Clasificación de Ingresos con Árboles de Decisión y Random Forest

Big Data e Inteligencia Artificial

Yeray Hurtado Dragón

Noviembre 2025

Índice

1. Introducción

En este informe se analiza y clasifica el nivel de ingresos de individuos utilizando técnicas de *Machine Learning*, específicamente Árboles de Decisión y Random Forest. El objetivo principal es entender cómo distintas variables demográficas y laborales, como edad, educación, ocupación y estado civil, influyen en la probabilidad de que una persona tenga ingresos superiores a 50K anuales.

Para ello, se utiliza el dataset **Adult Income**, que contiene información de más de 48.000 individuos, incluyendo datos sobre empleo, educación, relaciones familiares y características socioeconómicas.

A lo largo del informe se realizará un análisis exploratorio de los datos (EDA), se prepararán los datos para modelado, se entrenarán y evaluarán los modelos mencionados, y finalmente se compararán sus resultados para extraer conclusiones sobre su efectividad y las variables más influyentes en la predicción de ingresos.

2. Análisis Exploratorio de Datos (EDA)

El primer paso en el análisis de datos es explorar la información disponible para comprender su estructura, tipos de variables y posibles problemas de calidad. A continuación se muestra un resumen de las columnas del dataset y sus tipos de datos:

Columna	**Tipo de dato**	**Descripción**
age	int64	Edad de la persona
workclass	object	Tipo de empleo
fnlwgt	int64	Ponderación de la muestra
education	object	Nivel educativo
educational-num	int64	Nivel educativo en formato numérico
marital-status	object	Estado civil
occupation	object	Ocupación
relationship	object	Relación familiar
race	object	Raza o etnia
gender	object	Género
capital-gain	int64	Ganancia de capital
capital-loss	int64	Pérdida de capital
hours-per-week	int64	Horas trabajadas por semana
native-country	object	País de origen
income	object	Ingreso: <=50K o >50K (variable objetivo)

Cuadro 1: Resumen de las columnas del dataset Adult Income.

Como se puede observar, el dataset cuenta con 48.842 registros y 15 columnas, de las cuales 6 son numéricas y 9 son categóricas. Inicialmente no se detectan valores nulos; sin embargo, al examinar los datos más de cerca, se identificó que algunas entradas usan el carácter ‘¿ para representar valores desconocidos. Para manejar correctamente estos casos, se reemplazaron por valores nulos con la siguiente instrucción:

```
df.replace('?', np.nan, inplace=True)
```

Este paso permite tratar adecuadamente los valores faltantes en el análisis posterior.

2.1. age

La columna `age` nos dice la edad de cada persona y es de tipo `int64`. En el dataset, la edad más baja es **17 años** y la más alta **90 años**. Esto significa que la mayoría de las personas tienen entre jóvenes y adultos de mediana edad.

A continuación se muestra un histograma con la distribución de edades en el dataset:

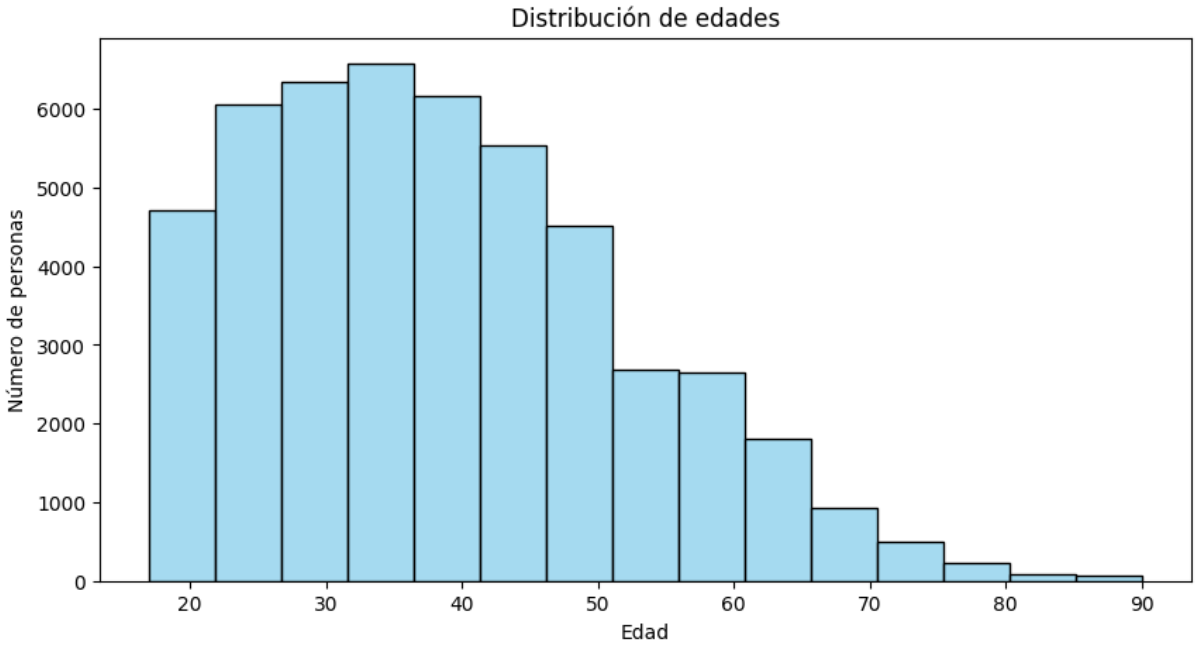

Figura 1: Distribución de edades en el dataset Adult Income.

Como se observa en la Figura 1, la mayoría de las personas se encuentra entre los 20 y 50 años. Se aprecia una ligera tendencia hacia las edades más jóvenes, con un número decreciente de individuos a medida que la edad aumenta. Este patrón es importante para el análisis de ingresos, ya que la edad puede estar correlacionada con la experiencia laboral y, por ende, con los ingresos.

2.2. workclass

La columna `workclass` nos indica el tipo de empleo o relación laboral de la persona. Existen ocho categorías posibles: *Private*, *Local-gov*, *Self-emp-not-inc*, *Federal-gov*, *State-gov*, *Self-emp-inc*, *Without-pay* y *Never-worked*. En general, esta variable nos dice si la persona trabaja en una empresa privada, para el gobierno o por cuenta propia. También sirve para identificar a quienes no trabajan o lo hacen sin remuneración.

En la Figura 2 se muestra la distribución de las diferentes categorías de `workclass`. Como se observa, la mayoría de las personas pertenece a la categoría *Private*, mientras que el resto de las clases laborales presentan frecuencias similares y relativamente bajas. Esto indica que el empleo en el sector privado domina ampliamente el dataset, mientras que las demás formas de empleo están mucho menos representadas.

