Predicción de los niveles de ingresos de los adultos usando Machine Learning

Juan Francisco Terán

November 2023

1 Introducción

El problema de predecir los ingresos de los adultos en función de sus atributos personales y profesionales es una tarea importante y desafiante que puede tener aplicaciones en diversos ámbitos, como la educación o el bienestar social.

En este proyecto abordaré este problema aplicando un algoritmo de aprendizaje automático supervisado llamado "Random Forest", que combina las predicciones de múltiples árboles de decisión para obtener un clasificador más preciso y exacto. Usaré una versión modificada del conjunto de datos "Adult", también conocido como "Census Income", que contiene información sobre 30,162 empleados en los Estados Unidos, como su edad, educación, clase laboral, ocupación, estado civil, ingresos y entre otros. Compararé el rendimiento de diferentes modelos de bosques aleatorios con distintos hiperparámetros y métodos de preprocesamiento y los evaluaré utilizando una matriz de confusión.

Mi objetivo es construir un modelo que pueda clasificar a cada empleado en una de dos categorías de ingresos por año: "low" (maximo USD 50.000) o "high" (mas de USD 50.000).

Mostraré cómo mi mejor modelo logra una alta puntuación de "accuracy" del 85.4% en los datos de prueba, lo cual es superior a algunos trabajos anteriores de otros autores. También discutiré acerca de las limitaciones y desafíos de mi trabajo y sugeriré una posible dirección para trabajos futuros relacionados a tareas de predicción sobre "Adult".

2 Trabajo Relacionado

Muchos autores en el mundo han escrito artículos científicos de sus trabajos sobre el conjunto de datos "Adult", también conocido como "Census Income", implementando variedad de algoritmos de aprendizaje.

- Thapa [1] compara el rendimiento de varios algoritmos de aprendizaje automático: "Random Forest", "K-Neighbors", "Logistic Regression", "Naive Bayes" y "Support Vector". El clasificador "Random Forest" obtuvo el mejor rendimiento con un "training accuracy" del 86.3% y un

86% de "test accuracy". Para ello, Thapa llegó a los siguientes hiperparámetros:

Hiperparametro	Mejor parametro	Train Accuracy	Test Accuracy
max_depth: range(3, 20, 2) n_estimators: range(1, 15) max_features: [2, 3, 5, 7]	max_depth: 11 max_features: 5 n_estimators: 14	86.3%	86%

Table 1: Hiperparámetros de Thapa

- Aplicando el algoritmo de aprendizaje automático "Random Forest", Bekena [2] obtiene un modelo con un "predictive accuracy" del 85% sobre el "test data", según él, un "accuracy" mayor en comparación al que proporcionan los clasificadores "Decision tree" y "Naive Bayes". Además, muestra que las características "marital-status", "capital-gain", "education", "age" y "hours-per-week" son las variables principales que representan una mayor proporción de "accuracy" del modelo.

3 Definición del Problema

3.1 Tarea

La tarea de predicción consiste en determinar si una persona tendrá un nivel de salario bajo "low" (maximo USD 50.000) o alto "high" (mas de USD 50.000) al año a partir de ciertos atributos y datos.

3.2 Algoritmo

Se construye un modelo de clasificación supervisado entrenado por "Bosques Aleatorios" para determinar si los ingresos de una persona superan los USD 50,000 al año a partir de información de empleados en los Estados Unidos.

Bosques aleatorios (o "Random Forest") es un algoritmo de aprendizaje conjunto que consiste en delegar la predicción final a partir de lo que dicen variedad de n árboles de decisión (o "Decision Trees") sobre un mismo conjunto de datos.

Un árbol de decisión es una estructura que divide los datos en subconjuntos basados en ciertas condiciones. Para la construcción de un árbol de decisión es importante determinar cuáles características son las más importantes del conjunto de datos, esto con el fin de ordenar la jerarquía de los nodos del árbol. Cuanto más importante es un atributo, más arriba estará en el árbol resultante, esto se hace utilizando diferentes mediciones estadísticas.

Por ejemplo, si se quiere clasificar algunas frutas según su sabor, forma y color, se puede utilizar un árbol como el siguiente:

```
Si el color es rojo, entonces
    Si la forma es redonda, entonces
        Si el sabor es dulce, entonces
            Es una manzana
        Si el sabor es ácido, entonces
            Es una fresa
    Si la forma es alargada, entonces
        Es una sandía
Si el color es amarillo, entonces
    Si la forma es redonda, entonces
        Si el sabor es dulce, entonces
            Es una banana
        Si el sabor es ácido, entonces
            Es un limón
    Si la forma es alargada, entonces
        Es una piña
```

En el ejemplo anterior, se pueden identificar diferentes elementos organizados: el nodo raíz (o atributo más importante) es el color, pues es la primera característica que se valida; las variables de forma y sabor son nodos intermedios; las hojas del árbol son las clases para etiquetar las frutas, en este caso manzana, fresa, sandía, banana y limón.

