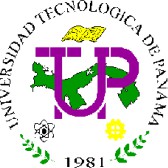
**UNIVERSIDAD TECNOLOGICA DE PANAMA**

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES CENTRO REGIONAL DE AZUERO

**PROYECTO FINAL**

**FC-FISC-1-8-2016**

Facilitador: **Luiyiana Pérez**

Asignatura: Analisis De Datos

Nombre de los estudiantes/cédula:

Reniher Murillo

Luis Navarro 8-1010-680

Grupo: 7IL121

# INTRODUCCIÓN

## Planteamiento del problema

En el mercado actual de automóviles, los consumidores enfrentan una gran variedad de modelos con diferentes características técnicas y precios. Factores como la seguridad, el costo de mantenimiento, el número de pasajeros o el tamaño del baúl son determinantes al evaluar si un vehículo se ajusta a las necesidades del comprador. Sin embargo, muchas veces, los usuarios carecen del conocimiento técnico necesario para interpretar todos estos criterios adecuadamente. Por otro lado, los fabricantes y distribuidores necesitan herramientas que les permitan identificar cómo la combinación de ciertas características afecta la percepción general del vehículo y su aceptabilidad.

El reto radica en encontrar una forma visual e intuitiva de representar todos estos factores, permitiendo a los usuarios finales y a los profesionales del sector automotriz tomar decisiones informadas y estratégicas. Aunque existen análisis estadísticos sobre la evaluación de automóviles, la mayoría son estáticos y poco accesibles para usuarios no técnicos.

### Objetivo general

Diseñar e implementar un tablero de control interactivo utilizando la biblioteca Dash de Python, que permita visualizar de forma dinámica la aceptabilidad de vehículos, a partir del análisis del conjunto de datos Car Evaluation de UCI Machine Learning Repository.  
El tablero permitirá filtrar por atributos como el precio, mantenimiento, número de puertas, capacidad, tamaño del baúl y nivel de seguridad, proporcionando herramientas visuales accesibles tanto para consumidores como para analistas del sector automotriz.

#### Descripción del proyecto

Este proyecto consiste en la construcción de un *dashboard* dinámico e interactivo que muestra información visual sobre cómo diversos atributos afectan la clasificación general de un automóvil (no aceptable, aceptable, bueno, excelente). Se implementó un proceso de limpieza, transformación y visualización de los datos utilizando Python, Pandas y la biblioteca Plotly Dash. La herramienta desarrollada permite analizar datos categóricos complejos mediante gráficos interactivos y predicciones, facilitando así la interpretación para usuarios sin conocimientos técnicos avanzados.Finalmente, el sistema será desplegado a través de GitHub y **Streamlit Cloud**, asegurando su disponibilidad en línea y demostrando su aplicabilidad en entornos reales.

# METODOLOGÍA

La metodología empleada fue de tipo **cuantitativa, exploratoria y descriptiva**, centrada en el análisis de un conjunto de datos categóricos de evaluación de automóviles. El desarrollo se llevó a cabo utilizando Python y la biblioteca **Streamlit** para la construcción de un tablero de control interactivo.

###### 2.1. Obtención del conjunto de datos

* **Fuente**: [Car Evaluation Dataset - UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/car+evaluation).
* Contiene 7 variables categóricas: buying, maint, doors, persons, lug\_boot, safety, y class.
* Se descargó y renombró el dataset en español para mejorar la comprensión (Precio\_Compra, Costo\_Mantenimiento, Num\_Puertas, etc.).

2.2. Preparación y limpieza de datos

* Se utilizó pandas para cargar, renombrar y explorar los datos.
* Aunque no presentaba valores nulos, se incorporó un proceso de verificación e interpolación preventiva (interpolate()), pensando en una futura extensión con datos reales.
* Se ajustaron etiquetas como "5more" a "5 o más" o "more" a "más de 4" para mayor claridad.

2.3. Construcción del dashboard interactivo

Se construyó el dashboard con **Streamlit**, que ofrece una experiencia intuitiva, reactiva y amigable. Las características implementadas fueron:

* **Filtros dinámicos** por precio, seguridad, capacidad, mantenimiento, etc.
* **Visualizaciones interactivas** con Plotly:
  + Histograma por niveles de aceptación.
  + Gráfico circular de distribución.
  + Comparativas por seguridad, capacidad, precio y aceptación.
  + Gráfico de dispersión entre precio, mantenimiento y seguridad.
* **Gráfico de radar personalizado** que compara cada auto con el promedio general.
* **Tabla filtrada de resultados**.
* **Sección de autos destacados**, ordenados por nivel de aceptación.
* **Mapa de calor** si existieran variables numéricas suficientes.
* **Gráfico de correlación interactivo** seleccionando pares de variables.
* **Descarga de datos en CSV** filtrados.
* **Sección predictiva**: se entrenó un modelo de árbol de decisión (DecisionTreeClassifier) para predecir la aceptabilidad de un vehículo en tiempo real a partir de las selecciones del usuario.
* **Árbol de decisión visual** y explicación de sus reglas (plot\_tree, export\_text).
* **Explicación detallada de cada variable** y su influencia.