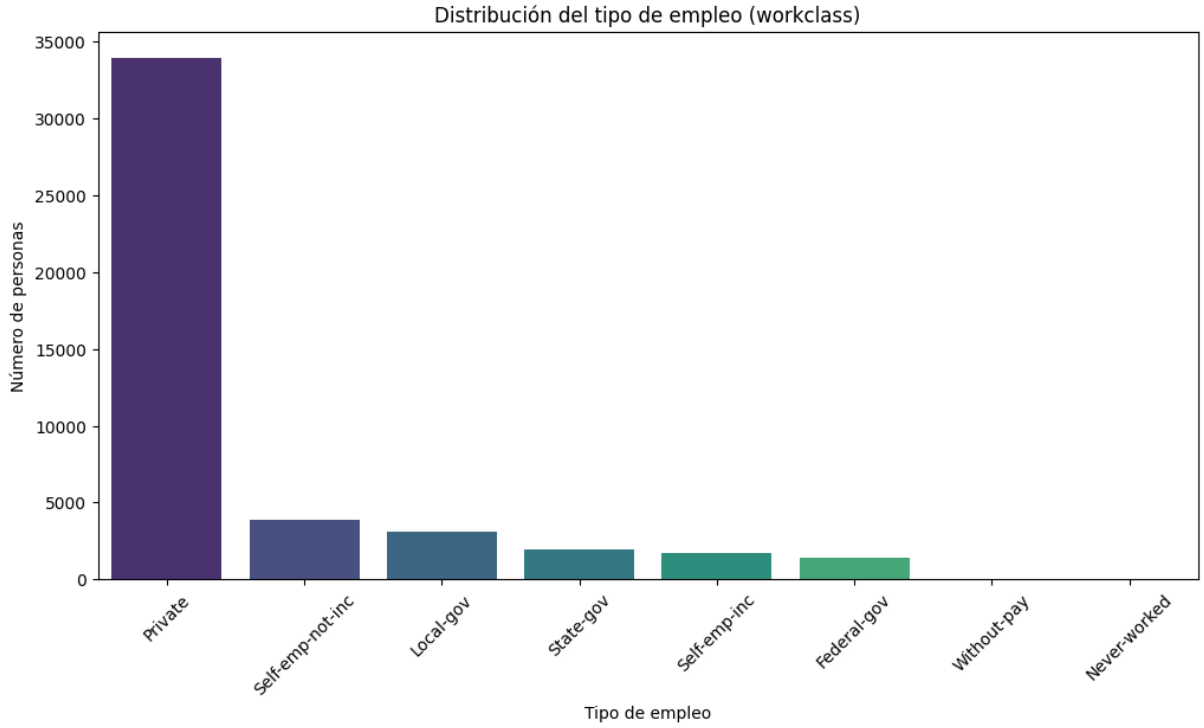

Figura 2: Distribución de la variable `workclass`. Se destaca la preponderancia del empleo privado (*Private*).

2.3. fnlwgt

La columna `fnlwgt` representa el peso final asignado a cada persona en la muestra del censo. Este número indica cuántas personas en la población real se estima que tienen las mismas características que ese registro. En otras palabras, el `fnlwgt` sirve para que los análisis estadísticos sean más representativos de la población total.

En este proyecto no se usa directamente para entrenar los modelos de clasificación, ya que no aporta información útil sobre los ingresos de una persona.

2.4. education

La columna `education` muestra el nivel educativo más alto alcanzado por cada persona. Tiene varios valores posibles, como *Preschool*, *1st-4th*, *5th-6th*, *9th*, *HS-grad*, *Some-college*, *Bachelors*, *Masters* o *Doctorate*, entre otros.

En general, esta variable indica el grado de estudios de la persona, desde educación básica hasta niveles universitarios o de posgrado. Por ejemplo, valores como *HS-grad* significan que la persona terminó la secundaria, mientras que *Bachelors* o *Masters* indican que tiene estudios universitarios. Es una variable muy importante para el análisis, ya que el nivel educativo suele estar relacionado con el nivel de ingresos.

En la Figura 3 se observa la distribución de los niveles educativos en el dataset. Se puede ver que la categoría *HS-grad* es la más frecuente, seguida por *Some-college*, mientras que los niveles superiores como *Bachelors*, *Masters* o *Doctorate* presentan frecuencias menores. Esto indica que la mayoría de los individuos tienen educación secundaria o algo de estudios universitarios, y pocos alcanzan niveles de posgrado.

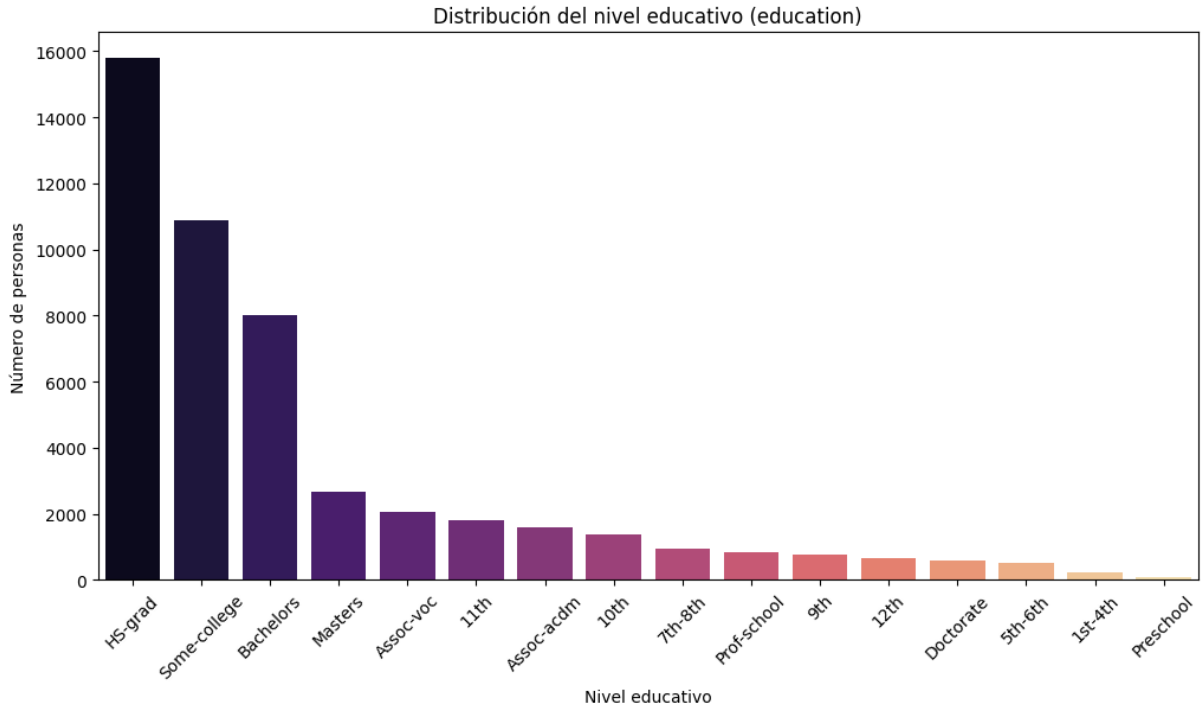

Figura 3: Distribución de la variable `education`, destacando la preponderancia de *HS-grad* y *Some-college*.

2.5. educational-num

La columna `educational-num` representa el nivel educativo en formato numérico. Cada número está asociado a una categoría del campo `education`. Por ejemplo, valores bajos como 1 o 2 corresponden a niveles de educación básica (*Preschool* o *1st-4th*), mientras que valores altos como 13, 14, 15 o 16 indican niveles universitarios o de posgrado (*Bachelors*, *Masters* o *Doctorate*).

Esta variable es útil porque traduce la educación a una forma numérica que los algoritmos de machine learning pueden entender directamente, sin necesidad de hacer transformaciones adicionales.

2.6. marital-status

La columna `marital-status` indica el estado civil de la persona. Los valores posibles son: *Never-married*, *Married-civ-spouse*, *Widowed*, *Divorced*, *Separated*, *Married-spouse-absent* y *Married-AF-spouse*.

En general, esta variable nos dice si la persona está casada, soltera, divorciada o viuda. Por ejemplo, *Never-married* corresponde a personas que nunca se han casado, *Married-civ-spouse* a matrimonios civiles, y *Married-AF-spouse* hace referencia a personas casadas con miembros de las fuerzas armadas.

Este campo puede influir en los ingresos, ya que el estado civil puede estar relacionado con el tipo de empleo o el número de horas trabajadas.

En la Figura 4 se observa la distribución de los estados civiles en el dataset. Se puede ver que las categorías más frecuentes son *Married-civ-spouse* y *Never-married*, mientras que *Married-AF-spouse* es la que menos registros tiene. Esto refleja que la mayoría de los individuos están casados o nunca se han casado, y solo un pequeño porcentaje pertenece a matrimonios con miembros de las fuerzas armadas.