Pueden haber distintos árboles de decisión para un mismo conjunto de datos, pues la jerarquía de los nodos puede cambiar según la métrica estadística que se use para determinar las características más importantes.

Un bosque aleatorio crea varios árboles de decisión, como el anterior pero cada uno con diferentes condiciones, y los combina para obtener una mejor predicción. En palabras simples, un bosque aleatorio es un bosque de varios árboles de decisión.

Antes de entrenar cada árbol de decisión, se genera una versión distinta del conjunto de datos utilizando "bootstrap resampling". Esta técnica estadística garantiza la misma cantidad de registros del conjunto de datos original, generando muestras con reemplazo de manera aleatoria (es decir, duplicar unos registros mientras que se eliminan otros). Cada muestra se usa para un árbol de decisión distinto. Luego, cada árbol de decisión es entrenado con un mismo criterio para determinar la jerarquía de los nodos de los árboles, esto con el fin de determinar cuáles características son las más importantes del conjunto de datos, cuanto más importante es un atributo más arriba estará en el árbol resultante. Finalmente, el árbol de decisión final será aquel con el promedio o la mayoría de los resultados de las predicciones de los n árboles de decisión en el bosque creado.

Los bosques aleatorios son un método de aprendizaje conjunto ya que agregan los resultados de varios árboles de decisión, pues cada árbol de decisión construido se considera como un modelo clasificador individual. Por tanto, los bosques aleatorios suelen entrenar mejores modelos que los que haría un solo árbol de decisión.

4 Evaluación Experimental

4.1 Datos

El conjunto de datos utilizado es una muestra del conocido "Adult", pero tiene variaciones, tiene valores faltantes.
El conjunto de datos presenta valores cuantitativos (números) y cualitativos (cadenas).

Variable	**Tipo de dato**	**Tipo de atributo**
Id	int64	Cualitativo nominal
age	float64	Cuantitativo discreto
fnlwgt	float64	Cuantitativo continuo
education-num	float64	Cualitativo ordinal
capital-gain	float64	Cuantitativo continuo
capital-loss	float64	Cuantitativo continuo
hours-per-week	float64	Cuantitativo discreto
workclass	object	Cualitativo nominal
education	object	Cualitativo nominal
marital-status	object	Cualitativo nominal
occupation	object	Cualitativo nominal
relationship	object	Cualitativo nominal
race	object	Cualitativo nominal
sex	object	Cualitativo nominal
native-country	object	Cualitativo nominal
income	object	Cualitativo nominal binario

Table 2: Tipos de atributos y datos

Estas variables se refieren al contenido de una extracción de la base de datos del censo de 1994 realizado en Estados Unidos. El conjunto de datos se obtiene de la información de la población estadounidense con al menos 17 años de edad y que trabaja mínimo 1 hora por semana.

Variable	**Descripcion**
Id	Identificador del registro
age	Edad de la persona
workclass	Clase de trabajo (Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked)
fnlwgt	Peso final
education	Nivel de educacion (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	Numero de nivel de la educacion
marital-status	Estado civil (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)
occupation	Ocupacion (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	Relacion (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
race	Raza (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)
sex	Sexo (Female, Male)
capital-gain	Ganancia capital
capital-loss	Perdida de capital
hours-per-week	Horas por semana
native-country	Patria (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, Trinadad&Tobago, Peru, Hong, Holand-Netherlands)
income	Ingreso anual (low, high)

Table 3: Descripción de atributos

- NA_fullTrain.csv: Conjunto de datos de entrenamiento. Se compone de 21,113 registros y 16 variables, incluyendo la clase "income" y el identificador "Id", con valores faltantes.

  Conteo de los valores faltantes de cada atributo del conjunto de datos de entrenamiento:

Variable	**Cantidad faltante (#)**	**Porcentaje faltante (%)**
Id	0	0.000000
age	896	4.243831
workclass	952	4.509070
fnlwgt	973	4.608535
education	964	4.565907
education-num	917	4.343296
marital-status	987	4.674845

occupation	906	4.291195
relationship	923	4.371714
race	997	4.722209
sex	918	4.348032
capital-gain	912	4.319614
capital-loss	878	4.158575
hours-per-week	948	4.490125
native-country	1026	4.859565
income	0	0.000000

Table 4: Valores faltantes para "NA_fullTrain"

- NA_fullTest.csv: Conjunto de datos de prueba. Se compone de 9,049 registros y 15 variables, con valores faltantes, incluyendo el identificador "Id" y sin la etiqueta de clase "income".

