



UNIVERSIDAD
TORCUATO DI TELLA

Licenciatura en Tecnología Digital

Tecnología Digital VI: Inteligencia Artificial

Librerías de análisis de datos

Clase práctica 1

Motivación

- ¿Qué herramienta nos conviene usar para **analizar** datos **numéricos**?
- ¿Y para **analizar** fácilmente **cualquier base** de datos?
- ¿Y para **visualizar** los resultados de esos análisis?

Organización

Etapas	Inicio		Duración
	Sección 1	Sección 2	
Introducción	08:00	09:50	05'
NumPy	08:05	09:55	30'
Pandas	08:35	10:25	30'
Visualización	09:05	10:55	30'
Cierre	09:35	11:25	05'
Fin de la clase	09:40	11:30	-

NumPy | Qué

¿Qué es NumPy?

Ante todo, NumPy es la versión corta de **Numerical Python**.

Provee una **interfaz eficiente para almacenar y operar** sobre un espacio de memoria donde se guardan datos de forma temporal.

NumPy | Por qué

¿Por qué seleccionamos NumPy?

Muchas **bases de datos** pueden ser representadas como **matrices de números**:

- Una imagen como una matriz 2D de números representando el brillo del píxel.
- Un sonido como una matriz 1D de intensidad versus tiempo.
- Un texto como dígitos representando la frecuencia de algunas palabras o frases.

Sin importar cuáles son los datos, el primer paso para analizarlos es **transformarlos** en una matriz de números.

En consecuencia, el almacenamiento y la manipulación eficientes de matrices numéricas son absolutamente **fundamentales** para hacer ciencia de datos.

NumPy | Por qué (continuación)

Las matrices de NumPy, si bien son similares al tipo `list`, proveen almacenamiento y operaciones de datos mucho **más eficientes, a medida que las matrices crecen** en tamaño.

Además, las matrices de NumPy son el **núcleo** de casi todas las herramientas de ciencias de datos en Python, entonces aprender a usar NumPy efectivamente es valioso sin importar qué aspecto de la ciencia de datos a uno le interesa más.

NumPy | Cómo

¿Cómo se compone NumPy?

Para conocer algunas de las principales estructuras y operaciones que componen a NumPy, veamos `td6-p01-c-numpy.ipynb`.

NumPy | Dónde

¿Dónde podemos encontrar recursos útiles sobre NumPy?

Primera opción: [documentación oficial](#).

Cheat sheets: [DataCamp](#) e [Intellipaat](#).

Tutoriales: [Real Python](#), [Stanford](#), [W3Schools](#), [Tutorials Point](#) y [GeeksforGeeks](#).

NumPy | Hasta qué

¿Hasta qué punto nos sirve NumPy?

NumPy provee aspectos esenciales **para los datos limpios y bien organizados**, típicamente vistos en tareas de computación **numérica**.

No obstante, sus **limitaciones** se tornan evidentes cuando necesitamos más flexibilidad, como asociar etiquetas a datos o trabajar con datos perdidos, y cuando necesitamos operaciones que no se corresponden bien con *broadcasting* elemento a elemento (la forma en que NumPy lidia con operaciones entre matrices de distinto tamaño), como agrupar o pivotar.

Esas tareas, que resultan difíciles de hacer con NumPy, son muy **importantes para analizar los datos menos estructurados** que nos rodean y ocupan gran parte del tiempo de un científico de datos.

Pandas | Qué

¿Qué es Pandas?

Es una librería más nueva, construida sobre NumPy, que provee una **implementación eficiente de un DataFrame**.

Los **DataFrames** son matrices multidimensionales con **etiquetas** para sus filas y columnas y, usualmente, con tipos **heterogéneos** y datos **faltantes**.

Pandas | Por qué

¿Por qué seleccionamos Pandas?

Ofrece una interfaz de **almacenamiento** conveniente para datos **etiquetados**.

