





UNIVERSIDAD
DE SANTIAGO
DE CHILE

Data Science Aplicado a las Finanzas

FACULTAD DE ADMINISTRACIÓN Y ECONOMÍA
MAGISTER EN FINANZAS
(Primer Semestre 2025)

Sesión 1 : Introducción a la Ciencia de Datos Financieros

Andrés C. Medina
Senior Data Scientist Engineer
anmedinas@gmail.com

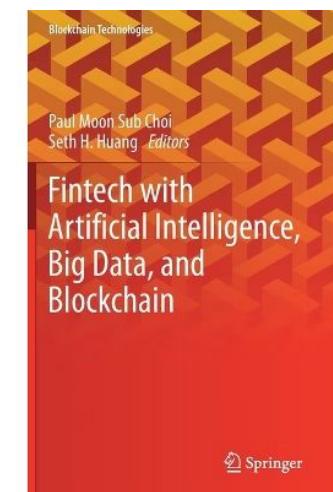
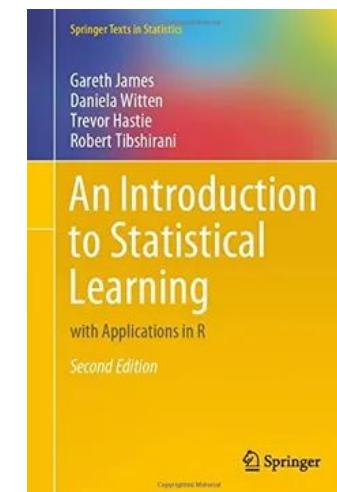
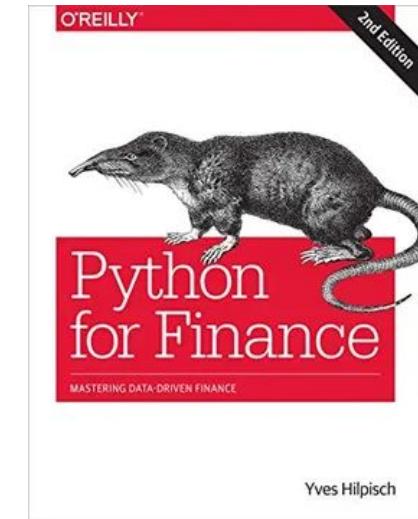
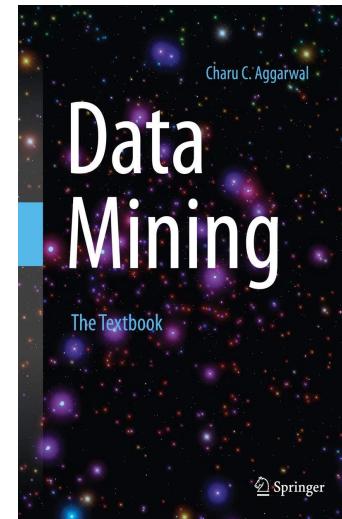
Agenda



Contenidos

Introducción a la Ciencia de Datos Financieros (unidad I)

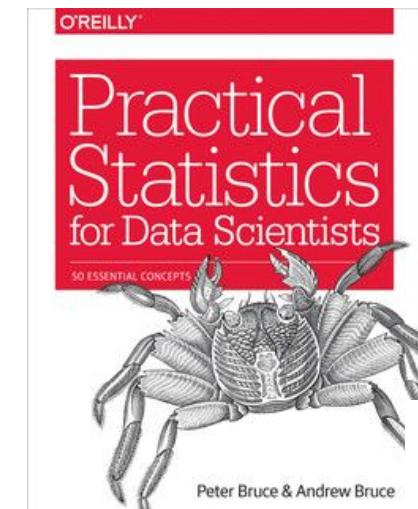
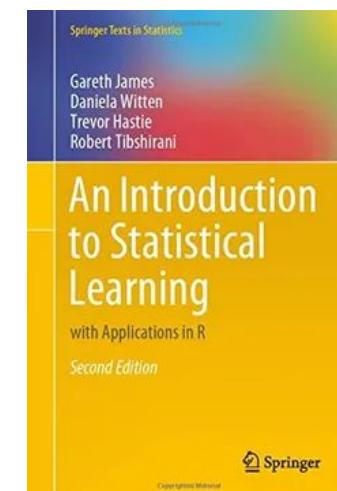
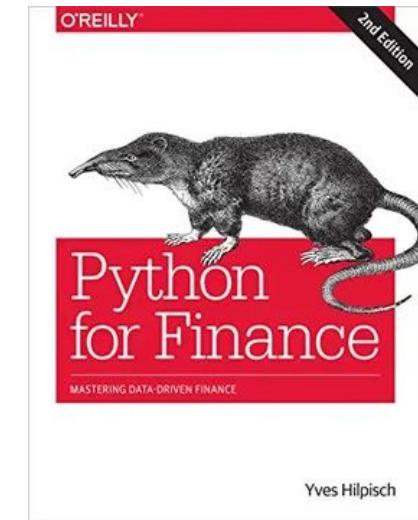
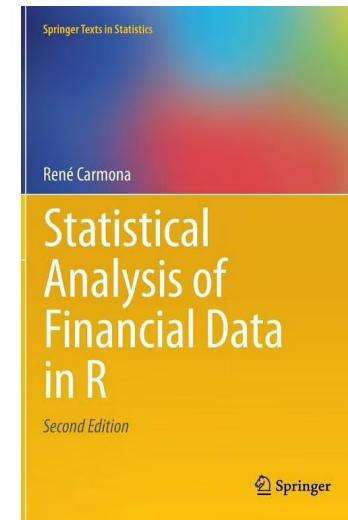
- ¿Qué es **Data Science** ?
- Conceptos Principales
- Proceso de Extracción del Conocimiento
 - Knowledge Discovery Databases (**KDD**)
 - Cross-Industry Standard Process for Data Mining (**CRISP-DM**)
 - Sample, Explore, Modify, Model and Assess (**SEMMA**)
- El valor del analytics en una industria financiera
- ¿Qué es **Data Mining** ?
- Aprendizaje no Supervisado
- Aprendizaje Supervisado
- Introducción a Python
 - ¿Qué es Python ?
 - Ecosistema de Python
 - Principales módulos utilizados
 - Entornos de programación
 - Instalación de anaconda, configuración y comandos principales
 - Instalacion de modulos utilizando pip y conda
 - Introducción a **numpy**, **scipy** y **pandas** .



Contenidos

Análisis Estadístico Aplicado (unidad II)

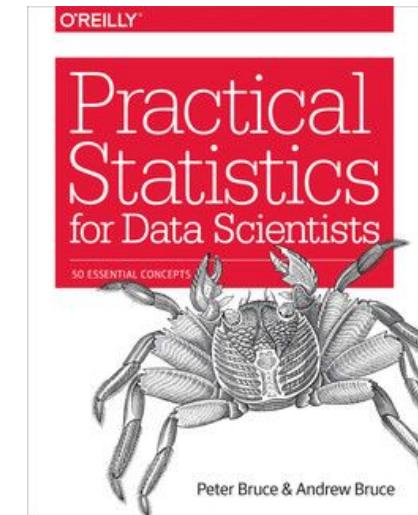
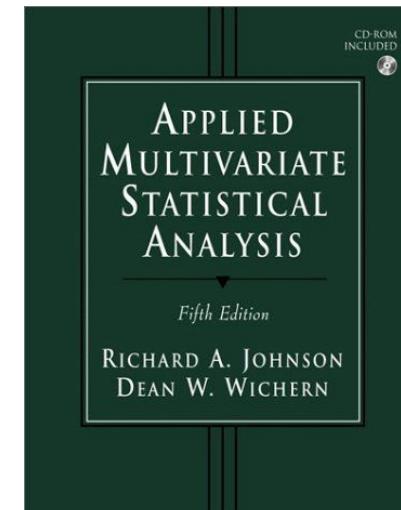
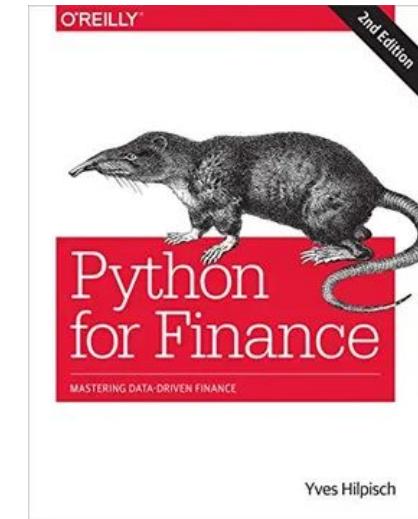
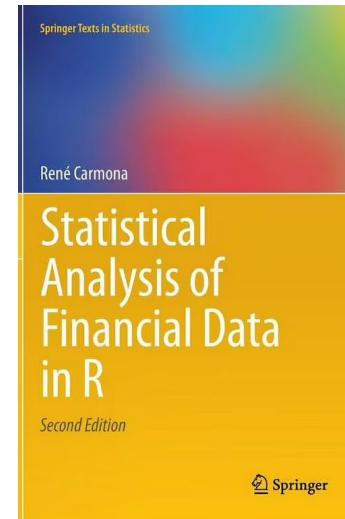
- Introducción al análisis estadístico financiero
 - Tipos de datos en finanzas
 - Exploración de fuentes de datos en Python (**yfinance**, **pandas**, **datareader**)
- Estadística descriptiva con **Python**
 - Medidas de tendencia central, dispersión, sesgo y curtosis.
- Visualización estadística
 - Histogramas, boxplots, KDE, series de tiempo, heatmaps de correlación entre activos.
- Transformación de datos financieros
 - Cálculo de retornos simples y logarítmicos, estandarización y normalización.
- Análisis de relaciones entre variables
 - Correlación de Pearson, Spearman y Kendall
 - Covarianza y varianza conjunta
- Comparación estadística de activos
 - Diferencia de medias y varianza entre grupos
 - Criterios para identificar activos líderes o estables.



Contenidos

Probabilidad y Distribuciones Aplicadas (unidad III)

- Fundamentos de Probabilidad
 - Espacio muestral y eventos, probabilidad conjunta, condicional e independencia, teorema de bayes.
- Variables aleatorias y funciones de probabilidad
 - Variables aleatorias discretas y continuas, Función de masa de probabilidad (**pmf**), densidad (**pdf**) y acumulada (**cdf**), momentos muestrales.
- Modelado de retornos financieros como variables aleatorias
 - Simulación de retornos.
- Distribuciones comunes en finanzas
 - Distribucion normal, t-student, exponencial, chi-cuadrado, log-normal y su uso en precios de activos.
- Ajuste de distribuciones a datos reales
 - Estimacion de parametros mediante (MLE), bondad de ajuste, gráficos QQ-PP
- Aplicaciones financieras
 - Value at Risk, Eventos extremos, colas pesadas.

