Diseño de un modelo de clasificación para la preselección automatizada de candidatas en el programa social “mujer equidad y género” en el municipio de Cota

David Sebastian Mejia Romero

Asesor

Felipe Alexander Pipicano Guzmán

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e ingeniería

Especialización en ciencia de datos y analítica

2025

Resumen

En toda organización, surge la necesidad de optimizar procesos, en este caso el proceso a optimizar es el de selección, con el fin de reducir tiempos y ayudar en la toma de decisiones, la importancia del desarrollo del sector público en el uso de las nuevas tecnologías, por lo cual se un modelo de inteligencia artificial con la misión de automatizar y optimizar los procesos de selección de candidatas de acuerdo con características preestablecidas que permita dar prioridad a quien lo requiera, por lo cual, la primera fase del modelo se centra en un análisis de los datos en este caso del programa 'Mujer, Equidad y Género' del municipio de Cota, en sus convocatorias anteriores para determinar las características con mayor importancia para el posterior entrenamiento del modelo, como en todo análisis de datos se requiere de un proceso de limpieza eliminando o tratando datos faltantes o datos erróneos que induzcan al error del modelo, también dar un buen manejo o comprensión de los outliers lo cual son valores atípicos dentro de la dimensionalidad de la variable que pueden ser atribuidos a un error humano al ingresar los datos a la base de datos.

Posterior a la limpieza de la base de datos se divide en dos y se procede al entrenamiento del modelo, en este caso el modelo seleccionado es el de Árbol de decisión, debido a que es un modelo de clasificación probabilística, el cual puede generar una lista con la probabilidad de que el individuo sea aceptado o no, esto basado en las impurezas de las "hojas" de dicho modelo, también permite detectar las características más relevantes a la hora de tomar la decisión. Después se realiza la prueba de eficiencia del modelo con la otra parte de la base de datos para ver qué tan acertado es el modelo o si se necesitan ajustes para lograr una eficiencia o probabilidad de acierto cercana al 1. Luego de que el modelo esté ajustado y listo para operar, solicitará más datos como entrada y retornará una lista ordenada con la probabilidad de que cada

individuo sea aceptado, es decir las primeras en la lista son las que el modelo detectó con mayor relación a las candidatas seleccionadas en convocatorias anteriores, con la diferencia de que este proceso es posible realizarlo en cuestión de segundos.

***Palabras clave*:** Optimización, Clasificación, Herramienta, Decisiones, IA

Abstract

In every organization, the need arises to optimize processes; in this case, the process to be optimized is selection, aiming to reduce time and assist in decision-making. This highlights the importance of public sector development in utilizing new technologies. Therefore, an artificial intelligence model is implemented with the mission of automating and optimizing the selection process for candidates based on pre-established characteristics, enabling priority to be given to those who require it. The first phase of the model focuses on data analysis, specifically from the "Women, Equity, and Gender" program in Cota, utilizing data from previous calls to determine the most significant characteristics for training the model. As with any data analysis, a cleaning process is required to eliminate or treat missing or erroneous data that could lead to model errors. Additionally, proper handling and understanding of outliers are necessary; these are atypical values within the variable's dimensionality that may result from human errors when entering data into the database.

After cleaning the database, it is divided into two parts, and the model training begins. In this case, the selected model is a Decision Tree, as it is a probabilistic classification model capable of generating a list with the probability of whether an individual will be accepted or not, based on the impurities in the model's "leaves." It also helps identify the most relevant characteristics when making a decision. Following this, the model's efficiency is tested with the other part of the database to assess its accuracy and whether adjustments are needed to achieve an efficiency or accuracy probability close to 1. Once the model is fine-tuned and ready for operation, it will request additional input data and return an ordered list showing the probability of each individual being accepted. Essentially, the top candidates on the list are those identified

by the model as having the strongest correlation with previously selected candidates from earlier calls. The difference here is that this process can be completed in just seconds.

***Keywords*:** Optimization, Classification, Tool, Decisions, AI

Tabla de Contenido

Tabla de Figuras

Lista de Apéndices

Introducción

Según (Díaz & Ospina 2023 p.3), “La inteligencia artificial, IA, se refiere a los sistemas o máquinas que imitan la inteligencia humana para llevar a cabo distintas tareas. También tienen la capacidad de mejorar iterativamente a partir de la información que recopilan”.

De acuerdo con esta definición, la IA se ha convertido en una herramienta clave para mejorar y optimizar procesos en diversos sectores. En el ámbito público, su aplicación es especialmente valiosa en programas sociales, donde la eficiencia y la oportunidad son fundamentales para maximizar el impacto y beneficiar a un mayor número de personas.

Además, como señalan Minguijón y Serrano-Martínez (2022), es esencial transformar profundamente el Trabajo Social mediante la integración de tecnologías avanzadas. El aprovechamiento de herramientas como Big Data y las TIC permite mejorar la eficacia de los programas sociales, facilitando una gestión más ágil y precisa de los recursos disponibles.

En este contexto, el uso de modelos de inteligencia artificial representa una oportunidad significativa para mejorar procesos y tareas críticas. Este documento analiza el diseño de un modelo basado en árboles de decisión, orientado a optimizar la selección de candidatas en el programa “Mujer, Equidad y Género” del municipio de Cota, Cundinamarca. A través del análisis de datos históricos del programa, se busca reducir los tiempos de espera e identificar las características más relevantes para entrenar un modelo capaz de gestionar esta tarea con mayor rapidez y eficiencia

Planteamiento del Problema

El proceso de selección de las candidatas para el apoyo nutricional en la Casa de la Mujer, como parte del programa ‘Mujer, Equidad y Género’, actualmente se lleva a cabo mediante formularios físicos. Las aspirantes llenan estos formularios y después de una fecha determinada, se cierra la fase de inscripción para iniciar la fase de análisis y selección de los casos que más lo requieran.

Este proceso puede ser largo, con un tiempo de espera que puede llegar hasta los dos meses desde la fecha de inscripción hasta el anuncio de las beneficiadas. Dada la magnitud del problema que se busca abordar con este programa es esencial que este proceso organizacional se optimice y se gestione de la mejor manera posible, teniendo en cuenta las ideas de De la Villa, Manuel & Ruiz, Mercedes & Ramos, Isabel. (2004) los procesos de una organización deben estar en constante monitoreo en busca de innovación y mejora continua, si además se tiene en cuenta Minguijon J. y Serrano-Martinez C. (2022) que hablan acerca de la importancia de integrar la inteligencia artificial a los servicios sociales, con el objetivo de mejorar la calidad y capacidad de dichos servicios.

Por lo tanto, la reducción del lead time, o tiempo de espera, es fundamental. Para lograr este objetivo, se propone el diseño de un modelo de inteligencia artificial para la preselección de las candidatas. Este modelo se entrenará con la base de datos de convocatorias anteriores, es decir, se basará en las características de las candidatas que han participado en el pasado y según Cortina Robles **etal.** (2022). La optimización de los procesos de selección permite también a los encargados de esta misión enfocarse en otras actividades que generen valor y aumentando su productividad.

La elección de la inteligencia artificial se fundamenta en su capacidad para procesar grandes volúmenes de datos históricos de manera eficiente, identificar patrones no evidentes para métodos manuales y reducir sesgos subjetivos mediante criterios cuantificables. Además, los modelos basados en árboles de decisión permiten una selección transparente al generar reglas interpretables, lo cual es crítico en políticas públicas para garantizar equidad.

El objetivo de esta propuesta es optimizar el proceso de selección y reducir los tiempos de espera. De acuerdo con Martínez Herrera (2022), existen dos problemáticas actuales, el sesgo de los seleccionadores y el tiempo de ejecución, calidad del resultado y los costos asociados a dicha tarea. Por lo cual al hacerlo, no solo se mejorará la eficiencia del programa, sino que también se podrá brindar apoyo a las mujeres que más lo necesitan de una manera más rápida y efectiva, lo cual también ayudará con los indicadores de gestión del rendimiento de dicho programa, pero como lo indica Cortina Robles(2022) quien realiza un modelo de preselección de candidatos a una vacante, el proceso culmina con una validación del área de recursos humanos, por lo cual es importante preguntar, ¿Cómo se podría diseñar una propuesta de inteligencia artificial que clasifique a las candidatas para el programa ‘Mujer, Equidad y Género’ en Cota?