  Conteo de los valores faltantes de cada atributo del conjunto de datos de prueba:

Variable	**Cantidad faltante (#)**	**Porcentaje faltante (%)**
Id	0	0.000000
age	397	4.387225
workclass	399	4.409327
fnlwgt	417	4.608244
education	427	4.718753
education-num	393	4.343021
marital-status	432	4.774008
occupation	396	4.376174
relationship	390	4.309868
race	429	4.740855
sex	405	4.475633
capital-gain	407	4.497735
capital-loss	414	4.575091
hours-per-week	397	4.387225
native-country	0	0.000000

Table 5: Valores faltantes para "NA_fullTest"

Cada característica tiene entre un 4-5% de valores faltantes. Por tanto, excluyendo el objetivo ("income") y el identificador ("Id") de cada registro, hay un total de 4-5% de datos faltantes en el conjunto de datos.
Los porcentajes de valores faltantes de todas las características están por debajo del 5% respecto a los valores de cada atributo. Este porcentaje no tendrá un impacto relevante sobre el análisis de los datos. Esto quiere decir

que la imputación es manejable y no se requieren métodos refinados para ello. Completar los datos faltantes es necesario para poder aplicar algunos algoritmos sobre los datos y tener una interpretación positiva (obtener "buenos" resultados del análisis de datos).

4.2 Metodología

CRISP-DM es una metodología (conjunto de buenas prácticas) de 6 pasos, propuestas por expertos, para la minería y análisis de datos. Esta basada en Dividir y Conquistar, es decir, dividir un problema en pequeños problemas, en este caso 6 pasos.

Utilizando la metodología CRISP-DM, realicé los siguientes pasos:

- Paso 1 y 2 (entender el negocio y los datos): Entender lo mínimo del negocio para poder comprender los datos. ¿Qué representa la información? ¿Qué significan las variables?

  En estos pasos me propuse describir el conjunto de datos. Esto incluye definir el origen de los datos, los tipos de atributos, la cantidad y porcentaje de valores faltantes que tiene cada variables y su impacto sobre el análisis y las estadísticas descriptivas de cada atributo.

  Para evidenciar la distribución de los datos, usé gráficos de barras para cada atributo cualitativo:

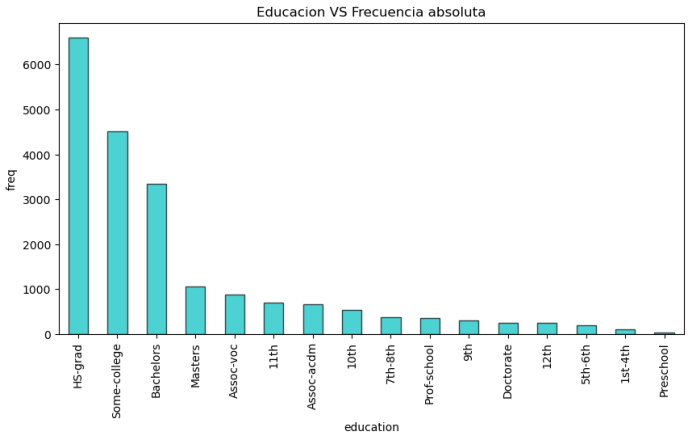

Figure 1: Gráfico de barras para "education"

  Por medio de histogramas para cada variable cuantitativa pude notar sus estadísticas descriptivas y distribución de los datos:

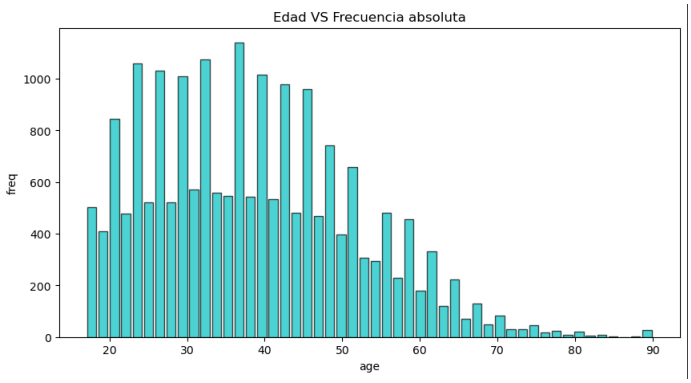

Figure 2: Histograma para "age"

A su vez, utilicé mapas de calor de los valores faltantes en el conjunto de datos con el fin de visualizar mejor la proporción de los datos que son nulos y los que no.

