Estudiantes
Julian Leonardo Robles Cabanzo
Diego Fernando Malagon Saenz

4. Ejercicios y Problemas

Ejercicios propuestos

1. **Considere la figura 6.1, tome una ecuación determinada, por ejemplo una raíz cúbica, o un seno, genere un data set con muchos valores. Con base en ese data set y utilizando una herramienta de ML, encuentre un modelo para el cálculo de la raíz cuadrada. Úselo con 10 ejemplos y compare los resultados con los que da la función del lenguaje.**

   Teniendo en cuenta que el ML consiste en entregarle datos y respuestas a estos datos para generar un modelo, entonces se tomó la raíz cuadrada(Una ecuación sencilla para no complicar el código) para generar un data set con muchos valores y de esta manera usar aprendizaje de maquina para generar un modelo que lo resuelva. El modelo define 3 capas ocultas con 128, 64 y 32 neuronas respectivamente,Esta arquitectura permite que la red aprenda patrones de diferentes niveles de abstracción:

   - 128 → aprende relaciones generales y amplias.
   - 64 → refina interacciones.
   - 32 → detecta detalles específicos

   Una vez entrenado el modelo se testeo con 10 ejemplos obteniendo el siguiente resultado expresados en la figura 1 y en la figura 2 donde se observa más fácilmente el bajo error obtenido.

	Valor	Raíz Real	Raíz Predicha	Diferencia	Error Relativo (%)
0	55.804794	7.470261	7.453278	0.016983	0.227338
1	182.729522	13.517748	13.535721	0.017972	0.132954
2	307.607295	17.538737	17.538849	0.000112	0.000640
3	451.910967	21.258198	21.262731	0.004534	0.021327
4	694.870050	26.360388	26.348680	0.011707	0.044413
5	769.288644	27.736053	27.713265	0.022788	0.082162
6	819.414532	28.625418	28.628056	0.002639	0.009217
7	870.260382	29.500176	29.471101	0.029075	0.098557
8	905.407316	30.089987	30.053852	0.036135	0.120089
9	920.897846	30.346299	30.310692	0.035607	0.117336

Figura 1: Resultados de prueba del modelo.

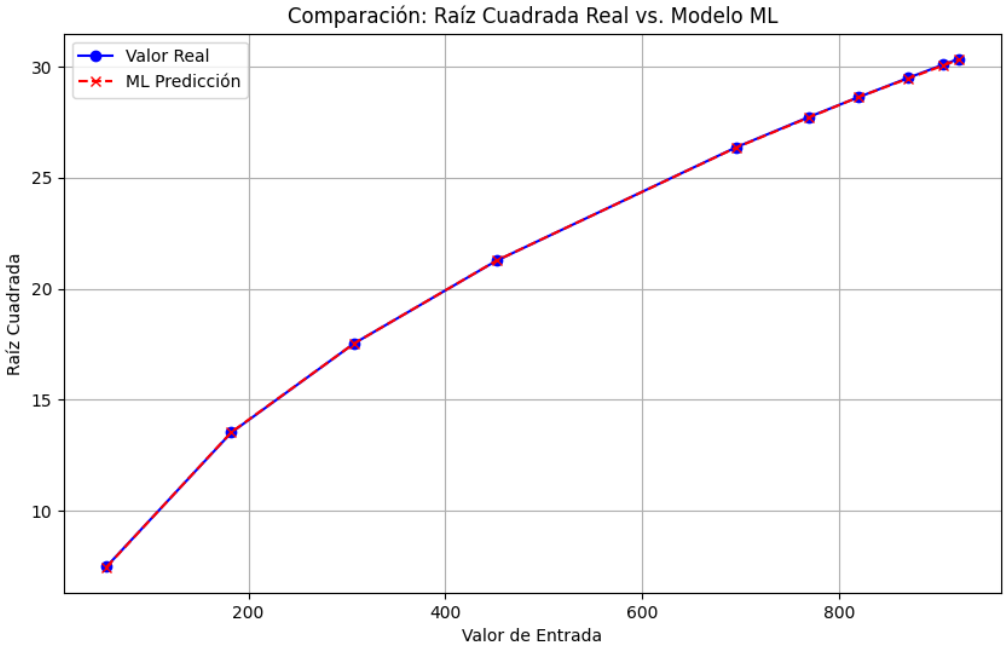

Figura 2: Comparación de la raiz cuadrada real con la del modelo de machine learning.