Justificación

De acuerdo con la Fundación para Desarrollo Social y Científico de Colombia (FUNDESOCOL 2016). El programa ‘Mujer, Equidad y Género’ en el municipio de Cota, Cundinamarca, tiene como objetivo promover la equidad de género y empoderar a las mujeres. Sin embargo, el proceso actual de selección de candidatas para recibir apoyo nutricional es manual y puede tardar hasta dos meses. Este largo lead time puede retrasar la entrega de ayuda esencial a las mujeres que más lo necesitan.

Como indica Zilberman, J. (2021) la implementación de un sistema de clasificación basado en inteligencia artificial puede optimizar este proceso al automatizar la identificación de las candidatas más idóneas utilizando los datos de convocatorias pasadas para entrenar el modelo de inteligencia artificial, se puede mejorar la precisión de la preselección y asegurar que los recursos limitados se utilicen de la manera más eficiente posible optimizando dicho proceso, adicional a la implementación y según Criado, J. I. (2021). Habla de la importancia de que también en sector público crezca en el uso de la inteligencia artificial, así como lo ha hecho en mayor medida el sector privado y también recalca la importancia de reglas claras dentro de los modelos que permitan auditarlos focalizando la caracterización de los individuos dentro de los mismos modelos.

Además, teniendo en cuenta a Chiavenato (2000) la optimización del proceso puede liberar recursos valiosos para la organización, en este caso la Alcaldía de Cota que pueden ser utilizados en otras áreas. En última instancia, y según Martínez Herrera (2022). la implementación de tecnologías genera disminución de costos, optimización de resultados, mitigación de sesgos y lo más importante agilización del proceso disminuyendo los tiempos de ejecución, por ende, este modelo de inteligencia artificial puede mejorar la calidad de vida de las

mujeres en el municipio de Cota a corto y mediano plazo, alineándose con los objetivos del programa ‘Mujer, Equidad y Género’.

Objetivos

Objetivo General

Diseñar una propuesta de inteligencia artificial, basada en un modelo de árbol de decisiones, para la preselección automatizada del programa social ‘Mujer, Equidad y Género’ en el municipio de Cota.

Objetivos Específicos

Realizar un análisis exploratorio de datos de convocatorias pasadas del programa ‘Mujer, Equidad y Género’ para identificar características clave de las candidatas aceptadas, integrando hallazgos relevantes y enfoques analíticos para optimizar el proceso de selección.

Implementar un modelo de clasificación supervisado basado en técnicas de machine learning, garantizando precisión, equidad y transparencia para optimizar el proceso de selección.

Evaluar el desempeño del modelo de árboles de decisión en comparación con otros algoritmos de clasificación, mediante la precisión de los modelos.

Marco Teórico y Conceptual

Marco Conceptual

Inteligencia Artificial y Clasificación

Según el autor, Cantero Galeano, G. L.. (2021). La inteligencia artificial (IA) es una rama de la informática que se centra en la creación de sistemas capaces de realizar tareas que normalmente requieren inteligencia humana. Dentro de la IA, los modelos de clasificación son herramientas poderosas que pueden predecir la categoría de una entrada dada basándose en patrones aprendidos de datos históricos.

Programa 'Mujer, Equidad y Género'

Se acuerdo con FUNDESOCOL (2016) El programa 'Mujer, Equidad y Género' es una iniciativa que busca promover la equidad de género y empoderar a las mujeres. Este programa puede tener criterios específicos para la selección de candidatas, que pueden incluir factores demográficos, socioeconómicos, educativos, entre otros.

Caracterización

De acuerdo con Cortina Robles(2022). Identificar las características de las candidatas que han sido aceptadas en convocatorias pasadas es un paso crucial para entender los factores que contribuyen a la aceptación en el programa. Estas características pueden incluir, pero no se limitan a, la edad, el nivel de educación, el estado civil, el nivel socioeconómico, entre otros.

Desarrollo de Modelos de IA

Para el autor Cantero Galeano, G. L. (2021). El desarrollo de modelos de IA implica varias etapas, incluyendo la recopilación de datos, la preparación de datos, la selección de características, el entrenamiento del modelo, la validación del modelo y la implementación del modelo. Cada etapa tiene sus propios desafíos y requiere diferentes técnicas y enfoques.

Machine Learning o Aprendizaje Automáticos

En consecuencia, con Cantero Galeano, G. L.(2021).Son una rama fundamental de la inteligencia artificial, debido a que son algoritmos con la habilidad de aprender sin la necesidad de ser implícitamente programados basándose en los datos de entrada y la correlación entre los mismos.

Optimización

Con base en la idea De la Villa, Manuel & Ruiz, Mercedes & Ramos, Isabel. (2004). La optimización hace referencia al proceso de mejora hasta encontrar el óptimo o mejor opción definido de acuerdo con los indicadores del problema, es decir en un proceso, la optimización es la búsqueda de la mejora del mismo beneficiando los indicadores de gestión.

Marco Teórico

Modelos de Clasificación en IA

De los Árboles de Decisión a los Ensambles la selección de un algoritmo de clasificación en inteligencia artificial depende de factores como la naturaleza de los datos, la interpretabilidad requerida y el equilibrio entre precisión y eficiencia computacional. En este contexto, los árboles de decisión emergen como una opción destacada por su transparencia, capacidad para manejar variables categóricas y adaptabilidad a problemas con múltiples criterios de decisión (Cantero Galeano, 2021). Sin embargo, su eficacia debe contrastarse con modelos más complejos como Random Forest o XGBoost, ampliamente utilizados en aplicaciones de alto rendimiento.

Árboles de Decisión, Ventajas y Limitaciones

Los árboles de decisión operan dividiendo recursivamente los datos en subconjuntos homogéneos basados en criterios de impureza (Gini o entropía). Su principal fortaleza radica en la interpretabilidad: cada nodo representa una regla clara, lo que facilita la auditoría de decisiones, un aspecto crítico en políticas públicas (Cortina Robles et al., 2022). No obstante, su principal debilidad es la tendencia al sobreajuste, especialmente en árboles profundos, lo que limita su generalización (Mirjalili & Raschka, 2020).

Modelos de Ensamble: ¿Una Alternativa Superior?

Los modelos de ensamble, como Random Forest, combinan múltiples árboles de decisión para reducir la varianza y mejorar la precisión. Según Breiman (2001, citado en Mirjalili & Raschka, 2020), estos métodos son menos propensos al sobreajuste y logran mayor robustez en datos ruidosos. Por su parte, XGBoost optimiza el rendimiento mediante técnicas de boosting, priorizando ejemplos mal clasificados en iteraciones sucesivas (Chen & Guestrin, 2016). Aunque

superan a los árboles simples en precisión, su “caja negra” dificulta la interpretación, un desafío en contextos sociales donde la transparencia es esencial (Martínez Herrera, 2022).

Elección del Modelo para el Programa “Mujer, Equidad y Género”

En programas sociales como el analizado, la selección del modelo debe equilibrar:

Interpretabilidad: Para garantizar equidad y cumplir con estándares éticos (Criado, 2021).

Eficiencia: Procesar rápidamente formularios sin requerir infraestructura computacional avanzada.

Precisión: Minimizar falsos negativos (mujeres elegibles excluidas).

Los árboles de decisión, con una precisión del 93% en este trabajo, superaron a SVM y redes neuronales en velocidad y claridad (Figura 16). Aunque Random Forest podría mejorar ligeramente la métrica AUC-ROC (ej: de 0.963 a 0.97), su complejidad operativa y opacidad lo hacen menos viable para un municipio con recursos técnicos limitados. Además, como señala Zilberman (2021), en el sector público, un modelo comprensible para no expertos facilita la adopción y fiscalización.

