

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ
DE CALDAS

Desarrollo de la tarea de investigación de librerías

Doctorado
en Ingeniería

<http://doctoradoingenieria.udistrital.edu.co>

Autor:

Victor Alfoso Diaz Gomez

Director:

PhD. Julián Andres Salamanca Bernal

Co-Director:

PhD. Nombre Co-director



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Grupo de Investigación
Grupo

de Investigación LIFAE

Énfasis: Ciencia de la
Información y el
Conocimiento

Doctorado en Ingeniería

19 /04/2025



Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, cuya filosofía enfatiza la legibilidad del código. Su sintaxis clara y sencilla permite un desarrollo eficiente y accesible tanto para principiantes como para expertos. Gracias a su estructura simple, Python facilita la escritura de código limpio y comprensible, reduciendo la cantidad de líneas necesarias en comparación con otros lenguajes de programación.

Python es un lenguaje multiparadigma, ya que admite diferentes estilos de programación, incluyendo la programación orientada a objetos, la programación imperativa y, en menor medida, la programación funcional. Su capacidad de tipado dinámico y su gestión automática de memoria lo hacen un lenguaje flexible y fácil de usar en diversos contextos. Además, es multiplataforma, lo que significa que el mismo código puede ejecutarse en distintos sistemas operativos sin necesidad de compilación previa.

El ecosistema de Python es vasto y cuenta con una extensa comunidad de desarrolladores que contribuyen constantemente con mejoras, paquetes y documentación. Su amplia biblioteca estándar proporciona módulos y herramientas que facilitan tareas como la manipulación de archivos, el desarrollo web, la gestión de bases de datos, el análisis de datos y la inteligencia artificial. Además, existen numerosos frameworks y bibliotecas de terceros, como Django y Flask para desarrollo web, Pandas y NumPy para ciencia de datos, TensorFlow y PyTorch para aprendizaje automático, y Selenium para automatización.

Python ha ganado una gran popularidad en diversas áreas de aplicación, incluyendo el desarrollo web, la automatización de tareas, la ciencia de datos, la inteligencia artificial, la ciberseguridad y la computación científica. Su versatilidad permite que se utilice tanto en proyectos pequeños como en soluciones empresariales a gran escala.

❖ Ejemplos y Aplicaciones de Python

❖ Desarrollo Web

- Uso de frameworks como Django y Flask para crear aplicaciones web dinámicas y escalables.
- Creación de API REST para conectar sistemas y servicios.

❖ Ciencia de Datos y Análisis

- Uso de bibliotecas como Pandas y NumPy para manipulación de datos.
- Creación de gráficos y visualizaciones con Matplotlib y Seaborn.
- Análisis de grandes volúmenes de datos y modelado estadístico.

❖ Inteligencia Artificial y Aprendizaje Automático

- Implementación de redes neuronales con TensorFlow y PyTorch.
- Procesamiento de lenguaje natural con NLTK y spaCy.
- Creación de chatbots y asistentes virtuales.

❖ Automatización y Scripting

- Automatización de tareas repetitivas con scripts en Python.
- Uso de Selenium para pruebas automatizadas en navegadores.
- Gestión de archivos y procesos del sistema de forma eficiente.

❖ Ciberseguridad y Hacking Ético

- Análisis de vulnerabilidades y pruebas de penetración con Scapy.
- Creación de herramientas para monitoreo de seguridad.
- Programación de scripts para escaneo de redes.

❖ Desarrollo de Videojuegos

- Uso de Pygame para crear videojuegos en 2D.
- Simulación de físicas y gráficos interactivos.

❖ Internet de las Cosas (IoT)

- Programación de microcontroladores con MicroPython.
- Creación de sistemas inteligentes conectados a la red.
- Gracias a su facilidad de aprendizaje y su potencia, Python se ha convertido en una de las opciones preferidas por empresas y desarrolladores de todo el mundo. Grandes compañías tecnológicas, como Google, Facebook, Netflix y Amazon, lo utilizan en sus sistemas y productos, demostrando su eficiencia y robustez en entornos productivos. Su capacidad de integración con otros lenguajes y tecnologías lo convierte en una herramienta indispensable para la programación moderna.
- Python es un lenguaje altamente versátil, con un diseño enfocado en la simplicidad y la productividad. Su comunidad activa y en constante crecimiento garantiza su evolución y relevancia en el mundo de la tecnología, asegurando que seguirá siendo una opción clave para el

desarrollo de software en el futuro.