2. **Estudie el algoritmo SVM con todo detalle, mejore su documentación y con base en el haga cambios para una aplicación.**

   El algoritmo Support Vector Machine (SVM) es una técnica supervisada utilizada principalmente para clasificación, aunque también puede aplicarse en regresión. Su objetivo principal es encontrar el hiperplano óptimo que separe los datos en distintas clases con el mayor margen posible. Este margen representa la distancia entre dicho hiperplano y los puntos más cercanos de cada clase, llamados vectores de soporte. Cuanto mayor sea el margen, más robusta suele ser la separación frente a ruido o datos nuevos.
   Una de las fortalezas de SVM es su capacidad para trabajar en espacios no lineales gracias a la técnica del kernel trick. Mediante funciones kernel (como el radial basis function (RBF) o el polinómico) SVM proyecta los datos a un espacio de mayor dimensión donde pueden ser separables linealmente. Esto lo hace especialmente útil en problemas complejos donde los datos presentan estructuras difíciles de modelar con métodos convencionales.
   En términos prácticos, SVM tiende a ofrecer buenos resultados incluso con conjuntos de datos limitados y alta dimensionalidad. Sin embargo, su rendimiento depende de la correcta elección del kernel, la penalización por errores (C) y los parámetros específicos del kernel. Cuando se ajusta adecuadamente, SVM puede lograr una clasificación precisa y generalizable, lo que lo convierte en una herramienta poderosa para tareas como detección de anomalías, reconocimiento de patrones y categorización de textos.

   Dicho esto se pensó en una aplicación de clasificación de imágenes, más específicamente imágenes de calor, dicho esto se encontró una base de datos en Kaggle con tres tipos de datos, imágenes de personas, imágenes de carros e imágenes de gatos, entonces para simplificar solo se tomó las imágenes de personas y gatos, con 1782 fotos para cada conjunto. Fue en este punto donde más problemas se tuvo a la hora de cargar los datos pues al ser un conjunto tan grande el software no lo reconocía, fue por eso que se tuvo que instalar e implementar la librería gdown.
   Los resultados de entrenamiento se pueden ver en la figura 3, donde se puede evidenciar que se logro una muy

buena precesión en la clasificación de ambos conjuntos de datos.

```
==================================================
REPORTE DE CLASIFICACIÓN
==================================================
              precision    recall  f1-score   support

      Humano       0.93      0.93      0.93       446
        Gato       0.93      0.93      0.93       445

    accuracy                           0.93       891
   macro avg       0.93      0.93      0.93       891
weighted avg       0.93      0.93      0.93       891
```

Figura 3: Resultado del entrenamiento con SVM

Se hicieron algunas pruebas del modelo, obteniendo buenos resultados y finalmente se plantearon dos gráficos para representar mejor los resultados, una matriz de confusión en la que se puede ver que la mayoría de resultados resultaron ser verdaderos positivos, y por otro lado se un gráfica de barras con la distribución de las perdiciones en la que se ve que la distribución es la misma para ambos conjunto lo cual tiene mucho sentido pues la cantidad de datos fue la misma para ambas etiquetas.

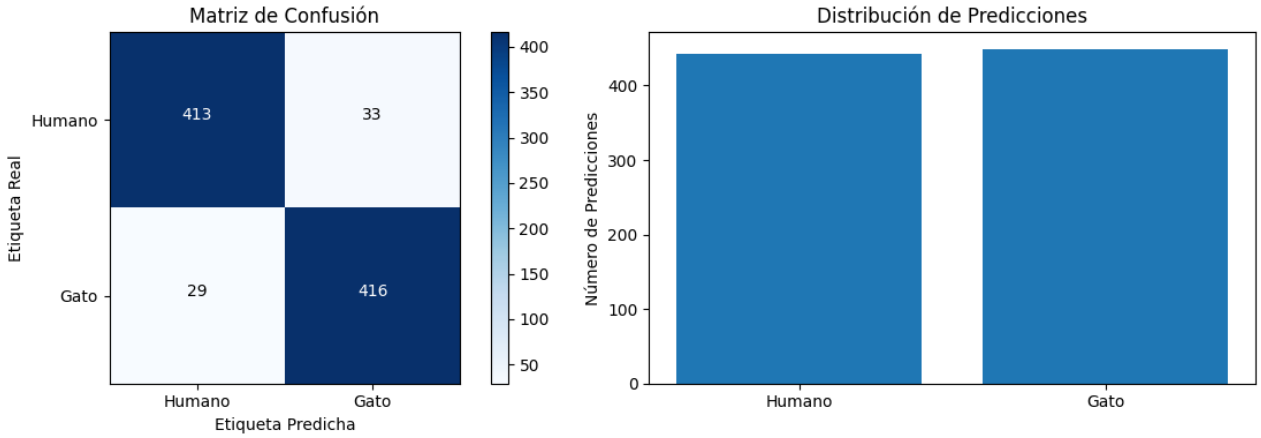

Figura 4: Matriz de confusión y gráfico de distribución.