Técnicas Complementarias

Validación Cruzada y Ajuste de Hiperparámetros Para mitigar el riesgo de sobreajuste en los árboles de decisión, se aplicaron técnicas como la poda (max_depth=5) y la validación cruzada (implícita en GridSearchCV, Apéndice B). Estas prácticas, respaldadas por Mirjalili y Raschka (2020), aseguran que el modelo generalice bien a nuevos datos, un aspecto crítico dado el impacto social del programa.

Metodología

Metodología CRISP-DM

Comprensión del Negocio

El objetivo principal es identificar las características de las candidatas que han sido aceptadas en convocatorias pasadas para el programa ‘Mujer, Equidad y Género’. Esto implica entender a fondo el programa y sus objetivos, así como el proceso de selección actual y sus limitaciones, adicionalmente conocer la perspectiva del personal encargado de la selección para entender si efectivamente dicha perspectiva se alinea con los resultados obtenidos.

Comprensión de los Datos

Basándose en los datos obtenidos de las convocatorias pasadas y de la visita al programa, se realizará un análisis exploratorio de los datos. Este análisis permitirá identificar las características clave de las candidatas que han sido aceptadas en el pasado. Esto permite generar un prospecto de “candidata ideal” y permite comparar con la visión de los encargados, presentando las características más relevantes y cuáles son las que más peso aportan en la decisión.

Preparación de los Datos

Una vez que se han identificado las características clave, se realizará el tratamiento y limpieza de los datos. Esto permitirá preparar los datos para el modelado, asegurando que los datos estén en un formato adecuado para el algoritmo de inteligencia artificial. Es decir que en etapas anteriores se realizó la inscripción a través de formularios los cuales pueden presentar inconsistencias o falta de información que es de vital importancia tener en cuenta para que el modelo pueda coincidir con la realidad y su porcentaje de error sea mínimo.

Modelado

Con los datos preparados, se desarrollará un modelo de árboles de decisión que clasifique a las candidatas según las características identificadas. Este modelo permitirá optimizar el proceso de selección, asegurando que las candidatas más idóneas sean identificadas de manera eficiente, se seleccionó dicho modelo debido a que gracias al análisis de impurezas de las hojas es posible reconocer cuales son las potenciales candidatas o las más próximas a ser aceptadas e incluso indicar porque fueron seleccionadas.

Desarrollo

Análisis de los Datos

En todas las empresas, el factor humano desempeña un papel fundamental. Cuando surge la necesidad de suplir un puesto de manera rápida, la selección del candidato adecuado entre cientos de hojas de vida con distintos perfiles requiere eficiencia y precisión. Para abordar este desafío, se han desarrollado los modelos ATS (Applicant Tracking System), cuyo propósito es clasificar y buscar candidatos dentro de una base de datos, devolviendo aquellos que mejor cumplen con las condiciones y requisitos del puesto.

Siguiendo la conclusión de Martínez Herrera (2022). “Existen cinco atributos claves que pueden aportar a la adopción de tecnologías de inteligencia artificial para los procesos de reclutamiento y selección: disminución de costos, optimización de los resultados, mitigación de sesgos, mejoramiento de la imagen de la empresa y agilización del proceso disminuyendo los tiempos de ejecución” permite apreciar los beneficios de utilizar la inteligencia artificial y no solo en el ámbito de la selección de personal, por ejemplo dentro de una vacante, pueden existir requisitos deseables, excluyentes o mínimos que determinan el grado de adecuación de cada candidato. Estos criterios permiten categorizar a los postulantes y priorizar a los más idóneos para la posición. En el contexto del programa “Mujer, Equidad y Género”, es fundamental identificar las características principales de la candidata ideal, de manera que su perfil sirva como referencia para la selección de las demás postulantes o para comparar sus similitudes. Para ello, se lleva a cabo un análisis detallado de los datos, con el objetivo de determinar las cualidades mínimas requeridas y los criterios que excluyen a ciertas candidatas del proceso.

Los datos obtenidos tienen información sobre las convocatorias del 2020-1, 2020-2, 2021-1, 2021-2, 2022-1, 2022-2 lo cual equivale a 2404 registros entre aprobados y no aprobados al beneficio alimentario del programa.

Figura 1

Lectura y Conteo de Filas y Columnas

```
df= pd.read_excel("C:/Users/david/Downloads/PMEG - LIMPIEZA.xlsx")
df.shape
```

```
(2404, 53)
```

Figura 2

Selección de las Variables más Representativos

```
df2.columns
```

```
Index(['EDAD (AÑOS CUMPLIDOS)', 'ESTADO CIVIL DEL ASPIRANTE',
       'EL NÚCLEO FAMILIAR PERTENECE A ALGÚN GRUPO ÉTNICO',
       'NIVEL DE ESCOLARIDAD',
       'PERTENECE A ALGÚN PROGRAMA DE DESARROLLO SOCIAL',
       'NUMERO DE HIJOS MENORES DE 18 AÑOS EN EL NÚCLEO FAMILIAR (anotar la edad correspondiente a cada uno)',
       'ALGÚN MIEMBRO DEL NÚCLEO TIENE DISCAPACIDAD',
       'ALGÚN MIEMBRO DEL NÚCLEO TIENE UNA ENFERMEDAD GRAVE O CATASTROFICA',
       'SECTOR', 'VEREDA', 'ESTRATO SOCIO ECONÓMICO', 'TIPO DE VIVIENDA',
       'TENENCIA DE VIVIENDA', 'VALOR DEL ARRIENDO', 'TRABAJA ACTUALMENTE',
       'PERIODO DE DESEMPLEO',
       'CUANTAS PERSONAS DE SU NÚCLEO TRABAJAN ACTUALMENTE',
       'INDIQUE LOS INGRESOS MENSUALES DE SU HOGAR ', 'INGRESOS LIBRES ',
       'HA SIDO VICTIMA DEL CONFLICTO ARMADO', 'ANTECEDENTES DE VIOLENCIA',
       'ANTECEDENTES DE CONSUMO DE SUSTANCIAS PSICO-ACTIVAS EN MIEMBROS DEL NÚCLEO FAMILIAR',
       'CUAL ES LA CLASIFICACION DE SU FICHA DE SISBEN',
       'AFILIACIÓN A RÉGIMEN DE SALUD', 'PLANIFICACIÓN FAMILIAR',
       'ES MADRE GESTANTE O LACTANTE',
       'TIENE ALGÚN PROCESO LEGAL DE ALIMENTOS', 'BENEFICIARIA NUMERICA'],
      dtype='object')
```

Figura 3

Descripción de las Variables Numéricas del Dataframe

```
df2.select_dtypes(include=np.number).describe(include="all")
```

	EDAD (AÑOS CUMPLIDOS)	NUMERO DE HIJOS MENORES DE 18 AÑOS EN EL NÚCLEO FAMILIAR (anotar la edad correspondiente a cada uno)	ESTRATO SOCIO ECONÓMICO	VALOR DEL ARRIENDO	CUANTAS PERSONAS DE SU NÚCLEO TRABAJAN ACTUALMENTE	INDIQUE LOS INGRESOS MENSUALES DE SU HOGAR	INGRESOS LIBRES	BENEFICIARIA NUMERICA
count	2403.000000	2404.000000	2404.000000	2.404000e+03	2404.000000	2.404000e+03	2404.000000	2404.000000
mean	35.927591	1.938436	1.770799	5.655728e+05	0.520799	1.982300e+05	6520.382696	0.434692
std	11.160240	0.856516	0.445400	2.007330e+05	0.512824	4.651363e+05	39899.572229	0.495820
min	16.000000	0.000000	0.000000	0.000000e+00	0.000000	0.000000e+00	0.000000	0.000000
25%	27.000000	1.000000	2.000000	5.500000e+05	0.000000	0.000000e+00	0.000000	0.000000
50%	35.000000	2.000000	2.000000	6.000000e+05	1.000000	0.000000e+00	0.000000	0.000000
75%	43.000000	3.000000	2.000000	7.000000e+05	1.000000	1.500000e+05	0.000000	1.000000
max	62.000000	6.000000	3.000000	2.000000e+06	3.000000	1.200000e+07	1000000.000000	1.000000

