

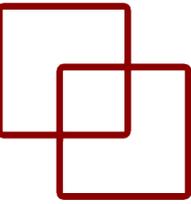
Tema 1. Introducción



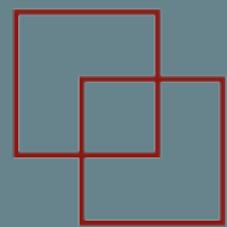
LAB CTIC UNI

Dr. Manuel Castillo-Cara
Intelligent Ubiquitous Technologies – Smart Cities (IUT-SCi)
Web: www.smartcityperu.org

Índice



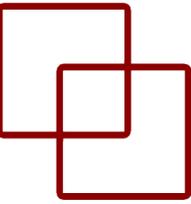
- Sílabo
- Machine Learning
- Contenidos
- Ecosistema de Python
- Gestión de entornos virtuales.



Smart City

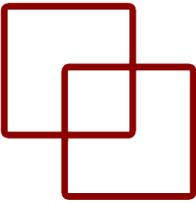
LAB CTIC  UNI

Sílabo



1. Datos Generales

- Nombre: Machine Learning con Python
- Modalidad: Virtual
- Requisitos:
 - Conocimientos intermedios de programación.
 - Conocimientos básicos de estadística inferencial y cálculo de probabilidades.



1. Datos Generales

- Nombre: Machine Learning con Python
- Modalidad: Virtual
- Requisitos:
 - Conocimientos intermedios de programación.
 - Conocimientos básicos de estadística inferencial y cálculo de probabilidades.

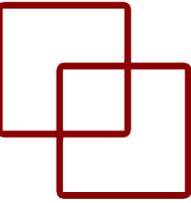
Profesor – Dr. Manuel Castillo-Cara



Profesor de la Universidad Nacional de Ingeniería (UNI) en la escuela de Ciencia de la Computación y Jefe del laboratorio IUT-SCi.

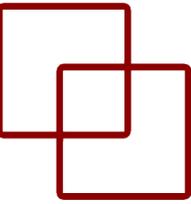
Doctorado en Ingeniería Informática en la Universidad de Castilla-La Mancha (España).

2. Competencias



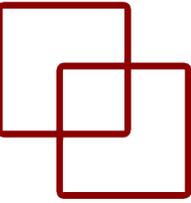
- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.

2. Competencias



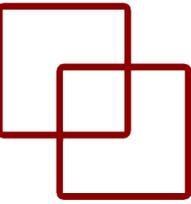
- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.
- Aplicar técnicas de tratamiento de datos en un conjunto de datos para mejorar la robustez y métrica de salida de los diferentes algoritmos de machine learning.

2. Competencias



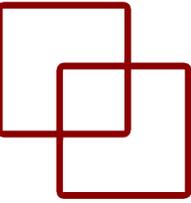
- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.
- Aplicar técnicas de tratamiento de datos en un conjunto de datos para mejorar la robustez y métrica de salida de los diferentes algoritmos de machine learning.
- Comprender los diferentes mecanismos y técnicas para aplicar analítica predictiva en problemas de machine learning e interpretar la salida obtenida por los modelos de predicción.

2. Competencias



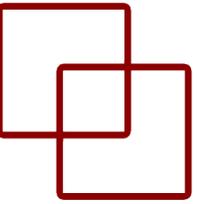
- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.
- Aplicar técnicas de tratamiento de datos en un conjunto de datos para mejorar la robustez y métrica de salida de los diferentes algoritmos de machine learning.
- Comprender los diferentes mecanismos y técnicas para aplicar analítica predictiva en problemas de machine learning e interpretar la salida obtenida por los modelos de predicción.
- Comprender y analizar la fase del análisis de datos previos al modelado algorítmico en machine learning.

