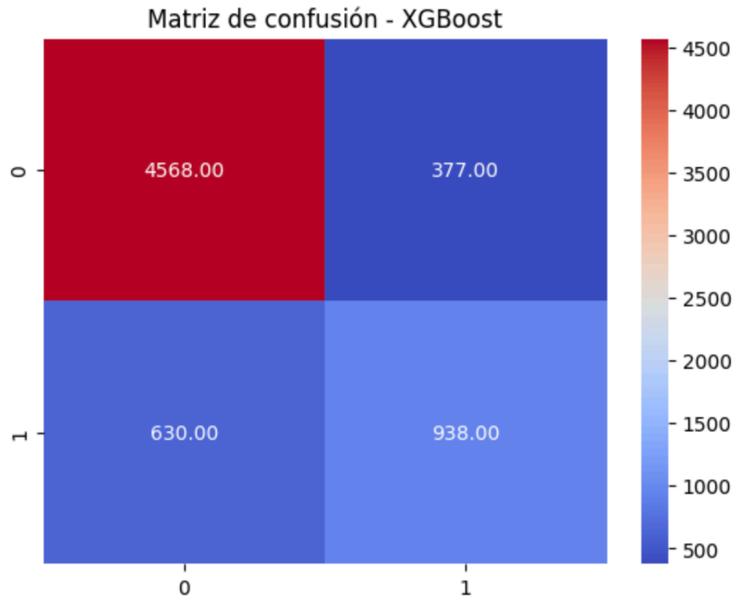
Modelo	Exactitud	AUC
Logístico Simple	0.76	0.51
Logístico Lasso	0.76	0.51
Logístico Ridge	0.76	0.51
Naive Bayes	0.83	0.87
Árbol de Decisión	0.79	0.72
Bosque Aleatorio	0.83	0.88
XGBoost	0.85	0.90
Support Vector	0.85	0.90

→ Mejores

Casa abierta al tiempo

XGBoost

	Valor
Exactitud	0.845386
Presición	0.713308
Sensibilidad	0.598214
Especificidad	0.923761
F1	0.650711




```python
cv = RandomizedSearchCV(
    XGBClassifier(),
    param_distributions={
        "max_depth": [None, 10, 100],
        "max_leaves": [4, 6, 8]
    }
)

cv.fit(X_train, y_train)

cv
```

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_max_leaves	param_max_depth	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
0	0.098335	0.021212	0.002607	0.000345	4	None	{'max_leaves': 4, 'max_depth': None}	0.855470	0.846833	0.847985	0.842772	0.844116	0.847435	0.004427	1
3	0.081611	0.001786	0.002461	0.000117	4	10	{'max_leaves': 4, 'max_depth': 10}	0.855470	0.846833	0.847985	0.842772	0.844116	0.847435	0.004427	1
6	0.085570	0.002425	0.002498	0.000235	4	100	{'max_leaves': 4, 'max_depth': 100}	0.855470	0.846833	0.847985	0.842772	0.844116	0.847435	0.004427	1
2	0.119268	0.018444	0.002688	0.000155	8	None	{'max_leaves': 8, 'max_depth': None}	0.854702	0.848369	0.842226	0.844692	0.846804	0.847359	0.004212	4
5	0.118118	0.014680	0.003323	0.000645	8	10	{'max_leaves': 8, 'max_depth': 10}	0.854702	0.848369	0.842226	0.844692	0.846804	0.847359	0.004212	4
8	0.125483	0.003768	0.003312	0.000432	8	100	{'max_leaves': 8, 'max_depth': 100}	0.854702	0.848369	0.842226	0.844692	0.846804	0.847359	0.004212	4
1	0.110361	0.003823	0.002892	0.000278	6	None	{'max_leaves': 6, 'max_depth': None}	0.854511	0.847217	0.843762	0.845652	0.844308	0.847090	0.003898	7
4	0.131262	0.044806	0.003776	0.001366	6	10	{'max_leaves': 6, 'max_depth': 10}	0.854511	0.847217	0.843762	0.845652	0.844308	0.847090	0.003898	7
7	0.123750	0.021208	0.003286	0.000321	6	100	{'max_leaves': 6, 'max_depth': 100}	0.854511	0.847217	0.843762	0.845652	0.844308	0.847090	0.003898	7

XGBoost

```
21: 0.0491
23: 0.0399
22: 0.0296
9: 0.0270
26: 0.0203
24: 0.0200
13: 0.0169
11: 0.0150
15: 0.0090
3: 0.0070
12: 0.0065
25: 0.0052
10: 0.0031
2: 0.0023
```

```
x21 = test_categorias(adult["native-country"], [16, 24, 32, 41])
```

16	Columbia
24	Dominican-Republic
32	Outlying-US(Guam-USVI-etc)
41	Holand-Netherlands

```
x23 = winzorizado(adult["fnlwgt"])
```

```
x22 = winzorizado(adult["age"])
```

```
x9 = test_categorias(adult["education"], [8, 4, 12, 2, 15])
```

8	7th-8th
4	9th
12	10th
2	11th
15	12th

```
x26 = (adult["capital-loss"] > 0).astype(int)
```

```
x24 = winzorizado(adult["education-num"])
```

```
x13 = test_categorias(adult["occupation"], [12, 10, 5])
```

5	Sales
10	Tech-support
12	Protective-serv

```
14: 0.0021
5: 0.0018
1: 0.0018
6: 0.0017
4: 0.0015
17: 0.0012
19: 0.0012
16: 0.0007
20: 0.0007
18: 0.0006
8: 0.0004
7: 0.0001
0: 0.0000
```

Conclusiones

En este proyecto final, hemos analizado el conjunto de datos sobre la población adulta en EU, para determinar quiénes son los que ganan más de $50k según características como la edad, el sexo, el color de piel, el tipo de trabajo, el nivel de estudios, entre otras características descritas.

Se generaron 27 variables predictivas para poder construir un modelo de clasificación capaz de predecir el valor de la respuesta binaria si ganará más de $50k.

Se compararon diferentes modelos de Regresión Logística, Árboles de Decisión, Bosques Aleatorios, Naive Bayes, Soportes Vectoriales y XGBoost, encontrando en los últimos dos los mejores niveles de presición y capacidad predictiva, analizando su matriz de confusión y la curva ROC/AUC.

Finalmente, optimizamos los hiperparámetros del mejor modelo seleccionado (XGBoost) mediante la validación cruzada, usando la búsqueda aleatoria.

Con esto, hemos desarrollado habilidades de exploración, análisis y validación de modelos, que nos permitirán adaptar nuevos tipos de problemas y generar modelos de valor para el análisis de datos.