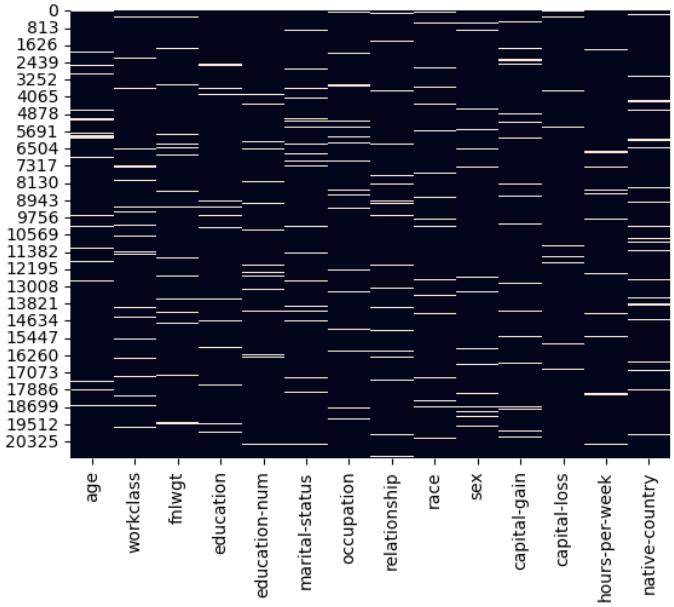

Figure 3: Mapa de calor de valores faltantes para "NA_fullTrain"

- Paso 3 (preparación de los datos): Mejorar la calidad de los datos a través de preprocesamiento.

  Para el preprocesamiento de los datos, consideré que los valores de los conjunto de datos no están limpios ni preparados para su análisis y por

tanto vi necesario aplicar algunas técnicas para mejorar su calidad.

- Imputación:
  - Existían registros con valores faltantes, por lo cual la información venía incompleta.
- Reducción de dimensionalidad
  - Existían atributos redundantes e irrelevantes, por lo cual sus datos no eran útiles para el análisis.
- Imputación por kNN (k = 230):
  - Primero reemplacé cada valor de las características cualitativas con códigos numéricos correspondientes a sus categorías originales. Esto debido a que "kNN" necesita que los datos de entrada sean cuantitativos.
  - Para aplicar esta técnica, tuve que omitir la variable cualitativa objetivo, en este caso el atributo "income".
  - Apliqué "kNN" para imputar los datos faltantes encontrando los 230 vecinos más cercanos, con todos sus valores completos. El mapa de calor de los valores faltantes en el conjunto de datos debería mostrar que no hay valores nulos, así:

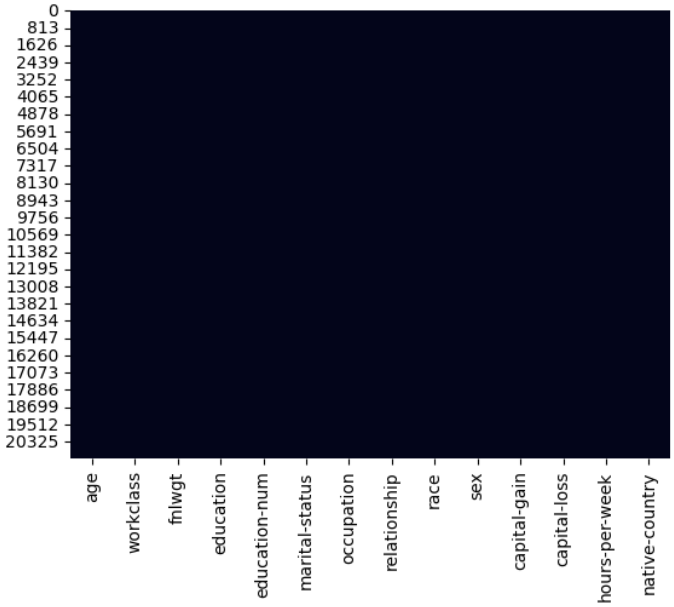

Figure 4: Mapa de calor de valores faltantes tras imputación

- Reducción de dimensionalidad (education-num, native-country):

    - Determiné que la característica "education-num" es redundante y dependiente, pues es una variable que representa la información contenida en el atributo "education".
    - Por otro lado, la variable "native-country" la consideré irrelevante ya que, por los pasos 1 y 2 de la metodología hechos anteriormente, el origen del conjunto de datos corresponde a un censo realizado a la población civil en Estados Unidos, por lo que la gran mayoría de los registros son de personas nativas de ese país (haciendo que esta característica brinde poca información y pueda ser causante de sesgos).
    - Por tanto, opino que estos datos no son útiles para el análisis, así que los reduje.