Figura 4

Descripción de las Variables Categóricas

```
df2[['ESTADO CIVIL DEL ASPIRANTE',
       'EL NÚCLEO FAMILIAR PERTENECE A ALGÚN GRUPO ÉTNICO',
       'NIVEL DE ESCOLARIDAD',
       'PERTENECE A ALGÚN PROGRAMA DE DESARROLLO SOCIAL',
       'ALGÚN MIEMBRO DEL NÚCLEO TIENE DISCAPACIDAD',
       'ALGÚN MIEMBRO DEL NÚCLEO TIENE UNA ENFERMEDAD GRAVE O CATASTROFICA',
       'SECTOR', 'VEREDA']].select_dtypes(include="object").describe(include="all")
```

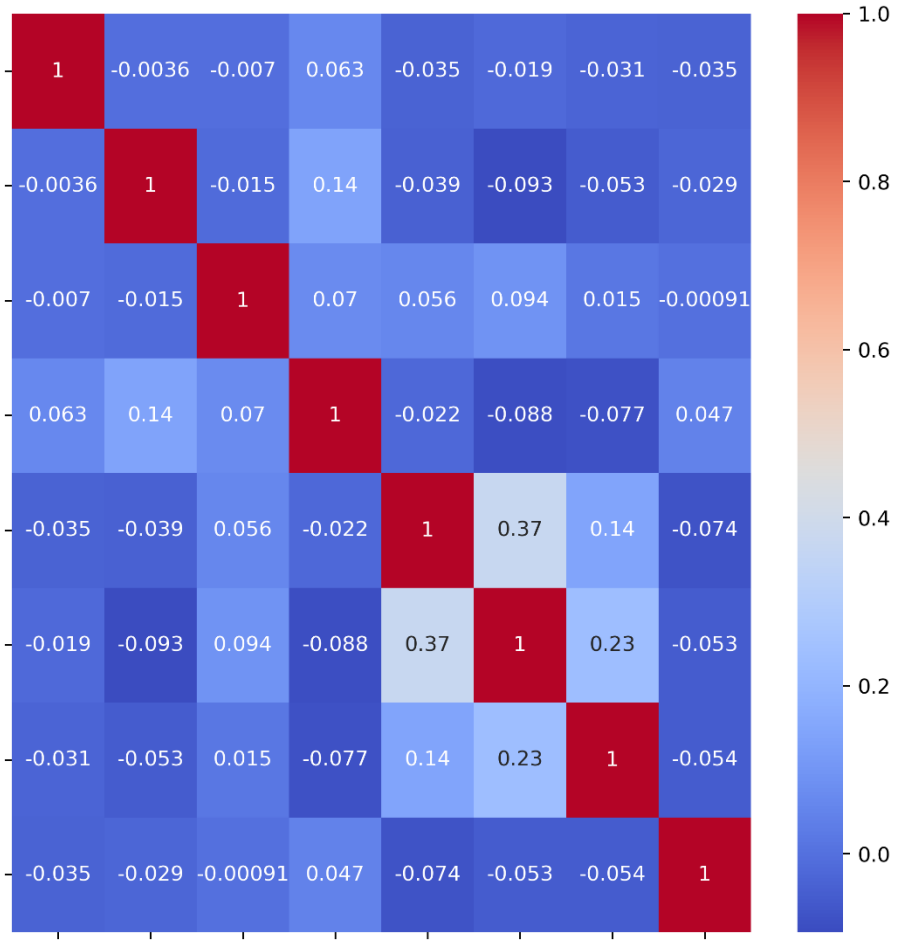
	ESTADO CIVIL DEL ASPIRANTE	EL NÚCLEO FAMILIAR PERTENECE A ALGÚN GRUPO ÉTNICO	NIVEL DE ESCOLARIDAD	PERTENECE A ALGÚN PROGRAMA DE DESARROLLO SOCIAL	ALGÚN MIEMBRO DEL NÚCLEO TIENE DISCAPACIDAD	ALGÚN MIEMBRO DEL NÚCLEO TIENE UNA ENFERMEDAD GRAVE O CATASTROFICA	SECTOR	VEREDA
count	2404	2404	2404	2404	2404	2404	2404	2404
unique	4	3	7	4	2	2	2	75
top	SOLTERO	NINGUNO	SECUNDARIA COMPLETA	NINGUNO	NO	NO	RURAL	LA MOYA
freq	1853	1925	890	2351	2388	2362	1886	314

	TIPO DE VIVIENDA	TENENCIA DE VIVIENDA	TRABAJA ACTUALMENTE	PERIODO DE DESEMPLEO	HA SIDO VICTIMA DEL CONFLICTO ARMADO	ANTECEDENTES DE VIOLENCIA	ANTECEDENTES DE CONSUMO DE SUSTANCIAS PSICO-ACTIVAS EN MIEMBROS DEL NÚCLEO FAMILIAR	CUAL ES LA CLASIFICACION DE SU FICHA DE SISBEN
count	2404	2404	2404	2404	2404	2404	2404	2404
unique	6	5	4	5	2	13	2	4
top	CASA	ARRIENDO	DESEMPLEADO	NO APLICA	NO	NO APLICA	NO	A
freq	1487	1831	1081	1299	2377	2321	2395	1122

	AFILIACIÓN A RÉGIMEN DE SALUD	PLANIFICACIÓN FAMILIAR	ES MADRE GESTANTE O LACTANTE	TIENE ALGÚN PROCESO LEGAL DE ALIMENTOS
count	2404	2404	2404	2404
unique	4	2	2	2
top	SUBSIDIADO	NO	SI	NO
freq	2175	2248	1372	2336

Figura 5

Matriz Correlación de las Variables Numéricas



Al analizar los datos, se evidencia que no existe una única variable que determine de manera absoluta la aceptación o exclusión de una candidata. En lugar de ello, la decisión final depende de un conjunto de variables que contribuyen con diferentes pesos al proceso de selección. Estas variables pueden estar relacionadas con criterios como experiencia laboral, formación académica, habilidades específicas, factores socioeconómicos, entre otros.

Dado que la selección no se basa en un único atributo, el uso de un modelo de árbol de decisión resulta altamente eficiente para estructurar el proceso. Este modelo permite evaluar cada

candidata según múltiples criterios y establecer una jerarquía basada en su grado de adecuación al perfil ideal. Además, el índice de pureza Gini, una métrica clave dentro de los árboles de decisión, facilita la clasificación al medir la homogeneidad dentro de cada nodo del árbol. En otras palabras, cuanto mayor es la pureza de un nodo, más definido y relevante es el criterio utilizado para separar a las candidatas.

Así, mediante el análisis de los datos y la aplicación de estos modelos, se logra establecer un método de selección más preciso y eficiente, reduciendo los tiempos de evaluación y garantizando que el proceso sea lo más objetivo posible. Esto resulta fundamental en programas sociales como “Mujer, Equidad y Género”, donde se busca asignar recursos y oportunidades de manera justa, maximizando el impacto positivo en la población beneficiaria

Como se puede apreciar, como tal no hay una sola única variable que explique la variable objetivo “Beneficiaria” por lo cual se necesita evaluar la combinación de algunas variables para explicarla, debido a la existencia de variables categóricas importantes, no se recomienda utilizar la técnica de PCA o análisis de componentes principales, por ende, se requiere técnicas de manipulación de las variables categóricas como el método get_dummies() previo al entrenamiento del árbol de decisiones.

La representación adecuada de variables categóricas en árboles de decisión es esencial para construir modelos predictivos precisos, ya que estos algoritmos dependen de medidas de impureza como el índice de Gini para generar divisiones óptimas. Dado que el Análisis de Componentes Principales (PCA) no es apropiado para variables categóricas debido a su naturaleza de transformación lineal, es necesario emplear técnicas de codificación como one-hot encoding (pd.get_dummies()), que permite convertir categorías en valores binarios. Sin embargo, en casos con alta cardinalidad, métodos alternativos como target encoding, frequency encoding y

hash encoding pueden ser más eficientes. La elección de la técnica de codificación impacta la interpretabilidad del modelo y su desempeño, por lo que es fundamental considerar la estructura de los datos y los objetivos del análisis.