2. Competencias



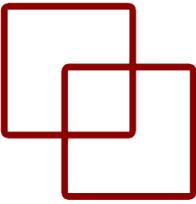
- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.
- Aplicar técnicas de tratamiento de datos en un conjunto de datos para mejorar la robustez y métrica de salida de los diferentes algoritmos de machine learning.
- Comprender los diferentes mecanismos y técnicas para aplicar analítica predictiva en problemas de machine learning e interpretar la salida obtenida por los modelos de predicción.
- Comprender y analizar la fase del análisis de datos previos al modelado algorítmico en machine learning.
- Realizar modelos algorítmicos robustos con una optimización de sus hiperparámetros para la fase de predicción

2. Competencias

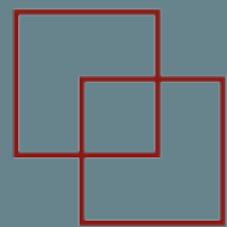


- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.
- Aplicar técnicas de tratamiento de datos en un conjunto de datos para mejorar la robustez y métrica de salida de los diferentes algoritmos de machine learning.
- Comprender los diferentes mecanismos y técnicas para aplicar analítica predictiva en problemas de machine learning e interpretar la salida obtenida por los modelos de predicción.
- Comprender y analizar la fase del análisis de datos previos al modelado algorítmico en machine learning.
- Realizar modelos algorítmicos robustos con una optimización de sus hiperparámetros para la fase de predicción
- Desarrollar y analizar proyectos de machine learning como regresión, clasificación y multiclase.

2. Competencias



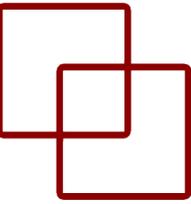
- Aplicar técnicas de análisis y visualización de datos en un conjunto de datos complejo para problemas de machine learning.
- Aplicar técnicas de tratamiento de datos en un conjunto de datos para mejorar la robustez y métrica de salida de los diferentes algoritmos de machine learning.
- Comprender los diferentes mecanismos y técnicas para aplicar analítica predictiva en problemas de machine learning e interpretar la salida obtenida por los modelos de predicción.
- Comprender y analizar la fase del análisis de datos previos al modelado algorítmico en machine learning.
- Realizar modelos algorítmicos robustos con una optimización de sus hiperparámetros para la fase de predicción
- Desarrollar y analizar proyectos de machine learning como regresión, clasificación y multiclase.
- Utilizar librerías específicas de Python como scikit-learn para trabajos de Machine Learning.



Smart City

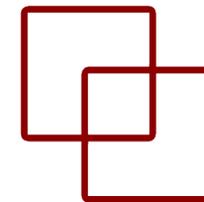
LAB CTIC  UNI

Machine Learning



1. Definición

- Machine Learning trata sobre la construcción de algoritmos que pueden aprender y hacer predicciones con datos; básicamente algoritmos que aprendan con la experiencia.



1. Definición

- Machine Learning trata sobre la construcción de algoritmos que pueden aprender y hacer predicciones con datos; básicamente algoritmos que aprendan con la experiencia.

Según Expertos

- Arthur Samuel (1954) Machine Learning: *“Área de estudio que les da a las computadoras la habilidad de aprender sin necesidad de ser explícitamente programadas”*.
- Según Tom M. Mitchel (1998): *“a un programa de computadora se le indica aprender de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P , si su rendimiento sobre T , medido por P , mejora con la experiencia de E .”*