- Paso 4 (modelado): Saber el tipo de análisis que hay que realizar y modelarlos por medio de tareas de modelado de Data Mining y Machine Learning.

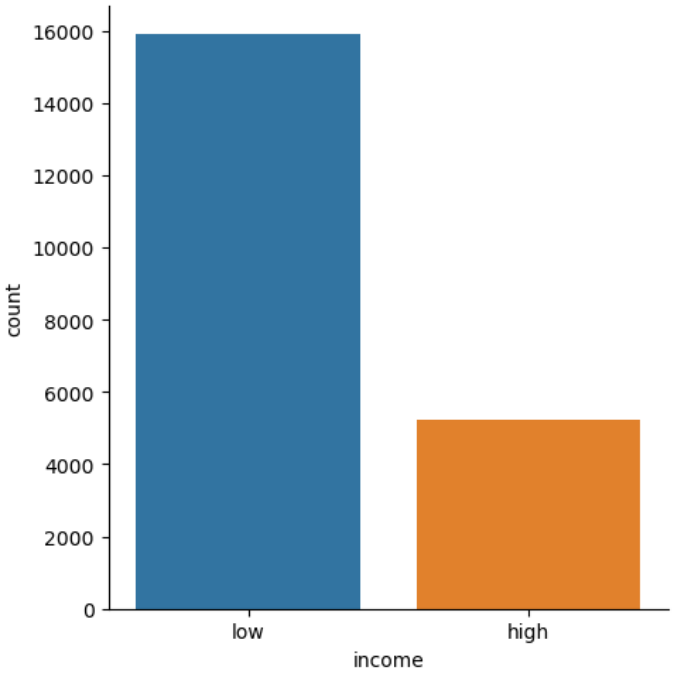

Figure 5: Distribución de "income"

Este gráfico de conteo permite visualizar que el número de empleados cuyo sueldo es bajo, es aproximadamente 3 veces la cantidad de trabajadores con sueldo alto.

Entrené un modelo de clasificación supervisado de bosques aleatorios para el conjunto de datos "Adult".

"Random Forest 1" y "Random Forest 2":

1. Dividí las características y el atributo de clase de todos los registros del conjunto de datos reducido.
2. Dividí el conjunto de datos reducido en un conjunto de entrenamiento (70%) y un conjunto de prueba (30%).
3. Construí el primer clasificador de bosque aleatorio llamado "Random Forest 1" y lo entrené.
4. Predije la respuesta para el conjunto de prueba según el modelo ajustado "Random Forest 1".
5. Construí el segundo clasificador de bosque aleatorio llamado "Random Forest 2" y lo entrené.
6. Predije la respuesta para el conjunto de prueba según el modelo ajustado "Random Forest 2".

- Paso 5 (evaluación): Evaluar que el análisis tenga sentido, sea coherente y tenga buen rendimiento.
  - Realicé la matriz de confusión de cada modelo ("Random Forest 1" y "Random Forest 2")
  - Calcule el "Accuracy Score" para cada modelo "Random Forest 1" y "Random Forest 2")
- Paso 5.1 (Reevaluación): Ajustar, entrenar y reevaluar los resultados.
- Paso 5.2 (Repreprocesamiento)
  - Discreticé sobre los atributos continuos ("age", "fnlwgt", "capital-gain", "capital-loss" y "hours-per-week") del conjunto de datos mediante el "Binning Method".
- Paso 5.3 (Remodelado)

  Entrené un modelo de clasificación supervisado de bosques aleatorios para el conjunto de datos "Adult".

  "Random Forest 3" y "Random Forest 4":

  1. Dividí las características y el atributo de clase de todos los registros del conjunto de datos reducido.
  2. Dividí el conjunto de datos reducido en un conjunto de entrenamiento (70%) y un conjunto de prueba (30%).
  3. Construí el tercer clasificador de bosque aleatorio llamado "Random Forest 3" y lo entrené.

4. Predije la respuesta para el conjunto de prueba según el modelo ajustado "Random Forest 3".
5. Construí el cuarto clasificador de bosque aleatorio llamado "Random Forest 4" y lo entrené.
6. Predije la respuesta para el conjunto de prueba según el modelo ajustado "Random Forest 4".

- Paso 5.4 (Reevaluación de los resultados)
  - Realicé la matriz de confusión de cada modelo ("Random Forest 3" y "Random Forest 4").
  - Calculé el "Accuracy Score" para los modelos "Random Forest 1" y "Random Forest 2".

