

**Universidad Internacional San Isidro Labrador**

Proyecto Final

Data Science

Modelo predictivo sobre los ingresos de la población adulta.

Prof. Dr. Samuel Saldaña Valenzuela

Fabián León Mendoza

Mayo, 2024

## Justificación

El desarrollo de sistemas capaces de predecir datos con eficiencia es sin duda uno de los temas en los que la ciencia y la tecnología hoy invierte más tiempo y recursos. La gran cantidad de cambios que vivimos como sociedad nos obliga a cambiar y evolucionar junto con ella, cada día vemos como personas que han dedicado su vida a una cierta actividad económica, a una habilidad específica o a una herramienta en específico son las que se quedan rezagadas o estancadas con mayor velocidad.

Somos observadores y agentes de cambio en uno de los momentos más cruciales de la historia de la humanidad, el mundo IT ha sufrido una revolución tras otra desde la popularización y el uso masivo del internet. Somos capaces de ver con nuestros propios ojos como las máquinas comienzan a ser independientes y comienzan a reemplazar personas en trabajos.

Muchas veces en algunos sectores tech las personas están tan acostumbradas al crecimiento sin freno que pasar un año sin un aumento económico o sin un ascenso es básicamente un sinónimo de estancamiento, si lo comparamos con la parábola bíblica de las vacas gordas y las vacas flacas el sector tecnológico ha visto cómo sus vacas crecen exponencialmente cada año desde hace prácticamente 15 años.

El minado y el análisis de datos es una de las nuevas industrias que se han generado producto de estos cambios abruptos y hoy día las empresas más grandes del mundo buscan la forma de poder sacar beneficio con los datos e información que las personas les brindan como si del mismísimo Santo Grial se tratara.

La forma en que el uso y el análisis de los datos influye en la cultura organizacional de una empresa es extraordinario y las organizaciones que logran sacarle el mayor provecho a este tema serán las que evolucionen y vivan para luchar en el mañana.

Más allá de un modelo de Machine Learning esta documentación se va a enfocar en cómo una organización común y cualquiera puede llegar a conseguir un sistema como el que se va a presentar a continuación y sobre todo se tratará de explicar cuales son los mayores beneficios de abrir las puertas de su organización a el análisis de los datos.

## **Objetivos**

### **Objetivo general**

1. Crear un modelo capaz de predecir efectivamente los ingresos de una persona según datos socioeconómicos de una persona

### **Objetivos específicos**

1. Estudiar el dataset dado para la creación del modelo mediante la búsqueda de medidas estadísticas y no estadísticas que ayuden a comprender la naturaleza de los datos.
2. Ejecutar la metodología de trabajo CRISP-DM explicando en cada proceso la importancia del mismo, esto con el fin de crear un entorno de desarrollo informado y eficiente.
3. Realizar una limpieza completa de los datos en búsqueda de datos incompletos, faltantes o redundantes.
4. Informar a las personas interesadas sobre la forma en la que se desarrolló el proyecto a un nivel técnico, involucrando a todas las partes en este proceso con el fin de poder contar con agentes de cambio con vistas y opiniones variadas.
5. Explicar los diferentes retos y obstáculos que una organización puede sufrir a la hora de aplicar un algoritmo de este tipo, asimismo destacando las fortalezas y el impacto que puede llegar a tener el aprovechamiento de los datos en cualquier organización.
6. Mostrar con gráficas y documentos visuales las características y el alcance de este proyecto, esto para favorecer una mayor comprensión de los temas propuestos en este documento.

## Tecnologías usadas en este proyecto

A continuación se enumeran las tecnologías utilizadas para el desarrollo del algoritmo

### Lenguajes de programación

- Python

### Frameworks o librerías

- Pandas
- NumPy
- SkLearn
- Matplotlib
- Seaborn

#### Python

Python es un **lenguaje de programación de alto nivel**, orientado a objetos, con una semántica dinámica integrada, principalmente para el **desarrollo web**, **aplicaciones informáticas** y **ciencia de datos**.

#### Pandas

Pandas es una biblioteca de Python que se utiliza para **trabajar con conjuntos de datos**. Tiene funciones para **analizar**, **limpiar**, **explorar** y **manipular datos**.

Pandas permite **analizar big data** y sacar conclusiones basadas en teorías estadísticas. Pandas puede **limpiar conjuntos de datos** desordenados y hacerlos legibles y relevantes.

## NumPy

NumPy es una biblioteca de Python que se utiliza para **trabajar con matrices**.

También cuenta con funciones para trabajar en el dominio del **álgebra lineal**, **transformada de Fourier** y **matrices**.

## SkLearn

Scikit-learn, también conocido como sklearn, es una biblioteca de código abierto, aprendizaje automático y modelado de datos para Python. Cuenta con varios algoritmos de clasificación, regresión y agrupamiento, incluidas máquinas de vectores de soporte, bosques aleatorios, aumento de gradiente, k-means y DBSCAN, y está diseñado para interoperar con las bibliotecas de Python, NumPy y SciPy.

## Matplotlib

Matplotlib es una biblioteca multiplataforma de visualización de datos y trazado gráfico (histogramas, diagramas de dispersión, gráficos de barras, etc.) para Python y su extensión numérica NumPy.

## Seaborn

Seaborn es una biblioteca de visualización de datos de Python basada en matplotlib. Proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos e informativos.

## **Que es CRISP-DM**

CRISP-DM, que son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar sus trabajos de minería de datos. El ciclo vital del modelo contiene seis fases con flechas que indican las dependencias más importantes y frecuentes entre fases. La secuencia de las fases no es estricta. De hecho, la mayoría de los proyectos avanzan y retroceden entre fases si es necesario.

CRISP-DM es ampliamente reconocida en la comunidad de analistas de datos y científicos de datos como la metodología más eficiente y efectiva para llevar a cabo proyectos de minería de datos. Su enfoque estructurado y paso a paso proporciona una hoja de ruta clara para el éxito. A continuación se presentan algunos motivos por la cual CRISP-DM es la metodología preferida por muchas personas para realizar proyectos de Data Mining:

### **1. Estructura Modular**

CRISP-DM divide el proceso de minería de datos en seis fases claramente definidas: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación y Despliegue. Esto facilita la administración y el seguimiento del progreso del proyecto además permite hacer entregas al negocio en cada uno de estos pasos lo cual agrega valor al proceso.



## 2. Enfocado en el Negocio

Coloca un fuerte énfasis en la comprensión de los objetivos y necesidades del negocio, lo que garantiza que los resultados sean relevantes y aplicables en el mundo real.

## 3. Flexibilidad

A pesar de su estructura sólida, CRISP-DM es lo suficientemente flexible como para adaptarse a una variedad de proyectos y tipos de datos.

## 4. Ciclo Iterativo

CRISP-DM es un ciclo continuo de mejora. Los resultados y conocimientos adquiridos en una fase pueden retroalimentar las anteriores, permitiendo un refinamiento constante.

### **Ejemplo de Aplicación: Segmentación de Clientes y Modelado Predictivo**

Supongamos que una empresa de servicios financieros desea segmentar a sus clientes en tres grupos en función de su historial de pagos: aquellos que pagan dentro de los 30 días, aquellos que pagan entre 31 y 60 días, y aquellos que pagan más de 60 días tarde. Además, la empresa busca identificar a tiempo a los clientes que podrían caer en mora y ser remitidos a cartera castigada. Aplicaremos CRISP-DM para abordar este problema.

## 1. Comprensión del Negocio

En esta fase, se colabora estrechamente con el equipo de negocios para definir los objetivos y criterios de éxito. En este caso, el objetivo es segmentar a los clientes y predecir retrasos en los pagos.

## 2. Comprensión de los Datos

Se recopilan y exploran los datos de historial de pagos. Se identifican las variables clave, como el historial de pagos, la antigüedad de la cuenta y la categoría del cliente.

## 3. Preparación de los Datos

Los datos se limpian y se transforman en un formato adecuado para el modelado. Se pueden utilizar herramientas como Pandas y NumPy en Python para este proceso.

## 4. Modelado

Se seleccionan y entrenan modelos predictivos, como regresión logística o árboles de decisión, utilizando bibliotecas de machine learning como Scikit-Learn.

## 5. Evaluación

Se evalúa el rendimiento del modelo utilizando métricas como precisión, recall y F1-score. Se ajustan los modelos según sea necesario.

## 6. Despliegue

Una vez que se ha alcanzado un modelo satisfactorio, se implementa en el entorno de producción para su uso continuo.

## **Comprensión del negocio**

Antes de adentrarnos en el desarrollo del modelo predictivo es esencial establecer los alcances y objetivos del proyecto. Esta etapa resulta fundamental para garantizar un proceso de desarrollo ordenado, enfocado y alineado con las expectativas del proyecto.

El objetivo principal de este proyecto es crear un modelo predictivo eficaz que pueda predecir si una persona es capaz de generar más de \$50,000 al año, basándose en variables como Estado Civil, Nivel de Educación, Edad y otras características relevantes.