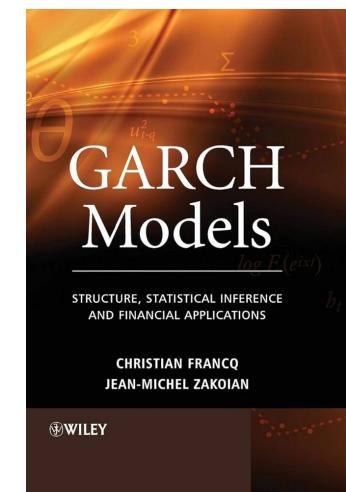
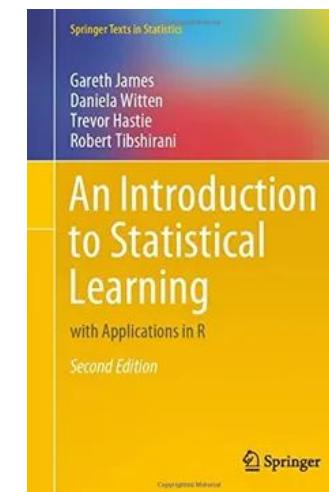
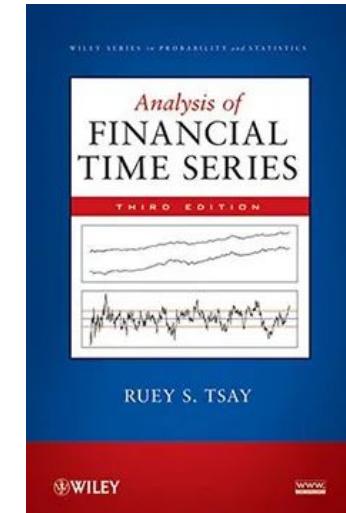
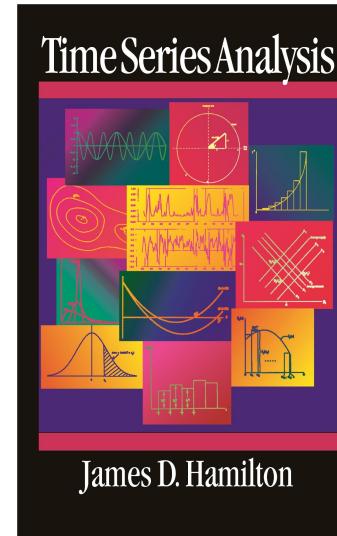




Contenidos

Series Temporales y Modelos Predictivos (unidad IV)

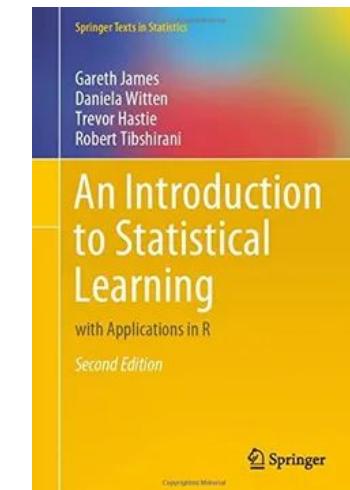
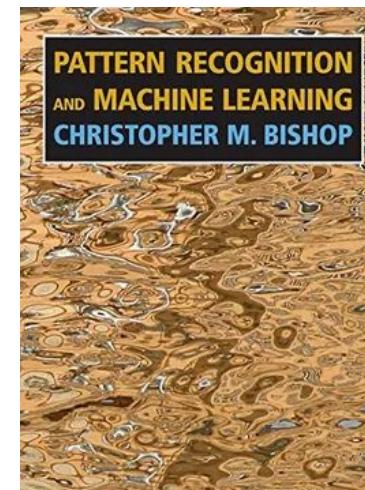
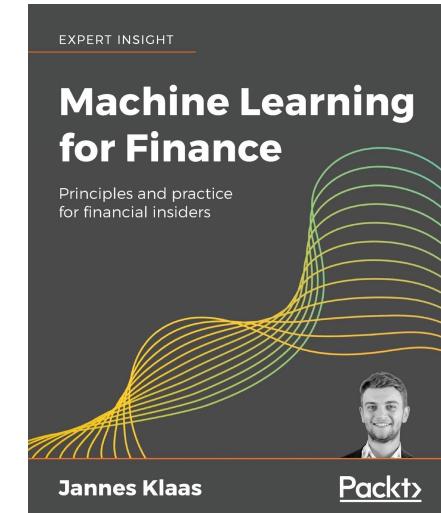
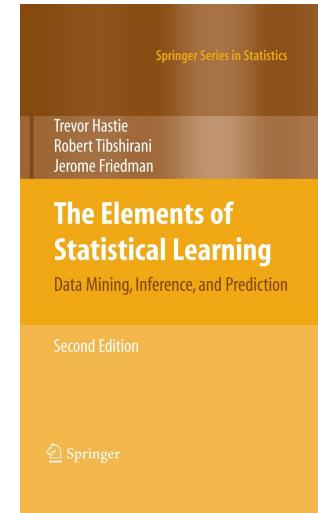
- Simulación Estadística
 - Introducción a la simulación de Monte Carlo
 - Modelos de Distribución Log-normal
 - Simulación de precios de activos
 - Proceso Wiener
 - Movimiento Browniano Geométrico
 - Simulación de precios de Opciones
- Series de Tiempo
 - Modelos Estacionarios
 - **White Noise , AR(p), MA(q) , ARMA(p,q) , ARIMA(p,d,q)**
 - Modelos de Volatilidad Estocástica
 - Hechos estilizados de series financieras
 - Fat Tails y no normalidad de retornos.
 - Leptokurtosis
 - Clustering
 - Asimetría de Volatilidades
 - otros
 - Modelos **ARCH(p) , GARCH(p, q) , EGARCH(p, q) y GJR-GARCH(p, q)**
 - Forecast



Contenidos

Introducción a Machine Learning en Finanzas (unidad V)

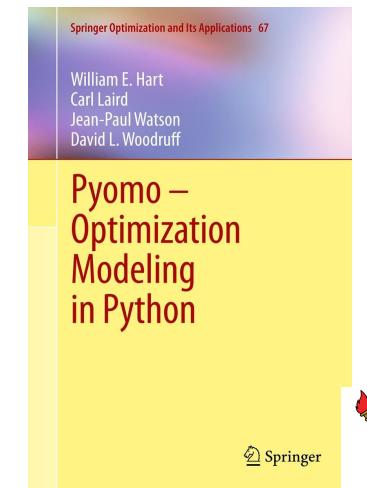
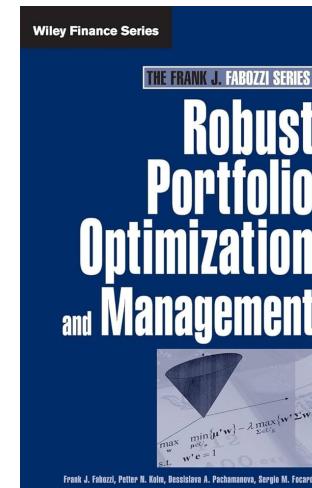
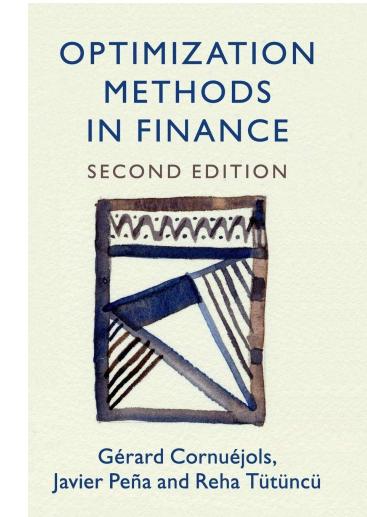
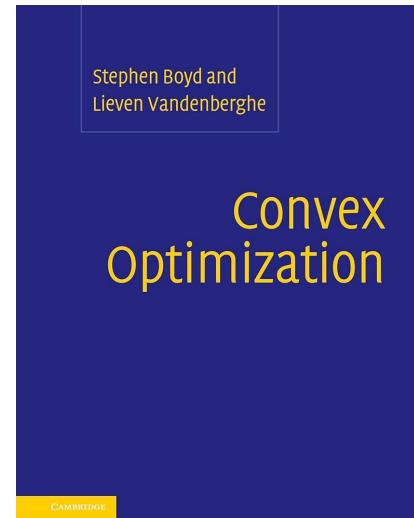
- Fundamentos de Machine Learning
 - Definición de ML : ¿Que lo diferencia de la estadística tradicional ?, aprendizaje supervisado vs no supervisado, ciclos de vida de proyectos en ML (elementos de MLOps)
- Preprocesamiento de datos financieros
 - Limpieza y transformación de datos temporales y transaccionales
 - Escalamiento y codificación
 - Feature Engineering
- Aprendizaje Supervisado
 - **Regresión lineal y regularizada** (Ridge, Lasso), **logistica** , **decision tree** , **random forest** y **boosting** .
- Métricas de evaluación
 - **accuracy** , **precision** , **recall** , **f1** , **ROC-AUC** para clasificación.
 - **MAE**,**MSE**, **RMSE** y **R^2** para regresión
- Aprendizaje No Supervisado
 - Hard Clustering : **KMeans** , **DBScan** y **Cluster Jerárquico**
 - Soft Clustering : **GMM** , **Fuzzy-Mean**



Contenidos

Optimización de Portafolios con Data Science (unidad VI)

- Introducción a la teoría moderna de portafolios (Markowitz)
 - Supuestos del modelo, retorno esperado, riesgo covarianza y frontera eficiente.
- Construcción de un portafolio
 - Procesamiento de datos con **yfinance**, cálculo de retornos y matriz de covarianzas
 - Simulación de montecarlo
- Visualización y análisis de portafolios simulados
 - Frontera eficiente, maximización ratio de Sharpe, identificación portfolio mínimo riesgo.
- Optimización con restricciones
 - restricciones típicas, introducción a optimización convexa , usos de **cvxpy** y **pyomo**
- Métricas financieras adicionales
 - Ratio de Sharpe, Sortino, Tracking Error
 - CVaR, VaR empíricos
- Optimización Robusta en Finanzas
 - Incertidumbre en los inputs , worst case, diferencia entre optimización estocástica y robusta, etc ...

