

# Cálculo 1

Elaboración de modelo en  
Jupyter Notebook o Google  
Colab

## Clase 15

Ingeniería en ciberseguridad

La excelencia no se improvisa



## Clase 15. Elaboración de modelo en Jupyter Notebook o Google Colab

Jupyter Notebook y Google Colab son entornos interactivos ampliamente utilizados para el desarrollo de modelos en Python.

Estos entornos son especialmente útiles en áreas como la ciencia de datos, el aprendizaje automático, la visualización de datos y el análisis exploratorio, ya que permiten combinar código, texto explicativo, ecuaciones matemáticas y visualizaciones en un solo documento interactivo.

### 15.1 Jupyter Notebook

Jupyter Notebook es una aplicación web que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Es especialmente popular en la comunidad de ciencia de datos debido a su capacidad para facilitar un flujo de trabajo interactivo.

Entre sus características principales se encuentran:

- **Celdas de código y texto:** Permiten la ejecución de fragmentos de código Python y la inclusión de texto explicativo, lo que es ideal para documentar el proceso de análisis y los resultados obtenidos.
- **Soporte multilinguaje:** Aunque es más conocido por su uso con Python, Jupyter también admite otros lenguajes de programación a través de diferentes “kernels”.
- **Visualización de datos:** Integra fácilmente bibliotecas de visualización como Matplotlib, Seaborn y Plotly, entre otras, para crear gráficos y visualizaciones interactivas (Gómez & Torres, 2019).

### 15.2 Google Colab

Google Colab es una plataforma gratuita basada en Jupyter Notebooks, pero con la ventaja añadida de estar en la nube. Esto permite a los usuarios acceder y ejecutar su código desde cualquier lugar sin necesidad de configuraciones locales complejas. Entre sus características principales se encuentran:

- **Acceso a hardware potente:** Ofrece acceso gratuito a GPUs y TPUs, lo cual es beneficioso para tareas que requieren altos recursos computacionales, como el entrenamiento de modelos de aprendizaje profundo.

- **Colaboración en tiempo real:** Permite a varios usuarios colaborar en el mismo notebook de manera simultánea, facilitando el trabajo en equipo y la enseñanza.
- **Integración con Google Drive:** Facilita el almacenamiento y la compartición de notebooks y datos, simplificando el manejo de archivos (Pérez & López, 2020).

### 15.3 Pasos para la elaboración de un modelo

#### Importación de bibliotecas y datos:

- El proyecto comienza importando las bibliotecas necesarias, como NumPy, Pandas, Matplotlib, Seaborn, entre otras.
- Se carga el conjunto de datos desde una fuente local o en línea.

#### Exploración y preprocesamiento de datos:

- Se realiza un análisis exploratorio de datos (EDA) para comprender la estructura y las características del conjunto de datos.
- Se limpian los datos, manejando valores nulos, duplicados y transformando variables si es necesario.

#### Desarrollo del modelo:

- Se selecciona y entrena un modelo utilizando algoritmos de aprendizaje automático disponibles en bibliotecas como Scikit-learn o TensorFlow.
- Se ajustan los hiperparámetros y se validan los resultados del modelo usando técnicas como la validación cruzada.

#### Evaluación y visualización de resultados:

- Se evalúan las métricas de rendimiento del modelo, como precisión, recall, F1-score, entre otras.
- Se crean visualizaciones para interpretar los resultados y los patrones identificados.

#### Documentación y presentación:

- Se documenta cada paso del proceso en celdas de texto, proporcionando contexto y explicaciones detalladas.
- Se generan informes y gráficos que facilitan la comunicación de los hallazgos a los stakeholders o colaboradores.

Por ejemplo, un analista de datos puede usar Jupyter Notebook o Google Colab para desarrollar un

modelo de predicción de ventas, explorando datos históricos, entrenando un modelo de regresión y visualizando las proyecciones futuras de ventas en diferentes escenarios (Martínez & Sánchez, 2021).