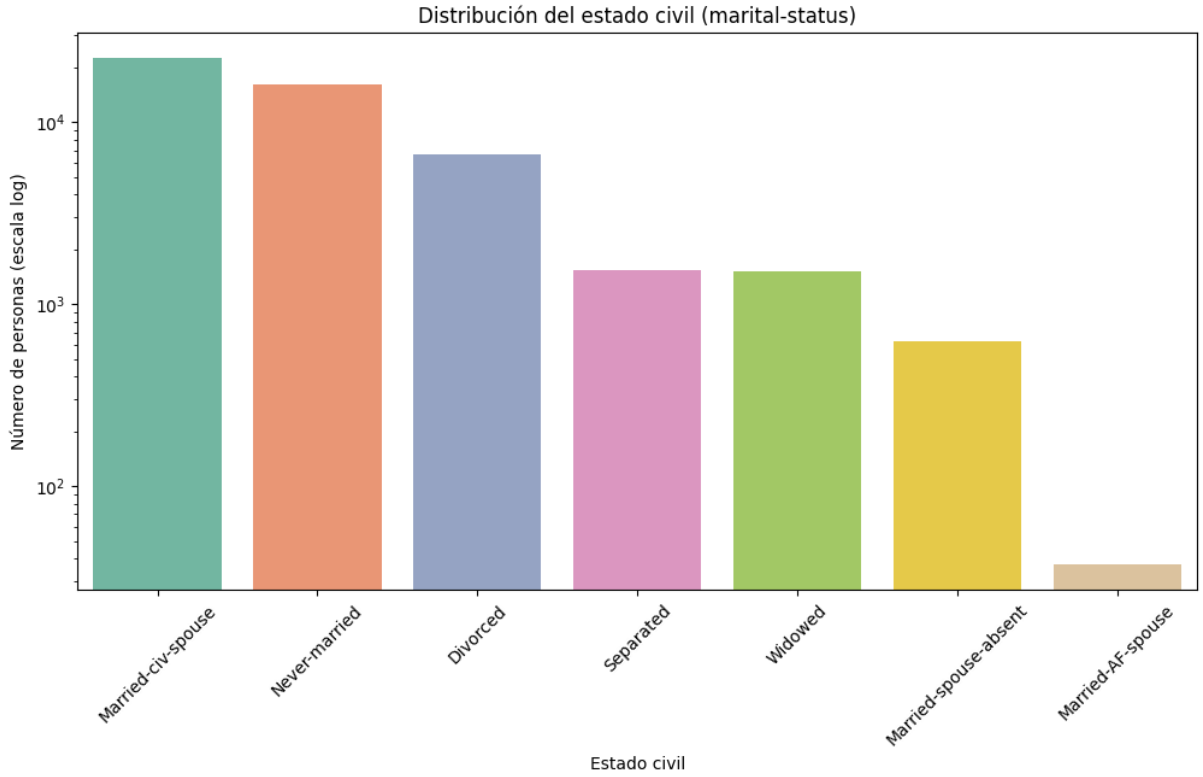

Figura 4: Distribución de la variable `marital-status`, mostrando que *Married-civ-spouse* y *Never-married* son las más frecuentes, y *Married-AF-spouse* la menos frecuente.

2.7. occupation

La columna `occupation` indica la ocupación principal de cada persona. Entre sus valores se encuentran categorías como *Prof-specialty*, *Craft-repair*, *Exec-managerial*, *Adm-clerical*, *Sales*, *Machine-op-inspct*, *Transport-moving*, *Handlers-cleaners*, *Farming-fishing*, *Tech-support*, *Protective-serv*, *Priv-house-serv* y *Armed-Forces*.

Esta variable es importante para el análisis de ingresos, ya que la ocupación suele estar directamente relacionada con el nivel salarial y las horas trabajadas.

En la Figura 5 se muestra la distribución de las ocupaciones en el dataset. Se observa que *Prof-specialty*, *Craft-repair* y *Exec-managerial* son las ocupaciones más frecuentes, con un número de registros muy similar entre ellas, mientras que *Armed-Forces* es la menos representada. Esto indica que la mayoría de los individuos trabaja en profesiones especializadas, manuales o de gestión, y que solo un pequeño porcentaje pertenece a las fuerzas armadas.

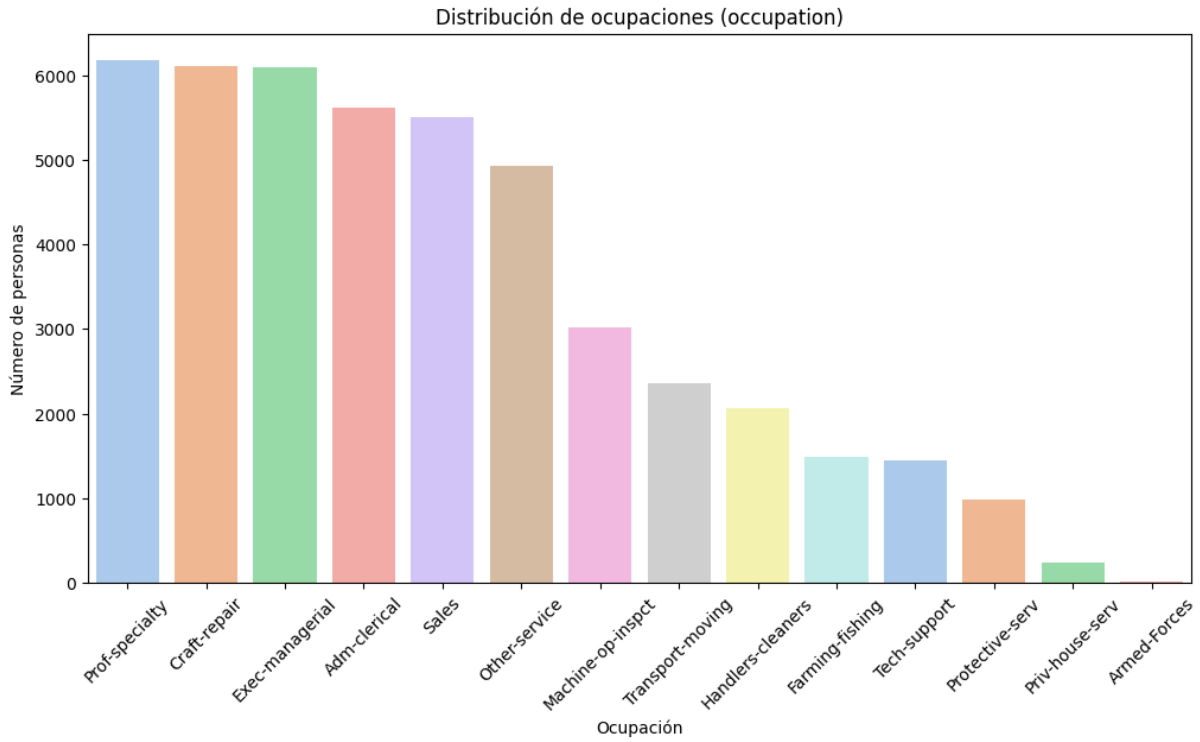

Figura 5: Distribución de la variable `occupation`, destacando que *Prof-specialty*, *Craft-repair* y *Exec-managerial* son las más frecuentes, y *Armed-Forces* la menos frecuente.

2.8. relationship

La columna `relationship` indica la relación de la persona con otros miembros del hogar. Los valores posibles incluyen *Husband*, *Not-in-family*, *Own-child*, *Unmarried*, *Wife* y *Other-relative*.

Esta variable es útil para entender la composición familiar de los individuos en el dataset y cómo se distribuyen las personas según sus vínculos familiares.

En la Figura 6 se muestra la distribución de las categorías de `relationship`. Se observa que las categorías más frecuentes son *Husband* y *Not-in-family*, mientras que *Other-relative* y *Wife* tienen menor representación. Esto indica que la mayoría de los individuos son cónyuges o personas que no están clasificadas dentro de una familia, mientras que los niños y otros parientes tienen menor presencia en el dataset.