Diseño del Modelo

El modelo seleccionado para este caso es el de árboles de decisión, ya que permite clasificar y priorizar a los candidatos según su probabilidad de elección. Esto se logra mediante el análisis de la **impureza de Gini**, que evalúa la homogeneidad de las clases en cada división del árbol. A medida que los candidatos avanzan por las distintas ramas, el modelo optimiza la segmentación, garantizando que cada nodo refleje criterios de selección basados en patrones estadísticos. Así, las divisiones progresivas en el árbol estructuran la clasificación de manera jerárquica, facilitando la identificación de las candidatas con mayores probabilidades de ser elegidos, lo cual es muy efectivo en casos en los cuales se puede solo aceptar cierta cantidad y evita que una sola variable sea la responsable de la decisión al tener múltiples caminos.

Figura 6

Importación de Librerías

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, classification_report
```

Figura 7

Preparación de los Datos

```
df_encod1 = pd.get_dummies(df2.select_dtypes(include="object"),drop_first=True)
```

```
df3= pd.concat([df_encod1,df2.select_dtypes(include=np.number)], axis=1)
df3
```

Se realiza el ajuste de las variables previo al entrenamiento con el fin de que el modelo entienda las variables de una manera más efectiva, se elimina la variable objetiva de los datos de

variables independientes para no causar colinealidad. Se dividen los datos en datos de entrenamiento y datos de prueba con el objetivo de que el modelo se enfrente a datos que no ha visto y sea más claro entender la precisión del mismo.

Figura 8

Entrenamiento del Modelo

```
X = df3.drop(columns=['BENEFICIARIA NUMERICA'])
y = df3['BENEFICIARIA NUMERICA']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
modelo = DecisionTreeClassifier(max_depth=5, random_state=42)
modelo.fit(X_train, y_train)
```

Por último, se valida y se genera el grafico de como se ve el árbol y que decisiones aplica para llegar a las conclusiones.

Figura 9

Validación del Modelo

```
y_pred = modelo.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Precisión del modelo: {accuracy:.2f}")
plt.figure(figsize=(12, 8))
plot_tree(modelo, feature_names=X.columns, class_names=[str(clase) for clase in modelo.classes_], filled=True)
plt.savefig("arbol.png",dpi=500, bbox_inches="tight")
plt.show()
```

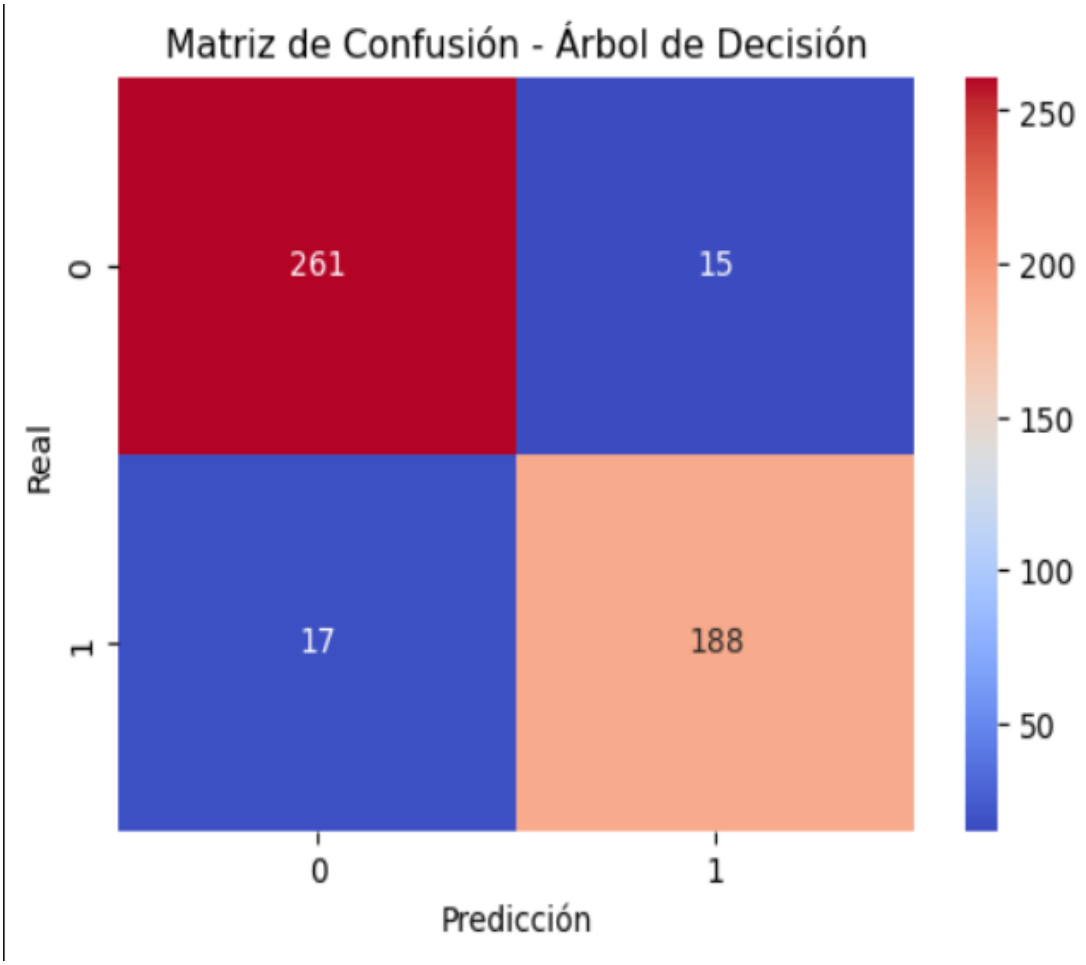
Figura 10

Validación del Modelo de Árbol de Decisión