Un modelo con estas capacidades tiene el potencial de convertirse en una herramienta de gran valor para empresas a nivel mundial. La capacidad de segmentar a la población según sus ingresos puede ser clave para aumentar las ventas de productos o servicios. No solo el poder adquisitivo de una persona influye en sus decisiones de compra, sino que también sus preferencias y gustos están estrechamente ligados a sus ingresos.

La implementación exitosa de este modelo predictivo puede tener un impacto significativo en el desempeño comercial de la empresa, permitiéndole maximizar sus esfuerzos de marketing, aumentar la satisfacción del cliente y mejorar la rentabilidad global.

## Comprensión de los Datos

El dataset llamado Adult Census Income será utilizado para desarrollar un modelo predictivo sobre los ingresos de la población adulta, este dataset en su origen contiene más de 32000 registros con una totalidad de 15 columnas, a continuación se hace un breve resumen de las columnas brindado por la fuente original del dataset.

**age:** continuous.

**workclass:** Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

**fnlwgt:** final weight, continuous.

**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.

**education-num:** continuous.

**marital-status:** Represents the responding unit's role in the family.

Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

**occupation:** Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

**relationship:** Represents the responding unit's role in the family. Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

**race:** White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

**sex:** Female, Male.

**capital-gain:** income from investment sources, apart from wages/salary, continuous.

**capital-loss:** losses from investment sources, apart from wages/salary, continuous.

**hours-per-week:** continuous.

**native-country:** United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&Tobago, Peru, Hong, Holand-Netherlands.

## Descripción columnas

Age: Edad del encuestado, mayor:90, menor:17, sin datos nulos o faltantes

Workclass: Clase trabajadora del encuestado, contiene datos faltantes  
fnlwgt: Cálculo realizada con el dataset tomando inclusive datos externos para realizarla, se elimina del dataset, sin datos nulos o faltantes  
Education: Nivel de educación del encuestado, columna eliminada del dataset, sin datos nulos o faltantes.  
education.num: Número entero representando el nivel de escolaridad 1 para el nivel más bajo, 16 para el nivel más alto.  
marital.status: Estado civil  
occupation: Profesión o trabajo realizado  
relationship: Representa el papel en la familia.  
sex: Sexo.  
capital.gain: Los ingresos provenientes de fuentes de inversión, aparte de sueldos/salarios.  
capital.loss: Pérdidas provenientes de fuentes de inversión, aparte de sueldos/salarios.  
hours-per-week: Horas trabajadas por semana.  
native.country: País de origen, contiene datos faltantes.

## **Origen de la información del Dataset**

Estos datos fueron extraídos de la base de datos de la oficina del censo de 1994 por Ronny Kohavi y Barry Becker. La tarea es determinar si una persona gana más de 50.000 dólares al año.

## Preparación de los datos

Es importante para la realización de un modelo adecuado acorde a los objetivos del proyecto hacer un análisis previo de los datos presentados y prepararlos para la etapa del modelado. En esta etapa es vital el hecho de no solo conocer los datos sino que también la relevancia o el valor que estos tienen con el fin de buscar ángulos de investigación para explorar en etapas posteriores del desarrollo.

```
data.shape  
  
(32561, 15)
```

El dataset original cuenta con 23560 registros y 15 columnas

```
[11] data['workclass'].value_counts()
```

workclass	
Private	22696
Self-emp-not-inc	2541
Local-gov	2093
?	1836
State-gov	1298
Self-emp-inc	1116
Federal-gov	960
Without-pay	14
Never-worked	7
Name: count, dtype: int64	

```
[14] data['occupation'].value_counts()
```

occupation	
Prof-specialty	4140
Craft-repair	4099
Exec-managerial	4066
Adm-clerical	3770
Sales	3650
Other-service	3295
Machine-op-inspct	2002
?	1843
Transport-moving	1597
Handlers-cleaners	1370
Farming-fishing	994
Tech-support	928
Protective-serv	649
Priv-house-serv	149
Armed-Forces	9
Name: count, dtype: int64	

```
data['native.country'].value_counts()
```

native.country	
United-States	29170
Mexico	643
?	583
Philippines	198
Germany	137
Canada	121
Puerto-Rico	114
El-Salvador	106
India	100
Cuba	95

Las columnas 'workclass', 'occupation' y 'native.country' tienen datos faltantes, estos son representados por un signo de interrogación. Estos valores son cambiados por NaN y cambiados por la moda o el valor más utilizado en cada una de estas características.

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

[ ] data = data.drop(['education', 'fnlwgt'], axis=1)
```

```
[ ] columnas_con_nan = ['workclass', 'occupation', 'native.country']

[ ] for col in columnas_con_nan:
    data[col].fillna(data[col].mode()[0], inplace=True)
```

```
#Se eliminan 2 columnas de el dataset que parecían innecesarias o que no eran útiles en este contexto
data = data.drop(['education', 'fnlwgt'], axis=1)
```

Se eliminaron 2 columnas del dataset original (Education. fnlwgt) debido a que Education era representada en 2 columnas y se decidió eliminar una de ellas. La razón por la cual se eliminó fnlwgt fue debido a que era un ponderado realizado en el momento de la recolección de los datos incluyendo datos de otras fuentes.

```
#Se cambian los datos de la columna income, si genera -50000
data['income'].replace({'<=50K':0, '>50K':1}, inplace=True)
```

Con el objetivo de hacer más sencillo el manejo de los datos se cambian los valores de la columna 'income' por 0 y 1. El primer valor para representar ingresos menores o iguales a \$50 000 y el segundo por ingresos por encima de esa cifra.

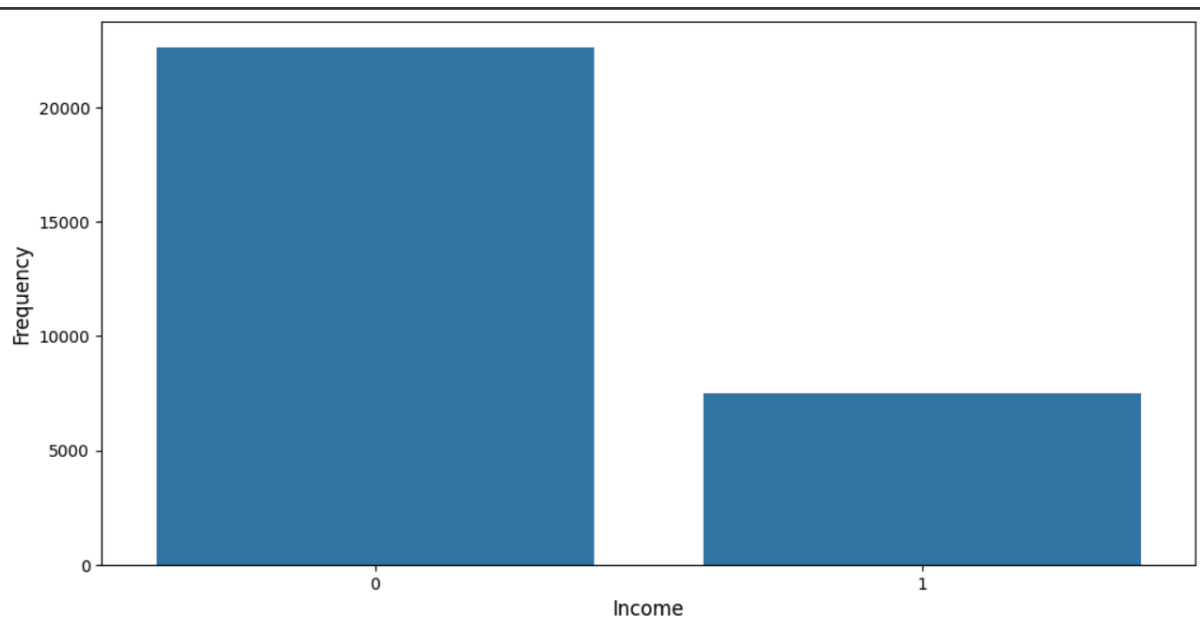