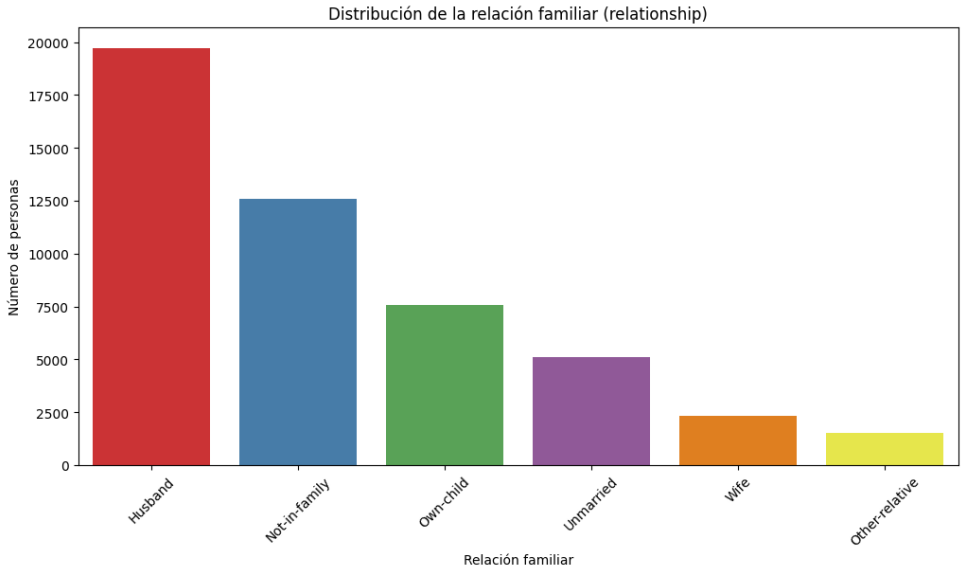

Figura 6: Distribución de la variable `relationship`, destacando que *Husband* y *Not-in-family* son las categorías más frecuentes.

2.9. race

La columna `race` indica la raza o etnia de cada persona en el dataset. Los valores posibles incluyen *White*, *Black*, *Asian-Pac-Islander*, *Amer-Indian-Eskimo* y *Other*.

Esta variable es útil para analizar la diversidad del dataset y estudiar posibles diferencias en distribución de ingresos o empleo según la etnia.

En la Figura 7 se muestra la distribución de las diferentes categorías de `race`. Se observa que la gran mayoría de los individuos se identifica como *White*, seguida de *Black*, mientras que las demás categorías (*Asian-Pac-Islander*, *Amer-Indian-Eskimo* y *Other*) tienen frecuencias significativamente menores. Esto refleja que el dataset está dominado por personas de raza blanca, con una representación más limitada de otras etnias.

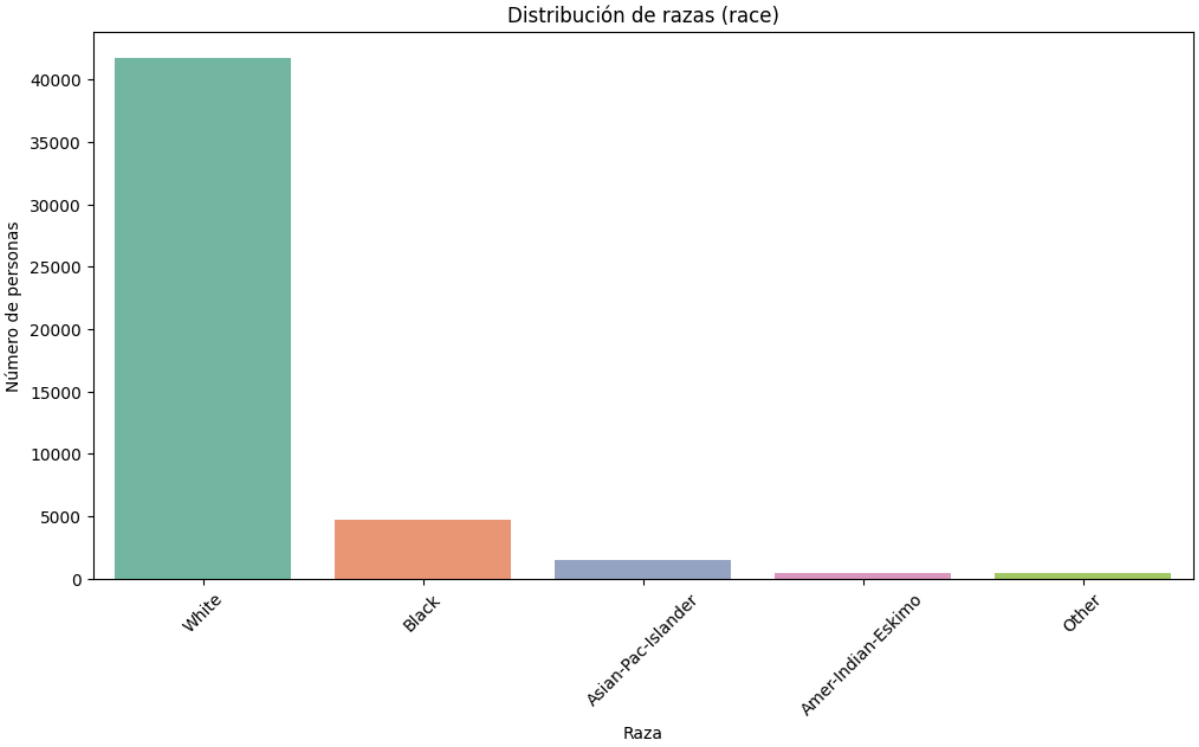

Figura 7: Distribución de la variable `race`, destacando la predominancia de la categoría *White*.

2.10. gender

La columna `gender` indica el género de cada persona en el dataset, con valores posibles *Male* y *Female*.

Esta variable es útil para analizar la distribución de hombres y mujeres en el dataset y explorar posibles diferencias en ingresos, ocupación o educación.

En la Figura 8 se observa que los individuos de género masculino duplican aproximadamente a los de género femenino. Esto refleja que hay una clara predominancia masculina en el dataset, lo cual puede influir en el análisis de otras variables relacionadas con empleo e ingresos.

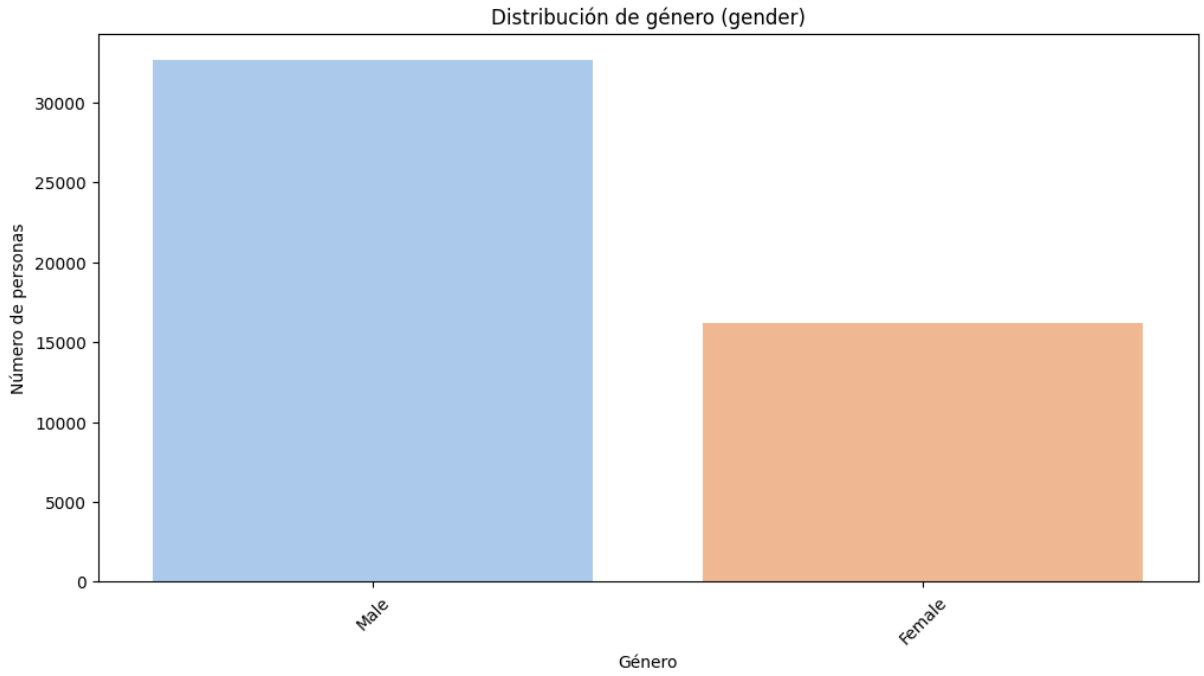

Figura 8: Distribución de la variable `gender`, mostrando la predominancia de *Male* sobre *Female*.