- Paso 6 (Despliegue): Llevar el análisis a producción.
  - Realicé el trabajo computacional en "Jupyter Notebook".

4.3 Resultados

Nombre	n_estimators	max_depth	max_features
Random Forest 1	100	12	log2
Random Forest 2	14	11	5

Table 6: Ajustes de Hiperparámetros ("RF1" y "RF2")

Nombre	n_estimators	max_depth	max_features
Random Forest 3	100	12	log2
Random Forest 4	14	11	5

Table 7: Ajustes de Hiperparámetros ("RF3" y "RF4")

Elegí determinar cuáles atributos son más importantes utilizando "Gini" como criterio para los árboles de decisión resultantes.

Para determinar la jerarquía de los nodos de los árboles, utilicé la impureza de Gini que aporta cada una de las características del conjunto de datos. Cuanto mayor es la impureza de un atributo, se considera más importante.

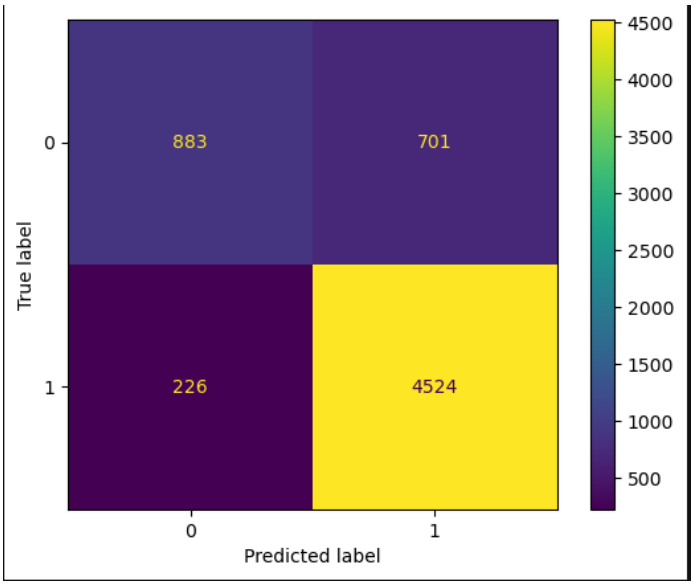

Figure 6: Matriz de Confusión para "Random Forest 1".

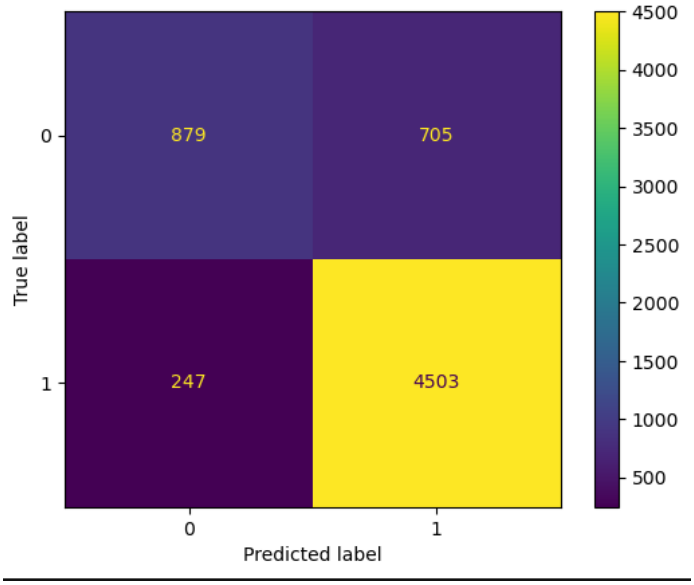

Figure 7: Matriz de Confusión para "Random Forest 2"

Teniendo en cuenta que cada clase ("low", "high") es una fila y columna de cada matriz de confusión, pude evidenciar que en el algoritmo de clasificación "Random Forest 1", la cantidad de predicciones correctas es mayor que

el número de etiquetas de clase predichas acertivamente por el modelo "Random Forest 2".

Nombre	Train Accuracy	Test Accuracy
Random Forest 1	0.8536469845279444	0.854459
Random Forest 2	0.8497000315756236	

Table 8: "Accuracy Score" para "RF1" y "RF2"

Se puede evidenciar que hay un mejor rendimiento en el modelo de aprendizaje "Random Forest 1".