2. Simplemente es...

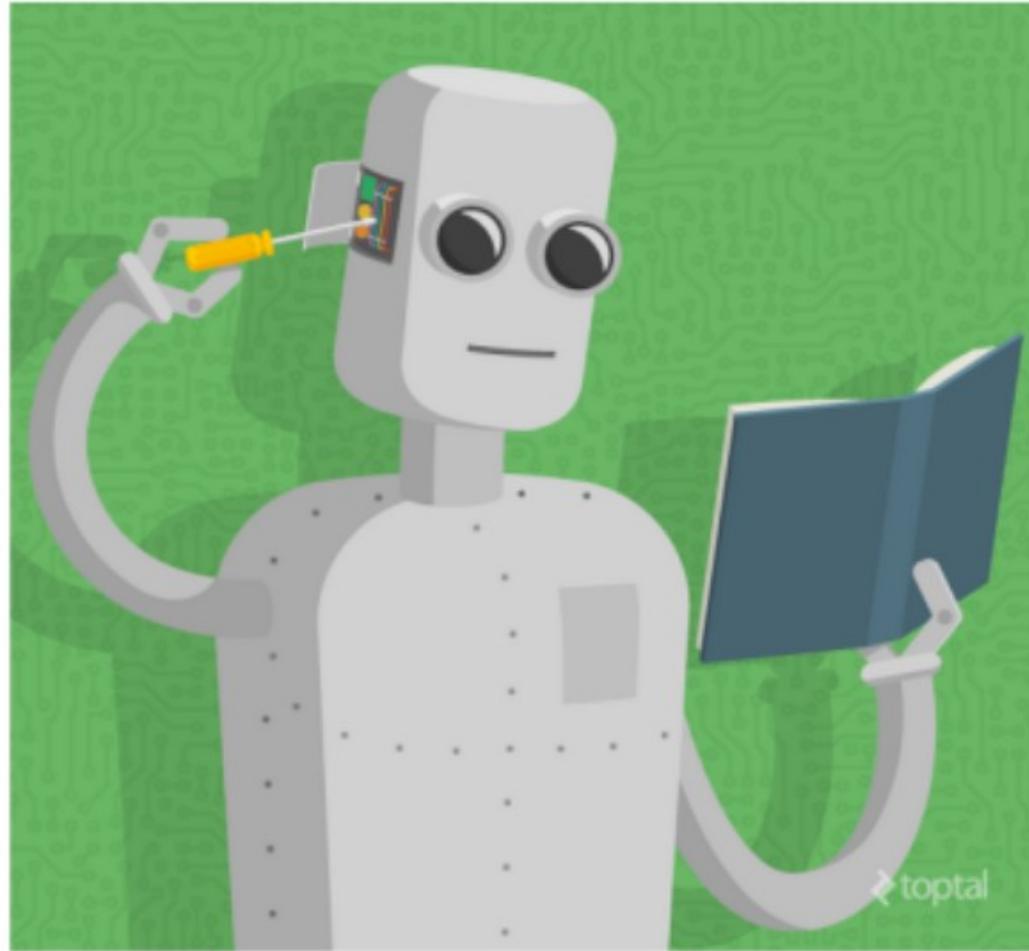
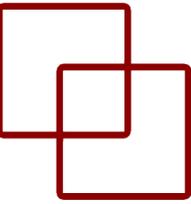
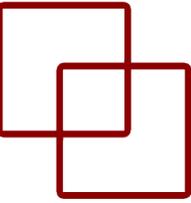
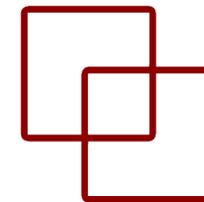


Figure: Máquina que aprende. Fuente: Toptal.

3. Rompiendo mitos



- No hay que ser bueno con las matemáticas.
- No hay que ser buenos programadores.
- Nos encontramos en la temática llamada modelado predictivo dentro del ML.
- ¿Diferencia entre estadística tradicional y modelado predictivo? ¿Donde estamos nosotros?

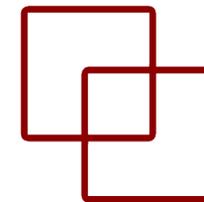


3. Rompiendo mitos

- No hay que ser bueno con las matemáticas.
- No hay que ser buenos programadores.
- Nos encontramos en la temática llamada modelado predictivo dentro del ML.
- ¿Diferencia entre estadística tradicional y modelado predictivo? ¿Donde estamos nosotros?

Estadística tradicional

Tiene la funcionalidad principal de llegar a comprender los datos.



3. Rompiendo mitos

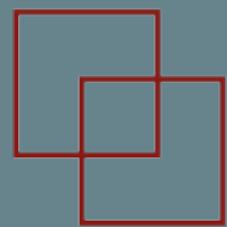
- No hay que ser bueno con las matemáticas.
- No hay que ser buenos programadores.
- Nos encontramos en la temática llamada modelado predictivo dentro del ML.
- ¿Diferencia entre estadística tradicional y modelado predictivo? ¿Donde estamos nosotros?

Estadística tradicional

Tiene la funcionalidad principal de llegar a comprender los datos.

Modelado predictivo

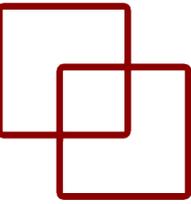
Técnica que tiene por objeto descubrir patrones de comportamiento de nuestros datos, en este caso tabulares como hojas de cálculo, para tener predicciones más precisas.



