INGENIERÍA
EN AUTOMATIZACIÓN

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Ingeniería

APLICACIÓN DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL EN AUTOMATIZACIÓN

EXAMEN 1: KNN TITANIC

Alumno:
Emilio Cabrera Mendoza

Periodo: 2025-2

Entrega: Martes 21 de octubre de 2025

Catedrático: Aceves Fernández Marco Antonio, Dr.

Índice

1 Marco Teórico

Esta sección presenta los fundamentos teóricos de los algoritmos y las técnicas de preprocesamiento utilizadas en este estudio.

1.1 K-Vecinos Más Cercanos (K-Nearest Neighbors, KNN)

El algoritmo KNN es un clasificador no paramétrico que se basa en el aprendizaje supervisado. La clasificación de un nuevo punto de datos se determina por el voto de mayoría de sus K vecinos más cercanos en el espacio de características. Es una técnica basada en la distancia, lo que hace que el escalado de características sea un paso crítico.

1.2 Análisis de Componentes Principales (Principal Component Analysis, PCA)

PCA es una técnica de reducción de dimensionalidad que transforma el conjunto original de variables en un conjunto más pequeño de nuevas variables, llamadas Componentes Principales. Estas componentes son combinaciones lineales de las variables originales y capturan la mayor varianza posible en los datos. Su principal uso es simplificar el modelo, reducir el ruido y entender la importancia relativa de las características.

1.3 Técnicas de Imputación

La imputación es el proceso de reemplazar valores faltantes en un conjunto de datos. La elección del método es crucial, ya que afecta la distribución y la varianza de la característica.

1.3.1 Imputación por Mediana o Media

Este es el método más simple, donde el valor faltante se reemplaza por la media (μ) o la mediana ($\tilde{x}$) de los valores observados de la característica.

- **Ventaja:** Fácil de implementar.
- **Desventaja:** Reduce la varianza natural de los datos y puede subestimar el error estándar. Fue evaluado pero descartado en este análisis.

1.3.2 Imputación Hot-Deck Aleatoria

Consiste en reemplazar un valor faltante con un valor muestreado al azar de los valores observados (no faltantes) de la misma característica.

- **Ventaja:** Preserva mejor la forma de la distribución original de la característica y su varianza en comparación con la media/mediana.
- **Uso en este Reporte:** Fue la técnica elegida para la imputación de la característica `Age` tras un análisis comparativo visual.

1.4 Normalización de Características (Min-Max Scaling)

La normalización Min-Max transforma los datos para que todos los valores numéricos queden en un rango específico, típicamente [0, 1].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Esto asegura que las características con diferentes escalas contribuyan de manera equitativa a la medida de distancia, siendo indispensable para el algoritmo KNN.

2 Objetivo

El objetivo de este análisis es doble:

1. Implementar y validar un clasificador de K-Vecinos Más Cercanos (KNN) de forma nativa en Python, utilizando únicamente las librerías `NumPy`, `Pandas`, `Collections`, `Math`, `Matplotlib` y `Seaborn`.

2. Evaluar el impacto de las técnicas de preprocesamiento, específicamente la Imputación Hot-Deck para la edad y la Normalización Min-Max de características, en el rendimiento predictivo del modelo sobre el conjunto de datos del Titanic.

3. Analizar la relevancia de las características mediante PCA y evaluar el rendimiento del modelo con un subconjunto reducido de ellas.

3 Metodología

La metodología de modelado sigue el flujo implementado en el Jupyter Notebook `titanic_knn.ipynb`.

3.1 Preprocesamiento y Limpieza de Datos

- **Carga y Selección Inicial:** Se cargó el dataset `Titanic-Dataset.csv`. Se descartaron las columnas `Name`, `PassengerId`, `Ticket`, `Cabin`, y `Embarked`. Las características retenidas fueron `Survived`, `Pclass`, `Sex`, `Age`, `SibSp`, `Parch`, y `Fare`.

- **Codificación Categórica:** La variable `Sex` se codificó numéricamente ('male': 0, 'female': 1).

- **Imputación de Valores Faltantes ('Age'):** Se detectaron 177 valores faltantes en `Age`. Se comparó visualmente la imputación por media y la imputación Hot-Deck Aleatoria (ver Figura 1 y 2). Se seleccionó **Hot-Deck Aleatoria** por preservar mejor la distribución original.