2.4. Entrenamiento del modelo de predicción

* Se aplicó OrdinalEncoder a las variables categóricas.
* Se entrenó el modelo con train\_test\_split y DecisionTreeClassifier.
* Se limitó la profundidad del árbol (max\_depth=3) para mantener la interpretabilidad.
* Se mostró el árbol visualmente y sus reglas textuales para ayudar al usuario a entender las decisiones del modelo.

2.5. Implementación y despliegue

* El código fue implementado en un único archivo app.py con soporte de assets para visuales.
* Se alojó el proyecto en GitHub para compartir el código.
* Se preparó una presentación en PDF (presentacion\_dash.pdf) como soporte explicativo.
* El despliegue final se realizó en **Streamlit Cloud** para permitir acceso libre desde la web.

# 3. RESULTADOS

La implementación del tablero permitió obtener los siguientes resultados clave:

* Los usuarios pueden interactuar con los datos de manera visual e intuitiva, seleccionando filtros y observando cómo varía la aceptabilidad del automóvil.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* **La variable “Seguridad” resultó ser la de mayor impacto** en la clasificación final del auto.

A screen shot of a video player

AI-generated content may be incorrect.

* El modelo predictivo incorporado permite anticipar si un automóvil será aceptado o no, según sus características.

A black and green line

AI-generated content may be incorrect.

* Se descubrieron patrones de comportamiento:
  + Autos con **alta seguridad** y capacidad para **4 o más personas** tienen **mayor probabilidad de ser aceptados**.
  + Aquellos con **alto costo de mantenimiento** y **poca seguridad** son generalmente **no aceptables**.
* A screen shot of a computer

  AI-generated content may be incorrect.El gráfico radar permitió comparar cualquier automóvil individual frente al promedio, facilitando su evaluación visual.
* El uso de Streamlit permitió construir una interfaz moderna sin requerir conocimientos técnicos por parte del usuario.

# **Conclusión**

El desarrollo del tablero de control interactivo para la evaluación de automóviles, basado en el conjunto de datos de UCI Machine Learning Repository, demuestra cómo la tecnología puede facilitar la comprensión de información compleja para distintos tipos de usuarios. Mediante el uso de herramientas como Python, Pandas, Plotly y Streamlit, fue posible transformar datos categóricos en visualizaciones claras, intuitivas y funcionales, accesibles incluso para personas sin conocimientos técnicos.

La integración de un modelo predictivo de árbol de decisión permitió enriquecer el análisis con capacidades de predicción en tiempo real, mejorando la utilidad práctica del sistema. Además, la posibilidad de filtrar, visualizar y comparar atributos clave como seguridad, capacidad, precio y mantenimiento ofrece un enfoque integral para tomar decisiones informadas en la elección de un vehículo.

Finalmente, la publicación del proyecto en GitHub y su despliegue en Streamlit Cloud aseguran su disponibilidad, transparencia y potencial de mejora continua. Este proyecto no solo cumple con su propósito técnico y visual, sino que también se presenta como una solución educativa, funcional y aplicable a escenarios reales del sector automotriz.

# Referencias

 Chacon, S., & Straub, B. (2014). Pro Git (2.ª ed.). Apress.  
Recuperado de: <https://git-scm.com/book/en/v2>

Referencia para el uso de Git y GitHub.

 The GitHub Docs. (2024). GitHub Documentation. GitHub, Inc.  
Recuperado de: https://docs.github.com/es

Documentación oficial para crear repositorios, subir archivos y usar GitHub Pages o GitHub Actions.

 Streamlit Inc. (2024). Streamlit Documentation.  
Recuperado de: https://docs.streamlit.io/

Documentación oficial para desplegar aplicaciones con Streamlit Cloud.

 Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Python Software Foundation.  
Recuperado de: https://docs.python.org/3/

Referencia técnica sobre el lenguaje de programación usado.

 Sommerville, I. (2016). Ingeniería de Software (10.ª ed.). Pearson Educación.

Fundamentos del desarrollo de software y buenas prácticas.

 McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2.ª ed.). O'Reilly Media.

Para trabajos con análisis de datos o visualización.