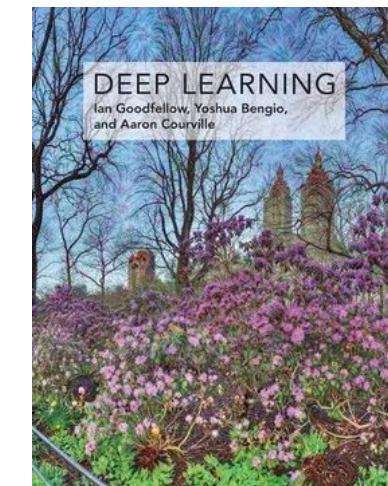
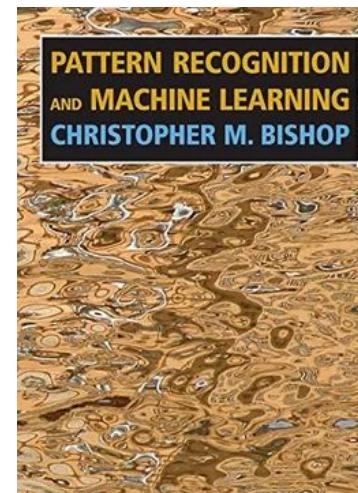
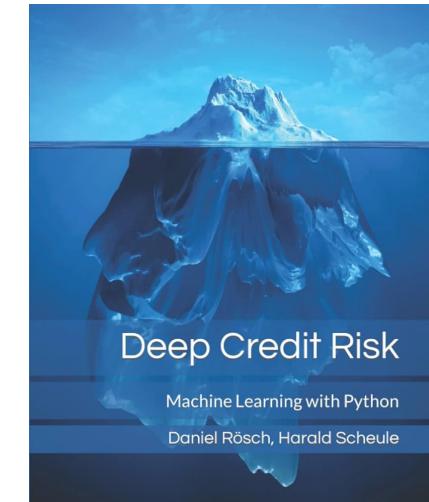
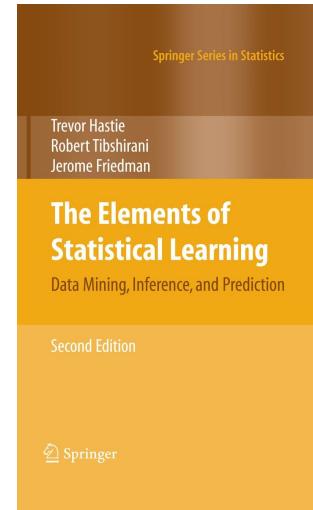




Contenidos

Técnicas Avanzadas de Machine Learning (unidad VII)

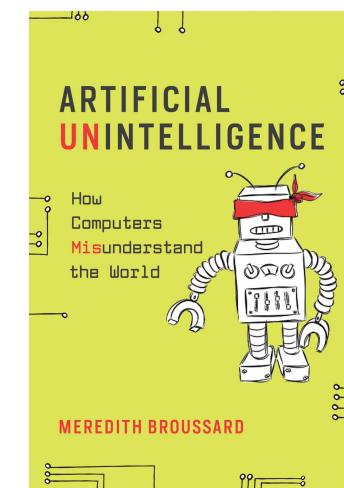
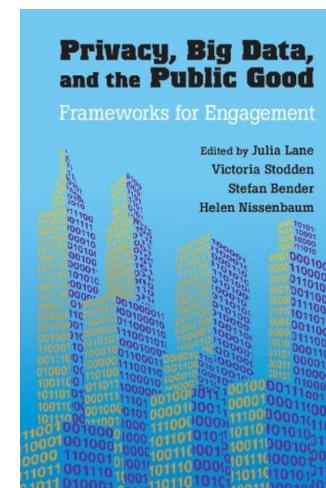
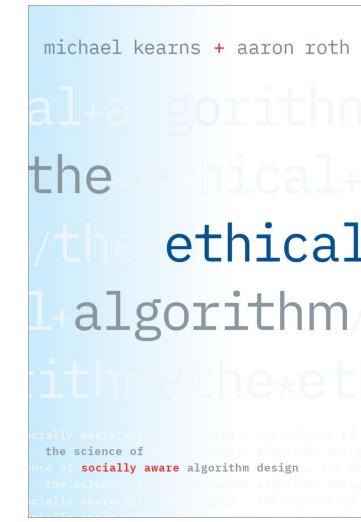
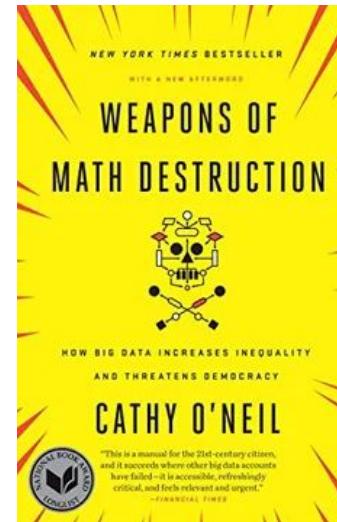
- Introducción a modelos complejos
 - Overfitting capacidad del modelo y regularización, ensambles y boosting.
- Redes Neuronales con [Lightning Pytorch](#)
 - Modelo de Perceptron, arquitectura basica.
- Entrenamiento , validación y evaluación
 - Batching, Forward and Backward pass
 - Métricas y visualización de curvas de aprendizaje.
- Interpretabilidad de redes neuronales
 - Feature Importance, [SHAP value](#)
- Modelos secuenciales en Lightning Pytorch
 - Modelos de RNN, LSTM y estructura de datos para series temporales
- Integración con pipelines de Machine Learning
 - integración de modelos en Lightning Pytorch con sklearn (wrappers)
 - Control de versiones, reproducibilidad y exportacion de modelos con torch.save y [ONNX](#).



Contenidos

Ética en Ciencia de Datos (unidad VIII)

- Introducción a la ética en ciencia de datos
- Principios éticos claves
- Privacidad y preservación de la confidencialidad
- Normativas y estándares relevantes
- Ética en el uso de modelos predictivos
- Transparencia en modelos y reportes
- Gobernanza de datos y modelos



Evaluación



Evaluaciones

- Unidad I : Introducción a la Ciencia de Datos (**HW01**)
- Unidad II : Análisis Estadístico Aplicado (**HW02**)
- Unidad III : Probabilidad y Distribuciones Aplicadas (**HW03**)
- Unidad IV : Series temporales y modelos predictivos (**HW04**)
- Unidad V : Introducción a Machine Learning en Finanzas (**HW05**)
- Unidad VI : Optimización de Portafolios en Finanzas (**HW06**)
- Unidad VII : Técnicas Avanzadas de Machine Learning (**HW07**)
- Unidad VIII : Ética en Ciencia de Datos (**HW08**)

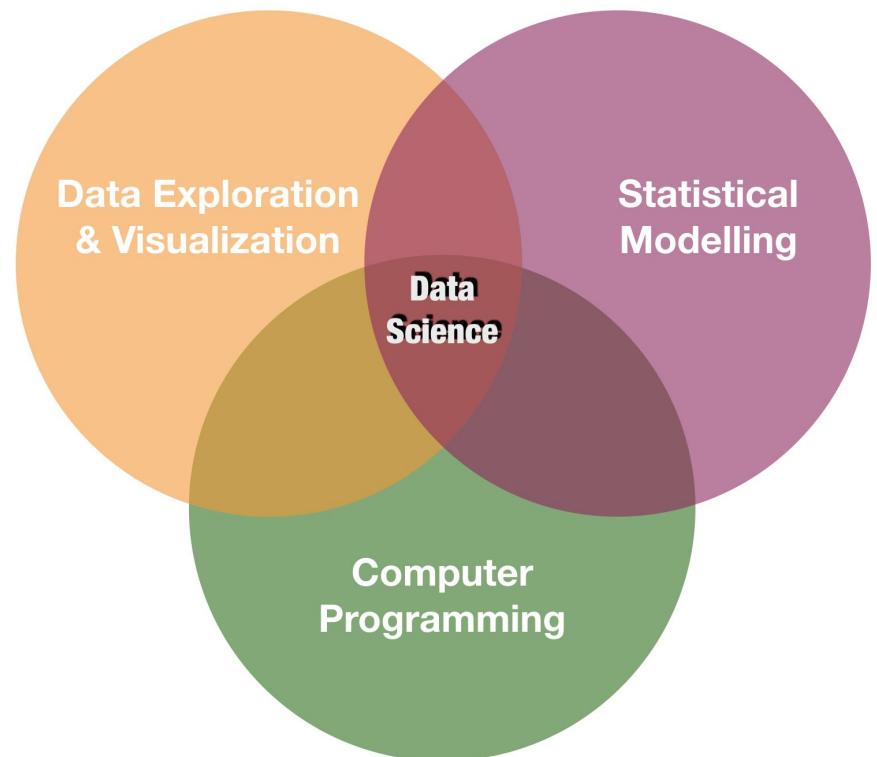
Nota Final : Promedio Simple de todas las actividades

Introducción a la Ciencia de Datos Financieros



¿Qué es Data Science ?