2.11. capital-gain

La columna `capital-gain` indica las ganancias de capital obtenidas por cada persona durante el año. La gran mayoría de los registros presenta un valor de cero, mientras que solo unos pocos individuos tienen ganancias significativas. Esto sugiere que la variable está muy sesgada y que unas pocas observaciones pueden influir de manera considerable en los ingresos totales.

2.12. capital-loss

La columna `capital-loss` representa las pérdidas de capital de cada persona durante el año. Al igual que en `capital-gain`, la mayoría de los valores son cero, con unos pocos casos de pérdidas relevantes. Esto indica que la variable también está muy sesgada y que las pérdidas afectan únicamente a un pequeño grupo de individuos.

2.13. hours-per-week

La columna `hours-per-week` indica el número de horas trabajadas por semana por cada persona. La mayoría de los individuos trabaja entre 35 y 45 horas semanales, aunque existen algunos valores extremos que representan jornadas más largas o muy cortas. Esta información es útil para analizar la relación entre carga laboral y nivel de ingresos.

2.14. native-country

La columna `native-country` indica el país de origen de cada individuo. La mayoría de los registros corresponde a *United-States*, mientras que otros países tienen representaciones mucho menores. Para no saturar el informe, en la Figura 9 se muestran únicamente los 10 países más frecuentes. Se observa claramente que *United-States* domina el dataset, seguido de países como *Mexico*, *Philippines* y *Germany*, mientras que los demás países tienen una presencia limitada.

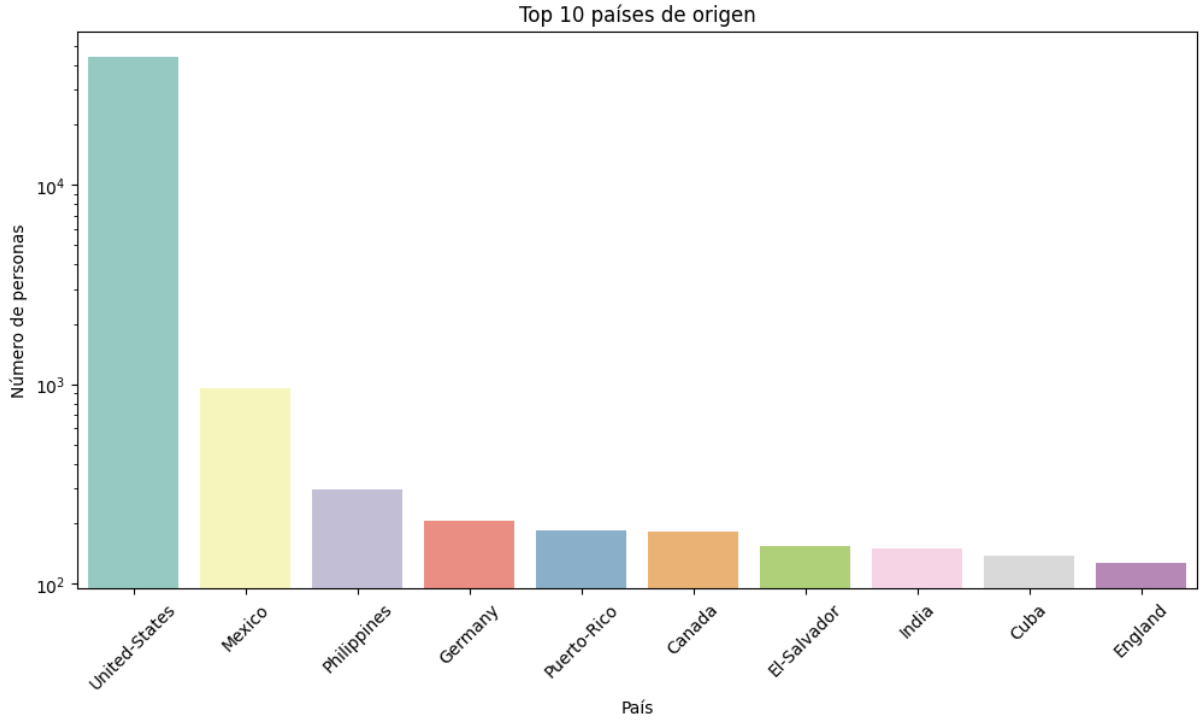

Figura 9: Distribución de los 10 países más frecuentes en la variable `native-country`, destacando la predominancia de *United-States*.

2.15. income

La columna `income` representa la variable objetivo del dataset, indicando si los ingresos de la persona son menores o iguales a 50K o superiores a 50K anuales. Se observa que la mayoría de los individuos tiene ingresos `<=50K`, mientras que un porcentaje menor supera los 50K. Esta variable es fundamental para el análisis posterior y la construcción de modelos predictivos.

3. Preprocesamiento de Datos

Una vez explorado el dataset, el siguiente paso consistió en preparar la información para que los modelos de Machine Learning pudieran trabajar correctamente. El preprocesamiento incluyó la detección y eliminación de valores faltantes, la codificación de variables categóricas y la simplificación de columnas redundantes.

3.1. Tratamiento de valores nulos

Al revisar el dataset se detectó que algunos valores desconocidos estaban representados con el carácter "?". Para tratarlos adecuadamente, primero se reemplazaron por valores nulos (`NaN`) y posteriormente se eliminaron las filas que los contenían. Esto se hizo de la siguiente forma:

```
df.replace('?', np.nan, inplace=True)
df.isnull().sum()

# Eliminar filas con valores nulos
df.dropna(inplace=True, ignore_index=True)

# Comprobar nuevamente
df.isnull().sum()
```

Tras ejecutar este proceso, el dataset quedó libre de valores faltantes. La decisión de eliminar las filas con valores nulos se tomó porque su proporción era pequeña en comparación con el total de registros (menos del 2 %), por lo que su eliminación no afectaba de manera significativa la representatividad de los datos.

3.2. Codificación de la variable objetivo

La variable `income` indica si una persona tiene ingresos `<=50K` o `>50K`. Dado que los modelos requieren datos numéricos, esta variable se transformó a formato binario mediante `LabelEncoder`, asignando 0 a `<=50K` y 1 a `>50K`:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['income'] = le.fit_transform(df['income'])
df.head()
```

3.3. Codificación de variables categóricas

Para el resto de variables categóricas, se aplicó *One-Hot Encoding*, una técnica que convierte cada categoría en una columna binaria (0 o 1). Este proceso evita introducir un orden artificial entre categorías y permite que los modelos interpreten correctamente los datos.

```
# One-hot encoding para variables categ ricas
categorical_cols = ['workclass', 'marital-status', 'occupation',
                    'relationship', 'race', 'gender', 'native-
                       country']

df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
```

Además, se convirtieron las columnas booleanas generadas por el *One-Hot Encoding* a valores enteros (0/1):

```
df = df.astype({col: int for col in df.columns if df[col].dtype ==
    'bool'})
```

3.4. Eliminación de columnas redundantes de educación

El dataset original contiene dos columnas relacionadas con el nivel educativo: `education` (categórica) y `educational-num` (numérica). Ambas transmiten la misma información, pero en formatos distintos. Para evitar redundancia y posibles problemas de multicolinealidad en los modelos, se realizaron los siguientes pasos:

1. Se eliminó la columna `education`.
2. Se renombró `educational-num` a `education`, para mantener un nombre claro y coherente.