3. **Estudie el algoritmo de árboles de decisión, con todo detalle mejore su documentación y con base en el haga cambios para una aplicación.**

   El algoritmo de árboles de decisión es una técnica supervisada utilizada tanto en clasificación como en regresión. Funciona como un modelo jerárquico que divide los datos en subconjuntos basándose en preguntas binarias sobre

sus características. Cada nodo del árbol representa una decisión, y cada rama indica el resultado de esa decisión, hasta llegar a una hoja que contiene la predicción final. Su estructura visual lo convierte en una herramienta intuitiva, fácil de interpretar, y útil para identificar relaciones y patrones entre variables.
A pesar de su claridad, los árboles de decisión simples tienden a ser frágiles ante pequeñas variaciones en los datos, lo que puede generar sobreajuste (overfitting) o bajo poder de generalización. Por esta razón, se han desarrollado enfoques más robustos como Random Forest, que mejora la estabilidad y precisión del modelo al construir múltiples árboles de decisión y combinar sus resultados. Este enfoque se basa en el principio del bagging (Bootstrap Aggregating), donde se crean subconjuntos de datos aleatorios para entrenar cada árbol de forma independiente.
En Random Forest, cada árbol contribuye con una predicción —en clasificación se toma la clase más votada (mayoría), mientras que en regresión se calcula el promedio de todas las salidas. Además, se utiliza una técnica llamada feature randomness, donde cada árbol selecciona un subconjunto aleatorio de variables en cada nodo, lo que introduce diversidad en la estructura del bosque y evita que todos los árboles aprendan lo mismo. Esta combinación de aleatoriedad y agregación permite que Random Forest sea altamente preciso, resistente al ruido y capaz de manejar conjuntos de datos complejos con muchas variables. Teniendo en cuenta esto, se desarrollo un código con asesoria de la IA Claude, que utilice Random Forest para resolver un problema de clasificación binaria de tarjetas de credito, usando un archivo csv que servirá de dataset con los detalles de información de diferentes personas y si le fue aprobado o no el credito, la ejecución del código tardó un aproximado de 3 minutos y medio.

De la figura 5 se rescatan varios resultados, El gráfico circular muestra que el 88.7 % de las solicitudes fueron aprobadas frente a un 11.3 % rechazadas. Esta alta tasa de aprobación podría indicar un sesgo en el conjunto de datos o criterios poco restrictivos, algo importante para ajustar el modelo si se quiere una clasificación más balanceada. Tambien se presenta un análisis por género revela tasas de aprobación similares, aunque sería útil calcular métricas específicas como el recall por clase para confirmar si hay un sesgo oculto. En cuanto a la educación, los niveles más altos están claramente asociados con mayor probabilidad de aprobación. Random Forest suele captar bien este tipo de relaciones jerárquicas.Por otro lado tambien se encontró que la mayoría de los aprobados se concentran en niveles bajos de ingreso anual, lo cual podría parecer contraintuitivo. Finalmente se tiene un mapa de calor permite ver cómo se relacionan variables como ingreso, número de hijos, días empleados, miembros familiares y la etiqueta de aprobación. Si alguna de estas tiene una fuerte correlación con la salida, puede considerarse una variable altamente predictiva, lo que se alinea con cómo Random Forest calcula la importancia de cada feature.