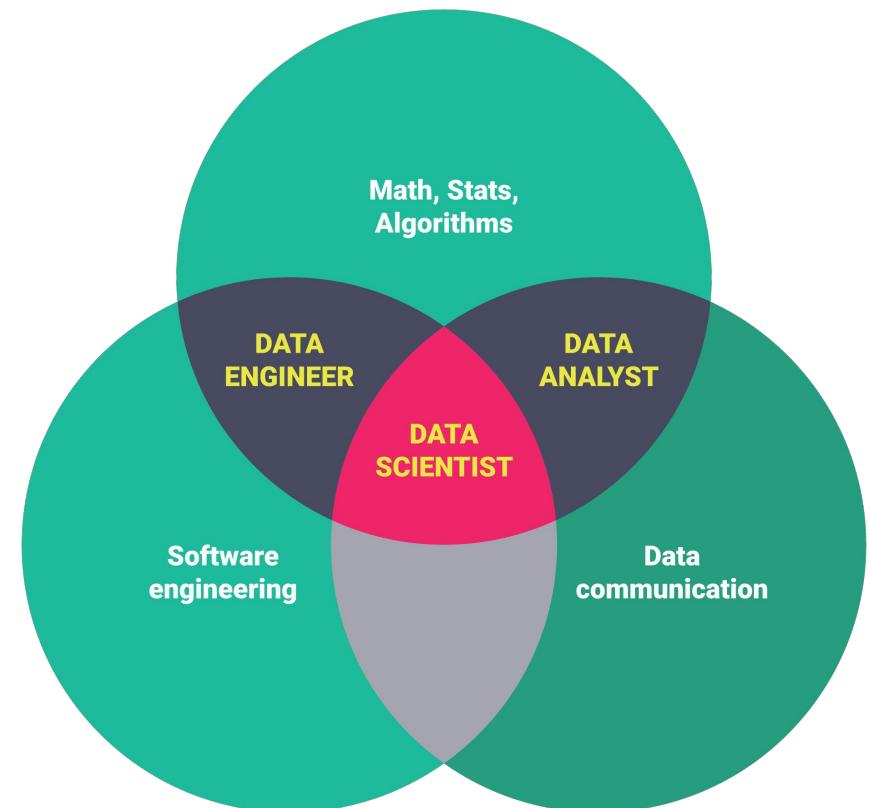
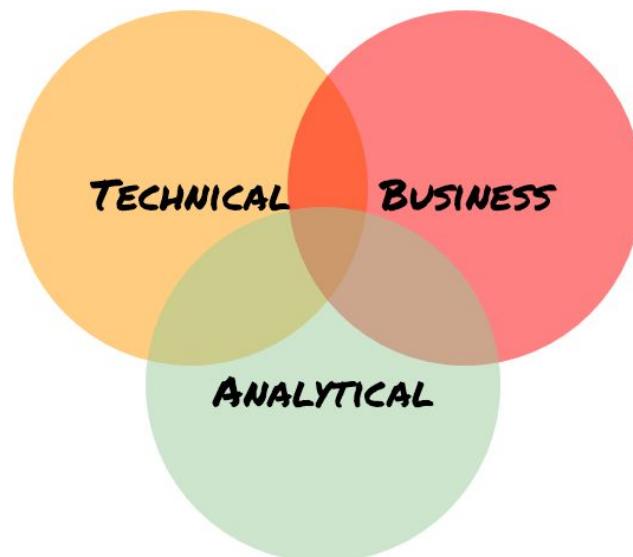
- La ciencia de datos es el estudio de datos con el fin de **extraer información significativa**.
- Es un enfoque multidisciplinario que combina principios y prácticas del campo de las matemáticas, la estadística, la inteligencia artificial y la ingeniería de computación para analizar grandes cantidades de datos.
- Este análisis permite que los científicos de datos planteen y respondan a preguntas como “qué pasó”, “por qué pasó”, “qué pasará” y “qué se puede hacer con los resultados”.
- La ciencia de datos es importante porque combina herramientas, métodos y tecnología para generar significado a partir de los datos.
- Aunque el término ciencia de datos no es nuevo, sus significados y connotaciones cambiaron con el tiempo. La palabra apareció por primera vez en los años 60 como nombre alternativo de la estadística. A finales de los 90, los profesionales de la computación formalizaron el término. Una propuesta de definición la consideraba un campo independiente con tres aspectos: diseño, recopilación y análisis de datos.





Conceptos Principales

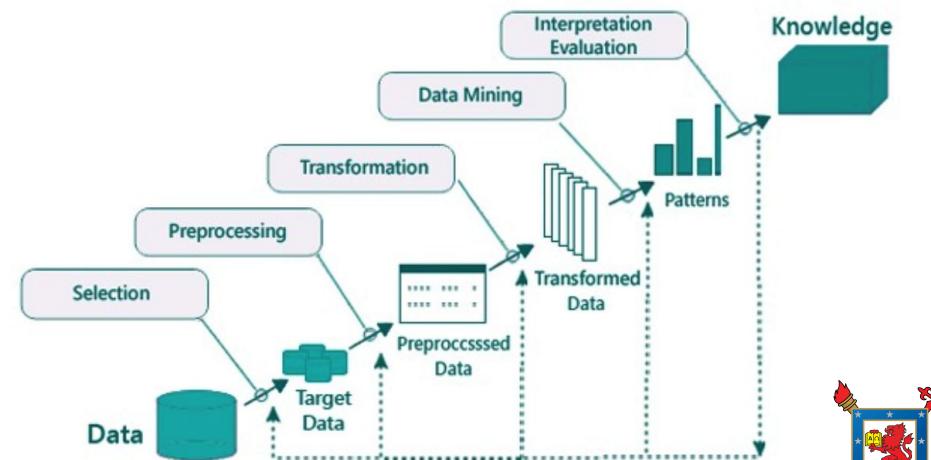
- **Business Analyst** Enfocado en el análisis de optimización de procesos de negocio utilizando datos, con fuerte énfasis en comunicación y la gestión de requerimientos.
- **Data Scientist** Enfocado en el análisis avanzado de datos y construcción de modelos predictivos con habilidades en estadística y machine learning
- **Data Engineer** Enfocado en la construcción y mantenimiento de la infraestructura de datos, con habilidades en ingeniería de software y manejo de base de datos.





Extracción del Conocimiento

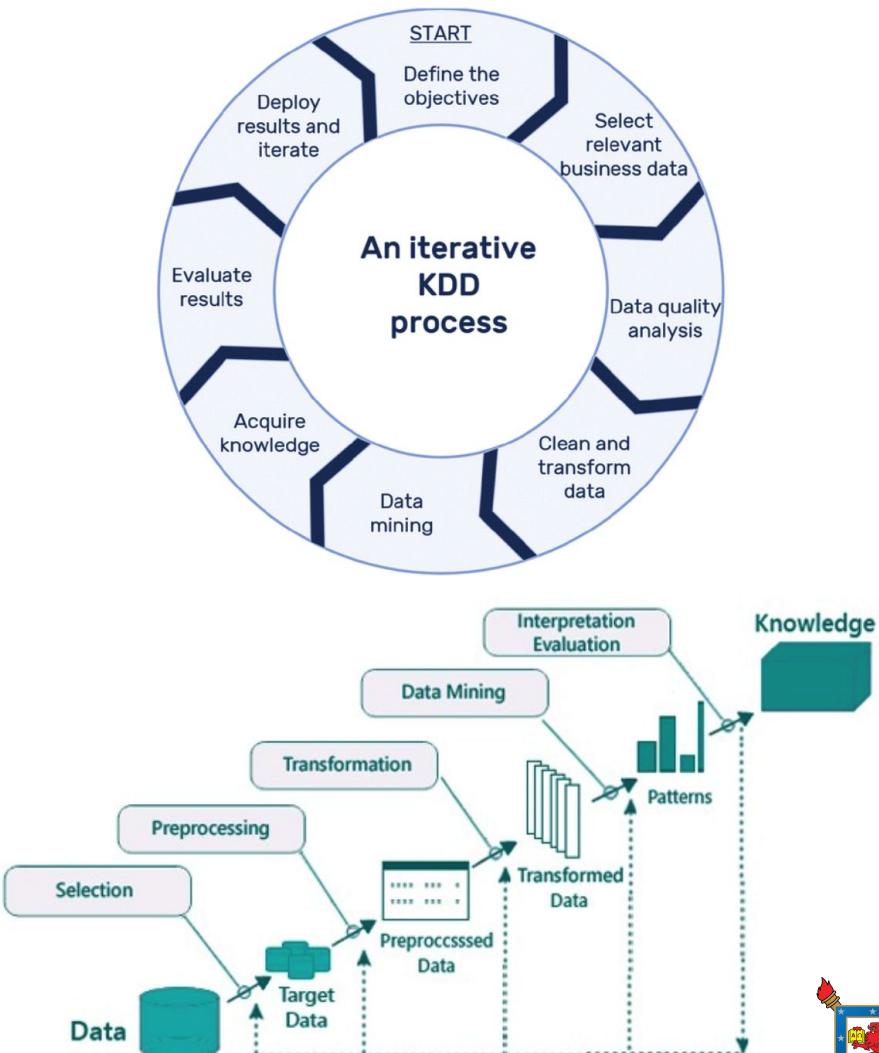
- El proceso **KDD** (Fayyad et al, 1996) es una metodología estructurada para descubrir patrones útiles y conocimiento a partir de grandes conjuntos de datos.
- **Paso 1. Selección**
 - Se recopila e identifican datos relevantes que se van a utilizar.
 - Identificación de Fuentes de datos
 - Extracción de datos
 - Definición de subconjunto de datos relevante para el análisis, basado en criterios específicos.
- **Paso 2. Preprocesamiento**
 - Paso clave para garantizar la calidad de los datos.
 - Limpieza de datos
 - Transformación de datos
 - Reducción de datos
- **Paso 3. Transformación**
 - Implica conversión de los datos en un formato adecuado para el análisis.
 - Generación de nuevas variables
 - Reducción de dimensionalidad
 - Discretización





Extracción del Conocimiento

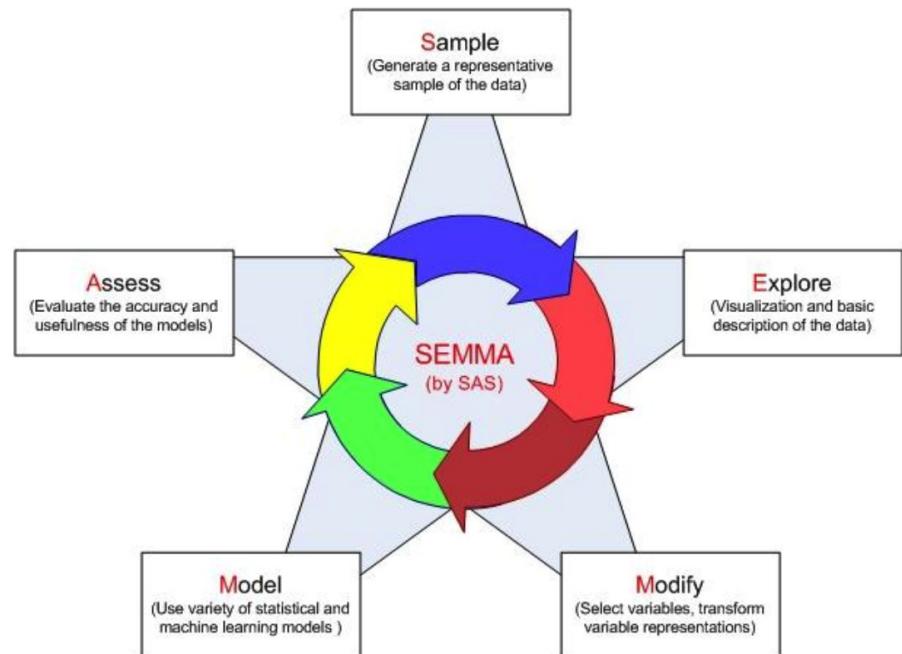
- El proceso **KDD** (Fayyad et al, 1996) es una metodología estructurada para descubrir patrones útiles y conocimiento a partir de grandes conjuntos de datos.
- **Paso 4. Datamining**
 - En esta etapa se aplican los algoritmos de machine learning para extraer patrones.
 - **Técnicas de clasificación**
 - **Regresión**
 - **Clustering**
 - **Reglas de Asociación**
- **Paso 5. Evaluación**
 - En esta etapa se interpretan y evalúan los patrones descubiertos para determinar su validez y utilidad.
 - **Evaluación de modelos**
 - **Interpretación de resultados**
 - **Validacion**