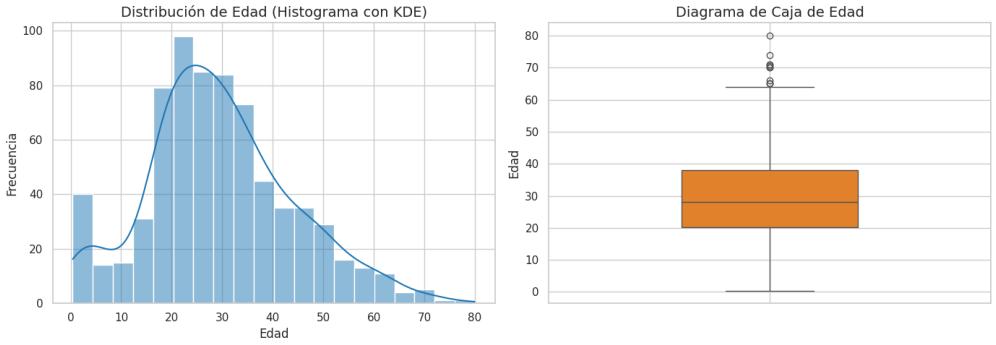

Figure 1: Distribución original de la característica 'Age' (antes de imputación).

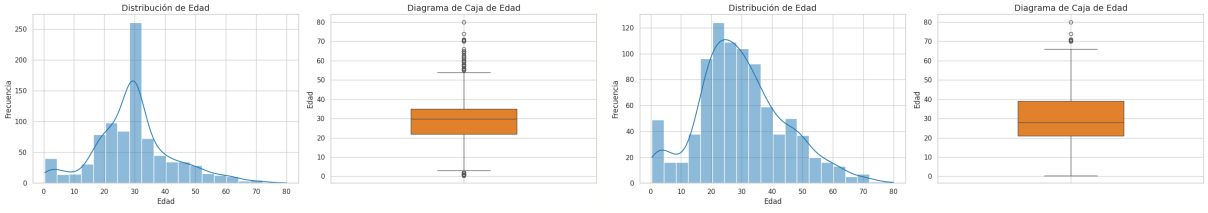

Figure 2: Comparación de la distribución de 'Age' tras imputación por Media (izquierda) y Hot-Deck (derecha).

3.2 Normalización de Características

Se utilizó la función customizada `normalize_dataframe` para aplicar la normalización Min-Max al rango [0, 1]. Esta técnica es fundamental para KNN ya que el modelo se basa en la Distancia Euclídea.

- **Fórmula de Normalización:** $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$
- **Aplicación:** Se aplicó a todas las características numéricas después de la imputación y codificación.

3.3 Análisis de Componentes Principales (PCA)

Se aplicó PCA sobre los datos normalizados (excluyendo la variable objetivo `Survived`) para evaluar la varianza explicada por cada componente principal derivado de las características `Pclass`, `Sex`, `Age`, `SibSp`, `Parch`, y `Fare`.

3.4 Modelo y Evaluación Customizados

- **División de Datos:** Los datos se dividieron aleatoriamente en 80% para entrenamiento y 20% para prueba.
- **Modelo KNN:** El clasificador `knn_model` se basa en la **Distancia Euclídea** (`euc_distance`) y determina la predicción final mediante **Voto de Mayoría** entre los K vecinos más cercanos.

- **Métricas Robustas:** La función `metrics` calcula Exactitud, Precisión, Sensitividad (Recall) y F1-Score, con manejo de errores de división por cero.

4 Resultados y Discusión

El modelo se entrenó con el 80% de los datos normalizados y se evaluó con el 20% restante. Se realizó una optimización de K evaluando el error en el conjunto de prueba para K de 1 a 40.

4.1 Resultados de PCA

El análisis de PCA sobre las 6 características predictoras normalizadas arrojó los siguientes porcentajes de varianza explicada por los eigenvalores:

Table 1: Varianza Explicada por Componente Principal

Componente	**Eigenvalor**	**Varianza Explicada (%)**
1	0.2415	50.05
2	0.1705	35.33
3	0.0320	6.62
4	0.0223	4.63
5	0.0106	2.20
6	0.0056	1.17

Los primeros 3 componentes, asociados principalmente a `Pclass`, `Sex` y `Age`, explican más del 90% de la varianza (50.05% + 35.33% + 6.62% = 91.99%). Basado en esto, se decidió evaluar el modelo KNN también con este subconjunto reducido de características.

4.2 Optimización de K

Se graficó la tasa de error (1 - Exactitud) contra el valor de K para el conjunto de prueba, tanto para el modelo con todas las características como para el modelo reducido.