Además, provee **formas eficientes de realizar las tareas** antes mencionadas (asociar etiquetas a datos, trabajar con datos perdidos, agrupar y pivotar).

Asimismo, implementa una gran cantidad de **poderosas operaciones** de datos, **familiares** para los usuarios tanto de herramientas de bases de datos como de programas de hojas de cálculo.

Pandas | Cómo

¿Cómo se compone Pandas?

Para conocer algunas de las principales estructuras y operaciones que componen a Pandas, veamos `td6-p01-e-pandas.ipynb`.

Pandas | Dónde

¿Dónde podemos encontrar recursos útiles sobre Pandas?

Primera opción: [documentación oficial](#).

Cheat sheets: [oficial](#), [DataCamp](#) y [otra de DataCamp](#).

Tutoriales: [W3Schools](#), [InteractiveChaos](#) y [LearnDataSci](#).

Pandas | Hasta qué

¿Hasta qué punto nos sirve Pandas?

Como vimos, Pandas ofrece una interfaz conveniente para **almacenar datos etiquetados** e implementa una gran cantidad de **poterosas operaciones** para realizar sobre los mismos. De hecho, para el día a día, Pandas suele ser una mejor opción que NumPy.

Al momento de mostrar los resultados obtenidos y comunicarlos de forma atractiva a otros, Pandas permite realizar algunas **visualizaciones**. **No obstante**, más de una vez se queda corto, respecto a librerías destinadas exclusivamente a la visualización.

Visualización |Cuál

¿Cuál es la mejor librería para realizar visualizaciones en Python?

No hay consenso. Varias son las candidatas. **Matplotlib**, **seaborn** y **Plotly** son algunas de las más populares.

Visualización |Cuál (continuación)

	Matplotlib	seaborn	Plotly
Pros	<ul style="list-style-type: none">• Independiente del sistema operativo o del formato de salida deseado.• Amplia base de usuarios.• Activa base de desarrolladores.	<ul style="list-style-type: none">• De alto nivel.• Integración con Pandas: usa inteligentemente las etiquetas de un DataFrame.	<ul style="list-style-type: none">• Visualizaciones interactivas por defecto.
Contras	<ul style="list-style-type: none">• De bajo nivel.• No diseñada para ser usada con objetos de tipo DataFrame.• Visualizaciones estáticas por defecto.	<ul style="list-style-type: none">• Visualizaciones estáticas por defecto.	<ul style="list-style-type: none">• Visualizaciones interactivas por defecto (no tan deseado si se quiere tener un mayor control de las conclusiones extraídas).
Modo	Veamos <code>td6-p01-f-matplotlib.ipynb</code> .	Veamos <code>td6-p01-g-seaborn.ipynb</code> .	Veamos <code>td6-p01-h-plotly.ipynb</code> .
Recursos	1º opción: documentación oficial . Cheat sheets: oficiales y DataCamp . Tutoriales: W3Schools , InteractiveChaos , GeeksforGeeks y Aprende con Alf .	1º opción: documentación oficial . Cheat sheets: DataCamp . Tutoriales: InteractiveChaos , GeeksforGeeks , Tutorials Point y EliteDataScience.com .	1º opción: documentación oficial . Cheat sheets: oficial y DataCamp . Tutoriales: oficial , GeeksforGeeks y Tutorials Point .

Visualización | Por último

Tres recursos que suelen ser útiles para la visualización en general:

- [PYTHON CHARTS](#),
- [from Data to Viz](#) y
- [ColorBrewer](#).

Cierre

Hoy vimos, entre otras cuestiones,

- qué herramienta nos conviene usar para **analizar** datos **numéricos**,
- cuál otra, para **analizar** fácilmente **cualquier base** de datos y
- cuáles otras, para **visualizar** los resultados de esos análisis.

Mañana vamos a estar poniendo en práctica todo esto, en el primer **taller**, resolviendo una serie de **ejercicios**.

Pueden darnos *feedback* de la clase [acá](#).