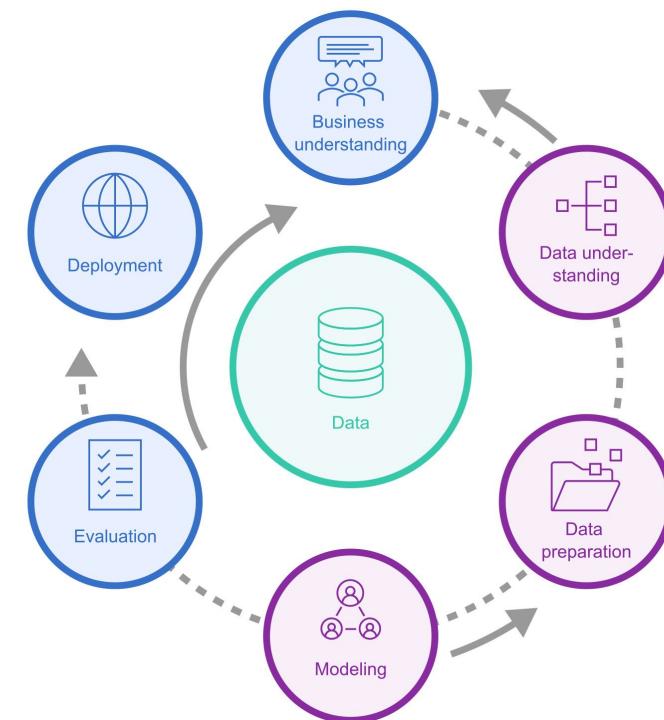
Extracción del Conocimiento

- **SEMMA** es una metodología desarrollada por SAS Institute para guiar el proceso técnico de minería de datos mediante su herramienta SAS Enterprise Miner.
- Su objetivo es construir modelos predictivos sólidos a través de una secuencia de pasos orientados a datos.
- **Paso 1. Sample** Seleccionar una muestra de datos suficiente para obtener información relevante y que permita su rápida manipulación
- **Paso 2. Explore** Explorar para buscar tendencias para comprender los datos.
- **Paso 3. Modify** Limpieza y transformación de datos para el modelado
- **Paso 4. Model** Aplicación de técnicas de modelamiento estadístico machine learning
- **Paso 5. Asses** Evaluación de los modelos en términos de su rendimiento. Se selecciona el modelo que mejor performance tenga según las métricas elegidas.



Extracción del Conocimiento

- El proceso **CRISP-DM** es una metodología estándar para llevar a cabo proyectos de minería de datos, desarrollada para ser aplicada en distintas industrias.
- Define un proceso estructurado y repetible para convertir problemas de negocio en soluciones analíticas útiles.
- **Paso 1. Entendimiento del Negocio** comprender y entender los requisitos y objetivos del proyecto
- **Paso 2. Entendimiento de los Datos** Obtener una comprensión preliminar de los datos
- **Paso 3. Preparación de los Datos** Incluye todas las actividades necesarias para construir el conjunto final a partir de los datos duros.
- **Paso 4. Modelado** Se seleccionan y aplican varias técnicas de modelamiento
- **Paso 5. Evaluación** Se evalúan diferentes y amplia gama de modelos tanto estadísticos como de aprendizaje de máquina (machine learning) y se revisa si cumplen con los objetivos del negocio.

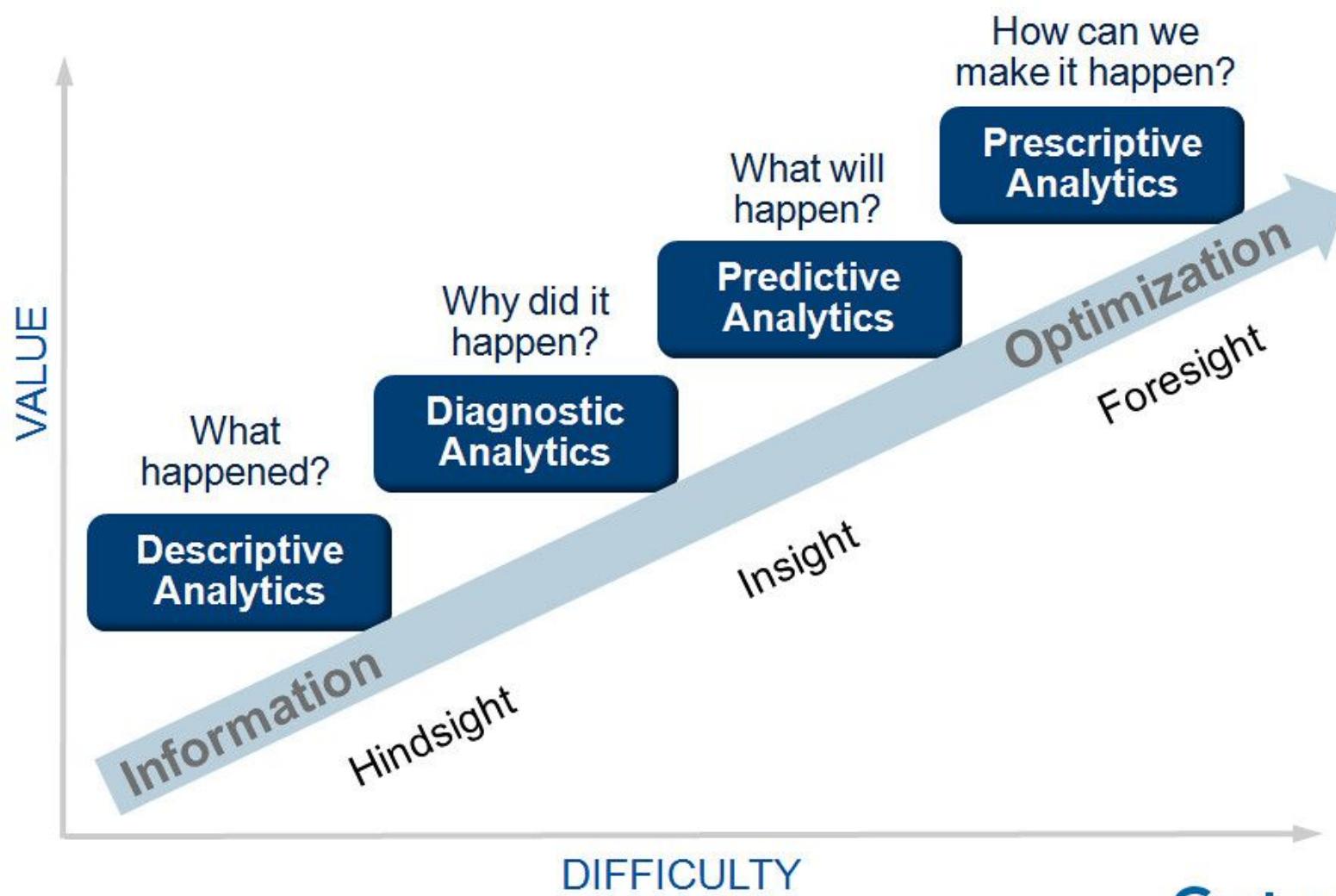


- **Paso 6. Deployment** Implementación del modelo en entornos de producción.