Smart City

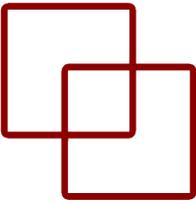
LAB CTIC  UNI

Contenidos



1. Introducción

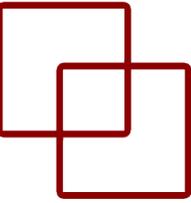
- Conceptos básicos de machine learning.
- Anaconda como nuestro gestor de trabajo.
- Jupyter Notebook como nuestro entorno de machine learning.
- Curso rápido de Python.



2. Análisis de datos

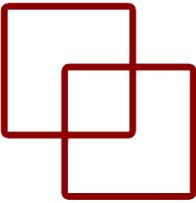
- Cargar un conjunto de datos.
- Estadística descriptiva.
- Visualización de datos.
- Taller: Trabajo de aplicación de diferentes técnicas analíticas de datos en un conjunto de datos seleccionado por el usuario e interpretar la salida obtenida.
- Examen tipo test sobre los contenidos del módulo.

3. Preprocesamiento de datos

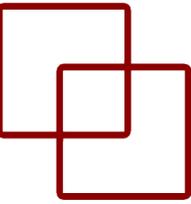


- Análisis exploratorio de datos.
- Preprocesamiento de datos.
- Métodos de remuestreo para estimar la precisión del modelo.
- Taller: Trabajo de aplicación de diferentes técnicas analíticas de datos en un conjunto de datos seleccionado por el usuario e interpretar la salida obtenida.
- Examen tipo test sobre los contenidos del módulo.

4. Fase de tratamiento de datos



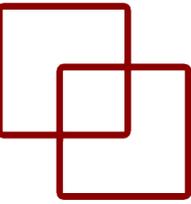
- Evaluación de las métricas.
- Feature Selection.
- Feature Importance.
- Reducción de dimensiones en un dataset.
- Taller: Aplicación de diferentes técnicas de tratamiento de datos en un conjunto de datos y verificación de su impacto en las métricas algorítmicas.
- Examen tipo test sobre los contenidos del módulo.



5. Fase de modelado

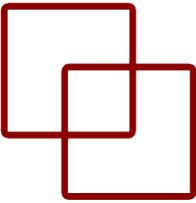
- Algoritmos de Machine Learning.
- Rendimiento de los algoritmos.
- Algoritmos Ensamblados
- Taller: Aplicación de diferentes algoritmos de machine learning en un conjunto de datos e interpretar la salida obtenida, así mismo, verificar el algoritmo que tenga mejor comportamiento.
- Examen tipo test sobre los contenidos del módulo.

6. Fase de optimización y forecasting

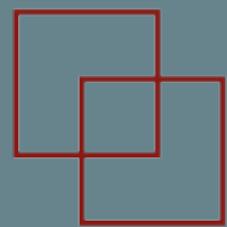


- Pipelines
- Procesamiento de datos avanzado
- Configuración de hiperparámetros
- Guardado e integración del modelo
- Taller: Una vez seleccionados los algoritmos candidatos a modelo realizar una optimización de estos a través de la configuración de sus hiperparámetros.
- Examen tipo test sobre los contenidos del módulo.

7. Proyectos de machine learning



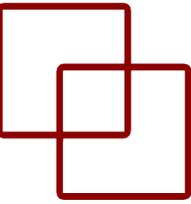
- Trabajar un proyecto de clasificación multiclase
- Trabajar un proyecto de regresión.
- Trabajar un proyecto de clasificación binaria.
- Proyecto: Realizar un proyecto completo analizando todas las fases estudiadas en los diferentes módulos.



Smart City

LAB CTIC  UNI

Lenguaje de programación Python

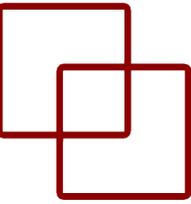


1. ¿Por qué Python?

- El uso de lenguajes de programación fuera del ámbito informático se ha vuelto imprescindible



- ¿Qué tienen en común estos lenguajes?



1. ¿Por qué Python?

- El uso de lenguajes de programación fuera del ámbito informático se ha vuelto imprescindible

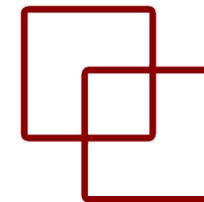


- ¿Qué tienen en común estos lenguajes?