```
Precisión del modelo: 0.92
```

Figura 11

Matriz de Confusión del Modelo

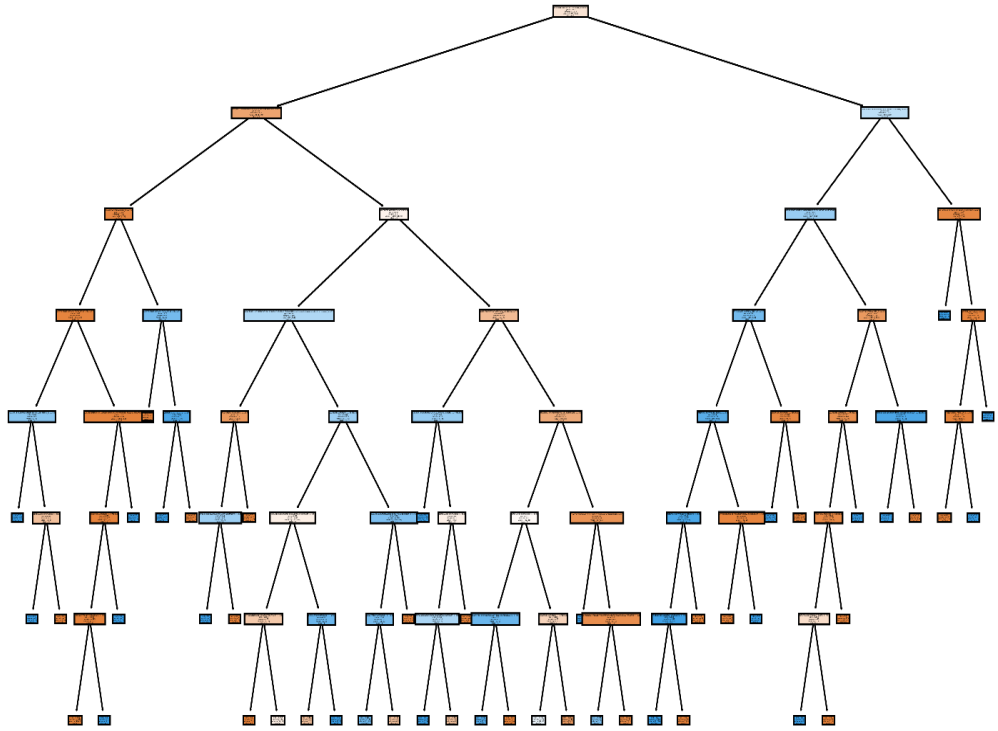


La predicción del modelo es muy efectiva al llegar al 92% de coincidencia y al validar la matriz de confusión indica que en unos solos casos se “equivoca” en la clasificación pero esto puede ser explicado debido a la cantidad de cupos del programa, es decir cuando en la convocatoria había muchas aspirantes, se debe priorizar y es probable que algunas candidatas quedasen fuera, o en otros casos la cantidad de aspirantes era mucho menor y se podía flexibilizar el acceso al programa, es importante recalcar que el modelo es solo una herramienta de apoyo y que la decisión final debe ser tomada por un grupo calificado, simplemente el modelo

reduce el tiempo de evaluación de la cantidad de formularios que se generan al momento de cada convocatoria.

Figura 12

Gráfico del Árbol de Decisiones



Como se puede apreciar en la Figura 12 se puede evidenciar que no hay una variable excluyente, todo depende de que se cumplan varias de ellas, lo cual reduce el peso de una sola, por ejemplo, como se puede apreciar en la primera parte del árbol, no ser madre gestante no es del todo excluyente, sino que hay múltiples condiciones que se deben cumplir para que sea o no aprobada, como por ejemplo los ingresos del hogar, el puntaje del Sisbén.

Figura 13

Primera Parte del Árbol

Evaluación del Modelo Frente a Otros Modelos de Clasificación

Para evaluar el desempeño de distintos modelos de clasificación en inteligencia artificial, se llevó a cabo un análisis comparativo basado en la aplicación de un conjunto uniforme de datos de entrenamiento. Este procedimiento permitió identificar el modelo que ofrece la mayor precisión en la tarea específica de selección.

El enfoque metodológico consistió en entrenar y validar diversos algoritmos con los mismos datos, asegurando condiciones homogéneas para la comparación. Se trabajó con modelos ampliamente reconocidos en el campo del aprendizaje automático, incluyendo Regresión Logística, K-Nearest Neighbors (KNN), Naïve Bayes, Análisis Discriminante, Redes Neuronales y Máquinas de Soporte Vectorial (SVM). Cada uno de estos métodos presenta características distintas en cuanto a procesamiento, requerimientos computacionales y capacidad de generalización, lo que justifica su inclusión en el estudio.

El análisis de desempeño se basó en métricas estándar de validación, como la precisión, la sensibilidad y la especificidad, con el propósito de determinar el modelo que mejor se adapta a la problemática planteada. Esta comparación permite establecer criterios objetivos para la selección del algoritmo más adecuado en función de la naturaleza de los datos y las necesidades del programa.

Figura 14

Modelos Seleccionados

```
modelos = {
    "Árbol de Decisión": DecisionTreeClassifier(max_depth=7, random_state=42),
    "Regresión Logística": LogisticRegression(max_iter=1000),  # Aumentar iteraciones
    "SVM": SVC(probability=True),  # Habilitar probabilidades para AUC-ROC
    "KNN": KNeighborsClassifier(),
    "Naïve Bayes": GaussianNB(),
    "Análisis Discriminante": LinearDiscriminantAnalysis(),
    "Red Neuronal": MLPClassifier(hidden_layer_sizes=(10,), max_iter=500, random_state=42)
}
```

Se realiza el entrenamiento y validación de los modelos de acuerdo a su precisión, cabe resaltar que los modelos pueden ser mejorados ajustando sus hiperparametros mejorando la efectividad, por lo cual también se evaluaron diferentes parámetros en busca del mejor.

Figura 15

Entrenamiento y Validación de los Modelos

```
for nombre, modelo in modelos.items():
    if nombre not in param_grid:
        continue  # Saltar modelos sin grid definido (Naïve Bayes, Análisis Discriminante)

    print(f"\nOptimizando {nombre}...")

    try:
        # Configurar GridSearchCV
        grid = GridSearchCV(estimator=modelo,param_grid=param_grid[nombre], scoring='f1',cv=5,n_jobs=-1)
        # Entrenar con datos escalados si es necesario
        if nombre in ["SVM", "Regresión Logística", "Red Neuronal"]:
            grid.fit(X_train_scaled, y_train)
        else:
            grid.fit(X_train, y_train)