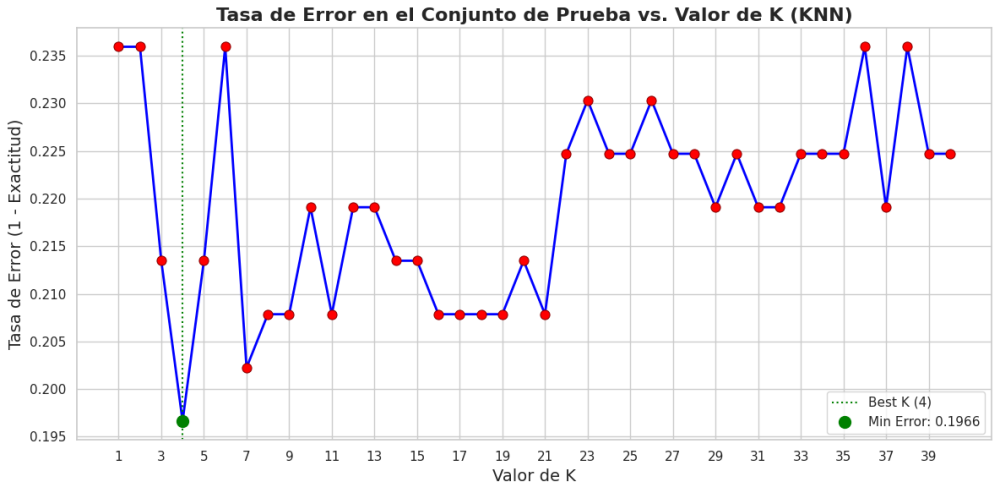

Figure 3: Tasa de error en el conjunto de prueba vs. el valor de K.

Ambos análisis (con todas las características y con las 3 principales) mostraron que el valor óptimo de K (impar, para evitar empates) que minimiza la tasa de error es **K=7**.

4.3 Métricas de Rendimiento (K=7)

A continuación, se presentan las métricas de rendimiento obtenidas con K=7 para ambos conjuntos de características.

Table 2: Métricas de Rendimiento del Modelo KNN (K=7)

Métrica	**Valor (Todas las Características)**	**Valor (Pclass, Sex, Age)**
Exactitud (Accuracy)	0.7978	0.7978
Precisión (Precision)	0.7846	0.7846
Sensitividad (Recall)	0.6986	0.6986
F1 Score	0.7391	0.7391

4.4 Discusión

La implementación de la imputación Hot-Deck fue clave para mantener una distribución de edad realista. La normalización Min-Max permitió que el KNN funcionara correctamente al dar igual peso a todas las características en el cálculo de la distancia.

El análisis PCA sugirió que `Pclass`, `Sex` y `Age` son las características más informativas, capturando casi el 92% de la varianza. Sorprendentemente, el rendimiento del modelo KNN con K=7 fue idéntico utilizando todas las características (6) o sólo estas 3 principales. Esto indica que las características adicionales (`SibSp`, `Parch`, `Fare`) no aportaron información adicional significativa para la clasificación con K=7 en este conjunto de prueba específico, o que su contribución fue redundante dada la presencia de las otras tres.

La exactitud final del 79.78% es un resultado competitivo y demuestra la viabilidad de implementar KNN de forma nativa con un preprocesamiento adecuado. La optimización de K mediante

la gráfica de tasa de error fue útil para seleccionar K=7, que ofreció el mejor balance en el conjunto de prueba.

5 Conclusión

Se implementó y evaluó exitosamente un clasificador KNN personalizado para predecir la supervivencia en el dataset del Titanic, logrando una exactitud del **79.78%** en el conjunto de prueba con K=7.

El preprocesamiento fue fundamental: la imputación Hot-Deck para la característica `Age` preservó mejor la distribución de los datos que la imputación por media, y la normalización Min-Max aseguró el correcto funcionamiento del algoritmo basado en distancia.

El análisis PCA reveló que las características `Pclass`, `Sex`, y `Age` concentran la mayor parte de la varianza. La evaluación del modelo KNN con sólo estas tres características arrojó un rendimiento idéntico al modelo con todas las características, sugiriendo que las variables restantes podrían ser menos relevantes o redundantes para este clasificador específico con K=7.

Este trabajo demuestra la efectividad del KNN implementado nativamente cuando se combina con técnicas de preprocesamiento apropiadas y validación mediante un conjunto de prueba.

Bibliografía

- Aceves Fernández, M. A. (2022). *Inteligencia artificial para programadores con prisa*. Universo de Letras. ISBN: 978-8418856723.
- Cabrera Mendoza, E. (2025). *Examen1_KNN* [Repositorio de código fuente]. GitHub. Recuperado de `https://github.com/EM50840/Examen1_KNN.git`