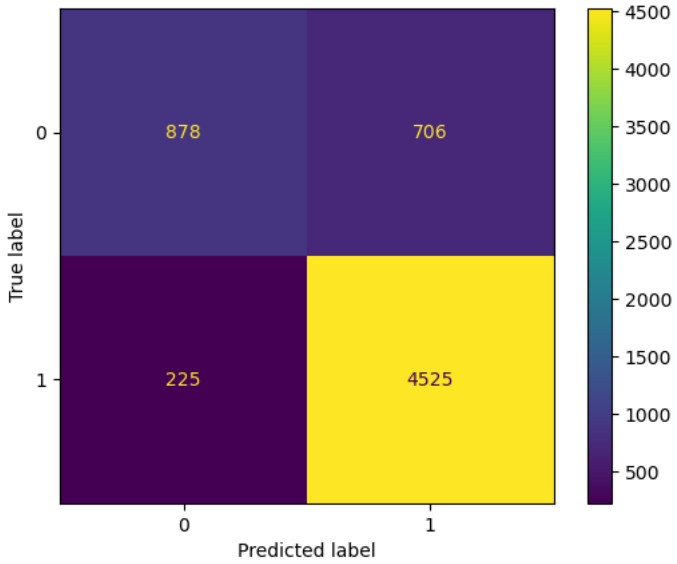

Figure 8: Matriz de Confusión para "Random Forest 3"

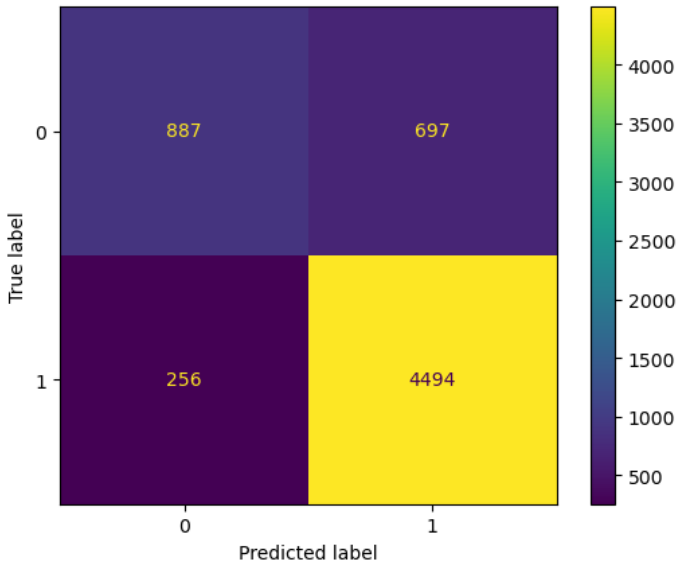

Figure 9: Matriz de Confusión para "Random Forest 4"

Pude evidenciar que en el algoritmo de clasificación "Random Forest 3", la cantidad de predicciones correctas es mayor que el número de etiquetas de clase predichas acertivamente por el modelo "Random Forest 4".

Por medio del "Binning Method", se usaron los siguientes valores para algunos de sus parámetros:

Variable	n_bins	enconde	strategy
age	60	ordinal	uniform
fnlwgt	18	ordinal	uniform
capital-gain	1000	ordinal	uniform
capital-loss	89	ordinal	uniform
hours-per-week	40	ordinal	uniform

Table 9: Parámetros de "Binning"

Nombre	Accuracy
Random Forest 3	0.853015472055573
Random Forest 4	0.8495421534575308

Table 10: "Accuracy Score" para "RF3" y "RF4"

Se puede evidenciar que el modelo llamado "Random Forest 3" predijo correctamente una mayor cantidad de clasificaciones, respecto al número total de ocurrencias correctas predichas por el clasificador entrenado "Random Forest 4". Por tanto, "Random Forest 3" tiene mejor rendimiento.

Transformar los valores cuantitativos en cualitativos para utilizar atributos discretos en vez de continuos permitió reducir los datos, no obstante, esto no quiere decir que los modelos tuvieran mayor rendimiento al ser evaluados. Es posible que aplicando otros métodos de discretización, como "1R", o distintos modelos de clasificación, como "Naive-Bayes", y experimentando con los diferentes parámetros para las técnicas y hiperparámetros para los algoritmos de aprendizaje, se puedan encontrar mejores resultados para el rendimiento de un modelo más adaptado.

4.4 Discusión

Pude encontrar un modelo con un rendimiento mejor, 85.4%, que el autor Bekena [2], el cual también implementó un algoritmo de aprendizaje "Random Forest", pero solo llegó al 85%. Sin embargo, no pude lograr superar el rendimiento mencionado por Thapa [1], del 86.3%, ni siquiera imitando la combinación de hiperparámetros que utilizó para entrenar su modelo.