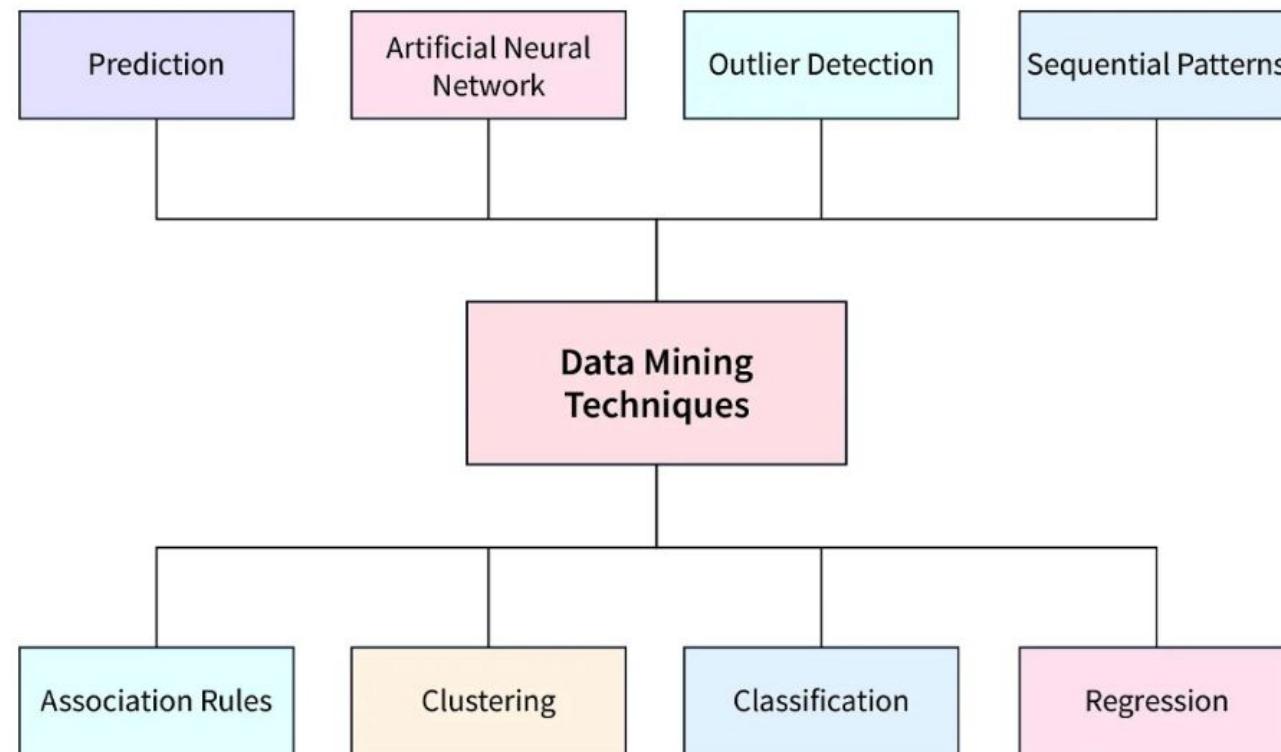
Gartner Map

Analytic Value Escalator



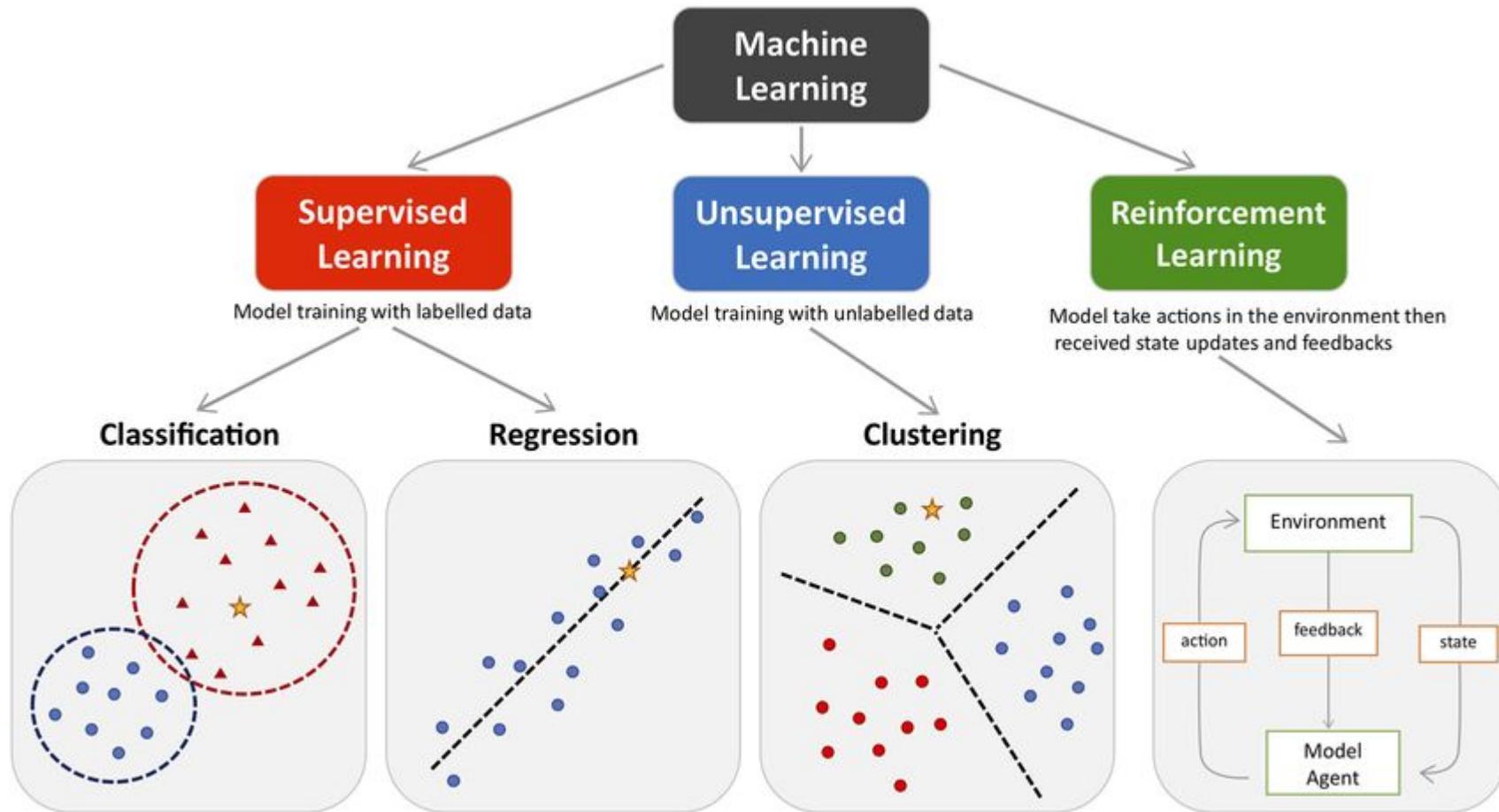


Data Mining



El **objetivo** principal del **Data Mining** es identificar patrones previamente desconocidos y extraer información útil que pueda ser utilizada para la toma de decisiones.

Machine Learning



El **objetivo** principal del **Machine learning** es construir modelos predictivos o de clasificación que puedan generalizar a datos nuevos o desconocidos.

Introducción Python



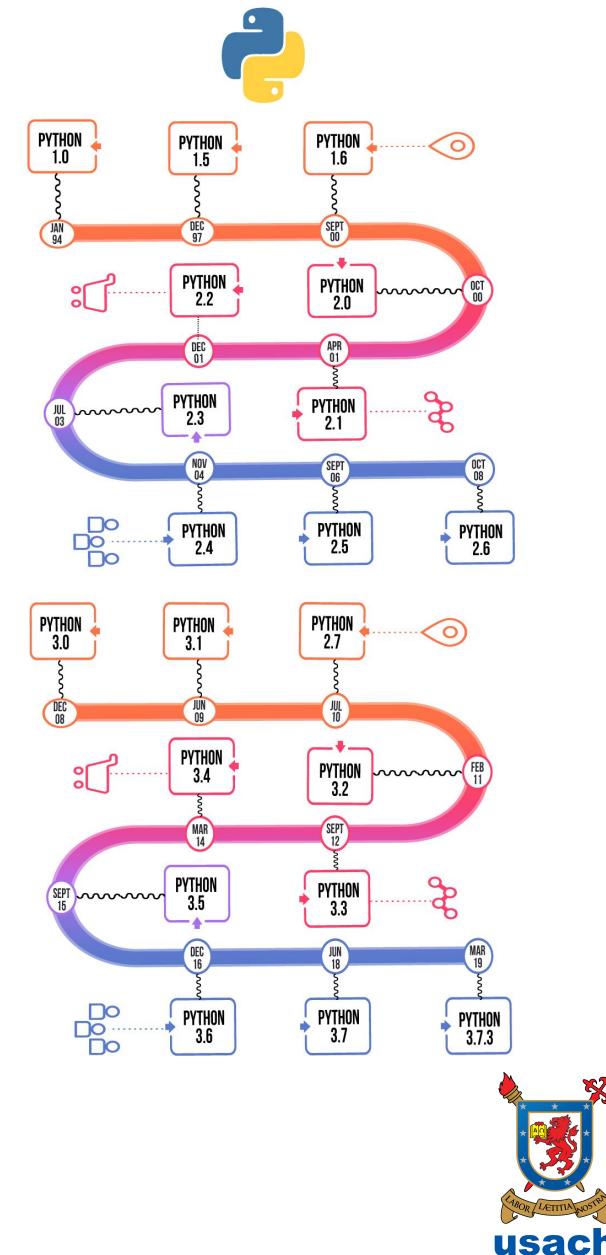
Introducción a Python

- ❑ Python es un lenguaje que data de los años 1990s, y su creación se atribuye al neerlandés Guido van Rossum.
- ❑ Su última versión es **Python 3.12**, sin embargo se recomienda utilizar versiones superiores a 3.9 ya que versiones anteriores ya no tienen soporte oficial.



Página Oficial <https://www.python.org/downloads/>

En resumen ... Python es un lenguaje de programación interpretado cuya principal filosofía es que sea legible por cualquier persona con conocimientos básicos de programación ...



Introducción a Python



Top 20 Data Science Tools in 2024

- | | | | |
|----|-----------------------------|----|-----------------|
| 01 | Python Programming Language | 01 | TensorFlow |
| 02 | R Programming Language | 02 | Scikit-learn |
| 03 | NumPy | 03 | SQL |
| 04 | Seaborn | 04 | MySQL |
| 05 | Pandas | 05 | MongoDB |
| 06 | Jupyter Notebooks | 06 | Microsoft Excel |
| 07 | R Studio | 07 | Tableau |
| 08 | Apache Spark | 08 | Power BI |
| 09 | Hadoop | 09 | IBM SPSS |
| 10 | Hugging Face | 10 | SAS |

Fuente : <https://www.geeksforgeeks.org/top-20-data-science-tools-in-2024/>

Introducción a Python

¿ Porque ha ganado tanta popularidad ?