```
data.describe()
```

	age	education.num	capital.gain	capital.loss	hours.per.week	income
count	30162.000000	30162.000000	30162.000000	30162.000000	30162.000000	30162.000000
mean	38.437902	10.121312	1092.007858	88.372489	40.931238	0.248922
std	13.134665	2.549995	7406.346497	404.298370	11.979984	0.432396
min	17.000000	1.000000	0.000000	0.000000	1.000000	0.000000
25%	28.000000	9.000000	0.000000	0.000000	40.000000	0.000000
50%	37.000000	10.000000	0.000000	0.000000	40.000000	0.000000
75%	47.000000	13.000000	0.000000	0.000000	45.000000	0.000000
max	90.000000	16.000000	99999.000000	4356.000000	99.000000	1.000000

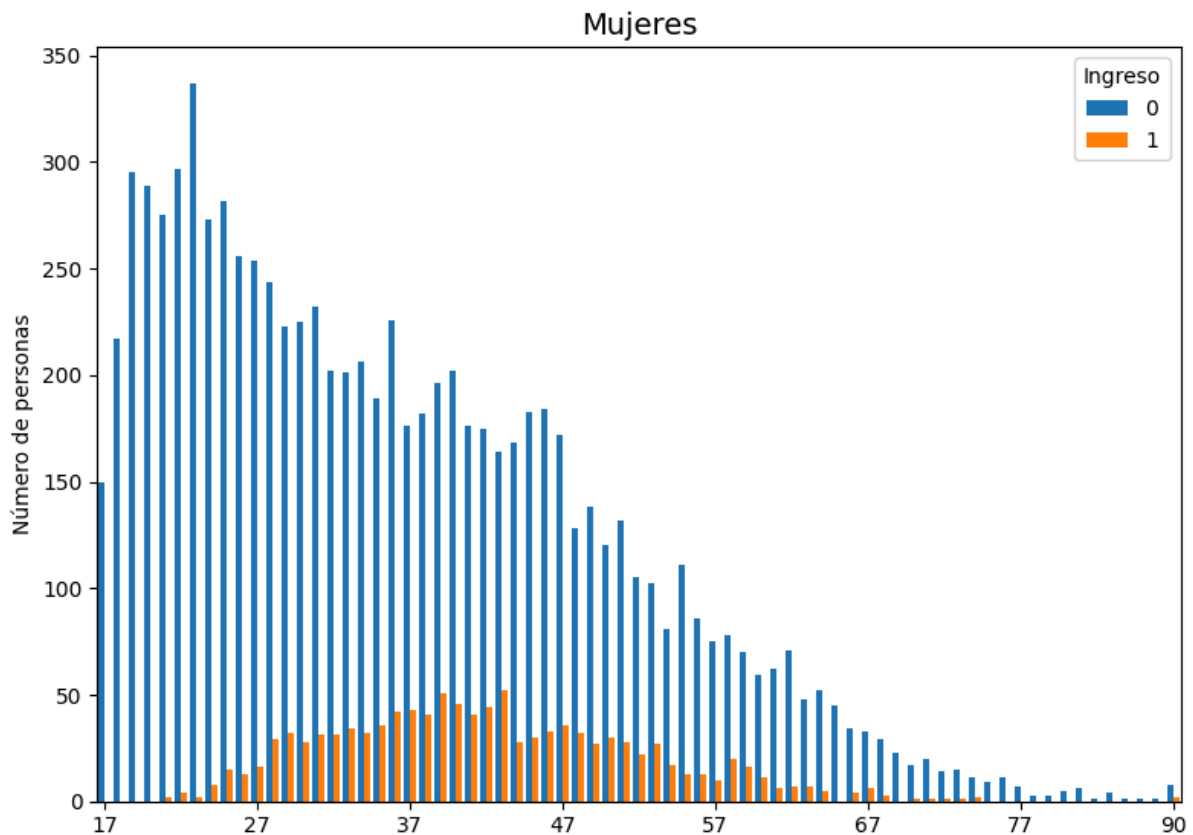
Medidas estadísticas luego de la limpieza de datos. La cual revela datos importantes para comprender los resultados finales del dataset como la edad media de los encuestados o la media de horas trabajadas por semana de la población.

## Visualización de datos

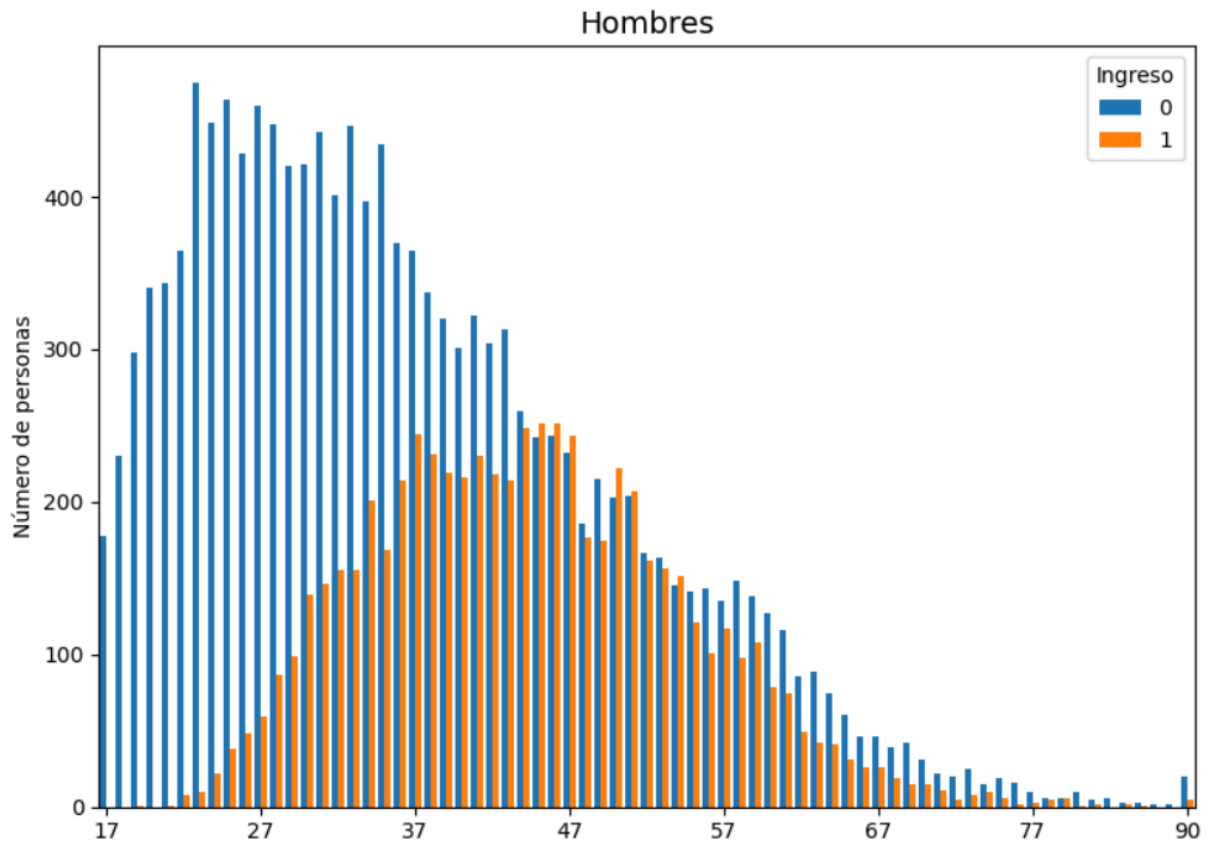


En esta matriz observamos 2 columnas, una representa las personas que ganan <\$50000 anuales (0) y la otra es para representar las personas que si logran superar esta barrera, vemos claramente un desbalanceo muy marcado de las clases lo que genera que la precisión de este dataset se vea afectado, esto se soluciona más adelante utilizando Random Oversampling, el proceso detrás de este cambio se explica más adelante en este documento.

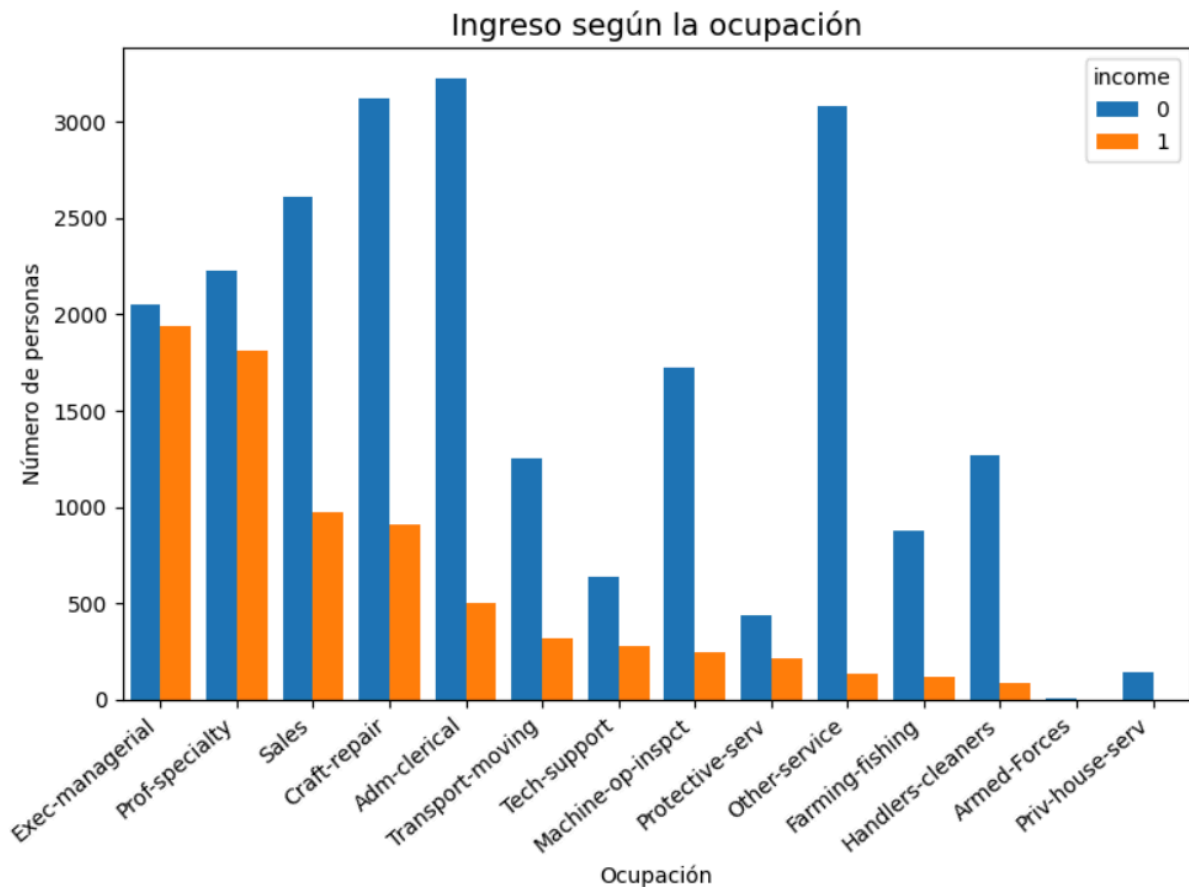
## Ingreso por edad



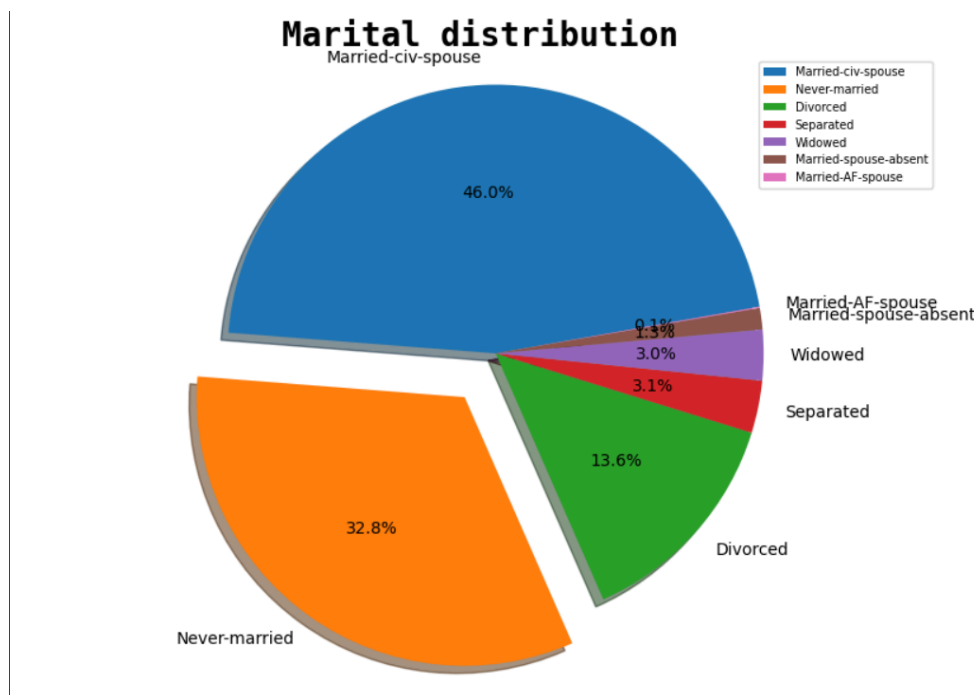
La siguiente métrica que vemos es una tabla que muestra los ingresos dividido por sexo, vemos que el porcentaje de mujeres que logran pasar la barrera de los \$50000 son muy pocas. Este fenómeno se puede deber a que usualmente las mujeres trabajan menos horas por semana lo cual se puede ver reflejado en este dataset y además que no suele ser tan común que las mujeres estudien carreras relacionadas a las ciencias o matemáticas las cuales tienen una remuneración más alta.



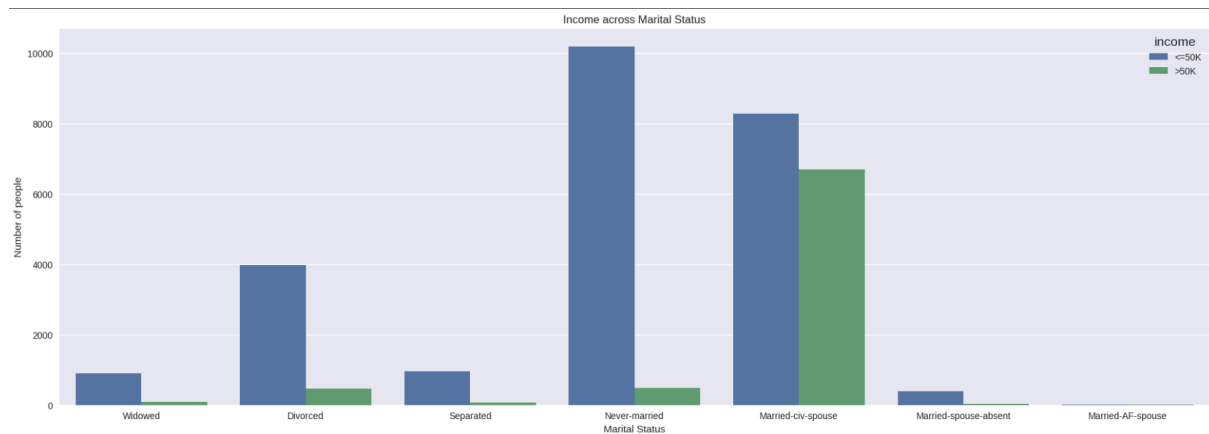
Analizando la métrica de hombres vemos algunas diferencias con respecto al género opuesto como que la cantidad de estos que generan >\$50000 es abismalmente superior, vemos una mayor cantidad de personas trabajadoras en todas las edades y logramos ver como de los 30s a los 50s son los años en donde las personas entran en su pico de productividad y es donde estas generan más dinero.



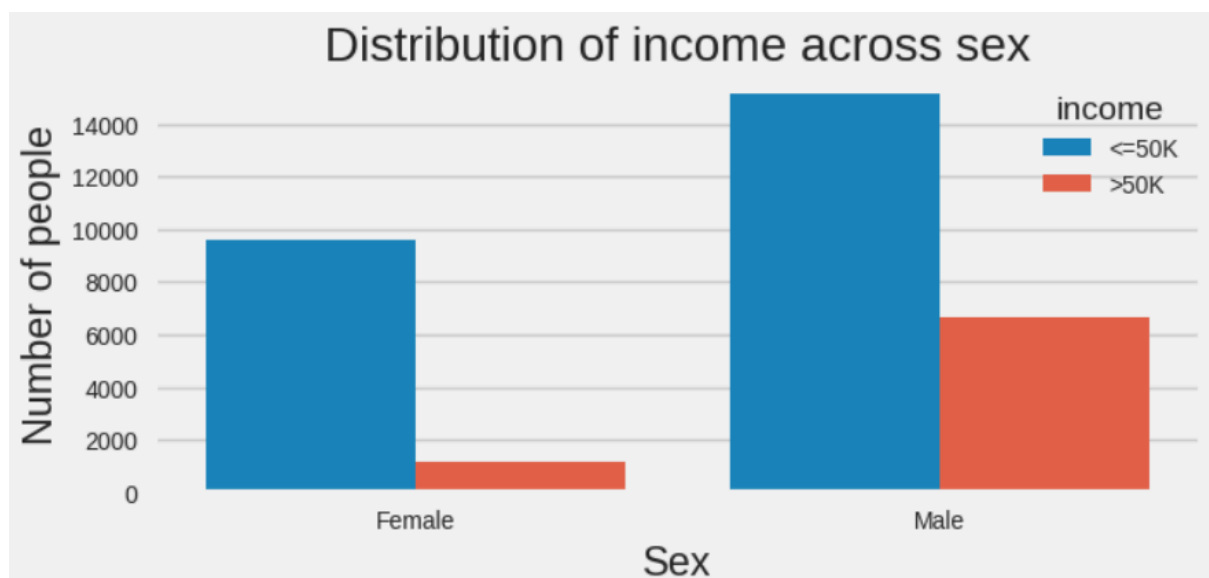
La última métrica que nos da información valiosa es la que nos muestra el ingreso de las personas según su ocupación, vemos que algunas de las ocupaciones que tienen más personas ganando >\$50000 son directores ejecutivos y profesores.



Estado civil de los encuestados



En este diagrama de barras vemos algo muy interesante, cruzando las variables de Estado Civil e Ingreso nos damos cuenta que las personas que nunca se han casado ganan significativamente menos dinero que las personas ya casadas.



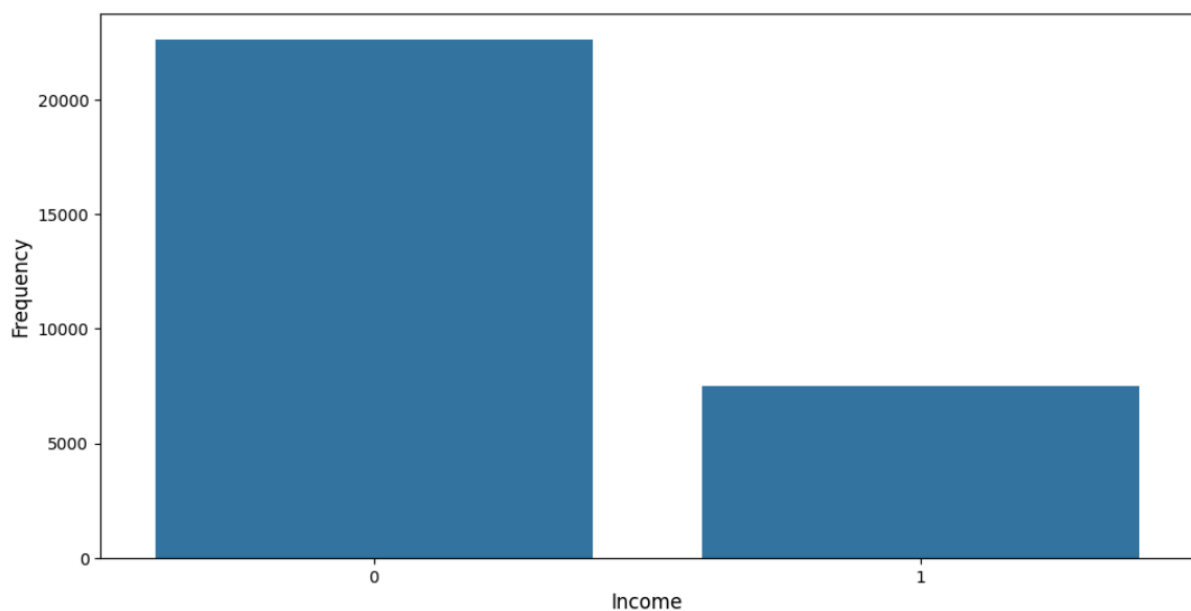
Acá volvemos a ver algo similar, es claro de observar que las mujeres aunque sean una población bastante más pequeña que la de los hombres ganan significativamente menos dinero.

## Preparación de los datos de entrenamiento

Luego de algunos ensayos erróneos se descubrió que se debía de hacer cambios al dataset antes de construir el modelo, en su estado original el dataset tenía algunos problemas que tuvieron que ser solucionados en esta etapa y que contribuyeron al producto final.

### Desbalanceo de clases

Al llegar a esta etapa de desarrollo el mayor problema que se encontró fue que las clases estaban muy desbalanceadas y esto afectaba el rendimiento del modelo, a continuación se muestra una gráfica



En esta gráfica se ve que el atributo income se encuentra muy desbalanceado (0 = Genera  $\leq$ \$50k, 1 = Genera  $>$ \$50k). Para representarlo con porcentajes el 75.9%

de la muestra estaba en el grupo 0 y esto se convirtió en una problemática, a continuación se enumeran brevemente las consecuencias del desbalance de clases.

- Sobreajuste a la clase mayoritaria.

Los modelos tienden a predecir con mayor precisión la clase mayoritaria y a ignorar la clase minoritaria, lo que puede llevar a un sobreajuste a la clase dominante y a una baja precisión en la clasificación de la clase minoritaria.

- Baja capacidad predictiva en la clase minoritaria.

Debido a que hay menos ejemplos de la clase minoritaria, los modelos pueden tener dificultades para aprender patrones significativos en esta clase.

- Desempeño sesgado hacia la clase dominante.

El desbalanceo de clases puede llevar a que el modelo tenga un desempeño sesgado hacia la clase mayoritaria, lo que puede afectar la generalización del modelo en datos no vistos.

- Dificultad para encontrar patrones en la clase minoritaria

Al tener menos ejemplos de la clase minoritaria, el modelo puede tener dificultades para identificar y generalizar patrones significativos en esta clase, lo que afecta su capacidad para realizar predicciones precisas.

- Impacto en las métricas de evaluación

Las métricas de evaluación como la precisión, el recall, la F1-score y el área bajo la curva ROC pueden ser engañosas en conjuntos de datos



desbalanceados, ya que pueden dar una falsa impresión de la efectividad del modelo.

## Oversampling

Para solucionar este problema un Data Scientist se encuentra con 2 alternativas las cuales son reducir la cantidad de muestras de la clase mayoritaria o aumentar las muestras de la clase minoritaria. Se eligió utilizar la 2 alternativa ya que el dataset es relativamente pequeño para los estándares de la industria y si se escogía reducirlo aún más podría ser contraproducente y empeorar aún más el desempeño del modelo. Se escogió utilizar la técnica de Random Oversampling, la cual se explica seguidamente:

Random Oversampling es un método de muestreo básico que se utiliza para aumentar el número de la clase minoritaria. Los puntos de datos de la clase menor se seleccionan al azar y se duplican exactamente en este método.

Se les conoce como métodos de “naive resampling” porque no asumen nada sobre los datos y no utilizan heurísticas. Esto los hace simples de implementar y rápidos de ejecutar.

Luego de aplicar el random oversampling estos fueron los resultados:

```
income
0      50.0 %
1      50.0 %
Name: proportion, dtype: object
```

## Labeling

Otro de los problemas que se encontró a la hora de crear el modelo fue que el dataset en su estado original fue el etiquetado ya que iba a ser utilizado un modelo supervisado. Luego de mucho análisis se terminó con 2 posibles alternativas para etiquetar los datos, a continuación se explican a profundidad cada uno:

### One-Hot Encode

Esta estrategia consiste en crear una columna binaria (que sólo puede contener los valores 0 o 1) para cada valor único que exista en la variable categórica que estamos codificando, y marcar con un 1 la columna correspondiente al valor presente en cada registro, dejando las demás columnas con un valor de 0. Por ejemplo, en el caso de la variable "sex" One Hot Encoding crearía dos columnas binarias (una para el valor "male" y otra para el valor "female"). Para cada persona, se asignaría un valor de 1 a la columna correspondiente a su género y un valor de 0 a la columna del género opuesto. De esta manera, cada registro queda representado por un vector binario que indica la presencia o ausencia de cada valor categórico.

### Label Encode

Label Encoding es una forma sencilla de asignar valores numéricos a las diferentes categorías de una variable categórica. Sin embargo, presenta una limitación importante, y es que estos valores numéricos pueden ser malinterpretados por algunos algoritmos de aprendizaje automático. Por ejemplo, si codificamos cuatro ciudades con los valores 0, 1, 2 y 3, es posible que un algoritmo interprete

erróneamente que, por ejemplo, la ciudad correspondiente al valor 3 tiene -según algún criterio- un valor tres veces mayor que la ciudad con el valor 1, lo cual no es cierto.

Label encode fue el escogido para realizar el proyecto ya que no aumentaba tanto la dimensión del dataset y funcionaba sin ningún problema con el modelo escogido.

### **Selección del modelo**

Igual o más importante que la calidad de los datos es la selección de los modelos, se debe de seleccionar un modelo capaz de poder trabajar con los datos de la mejor manera posible, aprovechando todo el potencial del dataset, una mala selección del modelo puede llevar al éxito o al fracaso de un proyecto ya que es la parte central del proyecto, cómo va a funcionar y como trata los datos.

Se corrieron simulaciones con 7 diferentes modelos, se seleccionó una terna final para valorar algunas características que podrían ser relevantes para el modelo y seleccionar 1 que sería el definitivo, a continuación se brinda más información de los 3 modelos con mejor rendimiento según la precisión de los mismos.

### **XGBoost Classifier**

XGBoost es un método de aprendizaje automático supervisado para clasificación y regresión. GBoost es la abreviatura de las palabras inglesas "extreme gradient boosting" (refuerzo de gradientes extremo). Este método se basa en árboles de

decisión y supone una mejora sobre otros métodos, como el bosque aleatorio y refuerzo de gradientes. Funciona bien con datasets grandes y complejos al utilizar varios métodos de optimización.

¿Cómo funciona?

- **Ensamble de Modelos Débiles:** XGBoost construye un modelo predictivo combinando las predicciones de múltiples modelos individuales, generalmente árboles de decisión, de manera iterativa. Estos modelos individuales se conocen como "modelos débiles" porque son relativamente simples y tienen una capacidad predictiva limitada por sí mismos.
- **Aumento:** XGBoost utiliza una técnica de aumento, donde cada modelo débil se entrena para corregir los errores cometidos por los modelos débiles anteriores. En otras palabras, cada modelo débil posterior se enfoca en los ejemplos que fueron clasificados incorrectamente por los modelos anteriores.
- **Optimización por Descenso de Gradiente:** XGBoost optimiza el modelo minimizando una función de pérdida específica. Utiliza la optimización por descenso de gradiente para encontrar los mejores parámetros para cada modelo débil. El algoritmo de descenso de gradiente calcula los gradientes de la función de pérdida con respecto a los parámetros del modelo y los actualiza de manera que se minimice la pérdida.
- **Regularización:** XGBoost incorpora técnicas de regularización para evitar el sobreajuste, que ocurre cuando el modelo funciona bien en los datos de

entrenamiento pero no se generaliza a nuevos datos no vistos. La regularización ayuda a controlar la complejidad del modelo y evitar una dependencia excesiva en cualquier característica o modelo débil individual.

- **Importancia de las Características:** XGBoost proporciona una medida de la importancia de las características, que indica la importancia relativa de cada característica en la realización de predicciones. Esta información puede ser útil para la selección de características y comprender los patrones subyacentes en los datos.

Rendimiento:

```
XGB Classifier:  
Accuracy score: 86.09  
F1 score: 86.57
```

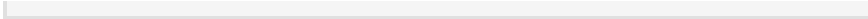
### Decision Tree Classifier

El modelo de árbol de decisión es un modelo computacional utilizado en aprendizaje automático y análisis de datos para tomar decisiones basadas en un conjunto de condiciones o características.

¿Cómo funciona?

- Estructura del Árbol: El modelo de árbol de decisión se representa como una estructura similar a un árbol, donde cada nodo interno representa una condición o característica, y cada nodo hoja representa una decisión o resultado.
- Criterios de división: El modelo determina las mejores condiciones o características para dividir los datos en cada nodo interno. Utiliza varios criterios de división, como la ganancia de información o la impureza de Gini, para medir la calidad de las divisiones y elegir las características más informativas.
- Particionamiento Recursivo: El modelo divide recursivamente los datos en función de las condiciones o características seleccionadas. Divide los datos en subconjuntos que son más homogéneos en términos de la variable objetivo o resultado.
- Decisiones en los Nodos Hoja: Una vez que los datos están particionados, el modelo asigna una decisión o resultado a cada nodo hoja en función de la clase mayoritaria o el resultado más común en ese subconjunto de datos.
- Predicción: Para hacer predicciones para nuevas instancias, el modelo sigue el camino desde el nodo raíz hasta un nodo hoja específico en función de los valores de las características. La decisión en el nodo hoja se utiliza como el resultado predicho.
- Interpretabilidad: Una de las ventajas del modelo de árbol de decisión es su interpretabilidad. El modelo se puede visualizar como un diagrama de árbol, lo que permite a los usuarios comprender el proceso de toma de decisiones y la importancia de diferentes características.

Rendimiento:



```
Decision Tree Classifier:  
Accuracy score: 91.61  
F1 score: 92.01
```

### Random Forest Classifier

El modelo Random Forest es como un "comité de expertos" formado por muchos árboles de decisión, donde cada árbol es un "experto" que da su opinión sobre una predicción. Luego, se toma la opinión de todos los "expertos" para hacer una predicción final más precisa. Este enfoque ayuda a reducir errores y a tomar decisiones más acertadas al combinar las opiniones de múltiples "expertos" (árboles de decisión) en lugar de depender solo de uno.

¿Cómo funciona?

- Creación de los árboles: Se construye un conjunto de árboles de decisión independientes entre sí. Cada árbol se entrena en un subconjunto aleatorio de los datos de entrenamiento, seleccionando características de forma aleatoria en cada división del árbol.
- Votación: Una vez que se han creado todos los árboles, se realiza una votación para determinar la predicción final. Cada árbol emite su propia predicción y la clase más frecuente se selecciona como la predicción final del Bosque Aleatorio.

- Importancia de las características: El Bosque Aleatorio también proporciona información sobre la importancia de cada característica en el proceso de predicción. Esto se calcula midiendo cuánto mejora la precisión del modelo al considerar cada característica en particular.
- Generalización: Debido a que el Bosque Aleatorio utiliza múltiples árboles, tiende a ser más resistente al sobreajuste que un solo árbol de decisión. Esto significa que puede generalizar mejor a nuevos datos y tener un rendimiento más estable.

Random Forest Classifier:  
Accuracy score: 92.6  
F1 score: 92.93

---

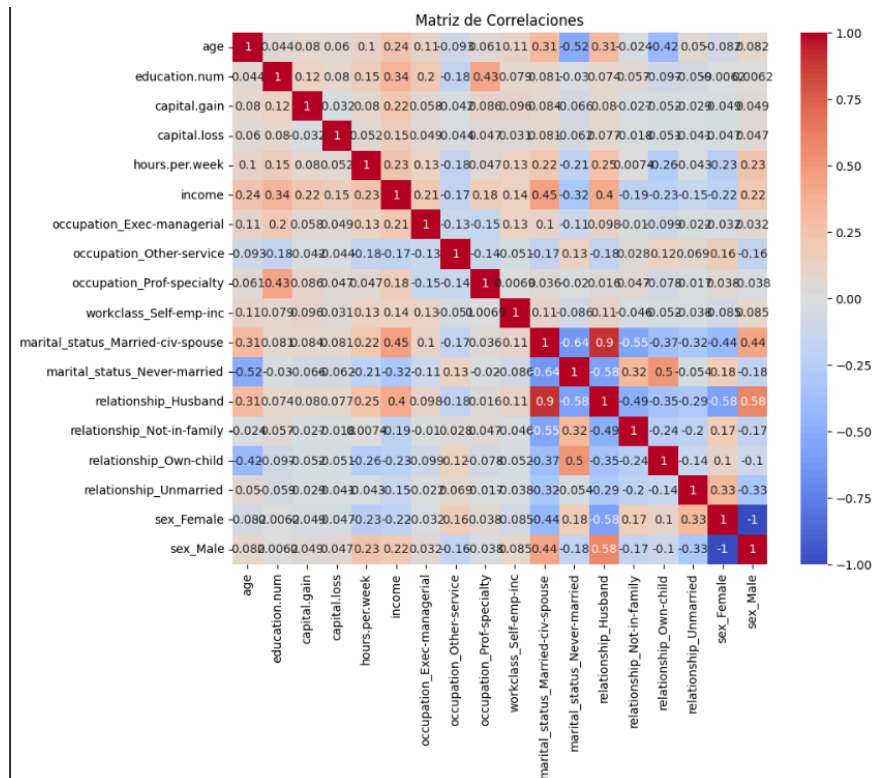
Para el desarrollo de este modelo se eligió utilizar el modelo de Random Forest ya que era el modelo que mejor resultados daba en precisión y F1 Score, el cual se abordará más adelante para comprender los alcances de este puntaje y que significa en términos de confiabilidad de los datos.

### **Correlación y causalidad de los datos**

En este segmento de la documentación se abordará el tema de la correlación y la causalidad de los datos, lo cual es un dato con mucho valor ya que nos presenta una idea de cuáles son los valores o datos que más relevancia tienen para el modelo, lo cual es vitalicio para el algoritmo de Random Forest



## Correlación de los datos



(Se utilizó el mismo dataset del modelo, simplemente se utilizó One-Hot Encoding para ver patrones más significativos)

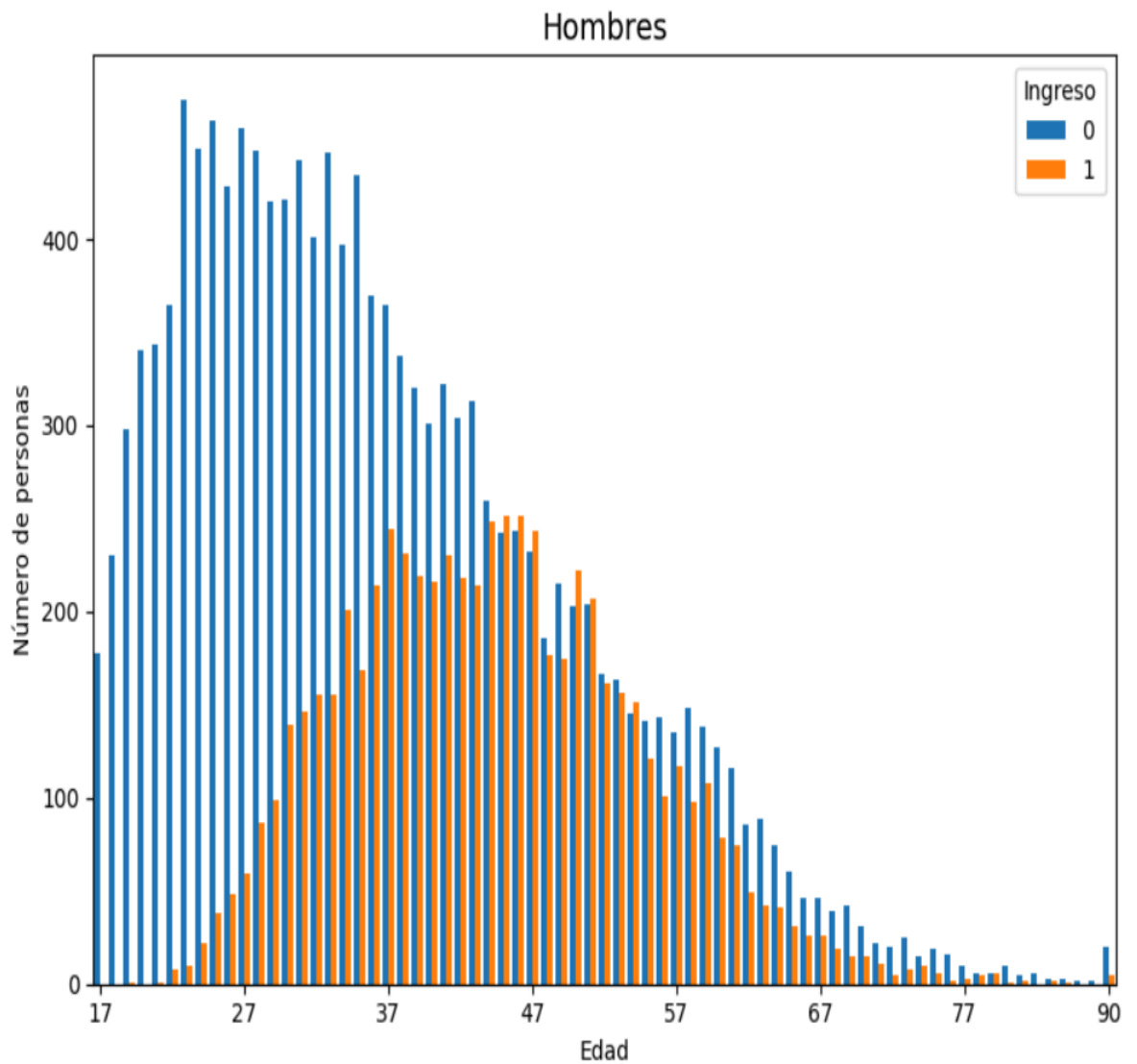
## Observaciones

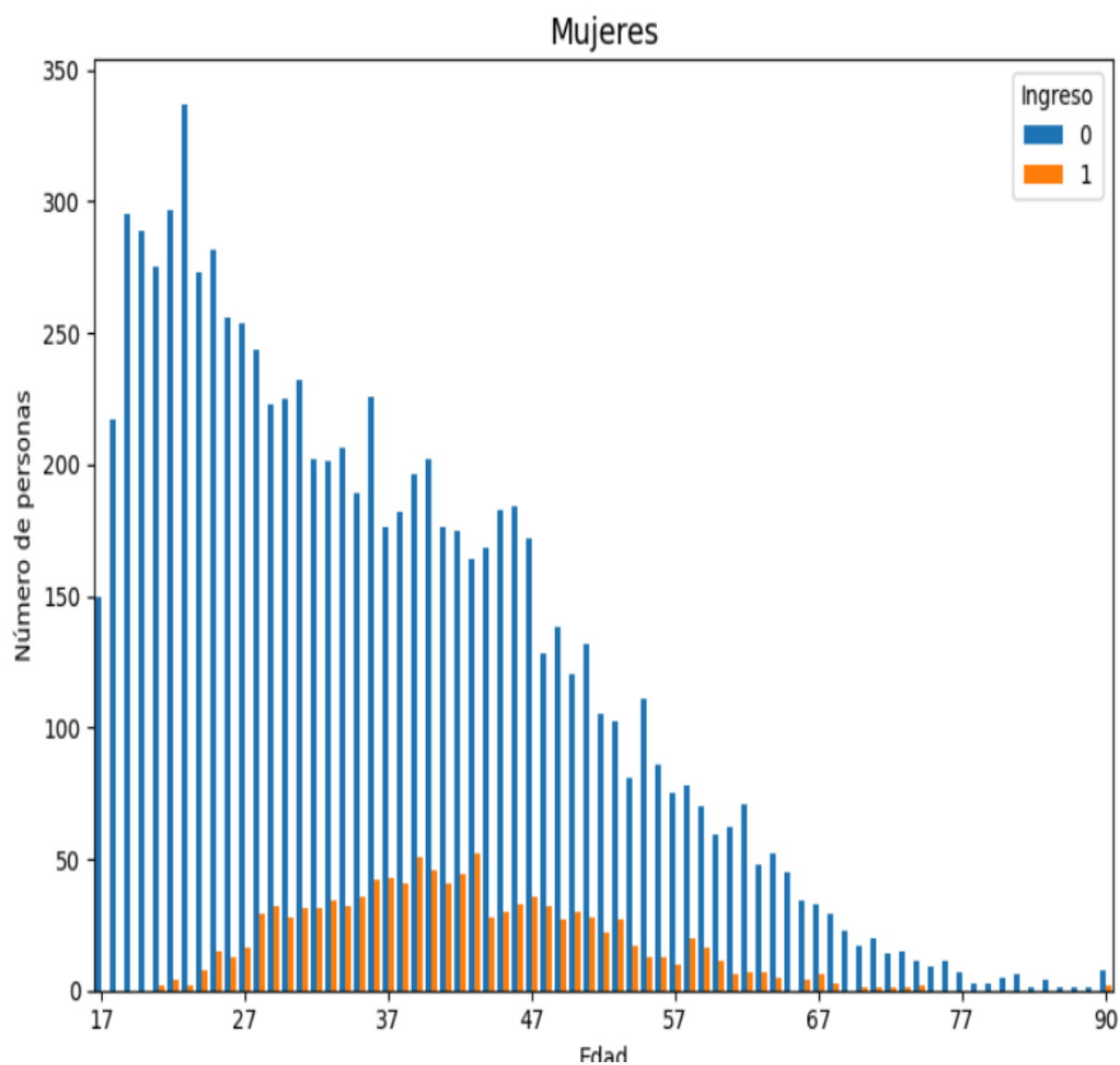
Si nos referimos únicamente a atributos que están correlacionados con la columna income podemos observar lo siguiente:

- Logramos observar que el atributo education\_num está muy fuertemente correlacionada al ingreso que una persona percibe, entre más alto sea el número mayor es el ingreso.
- Vemos que estar casado aumenta considerablemente el ingreso de una persona, con una correlación de 0.45

- Ser hombre aumenta un 22% el ingreso mientras que ser mujer lo disminuye un 22%
- Tener por lo menos un hijo tiene una correlación con el ingreso de -0.22 haciendo entender que tener un hijo reduce el ingreso de las personas.
- Logramos observar que no estar casado disminuye considerablemente el ingreso de una persona.
- El hecho de trabajar más horas por día puede hacer que una persona gane más dinero.

- Por último se denota que la edad es un punto que tiene influencia en los ingresos de una persona como habíamos visto en un gráfico anterior.





### Importancia de las características