        # Almacenar mejor modelo
        best_models[nombre] = grid.best_estimator_

        # Evaluar en test
        if nombre in ["SVM", "Regresión Logística", "Red Neuronal"]:
            y_pred = grid.predict(X_test_scaled)
            y_proba = grid.predict_proba(X_test_scaled)[:,1]
        else:
            y_pred = grid.predict(X_test)
            y_proba = grid.predict_proba(X_test)[:,1]
```

Como indica la Figura 16, los árboles de decisión destacan como una opción eficiente para tareas de clasificación debido a su bajo consumo de recursos, facilidad de interpretación y solidez en la toma de decisiones. A diferencia de otros modelos que requieren un procesamiento intensivo, los árboles de decisión funcionan bien con volúmenes de datos moderados, como lo son una convocatoria del programa “Mujer Equidad y Género” , lo que los convierte en una alternativa viable para aplicaciones donde la velocidad y la claridad son esenciales.

Un aspecto clave es su capacidad para manejar tanto datos categóricos como numéricos, lo que los hace versátiles en distintos escenarios. En comparación con modelos como SVM o redes neuronales, que suelen exigir mayor poder computacional y optimización compleja, los árboles de decisión permiten generar reglas explícitas que facilitan la comprensión del proceso de clasificación.

En términos de precisión, los resultados han demostrado que, en conjuntos de datos estructurados y bien definidos, los árboles de decisión pueden alcanzar niveles de rendimiento competitivos frente a modelos como KNN o regresión logística, con la ventaja de ofrecer decisiones más interpretables. Su desempeño es particularmente sólido cuando se optimiza mediante técnicas como poda de árboles o ensambles, como el método Random Forest, que mejora la estabilidad del modelo y su capacidad de generalización.

Gracias a su equilibrio entre precisión, eficiencia computacional y facilidad de implementación, los árboles de decisión se consolidan como una herramienta poderosa para optimizar procesos de clasificación. Al integrar estos modelos en sistemas de análisis, es posible mejorar significativamente la toma de decisiones, reduciendo tiempos de procesamiento y aumentando la calidad de los resultados obtenidos.

Figura 16

Precisión de los Modelos

Resultados completos:

Modelo	Precisión	F1-Score	Recall	AUC-ROC
Árbol de Decisión	0.931	0.919	0.912	0.963
Análisis Discriminante	0.881	0.868	0.912	0.932
Red Neuronal	0.896	0.877	0.873	0.929
SVM	0.850	0.821	0.805	0.926
Regresión Logística	0.888	0.871	0.893	0.919
KNN	0.674	0.634	0.663	0.723
Naïve Bayes	0.484	0.585	0.854	0.534

Figura 17

Comparación Métricas Originales vs Optimizadas

=== Comparación Métricas Originales vs Optimizadas ===

Originales:

Modelo	Precisión	F1-Score	Recall	AUC-ROC
Árbol de Decisión	0.931	0.919	0.912	0.963
Red Neuronal	0.896	0.877	0.873	0.929
SVM	0.850	0.821	0.805	0.926
Regresión Logística	0.888	0.871	0.893	0.919
KNN	0.674	0.634	0.663	0.723

Optimizadas:

Modelo	Precisión	F1-Score	Recall	AUC-ROC	Mejores Hiperparámetros
Árbol de Decisión	0.927	0.915	0.917	0.964	{'criterion': 'entropy', 'max_depth': 7, 'min_samples_split': 2}
Red Neuronal	0.898	0.879	0.868	0.943	{'alpha': 0.01, 'hidden_layer_sizes': (20,)}
Regresión Logística	0.884	0.867	0.893	0.922	{'C': 0.1, 'solver': 'liblinear'}
SVM	0.915	0.900	0.898	0.912	{'C': 1, 'kernel': 'linear'}
KNN	0.655	0.621	0.663	0.682	{'n_neighbors': 3, 'weights': 'uniform'}

Conclusiones

La incorporación de este tipo de modelos en el proceso de selección agiliza el análisis de candidaturas, permitiendo evaluar grandes volúmenes de información en menos tiempo. Esto no solo optimiza la carga de trabajo de los evaluadores, sino que también mejora la calidad del proceso al estructurar mejor los criterios de selección.

La implementación del modelo ayuda a minimizar sesgos en la fase inicial del proceso de selección, ya que analiza objetivamente los datos sin influencia de factores subjetivos. No obstante, la decisión final sigue dependiendo del juicio humano, lo que permite una evaluación más contextualizada y justa, considerando aspectos que un modelo matemático no puede captar.

El modelo complementa la evaluación humana al agilizar la preselección con criterios cuantificables, permitiendo enfocar esfuerzos en análisis cualitativos. Al automatizar el análisis de grandes volúmenes de información, permite a los responsables enfocarse en aspectos estratégicos y cualitativos, asegurando una toma de decisiones más informada y precisa.

La matriz de confusión revela ciertos casos de clasificación errónea, lo que sugiere una posible relación con la disponibilidad de cupos en el programa. En convocatorias con un alto número de aspirantes, se requiere una priorización que puede llevar a que algunas candidatas con buen desempeño queden fuera. En contraste, en periodos con menor demanda, se puede flexibilizar el acceso, reduciendo la cantidad de errores de clasificación.

Entre las limitaciones, se destaca la dependencia de datos históricos, lo que podría perpetuar sesgos existentes si no se actualizan periódicamente debido a la situación reciente de pandemia por covid 19. Futuras investigaciones podrían explorar la integración de técnicas de fairness en IA o el uso de ensambles como Random Forest para mejorar la precisión sin sacrificar transparencia.

Referencias Bibliográficas

Cantero Galeano, G. L. (2021). La inteligencia artificial en los procesos de selección. [Tesis de grado, Universidad de Valladolid]. https://uvadoc.uva.es/bitstream/handle/10324/48201/TFG-N.1601.pdf?sequence=1

Castro, J. F. G., & Jácome, T. C. M. (2019). AUTOMATIZACIÓN DE PROCESOS EN REDES DE DATOS MEDIANTE PROGRAMACIÓN EN PYTHON. APROPIACIÓN, GENERACIÓN Y USO EDIFICADOR DEL CONOCIMIENTO DE ESTUDIANTES SENTIPENSANTES, 163. https://uisrael.edu.ec/wp-content/uploads/2020/05/LIBRO-SENTIPENSANTES-FINAL-min.pdf#page=163

Chiavenato, I. (2000). Administración de recursos humanos (5a ed.). McGraw-Hill. https://frrq.cvg.utn.edu.ar/pluginfile.php/15522/mod_resource/content/0/Chiavenato%20Idalverto.%20Administraci%C3%B3n%20de%20Recursos%20Humanos.pdf

Colombiana, A. (2020). DISEÑO DE UN MODELO PARA LA OPTIMIZACIÓN DE LOS PROCESOS en el. https://www.redalyc.org/pdf/6735/673571164011.pdf

Cortina Robles, G. E., Valencia Borja, K. A. y Bayona Galindo, V. (2022). Automatización en los procesos de selección y reclutamiento, para simplificar el procedimiento de contratación [Tesis de pregrado, Universidad Cooperativa de Colombia]. Repositorio Institucional Universidad Cooperativa de Colombia. https://repository.ucc.edu.co/entities/publication/f8063090-729c-4446-957b-758f9a5d345a/full

Criado, J. I. (2021). Inteligencia artificial (y administración pública). EUNOMÍA. Revista en Cultura de la Legalidad, (20), 348-372. https://e-revistas.uc3m.es/index.php/EUNOM/article/view/6097

De la Villa, Manuel & Ruiz, Mercedes & Ramos, Isabel. (2004). Modelos de evaluación y mejora de procesos: Análisis comparativo. https://www.researchgate.net/publication/228925424_Modelos_de_evaluacion_y_mejora_de_procesos_Analisis_comparativo

Diaz, M. R. O., & Ospina, K. J. Z. (2023). Gobierno digital e inteligencia artificial, una mirada al caso colombiano. Administración & Desarrollo, 53(1), 1-34. https://dialnet.unirioja.es/servlet/articulo?codigo=9004212

Flores Vidal, J. G. (2015). Diseño de un sistema integrado para el mejoramiento de procesos y optimización gerencial y logístico en el programa social Qali Warma. https://www.academia.edu/80675822/Dise%C3%B1o_de_un_sistema_integrado_para_el_mejoramiento_de_procesos_y_optimizaci%C3%B3n_gerencial_y_log%C3%Adstico_en_el_programa_social_Qali_Warma

Fundación para Desarrollo Social y Científico de Colombia (FUNDESOCOL). (2016). Documento Política Pública Mujer y Equidad de Género (Versión PDF). https://cota-cundinamarca.gov.co/Transparencia/BancoDocumentos/DOCUEMENTO%20POLITICA%20PUBLICA%20MUJER%20Y%20EQUIDAD%20DE%20G%C3%89NERO.pdf

Martínez Herrera, H. (2022). Contratación de personal mediante inteligencia artificial: evidencia preliminar. Universitat Oberta de Catalunya. https://openaccess.uoc.edu/bitstream/10609/142389/6/hectormhTFM0122memoria.pdf

Minguijon J. y Serrano-Martinez C. (2022). La Inteligencia Artificial en los Servicios Sociales: estado de la cuestión y posibles desarrollos futuros. Cuadernos de Trabajo Social. https://doi.org/10.5209/cuts.78747

Mirjalili, V., & Raschka, S. (2020). Python machine learning. Marcombo. https://books.google.es/books?hl=es&lr=&id=5EtOEAAAQBAJ&oi=fnd&pg=PT5&dq=analisis+de+datos+con+python&ots=erF_OvWIH5&sig=-3perBAKSPextztkbCOvWgwRfw8#v=onepage&q&f=false

Ocaña-Fernández, Y., Valenzuela-Fernández, L. A., Vera-Flores, M. A., & Rengifo-Lozano, R. A. (2021). Inteligencia artificial (IA) aplicada a la gestión pública. Revista Venezolana de Gerencia, 26(94), 696-707. https://www.redalyc.org/journal/290/29069612013/29069612013.pdf

Pardo, A., Ruiz, M. Á., & San Martín, R. (2022). Análisis de datos en ciencias sociales y de la salud I. https://ideice.gob.do/documentacion/publicaciones-msg-set-id-2-art-p1-167-analisis-de-datos-en-ciencias-sociales-y-de-la-salud-i

Quinto, N. M. D., Villodas, A. J. C., Montero, C. P. C., Cueva, D. L. E., & Vera, S. A. N. (2021). La inteligencia artificial y la toma de decisiones gerenciales. Revista de Investigación Valor Agregado, 8(1), 52-69. https://revistas.upeu.edu.pe/index.php/ri_va/article/view/1631

Rivas, J. G. R., & Castillo, S. R. (2022). Uso de Python para el análisis de datos aplicado en la investigación. Investigación y Ciencia Aplicada a la Ingeniería. http://ojs.incaing.com.mx/index.php/ediciones/article/view/188

Salazar Marín, G. M. (2022). Aplicación del algoritmo de regresión lineal usando python para optimizar los recursos humanos en puestos de insumos químicos de SUNAT. https://repositorio.unfv.edu.pe/handle/20.500.13084/7507

Urrutia Labrín, C. I. (2015). Análisis de la implementación de programas sociales desde el enfoque de interfaz: El caso del Programa Acción del Fondo de Solidaridad e inversión social. https://repositorio.uchile.cl/handle/2250/137164

Zilberman, J. (2021). ¿Puede la inteligencia artificial optimizar el proceso de selección de talento?. Review of Global Management, 7(1), 10 22. https://revistas.upc.edu.pe/index.php/rgm/article/view/1925

Apéndices

Apéndice A

Link del Video

https://youtu.be/Q44wb5r03mY

Apéndice B

Código de Entrenamiento de los Modelos

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (accuracy_score, f1_score, recall_score, roc_auc_score)
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.neural_network import MLPClassifier


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