- Es el lenguaje más consultado en Stackoverflow.
- Es el lenguaje de referencia en Data Science, Machine Learning e Inteligencia Artificial
- Al ser el lenguaje de referencia, existe gran cantidad de documentación e información de solución de bugs, por tanto la curva de aprendizaje es menor.
- **Python** es un lenguaje de código abierto.
- Lenguaje Multiplataforma
- Al ser un lenguaje versátil, legible y multiplataforma, permite adaptar procesos complejos de finanzas que tradicionalmente eran complejos.
 - Aplicaciones Bancarias
 - Generación de pronósticos económicos
 - Big Data Financiero
 - Portfolio Management, etc



The infographic is titled "WHAT IS PYTHON?" and features a large Python logo icon. It lists several points about Python's popularity and applications:

- ▶ A back end programming language
- ▶ High-level & approachable for beginners
- ▶ Has a welcoming & established community

Used for tasks like:

- Web Development
- Scripting
- Web Scraping
- Data Analysis
- Automation

Used by companies like:

- Google
- Pinterest
- Spotify
- Dropbox

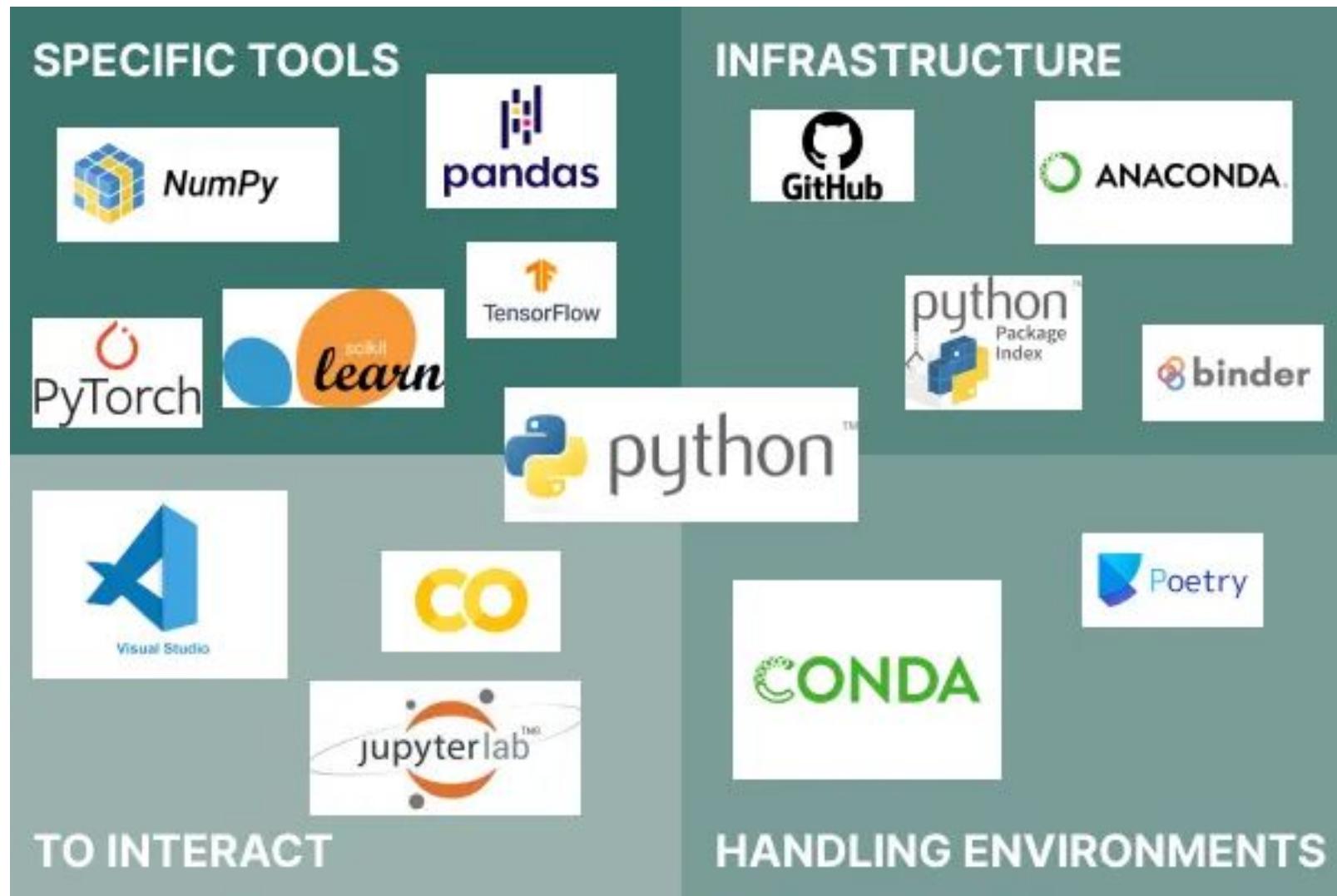
Used with frameworks like:

- django
- Flask
- Jupyter

COURSE REPORT + HACKBRIGHT ACADEMY



Ecosistema de Python





Ecosistema de Python

Procesamiento



NumPy



SciPy



pandas



Paquetes
Python DS

Modelos ML-AI



python™

Visualización



Minería



Scrapy

BeautifulSoup

Módulos de Python



FACULTAD DE
ADMINISTRACIÓN
Y ECONOMÍA

Módulos Esenciales

- Numpy (Análisis Matricial)**
- Pandas (Procesamiento de Datos)**
- Matplotlib (Visualización)**
- Seaborn (Visualización)**
- Statsmodels (Econometría)**
- Scipy (Cálculo Científico)**

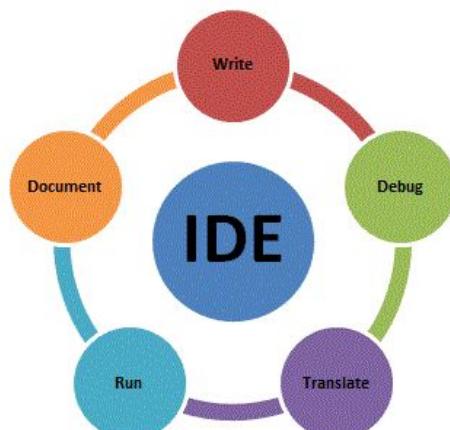
Módulos en Finanzas

- yfinance (Datos Financieros)**
- pyfolio (Reporte y Riesgos)**
- cvxpy (Portfolio Optimization)**
- arch (Volatilidad Estocástica)**

Entornos de Programación



- ❑ En palabras simples, un entorno de programación es un programa que nos ayuda a programar (a veces es conocido como **IDE**)
- ❑ Un entorno de desarrollo integrado (**IDE**) es un paquete de software que combina todas las herramientas de desarrollo en una única interfaz gráfica de usuario (**GUI**)
- ❑ **GUI** (Graphical User Interface) es una interfaz entre la persona y la máquina con el objetivo de representar el código del *backend* de un sistema de la forma más clara posible para el usuario.



```
automation_GUI.py
1 import pyautogui
2
3 pyautogui.moveTo(519, 1060, duration=1)
4 pyautogui.click()
5 pyautogui.moveTo(1717, 352, duration=1)
6 pyautogui.click()
7
8
```





Introducción a Anaconda

- ❑ **Anaconda Distribution** es una *Suite* de código abierto que abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la Ciencia de datos con Python.
- ❑ En líneas generales **Anaconda Distribution** es una distribución de Python que funciona como un gestor de entorno, un gestor de paquetes y que posee una colección de más de 720 paquetes de código abierto.
- ❑ **Anaconda Distribution** se agrupa en cuatro sectores o soluciones tecnológicas:
 - ❑ Anaconda Navigator
 - ❑ Anaconda Project
 - ❑ Librerías de Ciencia de datos
 - ❑ Conda
- ❑ Descargar Anaconda [aquí](#)
- ❑ Tutorial sencillo de instalación [aquí](#)

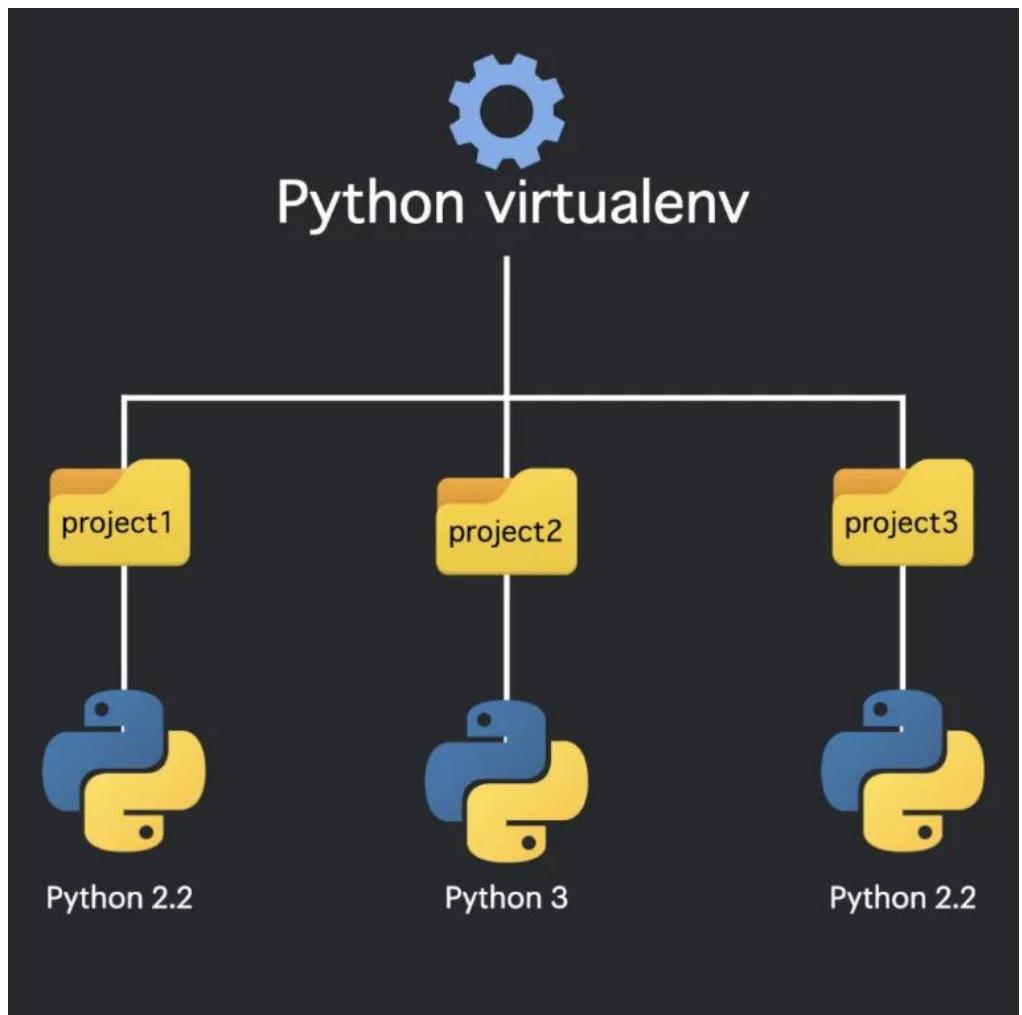
The screenshot displays the Anaconda Distribution homepage. It features four main sections: 1) **ANACONDA NAVIGATOR** (Desktop Portal to Data Science), 2) **ANACONDA PROJECT** (Portable Data Science Encapsulation), 3) **DATA SCIENCE LIBRARIES** (listing various tools like Jupyter, Spyder, NumPy, SciPy, Numba, pandas, DASK, Bokeh, Holoviews, Matplotlib, TensorFlow, and Theano), and 4) **CONDA** (Data Science Package & Environment Manager).