Quiero resaltar 2 aspectos relevantes:

1. El hecho de que, en el preprocesamiento, el valor

   $$\text{k} = 230$$

   fue el que mejor me funcionó. Este fue encontrado experimentalmente. Se hicieron muchas otras pruebas variando k hasta hallar uno que impulsara un "accuracy" de, al menos, 85% para el modelo.

2. El hecho de que, en la construcción del modelo, la combinación de los hiperparámetros

   $$\text{n\_estimators} = 100$$

   $$\text{max\_depth} = 12$$

   $$\text{max\_features} = \text{"log2"}$$

   fue hallado haciendo variedad de pruebas experimentales. Similar a como se obtuvo k para "kNN", en este caso los mejores parámetros fueron aquellos que más acercaron el "accuracy" al 86%.

Gracias a que "Random Forest 1" fue el algoritmo de aprendizaje que mejor se ajustó a las tuplas de entrada del conjunto de datos y a su conjunto de prueba, esta técnica debería estar mejor entrenada para relacionar nuevas características con su respectiva clase.

Por tanto, "Random Forest 1" es el método de clasificación más aceptable que construí para el conjunto de datos "Adult.data", con un

$$\text{Accuracy} = 0.8536469845279444.$$

No obstante, esto no quiere decir que no se puede ampliar este puntaje de calificación. Es posible que explorando los distintos parámetros que ofrece la técnica "kNN", aplicando otros modelos de clasificación como "Naive-Bayes", o buscando entre los diferentes hiperparámetros de distintos algoritmos de clasificación (incluyendo los del mismo "Random Forest"), se puedan encontrar mejores resultados.

Tras reevaluar los modelos "Random Forest 3" y "Random Forest 4" realizando los cambios sobre el conjunto de datos imputado y reducido, se puede concluir que el conjunto de datos preprocesado discretizado con la técnica "Binning method" no fue mejor entrenado que el primero "Random Forest 1".

5 Conclusiones

- Logré obtener un modelo con mejor rendimiento que Bekena [2]. Esto puede ser debido a la gran variedad de combinaciones de hiperparámetros que exploré, las cuales me terminaron llevando a tener mejores resultados.

- No pude implementar un modelo con un rendimiento como el que referencia Thapa [1]. Esto pudo deberse a que, a pesar de que experimenté bastante con los hiperparámetros de los "Random Forest", ambos realizamos distintas técnicas en el preprocesamiento de los datos, siendo evidente que lo más probable es que los métodos utilizados por él causen una mayor calidad de los datos y, por tanto, mejores resultados en el análisis.

6 Bibliografía

[1] Thapa, S. (2023). Adult income prediction using various ML algorithms. Social Science Research Network. https://doi.org/10.2139/ssrn.4325813

[2] Bekena, S. (2017). Using decision Tree classifier to predict income levels Munich Personal REPEC Archive. (s. f.). https://mpra.ub.uni-muenchen.de/id/eprint/83406

7 Anexos

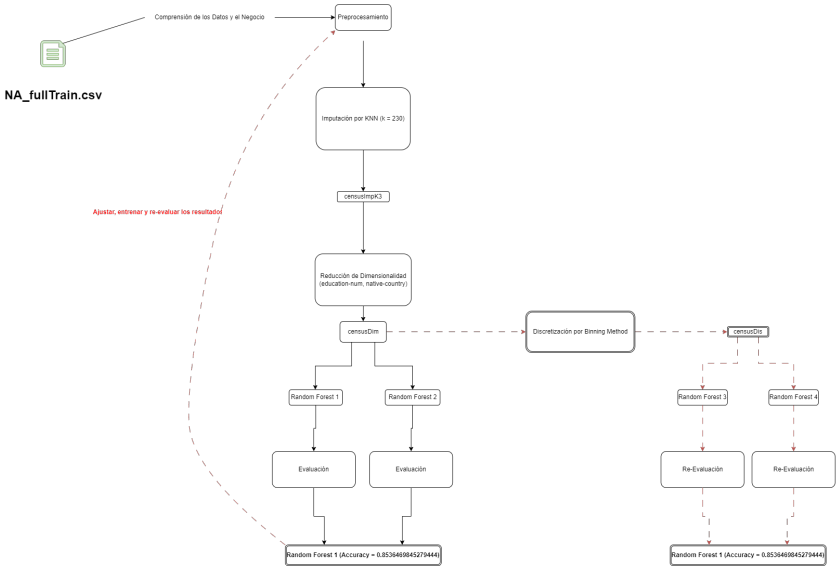

Figure 10: Flujo de trabajo general del proyecto