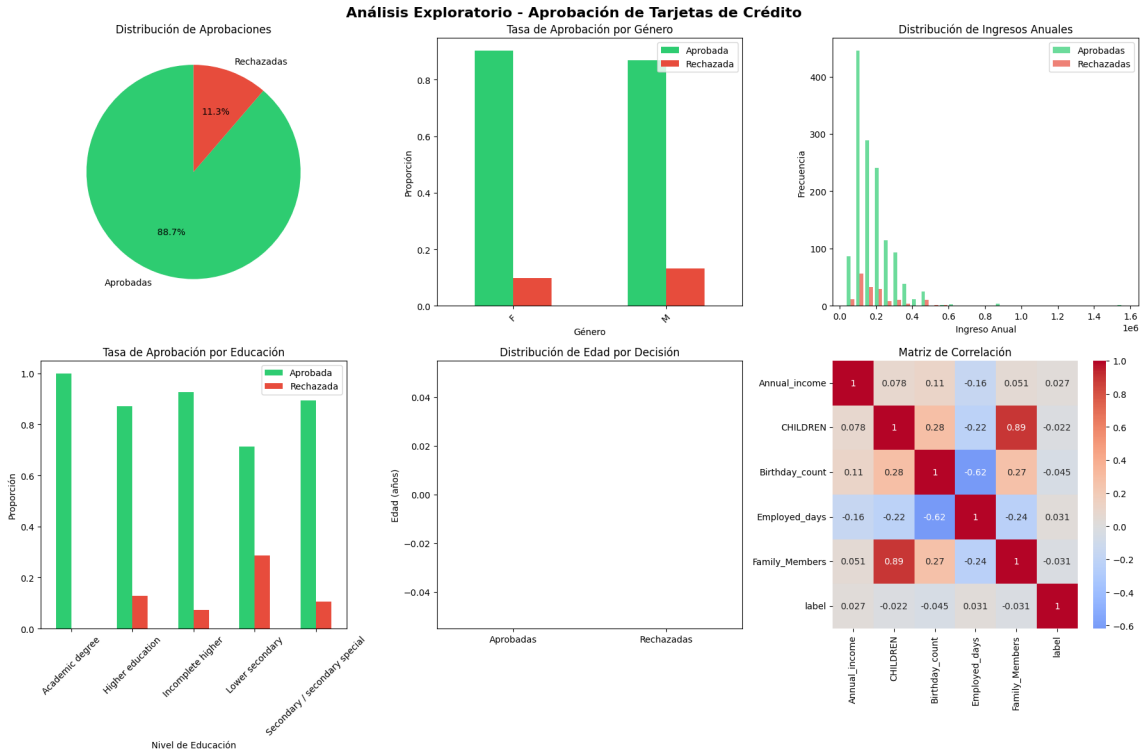

Figura 5: Resultados importantes de estudio con random forest.

Luego de esto se opto por optimizar el modelo con los hiperparametros llegando a los resultados mostrados en el la ultima figura, de cual se dastacan los resutados finales los cuales fueron la determinación de que el factor que más influye en la aprobación de los créditos es la edad, y el que menos es la educación y el genero.

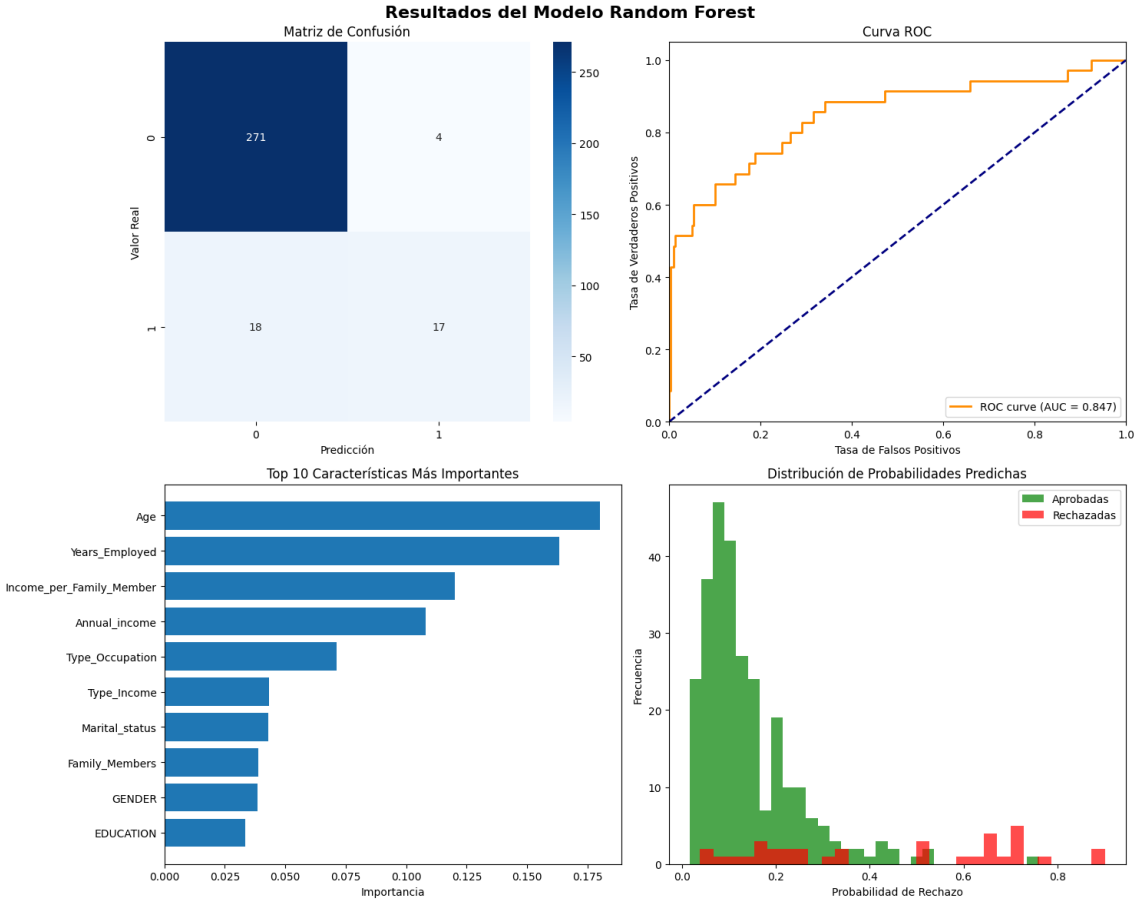

Figura 6: Resultados luego de la optimización.