Interactivo

Gran comunidad

Aprendizaje



1. ¿Por qué Python?

- El uso de lenguajes de programación fuera del ámbito informático se ha vuelto imprescindible



- ¿Qué tienen en común estos lenguajes?

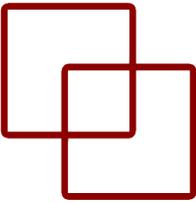
Interactivo

Gran comunidad

Aprendizaje

- Python es un potente lenguaje de programación comparte estas características, y además de cara al curso encaja a la perfección.

1. Rankings

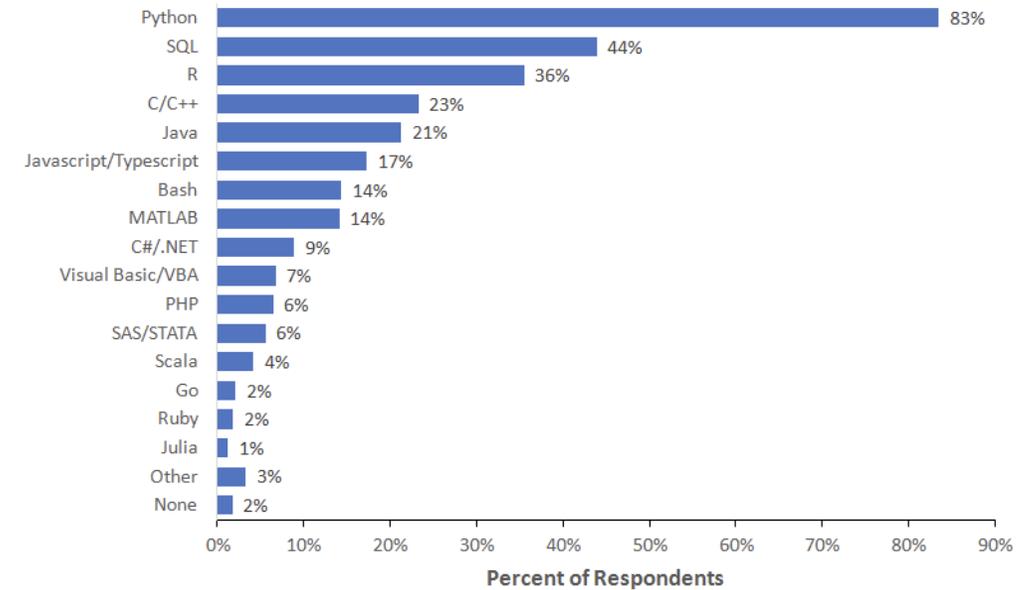


Language Types (click to hide)

Web
 Mobile
 Enterprise
 Embedded

Language Rank	Types	Spectrum Ranking
1. Python	Web, Enterprise, Embedded	100.0
2. C++	Mobile, Enterprise, Embedded	99.7
3. Java	Web, Mobile, Enterprise	97.5
4. C	Mobile, Enterprise, Embedded	96.7
5. C#	Web, Mobile, Enterprise	89.4
6. PHP	Web	84.9
7. R	Enterprise	82.9
8. JavaScript	Web, Mobile	82.6
9. Go	Web, Enterprise	76.4
10. Assembly	Embedded	74.1
11. Matlab	Enterprise	72.8
12. Scala	Web, Mobile	72.1
13. Ruby	Web, Enterprise	71.4

What programming language do you use on a regular basis?



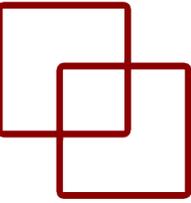
Note: Data are from the 2018 Kaggle Machine Learning and Data Science Survey. You can learn more about the study here: <http://www.kaggle.com/kaggle/kaggle-survey-2018>. A total of 18827 respondents answered the question.



Copyright 2019 Business Over Broadway

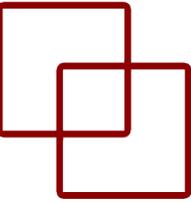
DATA SCIENCE | CUSTOMER ANALYTICS | MACHINE LEARNING

3. Descripción

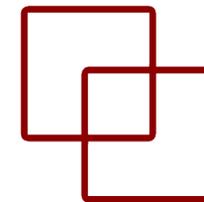


- **Scripting:** no necesita compilación. Permite ejecutar código directamente en un intérprete.