```
age          0.217761
workclass    0.040544
education.num 0.137848
marital.status 0.102629
occupation   0.077410
relationship 0.148694
race         0.017328
sex          0.018863
capital.gain  0.087945
capital.loss  0.027272
hours.per.week 0.106569
native.country 0.017137
dtype: float64
```

(Se utilizó el mismo dataset del modelo, simplemente se utilizó One-Hot Encoding para ver patrones más significativos)

Logramos observar que aunque sean los mismos datos los valores cambian bastante, el género pierde mucha importancia, la edad pasa a ser la más importante, todas los atributos en general pierden mucha importancia.

Esto se debe a la diferencia que hay entre correlación y causalidad.

Correlación:

- La correlación mide la relación o asociación estadística entre dos variables.
- Indica la fuerza y dirección de la relación entre variables, pero no implica causalidad.
- La correlación puede ser positiva (ambas variables aumentan o disminuyen juntas), negativa (una variable aumenta mientras la otra disminuye) o cero (no hay relación).
- La correlación se mide utilizando un coeficiente de correlación, como el coeficiente de correlación de Pearson o el coeficiente de correlación de rangos de Spearman.
- La correlación es útil para identificar patrones y predecir una variable en función de otra, pero no explica la causa subyacente de la relación.

#### Causalidad:

- La causalidad se refiere a una relación de causa y efecto entre dos variables, donde los cambios en una variable influyen directamente en los cambios en la otra.
- La causalidad implica que una variable es responsable de la ocurrencia de la otra variable.
- Establecer la causalidad requiere evidencia más rigurosa, como estudios experimentales o estudios observacionales bien diseñados que controlen los factores de confusión.
- La causalidad es importante para comprender los mecanismos y las razones detrás de las relaciones observadas.

## **Puntajes del modelo**

Para calificar la precisión del modelo se utilizaron 2 medidas, accuracy score y P1 score, ambos de la librería SKlearn.

A continuación se brinda más información sobre estos 2 puntajes.

El accuracy score se calcula comparando las etiquetas predichas de una muestra con las etiquetas reales de esa muestra. En la clasificación binaria, el accuracy score es la proporción entre el número de predicciones correctas y el total de predicciones.

El F1 score se define como la media armónica de la precisión y la exhaustividad. La precisión es la proporción de verdaderos positivos sobre el total de predicciones positivas, mientras que la exhaustividad es la proporción de verdaderos positivos sobre el total de casos positivos reales.

La media armónica se utiliza en lugar de la media aritmética para el cálculo del F1 score porque penaliza más los valores bajos en comparación con la media aritmética. Esto significa que el F1 score será bajo si tanto la precisión como la exhaustividad son bajas.

El F1 score varía entre 0 y 1, donde 1 representa un modelo perfecto y 0 representa un modelo sin capacidad de clasificación.

El modelo final tiene estos puntajes:

```
Random Forest Classifier:  
Accuracy score: 91.77  
F1 score: 92.1
```

¿Por qué estos puntajes?

Estos puntajes son producto de horas de análisis y diseño de la arquitectura ideal para el tipo de datos presentados en el dataset, dado los estándares de la industria estos porcentajes son aceptables más no ideales pero también se deben de considerar aspectos como la naturaleza original del dataset, recordamos que este es bastante pequeño en comparación con datasets normales, aún utilizando oversampling las mayores falencias del modelo se deben a una limitada cantidad de datos, se considera que siguiendo los parámetros establecidos para este proyecto este es el mejor resultado que se podía obtener ya que la arquitectura y las características de los datos están previamente analizadas para lograr dar el mayor desempeño posible.

Como fue mostrado previamente en la sección de selección del modelo se puede corroborar como Random Forest tiene el mejor rendimiento.

Modelo a nivel empresarial



A continuación se presentan algunas de las implicaciones de este modelo a nivel empresarial, dejando de lado la parte técnica del algoritmo.

Utilizar un modelo de este estilo en una empresa encargada de vender cualquier tipo de servicio a 3ros puede tener estas ventajas:

- Segmentación de clientes: Con un modelo preciso, la empresa puede segmentar a sus clientes en función de su capacidad adquisitiva. Esto puede ayudar a adaptar estrategias de marketing de manera más efectiva, ofreciendo productos y servicios específicos dependiendo de la capacidad adquisitiva de la persona.
- Personalización de ofertas: Al conocer el nivel de ingresos estimado de un individuo, la empresa puede personalizar sus ofertas. Esto con el fin de que se ajusten mejor a las necesidades y presupuesto de cada cliente.
- Optimización de precios: Con la información sobre los ingresos de los clientes potenciales, la empresa puede ajustar sus estrategias de precios de manera más precisa, ofreciendo precios competitivos para cada segmento de ingresos.
- Mejora de la rentabilidad: Al dirigirse a clientes con un potencial de ingresos más alto, la empresa puede aumentar su rentabilidad ya que se pueden centrar los esfuerzos en llegar a aquellas personas que tienen más posibilidades de ser compradores.

- Reducción de costos de adquisición de clientes: Al dirigirse de manera más precisa a clientes potenciales con un mayor poder adquisitivo, la empresa puede reducir los costos asociados con la adquisición de clientes.

Alguno de los retos más importantes con los que una organización se puede enfrentar a la hora de implementar una algoritmo de esta índole son:

### **Privacidad de datos**

- Anonimización y pseudonimización: Antes de utilizar los datos en el modelo, se deben aplicar técnicas de anonimización o pseudonimización para proteger la identidad de los individuos en los datos.
- Consentimiento informado: Es fundamental obtener el consentimiento informado de los usuarios cuyos datos se utilizarán en el proyecto. Deben ser conscientes de cómo se utilizarán sus datos y dar su aprobación.
- Cumplimiento normativo: En donde sea requerido, asegurarse de cumplir con regulaciones como el Reglamento General de Protección de Datos (GDPR) en la Unión Europea o la Ley de Privacidad del Consumidor de California (CCPA) en Estados Unidos.

### **Seguridad de datos**

- **Encriptación de datos:** Los datos deben ser encriptados en reposo y en tránsito para protegerlos de accesos no autorizados.
- **Control de acceso:** Limitar el acceso a los datos solo a las personas autorizadas. Implementar políticas de control de acceso y autenticación robustas.
- **Auditoría de datos:** Registrar y monitorear las actividades relacionadas con los datos, para detectar posibles brechas de seguridad o accesos no autorizados.
- **Seguridad del modelo:** Proteger el modelo de Machine Learning de ataques adversarios, como envenenamiento de datos o ataques de inferencia.

## **Ética y responsabilidad**

- **Equidad y sesgo algorítmico:** Es fundamental garantizar que los modelos de Machine Learning sean equitativos y no discriminatorios. Es importante analizar y mitigar el sesgo algorítmico que puede surgir de los datos de entrenamiento sesgados o de características sensibles que pueden llevar a decisiones injustas.
- **Transparencia y explicabilidad:** Los modelos de Machine Learning a menudo son cajas negras difíciles de interpretar. Es esencial esforzarse por aumentar la transparencia y explicabilidad de los modelos para comprender cómo toman decisiones y poder aplicarlas a las partes interesadas y usuarios.
- **Responsabilidad y rendición de cuentas:** Los desarrolladores y propietarios de modelos de Machine Learning deben asumir la responsabilidad de las

decisiones tomadas por sus modelos. Esto incluye identificar y corregir posibles sesgos, errores o consecuencias no deseadas de los modelos.

- **Formación y sensibilización:** Es importante capacitar a los equipos de desarrollo en ética y responsabilidad en Machine Learning para fomentar una cultura de responsabilidad ética en todos los aspectos del proyecto.
- **Revisión y actualización continua:** Los modelos de Machine Learning deben ser revisados y actualizados regularmente para garantizar que sigan siendo éticos y responsables a medida que cambian las circunstancias y los datos.

## Conclusiones

1. Se investigó el dataset a fondo para poder lograr un entendimiento profundo del mismo y así poder tomar decisiones eficientes e informadas en pro del modelo y su resultado final.
2. Se explicó y realizó el proyecto siguiendo los pasos y normas sugeridas por la metodología CRISP-DM y se explicó a profundidad el por qué de esta metodología a fondo.
3. Se realizó la eliminación de datos dummy o datos incompletos los cuales son perjudiciales para la creación de métricas y para el análisis general de los datos. Asimismo se realizaron los cambios necesarios para asegurar que el manejo de los datos sea fácil y cómodo, eliminando columnas redundantes, cambiando tipos de datos y agregando data según fuera necesario.
4. Se logró crear un modelo efectivo y preciso utilizando Random Forest Classifier utilizando Random Oversampling y Label Encoding para poder mejorar el aprovechamiento de los datos disponibles.
5. Se mostró gran cantidad de información relevante acerca de los alcances y las fortalezas de este modelo, siempre tratando de usar la menor cantidad de lenguaje técnico posible, con el fin de que esta sea una documentación que cualquier persona pueda leerlo y sea parte de este proceso.

6. Se hizo hincapié en mostrar no solo las fortalezas y virtudes del algoritmo sino también los puntos débiles, siempre con la mirada en las posibles soluciones y en el manejo general del algoritmo y todas sus implicaciones como proyecto, esto con el propósito de que la cúpula gerencial esté completamente informada de todo lo que conlleva el tratamiento y el análisis de datos en una organización.

## Recomendaciones

1. **Definir los alcances del modelo:** Con cada proyecto que se vaya a implementar es muy importante antes de todo definir los alcances reales del algoritmo esto para evitar futuros inconvenientes con el mismo, se debe de comprender las fortalezas y las debilidades del mismo para hallar un rol en el que le aporte valor a la empresa.
2. **Enriquecer el dataset con datos propios:** Siempre es una alternativa la cual se puede estudiar el hecho de enriquecer el dataset con datos propios recopilados por el ERP de la organización, tomando en cuenta que estos datos deben de pasar por varias etapas de normalización y transformación para poder ser agregados al dataset original.
3. **Capacitar a los colaboradores de la empresa sobre este tema:** Una de las mejores maneras de conseguir que un cambio importante en la rutina y en la forma de trabajo sea bien recibido es enseñando y concientizando a los colaboradores que se verán involucrados en el manejo del algoritmo sobre el mismo, además puede mejorar el ambiente y la cultura empresarial el hecho de incluir a los colaboradores en charlas y capacitaciones diseñadas para lograr comprender aunque sea de una forma superficial cómo funciona este modelo.
4. **Estar al pendiente de las consideraciones éticas del modelo:** Si se llega al punto en el que se comienzan a incluir datos de clientes pasados y

actuales en el dataset de entrenamiento como forma de aumentar el rendimiento del algoritmo siempre hay que estar al corriente de cambios en directrices y leyes que rigen el tratamiento de datos sensibles de las personas.

- 5. Tener un grupo especializado de personas que trabajen en mejorar el modelo:** En Data Science es común que los modelos dejen de predecir tan efectivamente debido a muchos factores, también es común que un modelo desatendido termine cayendo en sesgos que afecte su rendimiento a futuro, además la tecnología avanza muy rápidamente, vemos cambios y mejoras cada día, la velocidad en la que los algoritmos pueden quedar desfasados o desactualizados en el mundo IT es muy alta por lo cual es indispensable crear un departamento dedicado a el análisis y el tratamiento de datos si la organización decide emprender este camino.