```
# Normalizar datos para modelos sensibles a escalas
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

modelos = {
    “Árbol de Decisión”: DecisionTreeClassifier(max_depth=7, random_state=42),
    “Regresión Logística”: LogisticRegression(max_iter=1000),  # Aumentar iteraciones
    “SVM”: SVC(probability=True),  # Habilitar probabilidades para AUC-ROC
    “KNN”: KNeighborsClassifier(),
    “Naïve Bayes”: GaussianNB(),
    “Análisis Discriminante”: LinearDiscriminantAnalysis(),
    “Red Neuronal”: MLPClassifier(hidden_layer_sizes=(10,), max_iter=500, random_state=42)
}

# Diccionario para almacenar métricas
metricas = {
    'Modelo': [],
    'Precisión': [],
    'F1-Score': [],
    'Recall': [],
    'AUC-ROC': []
}
```

```
for nombre, modelo in modelos.items():
    try:
        # Entrenamiento con datos escalados para modelos que lo requieren
        if nombre in [“SVM”, “KNN”, “Regresión Logística”, “Red Neuronal”]:
            modelo.fit(X_train_scaled, y_train)
            y_pred = modelo.predict(X_test_scaled)
            y_proba = modelo.predict_proba(X_test_scaled)[:, 1] if hasattr(modelo, “predict_proba”)
else None
        else:
            modelo.fit(X_train, y_train)
            y_pred = modelo.predict(X_test)
            y_proba = modelo.predict_proba(X_test)[:, 1] if hasattr(modelo, “predict_proba”) else
None

        # Cálculo de métricas
        metricas['Modelo'].append(nombre)
        metricas['Precisión'].append(accuracy_score(y_test, y_pred))
        metricas['F1-Score'].append(f1_score(y_test, y_pred))
        metricas['Recall'].append(recall_score(y_test, y_pred))
        metricas['AUC-ROC'].append(roc_auc_score(y_test, y_proba) if y_proba is not None else
np.nan)
```

```
    except Exception as e:
        print(f"Error en {nombre}: {str(e)}")
        continue
# Crear DataFrame con resultados
df_metricas = pd.DataFrame(metricas)
df_metricas = df_metricas.round(3)


# Mostrar resultados ordenados por AUC-ROC
print("\nResultados completos:")
print(df_metricas.sort_values(by='AUC-ROC',ascending=False).to_string(index=False))
```

```
Resultados completos:
                Modelo  Precisión  F1-Score  Recall  AUC-ROC
     Árbol de Decisión      0.931     0.919   0.912    0.963
Análisis Discriminante      0.881     0.868   0.912    0.932
          Red Neuronal      0.896     0.877   0.873    0.929
                   SVM      0.850     0.821   0.805    0.926
   Regresión Logística      0.888     0.871   0.893    0.919
                   KNN      0.674     0.634   0.663    0.723
           Naïve Bayes      0.484     0.585   0.854    0.534
```

Apéndice C

Código de Optimización de Modelos

```
 from sklearn.model_selection import GridSearchCV


# Definir hiperparámetros a optimizar para cada modelo
param_grid = {
   “Árbol de Decisión”: {
      'max_depth': [3, 5, 7, 10],
      'min_samples_split': [2, 5, 10],
      'criterion': ['gini', 'entropy']
   },
   “Regresión Logística”: {
      'C': [0.1, 1, 10],
      'solver': ['liblinear', 'saga']
   },
   “SVM”: {
      'C': [0.1, 1, 10],
      'kernel': ['linear', 'rbf']
   },
   “KNN”: {
      'n_neighbors': [3, 5, 7],
      'weights': ['uniform', 'distance']
   },
```

```
    “Red Neuronal”: {
        'hidden_layer_sizes': [(10,), (20,), (10,10)],
        'alpha': [0.0001, 0.001, 0.01]
    }
}

# Diccionario para almacenar mejores modelos y métricas
best_models = {}
optimized_metrics = {
    'Modelo': [],
    'Precisión': [],
    'F1-Score': [],
    'Recall': [],
    'AUC-ROC': [],
    'Mejores Hiperparámetros': []
}

for nombre, modelo in modelos.items():
    if nombre not in param_grid:
        continue  # Saltar modelos sin grid definido (Naïve Bayes, Análisis Discriminante)

    print(f”\nOptimizando {nombre}...”)
```

```
try:
    # Configurar GridSearchCV
    grid = GridSearchCV(
        estimator=modelo,
        param_grid=param_grid[nombre],
        scoring='f1',
        cv=5,
        n_jobs=-1
    )

    # Entrenar con datos escalados si es necesario
    if nombre in [“SVM”, “Regresión Logística”, “Red Neuronal”]:
        grid.fit(X_train_scaled, y_train)
    else:
        grid.fit(X_train, y_train)

    # Almacenar mejor modelo
    best_models[nombre] = grid.best_estimator_

    # Evaluar en test
    if nombre in [“SVM”, “Regresión Logística”, “Red Neuronal”]:
        y_pred = grid.predict(X_test_scaled)
        y_proba = grid.predict_proba(X_test_scaled)[:,1]
```

```
        else:
            y_pred = grid.predict(X_test)
            y_proba = grid.predict_proba(X_test)[:,1]

        # Registrar métricas
        optimized_metrics['Modelo'].append(nombre)
        optimized_metrics['Precisión'].append(accuracy_score(y_test, y_pred))
        optimized_metrics['F1-Score'].append(f1_score(y_test, y_pred))
        optimized_metrics['Recall'].append(recall_score(y_test, y_pred))
        optimized_metrics['AUC-ROC'].append(roc_auc_score(y_test, y_proba))
        optimized_metrics['Mejores Hiperparámetros'].append(grid.best_params_)

        print(f"¡Optimización exitosa! Mejores parámetros: {grid.best_params_}")

    except Exception as e:
        print(f"Error en {nombre}: {str(e)}")
        continue

# Crear DataFrame con resultados optimizados
df_optimized = pd.DataFrame(optimized_metrics).round(3)

# Mostrar resultados comparativos
print("\n=== Comparación Métricas Originales vs Optimizadas ===")
```

```
print(“Originales:”)
print(df_metricas[df_metricas['Modelo'].isin(optimized_metrics['Modelo'])].sort_values('AUC-ROC', ascending=False).to_string(index=False))
print(“\nOptimizadas:”)
print(df_optimized.sort_values('AUC-ROC', ascending=False).to_string(index=False))
```

```
=== Comparación Métricas Originales vs Optimizadas ===
Originales:
            Modelo  Precisión  F1-Score  Recall  AUC-ROC
  Árbol de Decisión      0.931     0.919   0.912    0.963
      Red Neuronal      0.896     0.877   0.873    0.929
               SVM      0.850     0.821   0.805    0.926
Regresión Logística      0.888     0.871   0.893    0.919
               KNN      0.674     0.634   0.663    0.723

Optimizadas:
            Modelo  Precisión  F1-Score  Recall  AUC-ROC                                    Mejores Hiperparámetros
  Árbol de Decisión      0.927     0.915   0.917    0.964 {'criterion': 'entropy', 'max_depth': 7, 'min_samples_split': 2}
      Red Neuronal      0.898     0.879   0.868    0.943                    {'alpha': 0.01, 'hidden_layer_sizes': (20,)}
Regresión Logística      0.884     0.867   0.893    0.922                              {'C': 0.1, 'solver': 'liblinear'}
               SVM      0.915     0.900   0.898    0.912                                  {'C': 1, 'kernel': 'linear'}
               KNN      0.655     0.621   0.663    0.682                      {'n_neighbors': 3, 'weights': 'uniform'}
```