Guía de Librerías Básicas de Python para Machine Learning

Este documento presenta una introducción práctica a las librerías más utilizadas en Python para el análisis de datos y el desarrollo de soluciones de Machine Learning. Incluye ejemplos de código, visualizaciones y casos de uso para facilitar el aprendizaje y la aplicación de estas herramientas en proyectos reales.

1. NumPy

NumPy es una librería especializada en el cálculo numérico y el análisis de datos. Su estructura principal, los arrays, permiten almacenar datos del mismo tipo en múltiples dimensiones. Ejemplos:

Array unidimensional:

```
np.array([1, 2, 3])
```

Array bidimensional:

```
np.array([[1, 2, 3], [4, 5, 6]])
```

Otras funciones útiles incluyen:

- `np.empty(dimensiones)`, `np.zeros(dimensiones)`, `np.ones(dimensiones)`
- `np.full(dimensiones, valor)`, `np.identity(n)`
- `np.arange(inicio, fin, paso)`, `np.linspace(inicio, fin, n)`
- `np.random.random(dimensiones)`

2. Matplotlib

Matplotlib es una librería de código abierto para visualización de datos. Fue creada en 2002 por John Hunter para visualizar señales cerebrales. Ofrece funciones similares a Matlab y permite crear gráficos detallados y personalizados.

Introducción a Librerías Básicas de Python para Machine Learning

3. Pandas

Pandas facilita la manipulación de datos estructurados. Sus principales características son: - Nuevas estructuras basadas en NumPy, como DataFrames y Series

- Soporte para lectura/escritura de CSV, Excel y SQL
- Acceso a datos mediante índices o nombres
- Métodos para reordenar, dividir y combinar datos
- Soporte para series temporales y alta eficiencia gracias a C/C++

4. Polars

Polars es una librería rápida para el manejo de dataframes. Sus ventajas incluyen:

- Uso de múltiples núcleos del procesador
- Consultas optimizadas para bajo consumo de memoria
- Capacidad para grandes conjuntos de datos
- API robusta y consistente
- Tipado estricto previo a ejecución