```
df.drop(columns=['education'], inplace=True)
df.rename(columns={'educational-num': 'education'}, inplace=True)
df.head()
```

De esta forma, se conserva una única columna educativa con formato numérico, que resulta más útil para los modelos de clasificación.

3.5. Exportación del dataset limpio

Finalmente, el dataset preprocesado se guardó en un nuevo archivo CSV para su posterior uso en la fase de modelado:

```
df.to_csv("../data/processed/adult_clean.csv", index=False)
```

Con esto, se obtuvo una versión completamente limpia y lista del dataset, sin valores nulos, con variables codificadas correctamente y sin redundancias. Este conjunto de datos servirá como base para entrenar y evaluar los modelos de Árbol de Decisión y Random Forest.

4. Preparación de los datos para modelado

Para predecir la variable objetivo `income`, se entrenó un modelo de *Árbol de Decisión* y *Random Fores* utilizando las variables explicativas del dataset. Separamos las características y la variable objetivo de la siguiente manera:

```
X = df.drop(['income', 'fnlwgt'], axis=1)
y = df['income']
```

En este paso, se excluyen las columnas `income` y `fnlwgt` de las variables explicativas. La columna `income` corresponde a la variable objetivo que el modelo debe clasificar; incluirla como entrada invalidaría el entrenamiento, ya que el modelo tendría acceso directo a la respuesta correcta. Por su parte, `fnlwgt` representa el peso de cada registro en la muestra del censo y no aporta información relevante sobre los ingresos individuales. Incluirla podría introducir ruido y sesgar el modelo. Por tanto, la selección de `X` se centra únicamente en variables explicativas que realmente pueden influir en la clasificación de los ingresos.

4.1. División en conjunto de entrenamiento y prueba

Se dividió el dataset en conjuntos de entrenamiento y prueba utilizando un 80 % de los datos para entrenar y un 20 % para evaluar el modelo:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

5. Árbol de Decisión

5.1. Entrenamiento del modelo

Se creó un clasificador de Árbol de Decisión con una profundidad máxima de 8 para evitar sobreajuste:

```
from sklearn.tree import DecisionTreeClassifier

dtc = DecisionTreeClassifier(max_depth=12, random_state=42)
dtc.fit(X_train, y_train)
```

La elección de `max_depth=12` busca un equilibrio entre complejidad y generalización. Un árbol muy profundo podría ajustarse demasiado a los datos de entrenamiento (*overfitting*), capturando ruido y patrones irrelevantes, lo que reduciría su capacidad de predecir correctamente sobre datos nuevos. Por otro lado, un árbol demasiado superficial podría no capturar relaciones importantes entre variables (*underfitting*). Tras pruebas preliminares, se encontró que una profundidad de 12 ofrecía un buen compromiso, logrando un alto desempeño en el conjunto de prueba sin sobreajustar los datos.

5.2. Evaluación del modelo

Se evaluó el rendimiento del modelo sobre el conjunto de prueba utilizando las métricas de **exactitud** (*accuracy*), **precisión** (*precision*), **recuperación** (*recall*) y **F1-score**.

La exactitud indica la proporción de predicciones correctas sobre el total de muestras, mientras que la precisión y la recuperación se interpretan por clase:

- **Precisión:** de todas las predicciones que el modelo realizó para una clase determinada, qué proporción fue correcta.
- **Recuperación (Recall):** de todos los casos reales de una clase determinada, qué proporción el modelo logró identificar correctamente.
- **F1-score:** media armónica entre precisión y recall, útil para evaluar el desempeño en casos de clases desbalanceadas.

El modelo obtuvo los siguientes resultados:

Clase	**Precisión**	**Recall**	**F1-score**
0 (`<=50K`)	0.88	0.93	0.91
1 (`>50K`)	0.75	0.60	0.67
Exactitud total		0.853	

Cuadro 2: Reporte de clasificación del Árbol de Decisión, mostrando las métricas por clase y la exactitud global del modelo.

Como se observa, la clase 0 (`<=50K`) presenta alta precisión y recuperación, lo que significa que el modelo identifica correctamente la mayoría de los ingresos bajos y realiza pocas predicciones incorrectas. En cambio, la clase 1 (`>50K`) tiene menor recuperación, lo que indica que muchos individuos con ingresos altos no son correctamente clasificados. Esto se debe principalmente al **desbalance de clases** en el dataset: hay muchas más personas con ingresos bajos que altos, lo que hace que el modelo tienda a favorecer la clase mayoritaria.

5.2.1. Matriz de Confusión

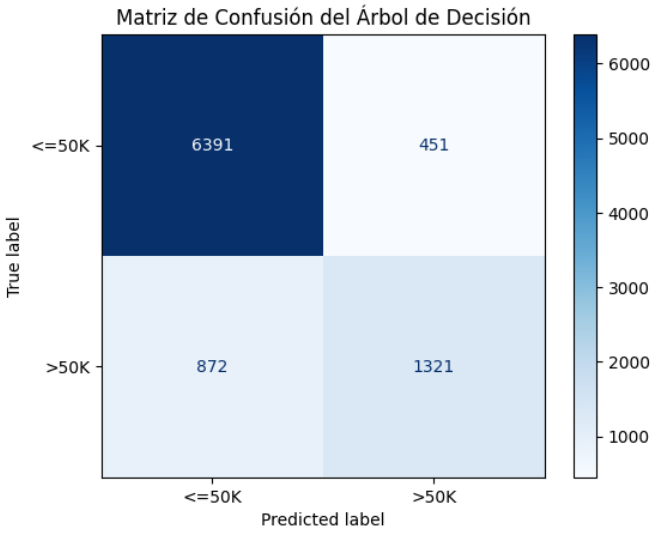

Figura 10: Matriz de confusión del Árbol de Decisión.

La Figura 10 muestra la matriz de confusión del modelo, que permite visualizar con mayor detalle la distribución de aciertos y errores:

- Para la clase `<=50K`, el modelo obtuvo **6391 aciertos** y solo **451 falsos positivos**, lo que confirma su buen desempeño.
- Para la clase `>50K`, el modelo logró **1321 aciertos**, pero presentó **872 falsos negativos**, es decir, personas con ingresos altos fueron clasificadas incorrectamente como si ganaran menos.

Este comportamiento refleja que el modelo funciona bien para la clase mayoritaria, pero tiene dificultades al identificar correctamente ingresos superiores a 50K, algo esperable debido al desbalance del dataset.

A pesar de ello, la **exactitud global** del modelo es del 85.3 %, lo que indica un desempeño general sólido. La combinación de métricas y la matriz de confusión permite comprender no solo cuán preciso es el modelo, sino también en qué tipo de casos tiende a fallar, lo cual es esencial para futuras mejoras y ajustes.

5.3. Visualización e Interpretación

Para interpretar el modelo, se visualizó el árbol de decisión hasta una profundidad de 3, mostrando las características más importantes y las reglas de decisión principales

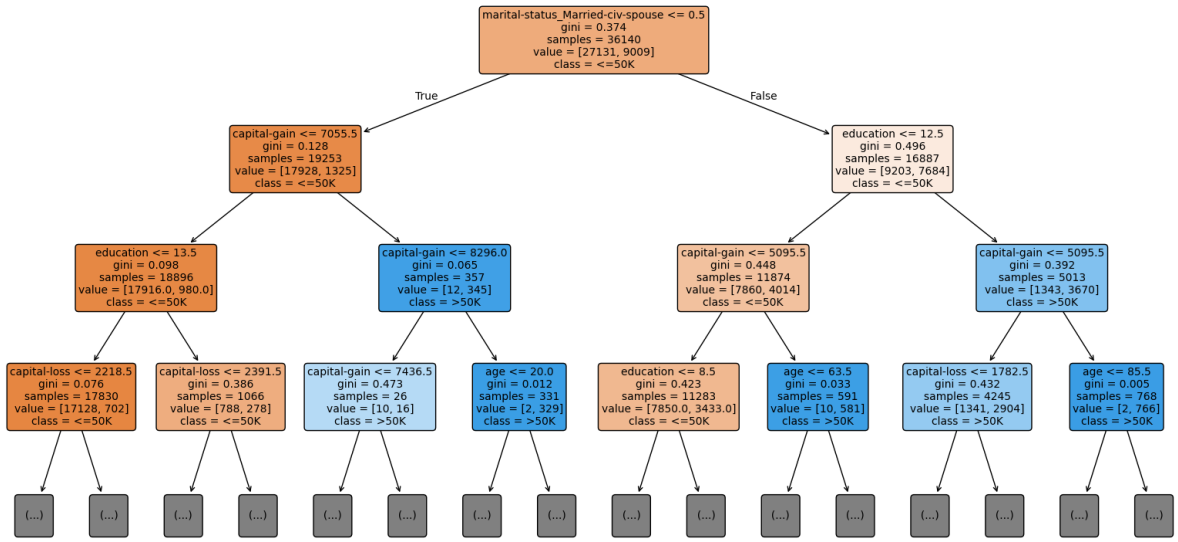

Figura 11: Visualización del Árbol de Decisión entrenado sobre el dataset Adult Income. Se muestran las primeras 3 capas del árbol para mayor claridad.

El árbol mostrado en la Figura 11 representa la estructura del modelo entrenado con el dataset *Adult Income*, truncado a una profundidad de tres niveles para facilitar su interpretación. Cada nodo interno corresponde a una decisión basada en una variable explicativa, y cada hoja indica la clase predominante (`<=50K` o `>50K`) junto con el número de muestras que llegan a ese punto.

Nodo raíz

La primera división se realiza con la variable `marital-status_Married-civ-spouse`, es decir, si la persona está casada civilmente. Esta característica es la más importante del modelo, con un índice de impureza *Gini* de 0.374. El árbol separa claramente a los individuos casados de los no casados:

- Los que **no están casados civilmente** tienden a pertenecer a la clase `<=50K`.
- Los que **sí están casados civilmente** tienen una probabilidad mucho mayor de pertenecer a la clase `>50K`.

Rama izquierda (no casados civilmente)

El siguiente criterio en esta rama es `capital-gain <= 7055.5`. Este umbral diferencia a las personas con pocas o ninguna ganancia de capital (la mayoría) de aquellas con ganancias significativas. Los individuos con baja o nula ganancia de capital suelen ganar `<=50K`. Dentro de este grupo, la variable `education` también juega un papel relevante: niveles educativos superiores a 13.5 se asocian con mayores ingresos, aunque en este subconjunto los casos de `>50K` siguen siendo minoritarios.

Rama derecha (casados civilmente)

En la rama correspondiente a personas casadas, el árbol divide nuevamente según `education <= 12.5`. Los individuos con menor nivel educativo presentan una mezcla de ingresos, mientras que aquellos con educación superior a ese umbral tienden claramente a tener ingresos `>50K`.

Además, las variables `capital-gain` y `age` refuerzan la separación:

- Un `capital-gain` mayor a 5095.5 está fuertemente asociado con la clase `>50K`.
- En los nodos más profundos, la variable `age` (edad) contribuye a refinar la predicción, indicando que las personas mayores suelen presentar ingresos más altos.

Conclusión interpretativa

El análisis del árbol de decisión permite identificar las variables más influyentes para predecir el nivel de ingresos:

1. `marital-status` (estado civil),
2. `capital-gain` (ganancias de capital),
3. `education` (nivel educativo),
4. y `age` (edad).

Estas variables son las que más contribuyen a discriminar entre los dos niveles de ingreso. En particular, estar casado civilmente, tener una educación universitaria o superior, y presentar ganancias de capital positivas son factores determinantes para pertenecer a la clase de ingresos `>50K`.

En cambio, las personas no casadas y con pocas o ninguna ganancia de capital presentan alta probabilidad de ubicarse en la clase `<=50K`. El árbol, aunque simplificado en profundidad 3, permite observar de manera intuitiva las reglas que el modelo utiliza para la clasificación, y confirma que las variables socioeconómicas y demográficas desempeñan un papel central en la predicción de los ingresos.

6. Random Forest

Para mejorar la robustez y reducir el sobreajuste de un solo árbol, se entrenó un modelo de *Random Forest* utilizando el mismo conjunto de características y división de datos que el Árbol de Decisión.

6.1. Entrenamiento del modelo

Se utilizó el clasificador `RandomForestClassifier` con `random_state=42` para garantizar reproducibilidad:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

6.2. Evaluación del modelo

El rendimiento del modelo **Random Forest** se evaluó utilizando las mismas métricas que en el Árbol de Decisión: **precisión**, **recuperación** (*recall*), **F1-score** y **exactitud**. Los resultados obtenidos fueron los siguientes:

Clase	Precisión	Recall	F1-score
0 (<=50K)	0.88	0.91	0.90
1 (>50K)	0.70	0.63	0.66
Exactitud total		0.844	

Cuadro 3: Reporte de clasificación del modelo Random Forest, mostrando las métricas por clase y la exactitud global.

Como se observa, la clase 0 (<=50K) presenta valores altos en precisión y recall, indicando que el modelo identifica correctamente la mayoría de los individuos con ingresos bajos. Por otro lado, la clase 1 (>50K) muestra un rendimiento menor, especialmente en la métrica de *recall*, lo que indica que el modelo sigue teniendo dificultades para identificar correctamente algunos casos de ingresos altos, aunque su desempeño es ligeramente mejor que el del Árbol de Decisión.