The screenshot shows the Anaconda Navigator application window. On the left is a sidebar with links to Home, Environments, Projects (beta), Learning, and Community. The main area is a grid of cards representing different scientific communities and projects, such as SciPy Conferences, ANACONDA Forum, Stack Overflow: Python, Conda Forum, Bokeh Forum, Blaze Dev Forum, Numba Dev Forum, Matplotlib Forum, NumPy and SciPy Project Mailing Lists, Continuum on Twitter, and Engage.



Environments

- ❑ Un **Entorno Virtual** es un entorno Python en el que el intérprete, las bibliotecas y los scripts instalados en él están aislados de los instalados en otros entornos virtuales, y (por defecto) cualquier biblioteca instalada en un «sistema» Python, es decir, uno que esté instalado como parte de tu sistema operativo.
- ❑ En palabra sencillas, un entorno virtual es un espacio independiente a la instalación local, con el objetivo de aislar los recursos y librerías. Gracias a este concepto podemos tener distintos entornos virtuales con diferentes versiones de Python o de una librería concreta.
- ❑ ¿Porque hacer esto?
 - ❑ Organización
 - ❑ Estabilidad
 - ❑ Diferencia de versiones en Python
 - ❑ Pensar más allá de la caja





Instrucciones Conda

```
# Gestión de Entornos

# Crear un entorno
conda create --name nombre_entorno

# Activar un entorno
conda activate nombre_entorno

# Desactivar un entorno
conda deactivate

# Eliminar un entorno
conda remove --name nombre_entorno --all

# Listar entornos disponibles
conda env list
```



Instrucciones Conda



```
# Gestión de Paquetes

# Instalar un paquete
conda install nombre_paquete

# Instalar un paquete desde canal específico
conda install -c conda-forge nombre_paquete

# Actualizar un paquete
conda update nombre_paquete

# Eliminar paquete
conda remove nombre_paquete

# Listar todos los paquetes instalados en el entorno activo
conda list
```



Instrucciones Conda

```
● ● ●

# Actualizar Conda
conda update conda

# Ver la versión de conda instalada
conda --version

# Obtener ayuda comando especial
conda help nombre_comando

# Clonar entornos
conda create --name nuevo_entorno --clone nombre_entorno_existente

# Crear un entorno con una versión específica de Python
conda create --name mi_entorno python=3.8
```



¿Qué es pip ?

Los paquetes de software son conjuntos de módulos y bibliotecas que pueden ser reutilizados en diferentes proyectos. Los paquetes pueden proporcionar funcionalidades que van desde operaciones matemáticas hasta acceso a bases de datos y análisis de datos, entre otros.

La ventaja de usar un gestor de paquetes como **pip** es que simplifica el proceso de instalación y actualización de paquetes, así como la gestión de dependencias entre ellos



```
● ● ●

# Instalacion de Paquete
pip install nombre_paquete

# Instalar version especifica de un paquete
pip install nombre_paquete==X.Y.Z

# Instalar multiples paquetes desde un archivo
pip install -r requirements.txt

# Actualizar un paquete
pip install --upgrade nombre_paquete

# Actualizar pip
pip install --upgrade pip

# Desinstalar un paquete
pip uninstall nombre_paquete

# Listar todos los paquetes
pip list

# Mostar informacion sobre paquete especifico
pip show nombre_paquete

# Verificar si existen actualizaciones disponibles
pip list --outdated

# Exportar listado de paquetes a archivo texto
pip freeze > requirements.txt

# Verificacion instalacion correcta
pip check
```

Jupyter

Jupyter Notebook es una aplicación interactiva basada en web que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto explicativo.

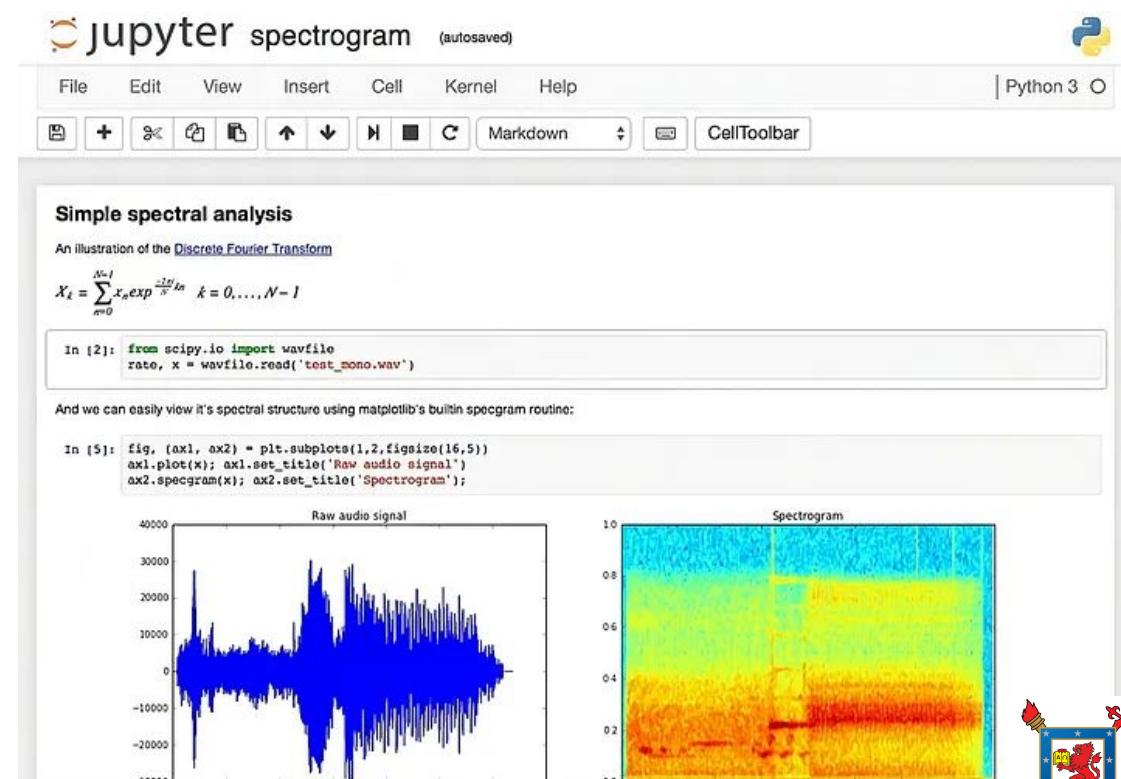
Jupyter Notebook es ampliamente utilizado en el ámbito de la ciencia de datos, matemáticas, la investigación científica y la educación debido a su capacidad para combinar código ejecutable, resultados visuales y narrativa en un solo lugar.

Para instalar y luego, iniciar **Jupyter Notebook**

```
# Instalacion de jupyter con pip
pip install jupyter

# Instalacion de jupyter con conda
conda install jupyter

# Iniciar Jupyter
jupyter notebook
```



Colab (Google Colaboratory)



Google Colab es un documento ejecutable que permite escribir, ejecutar y compartir código dentro de Google Drive. Utiliza documentos notebook que se componen de células, cada una de las cuales puede contener código, texto, imágenes y demás.

¿ Que se puede hacer en **Google Colab** ?

- Escribir y ejecutar código en **Python**.
- Documentar códigos que admiten ecuaciones matemáticas.
- Crear y compartir notebooks.
- Importar y guardar archivos desde Google Drive, etc...

¿ Ventajas de **Google Colab** ?

- No requiere configuración y las bibliotecas ya están pre-instaladas (las mas fundamentales)
- Se guarda en Cloud
- Permite colaboracion
- Acceso gratis a GPU
- etc...

Drive

+ Nuevo

Página principal

Mi unidad

Ordenadores

Compartido conmigo

Reciente

Destacados

Spam

Papelera

Almacenamiento (usado: 12,09 GB de 15 GB usado)

Mi unidad

1 seleccionados

Nombre ↑

Nombre	Tipo	Último ...
Nueva carpeta	Carpeta	yo
Subir archivo	Archivo	yo
Subir carpeta	Carpeta	yo
Capsulas_R	Carpeta	yo
CausalCenco	Carpeta	yo
CFA	Documento de Google	yo
CMF	Hojas de cálculo de Google	yo
Presentaciones de Google	Presentación de Google	yo
Colab Notebo	Formularios de Google	yo
CS_329S	Más	yo

Drive

+ Nuevo

Página principal

Mi unidad

Ordenadores

Compartido conmigo

Reciente

Destacados

Spam

Papelera

Almacenamiento (usado: 12,09 GB de 15 GB usado)

Obtener más almacenamiento

Mi unidad

1 seleccionados

Nombre ↑

Nombre	Tipo	Último ...
Nueva carpeta	Carpeta	yo
Subir archivo	Archivo	yo
Subir carpeta	Carpeta	yo
Capsulas_R	Carpeta	yo
CausalCenco	Carpeta	yo
CFA	Documento de Google	yo
CMF	Hojas de cálculo de Google	yo
Presentaciones de Google	Presentación de Google	yo
Colab Notebo	Formularios de Google	yo
CS_329S	Más	yo
curriculum	Imagen	yo
DM_beuchef	Imagen	yo
Docencia	Imagen	yo
Documento	Imagen	yo
Dibujos de Google	Imagen	yo
Google My Maps	Imagen	yo
Google Sites	Imagen	yo
draw.io	Imagen	yo
Google Apps Script	Imagen	yo
Google Colaboratory	Imagen	yo
Google Jamboard	Imagen	yo



Colab (Google Colaboratory)

Google Colab es un documento ejecutable que permite escribir, ejecutar y compartir código dentro de Google Drive. Utiliza documentos notebook que se componen de células, cada una de las cuales puede contener código, texto, imágenes y demás.

¿ Que se puede hacer en **Google Colab** ?

- Escribir y ejecutar código en **Python**.
- Documentar códigos que admiten ecuaciones matemáticas.
- Crear y compartir notebooks.
- Importar y guardar archivos desde Google Drive, etc...

¿ Ventajas de **Google Colab** ?

- No requiere configuración y las bibliotecas ya están pre-instaladas (las mas fundamentales)
- Se guarda en Cloud
- Permite colaboracion
- Acceso gratis a GPU
- etc...

The image contains two screenshots of the Google Colab interface. The top screenshot shows a notebook titled '03_Regression_life_satisfaction.ipynb' with a code cell containing Python code to clone a repository. The bottom screenshot shows a notebook titled 'prueba.ipynb' with a code cell containing a command to check the version of TensorFlow using pip.

 [Bienvenida a Colab](#)
 [Ejemplo Colab](#)