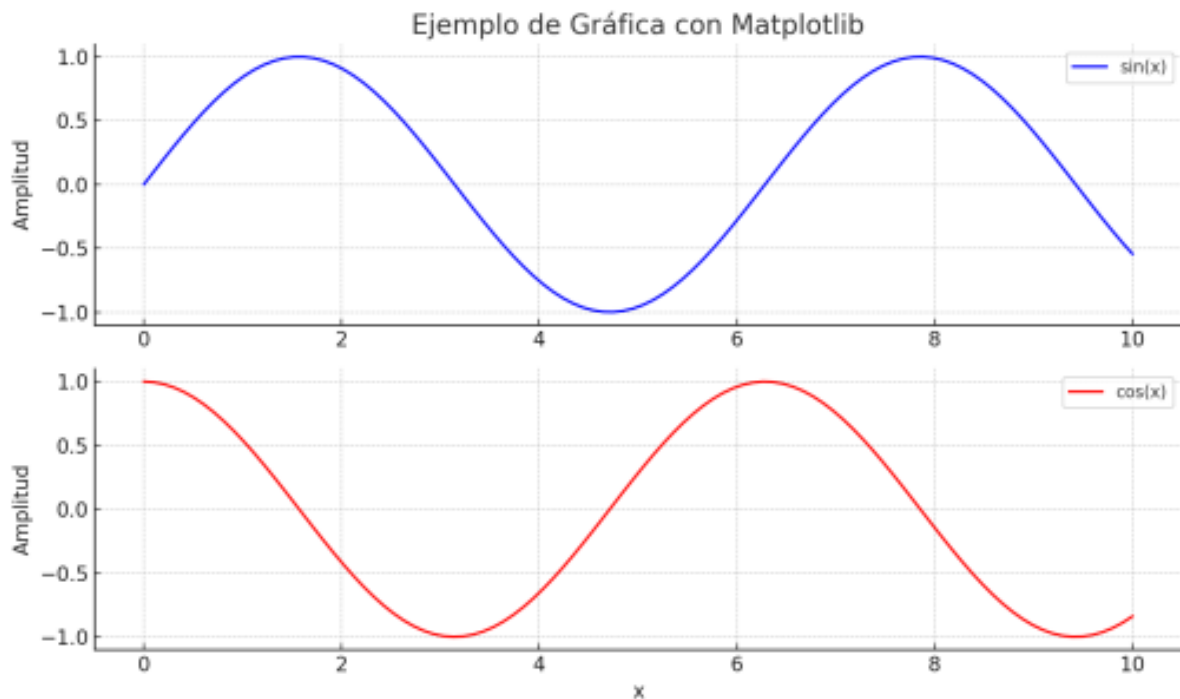
5. hvPlot

hvPlot es una herramienta de visualización de alto nivel compatible con múltiples fuentes de datos (pandas, polars, xarray, etc.) y backends gráficos (Bokeh, Matplotlib, Plotly). Permite explorar datos, generar informes y crear aplicaciones visuales interactivas.

Introducción a Librerías Básicas de Python para Machine Learning

6. Ejemplos de Aplicación y Visualización

A continuación, se presenta una gráfica de ejemplo generada con Matplotlib, que ilustra cómo se pueden visualizar funciones matemáticas como el seno y el coseno.



Este tipo de visualizaciones son fundamentales en el análisis de datos exploratorio (EDA), ya que permiten identificar patrones, tendencias y anomalías en los datos antes de aplicar modelos de machine learning.

1. Descripción de Casos de Uso Común

Podemos incluir ejemplos prácticos de cómo estas librerías se utilizan en el mundo real.
Ejemplos de uso en:

Análisis Exploratorio de Datos (EDA) con pandas y matplotlib

Entrenamiento de Modelos de Machine Learning con numpy y pandas

Visualización de Datos Complejos usando hvPlot

2. Aplicación con Machine Learning

Podemos hacer un ejemplo simple de cómo preparar datos para un modelo de machine learning usando pandas y numpy. Este ejemplo podría incluir:

Preparación de datos

Limpieza de datos

Normalización de datos

Dividir los datos en conjunto de entrenamiento y prueba

Usar un modelo de regresión o clasificación básica

3. Optimización del Rendimiento

Podemos hablar sobre cómo usar polars para manejar grandes volúmenes de datos de manera más eficiente que pandas, especialmente cuando se trabaja con datasets muy grandes (por ejemplo, más de 1 millón de filas).

4. Más Gráficas y Visualizaciones

Podemos agregar gráficos adicionales, como:

Gráficas de Dispersión para mostrar relaciones entre variables.

Histogramas para ver la distribución de datos.

Diagramas de Caja para detectar valores atípicos (outliers).

5. Pipelines de Machine Learning

Incluir un ejemplo de cómo utilizar librerías como scikit-learn junto con pandas y numpy para crear un pipeline de Machine Learning básico.

6. Manejo de Datos Faltantes y Limpieza de Datos

Podemos agregar un pequeño segmento sobre cómo manejar los valores faltantes, detectar valores nulos y limpiarlos eficientemente utilizando pandas.

7. Estadísticas Básicas con NumPy y Pandas

Proveer ejemplos de cómo realizar análisis estadísticos básicos, como:

Media

Mediana

Desviación estándar

Correlaciones

8. Consejos y Buenas Prácticas

Tips útiles para trabajar con grandes volúmenes de datos y optimizar el rendimiento de los modelos de machine learning:

Uso de chunking en pandas para leer archivos grandes.

Evitar usar bucles for y usar operaciones vectorizadas en NumPy.

Preprocesamiento eficiente de datos con Polars.

Referencias bibliográficas

Bahit, E. U. G. E. N. I. A. (2018). Introduccion al lenguaje python. *Libro de estudio del curso de la Escuela de Informática Eugenia Bahit*.

Marzal, A., & Gracia, I. (2020). *Introducción a la programación con Python*. Universitat Jaume I..

Torres, J. (2020). *Python deep learning: Introducción práctica con Keras y TensorFlow 2*. Alpha Editorial.

Varó, A. M., Luengo, I. G., & Sevilla, P. G. (2014). *Introducción a la programación con Python 3*. Universitat Jaume I. Servei de Comunicació i Publicacions.

Buriticá, O. I., & Guerrero, L. E. M. (2021). *Introducción a la programación con Python*. Ra-Ma Editorial.