6.2.1. Matriz de Confusión

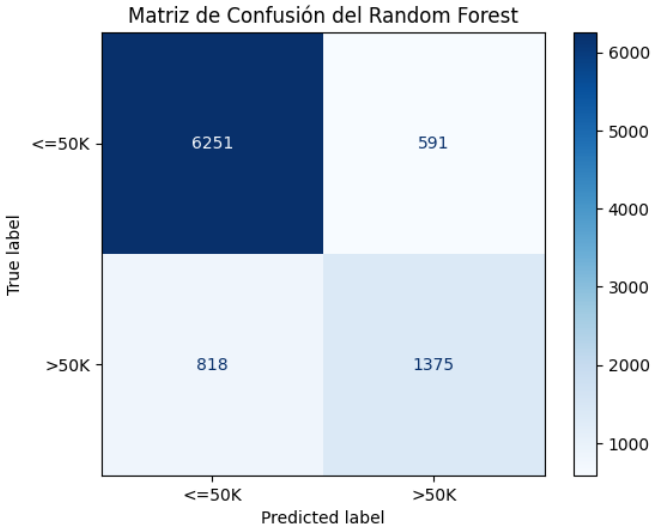

Figura 12: Matriz de confusión del modelo Random Forest.

La matriz de confusión de la Figura 12 permite analizar con más detalle los aciertos y errores del modelo:

- Para la clase `<=50K`, el modelo obtuvo **6251 aciertos** y **591 falsos positivos**, lo que confirma un rendimiento estable en la clase mayoritaria.
- Para la clase `>50K`, se registraron **1375 aciertos** y **818 falsos negativos**. Aunque sigue habiendo una cantidad significativa de errores, el modelo mejora ligeramente respecto al Árbol de Decisión en la correcta identificación de ingresos altos.

En general, el modelo Random Forest mantiene un buen rendimiento global con una **exactitud del 84.4 %**. Sin embargo, al igual que el modelo anterior, muestra dificultades al clasificar correctamente la clase minoritaria (`>50K`), lo cual está relacionado con el **desbalance de clases** presente en el conjunto de datos. Aun así, el uso de múltiples árboles permite al Random Forest conseguir una ligera mejora en los verdaderos positivos para la clase `>50K` en comparación con el Árbol de Decisión.

6.3. Visualización e Interpretación

Aunque el modelo **Random Forest** está compuesto por cientos de árboles, es posible visualizar uno de ellos para comprender de forma intuitiva cómo el conjunto de árboles aprende las relaciones presentes en los datos. Para ello, se graficó uno de los árboles individuales del bosque, truncado a una profundidad de 3 niveles con el fin de facilitar la interpretación de sus reglas de decisión.

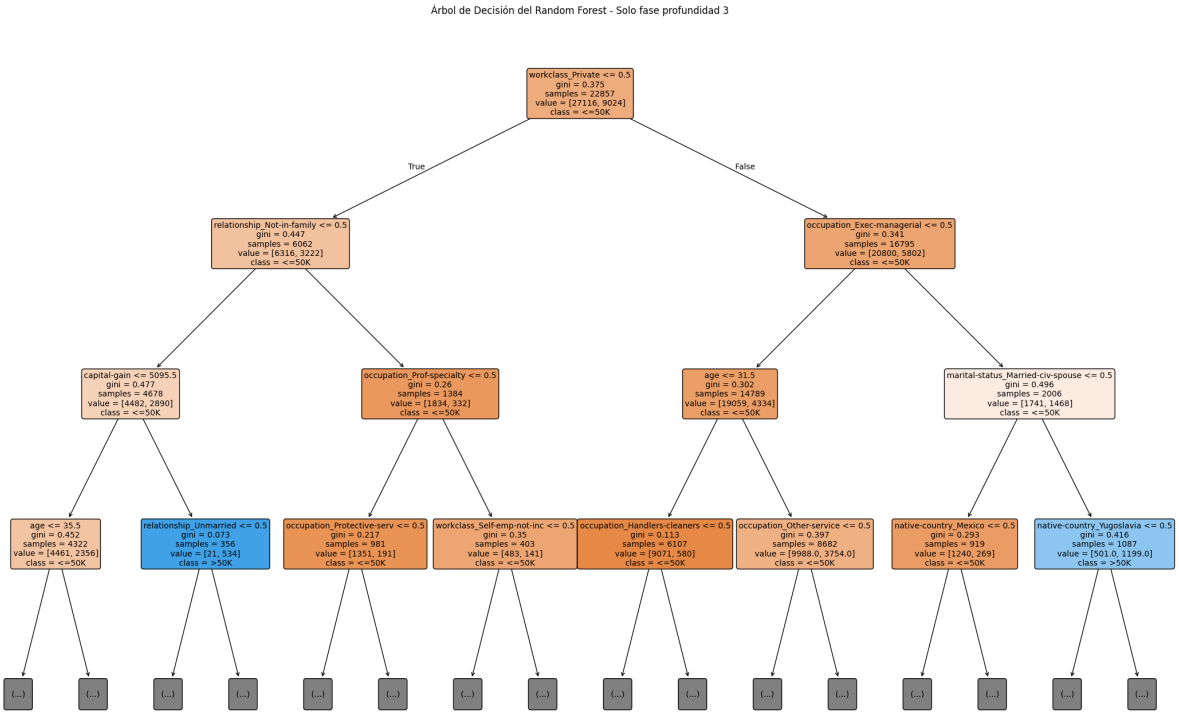

Figura 13: Visualización de un árbol de decisión perteneciente al Random Forest, truncado a una profundidad de 3.

La Figura 13 muestra uno de los árboles que componen el modelo Random Forest entrenado sobre el dataset *Adult Income*. Aunque cada árbol es diferente debido a la selección aleatoria de muestras y características (*bootstrap* y *feature bagging*), este árbol permite observar las reglas más comunes que el bosque tiende a aprovechar para tomar decisiones.

Cada nodo interno representa una división basada en una característica, mientras que cada hoja indica la clase mayoritaria (`<=50K` o `>50K`) y el número de muestras que caen en ese nodo.

Nodo raíz

El árbol comienza dividiendo por la variable `workclass_Private`. Esta característica distingue entre individuos que trabajan en el sector privado y quienes pertenecen a otros tipos de empleo. El nodo raíz presenta un índice Gini de 0.273, lo que indica que la división inicial ofrece una separación relativamente clara entre ambas clases, favoreciendo la clasificación hacia `<=50K`.

Rama izquierda (trabajadores no privados)

En el lado izquierdo del árbol, el siguiente criterio utilizado es `relationship_Not-in-family`. Esto refleja que el modelo considera relevante si la persona vive o no en el núcleo familiar directo. En esta rama:

- Las personas que **no pertenecen** a un núcleo familiar tradicional tienden a clasificarse como `<=50K`.
- Dentro de esta subdivisión, variables como `capital-gain` y `occupation_Prof-specialty` refinan la predicción.

Por ejemplo:

- Un `capital-gain <= 5095.5` conduce a predicciones principalmente hacia `<=50K`.
- La característica `relationship_Unmarried` aparece como un determinante importante: las personas solteras muestran fuerte tendencia a estar en la clase `<=50K`.

En los nodos más profundos, variables como `occupation_Protective-serv` o `Self-emp-not-inc` permiten separar pequeños subgrupos con leves diferencias en la distribución de ingresos.

Rama derecha (trabajadores del sector privado)

En la rama derecha, el modelo divide según `occupation_Exec-managerial`. Esta característica identifica ocupaciones de tipo ejecutivo o gerencial, que suelen estar asociadas a mayores niveles de ingreso.

A continuación, la variable `age` juega un papel decisivo:

- Para edades menores o iguales a 31.5 años, la mayoría de los individuos se clasifica como `<=50K`.
- Para edades mayores, la proporción de ingresos `>50K` aumenta, aunque el árbol sigue mostrando predominio de la clase `<=50K` en los nodos analizados.

Más adelante, aparecen variables relacionadas con la ocupación específica del individuo, como `Handlers-cleaners`, `Other-service` o características del país de origen. Aunque estas divisiones generan nodos con baja impureza, reflejan patrones específicos de subpoblaciones dentro del conjunto de datos.

Conclusión interpretativa

Este árbol individual del Random Forest revela varios aspectos clave:

1. Características laborales como `workclass` y `occupation` son altamente relevantes en la separación entre ingresos bajos y altos.
2. La edad y el estatus relacional (`relationship`) contribuyen de manera consistente a refinar la predicción.
3. La mayoría de las ramas del árbol tienden a clasificar hacia `<=50K`, lo cual está alineado con el desbalance de clases del dataset.

Aunque cada árbol del Random Forest es distinto, el análisis de este ejemplar permite comprender las reglas generales que el modelo emplea: factores laborales, edad, nivel de integración familiar y características ocupacionales son determinantes esenciales para predecir los ingresos de un individuo. Este tipo de visualización facilita interpretar la lógica interna del modelo, aun cuando su verdadero poder proviene de la agregación de muchos árboles individuales.

7. Comparación de Modelos

Se comparan los modelos de Árbol de Decisión y Random Forest utilizando métricas de desempeño por clase y exactitud global. Esta presentación vertical facilita la lectura y evita que los títulos se colapsen.

Modelo	**Clase**	**Precisión**	**Recall**	**F1-score**
Árbol de Decisión	<=50K	0.88	0.93	0.91
	>50K	0.75	0.60	0.67
Random Forest	<=50K	0.88	0.91	0.90
	>50K	0.70	0.63	0.66
Exactitud global		Árbol de Decisión 0.85 | Random Forest 0.84		

Cuadro 4: Comparación vertical de desempeño de Árbol de Decisión y Random Forest por clase y exactitud global.

Análisis de la tabla

- **Clase <=50K:** Ambos modelos identifican correctamente la mayoría de ingresos bajos. Árbol de Decisión tiene recall ligeramente superior (0.93 vs 0.91).
- **Clase >50K:** Random Forest detecta más ingresos altos (recall 0.63 vs 0.60), aunque Árbol de Decisión mantiene mayor precisión (0.75 vs 0.70).
- **Exactitud global:** Árbol de Decisión ligeramente mejor (0.85 vs 0.84).
- **Interpretabilidad y robustez:** Árbol de Decisión más simple de interpretar; Random Forest más robusto y menos propenso a sobreajuste.
- **Variables importantes:** Coinciden en la relevancia de estado civil, educación, ganancias de capital, edad y ocupación.

Conclusión

La elección del modelo depende del objetivo: rapidez e interpretabilidad (Árbol de Decisión) o robustez y mejor detección de la clase minoritaria (Random Forest).

8. Conclusiones

Tras el análisis realizado, se pueden extraer las siguientes conclusiones:

1. Las variables que más influyen en la predicción de ingresos son el **estado civil**, la **educación**, las **ganancias de capital**, la **edad** y características laborales específicas como `workclass` y `occupation`.

2. Ambos modelos clasifican correctamente la mayoría de los ingresos bajos (`<=50K`), pero presentan dificultades para predecir correctamente los ingresos altos (`>50K`) debido al desbalance de clases.

3. El Árbol de Decisión es más interpretable y permite visualizar claramente reglas de decisión, mientras que Random Forest ofrece mayor robustez y ligera mejora en la identificación de ingresos altos, aunque a costa de complejidad interpretativa.

4. El preprocesamiento de datos, incluyendo manejo de valores nulos y codificación de variables categóricas, es clave para lograr un buen desempeño de los modelos.

5. Este análisis demuestra que técnicas de *Machine Learning* pueden ser efectivas para predecir niveles de ingresos a partir de variables demográficas y laborales, siempre considerando el balance de clases y la interpretación de resultados.