### 3.9.4. Modelo matemático de negocio con análisis marginal y producción total completo con Python.

El modelo matemático de negocio es una herramienta crucial para entender y optimizar el funcionamiento de una empresa. Este modelo puede incluir componentes como el análisis marginal y la producción total, utilizando Python para realizar cálculos precisos y visualizaciones detalladas.

El análisis marginal se centra en cómo cambian los costos y beneficios con respecto a pequeñas variaciones en el nivel de producción o ventas. Es fundamental para la toma de decisiones empresariales, ya que ayuda a determinar el punto óptimo de producción o fijación de precios.

- **Costo marginal (CM):** Es el costo adicional incurrido al producir una unidad adicional de un producto. Matemáticamente, se define como la derivada del costo total con respecto a la cantidad producida, es decir:

donde  $C(q)$  es la función de costo total y  $q$  es la cantidad producida.

- **Ingreso marginal (IM):** Es el ingreso adicional obtenido al vender una unidad adicional. Se calcula como la derivada del ingreso total respecto a la cantidad vendida:

donde  $R(q)$  es la función de ingreso total (Pérez & Martínez, 2019).

La producción total se refiere a la cantidad total de bienes producidos por una empresa en un periodo determinado. Es crucial para evaluar la eficiencia y capacidad de producción de una empresa.

**Función de producción:** Describe la relación entre los insumos utilizados (trabajo, capital, etc.) y la cantidad de producción generada. Puede expresarse como:  $Q=f(L,K)$ , donde  $Q$  es la producción total,  $L$  representa el trabajo y  $K$  el capital. Esta función ayuda a determinar cómo los cambios en los insumos afectan la producción total (Gómez & Torres, 2020).

Python, con sus bibliotecas como NumPy, Pandas y Matplotlib, permite la implementación eficiente de modelos matemáticos de negocios. A continuación, se describe un flujo de trabajo típico:

- **Definición de funciones de costo e ingreso:** Se definen las funciones de costo total  $C(q)$  e ingreso total  $R(q)$  en términos de la cantidad  $q$ . Estas funciones pueden incluir componentes fijos y variables.
- **Cálculo de derivadas:** Se utilizan herramientas como SymPy para calcular simbólicamente las derivadas, obteniendo así las funciones de costo marginal e ingreso marginal.
- **Optimización de beneficios:** Se determina el nivel de producción  $q$  que maximiza el beneficio, encontrado cuando  $IM = CM$ . Esto se puede hacer utilizando métodos numéricos o de optimización, como `scipy.optimize`.
- **Visualización de resultados:** Se utilizan bibliotecas de visualización como Matplotlib para graficar las funciones de costo, ingreso, beneficio y sus respectivas derivadas. Esto facilita la interpretación de los resultados y la toma de decisiones.

Este modelo puede ser aplicado, por ejemplo, en una empresa manufacturera que busca optimizar su producción para maximizar el beneficio. El análisis marginal ayudará a determinar cuántas unidades adicionales deben producirse o no para lograr la máxima rentabilidad (López & Sánchez, 2021).

## Referencias

Gómez, M., & Rodríguez, J. (2019). *GeoGebra y su aplicación en la enseñanza de las matemáticas*. Editorial Educativa.

Martínez, S., & Torres, A. (2020). *Simulación y modelado en ingeniería: herramientas y aplicaciones*. Editorial Técnica.

Pérez, L., & López, J. (2021). *Tecnologías educativas: integración de software en la enseñanza de las ciencias*. Editorial Académica.

## Glosario de los términos citados

**Bibliotecas:** Conjunto de módulos y funciones predefinidos que pueden ser reutilizados en diferentes programas para realizar tareas específicas. En programación, las bibliotecas permiten simplificar el desarrollo al ofrecer herramientas ya creadas para manejar desde cálculos matemáticos hasta interfaces de usuario.

**Código:** Conjunto de instrucciones escritas en un lenguaje de programación que una computadora puede interpretar y ejecutar para realizar una tarea específica. El código puede variar desde simples secuencias de comandos hasta complejos sistemas de software.



**La excelencia no se improvisa**

síguenos