3. Descripción



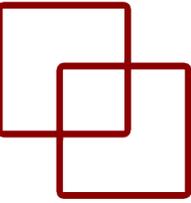
- **Scripting:** no necesita compilación. Permite ejecutar código directamente en un intérprete
- **POO:** la programación orientada a objetos es útil para describir modelos, datasets, etc.



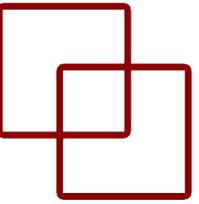
3. Descripción

- **Scripting:** no necesita compilación. Permite ejecutar código directamente en un intérprete
- **POO:** la programación orientada a objetos es útil para describir modelos, datasets, etc.
- **Orden superior:** Elementos de la programación funcional y funciones de orden superior permiten simplificar y optimizar las ejecuciones sobre un conjunto de datos

4. ¿Que es un intérprete

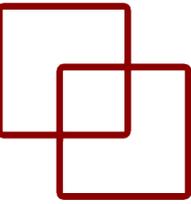


- En Python todo código se ejecuta dentro de un **intérprete**.



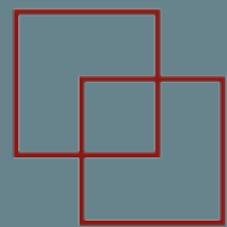
4. ¿Que es un intérprete

- En Python todo código se ejecuta dentro de un **intérprete**.
- Es un **entorno de trabajo** que contiene la información de lo que se está ejecutando (variables, funciones, paquetes importados, etc).



4. ¿Que es un intérprete

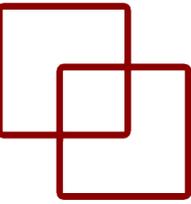
- En Python todo código se ejecuta dentro de un **intérprete**.
- Es un **entorno de trabajo** que contiene la información de lo que se está ejecutando (variables, funciones, paquetes importados, etc).
- Estos entornos de trabajo son **aislados**: si abrimos dos intérpretes de Python, ni las variables ni el resto de elementos se comparten.



Smart City

LAB CTIC  UNI

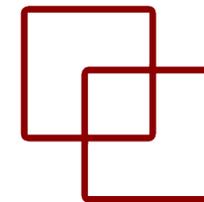
Ecosistema de Python



1. Tratamiento de datos

- Tidy data: “Wickham, H. (2014). Tidy data. Journal of Statistical Software, 59(10), 1-23.”





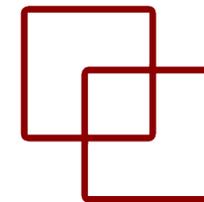
1. Tratamiento de datos

- Tidy data: “Wickham, H. (2014). Tidy data. *Journal of Statistical Software* 59(10), 1-23.”



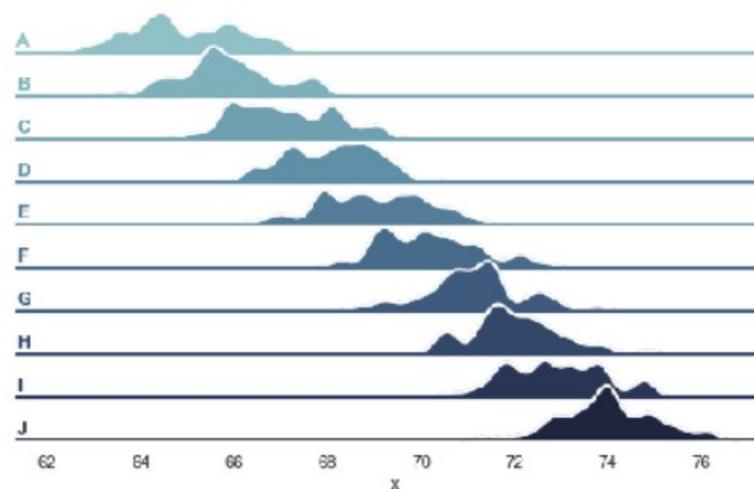
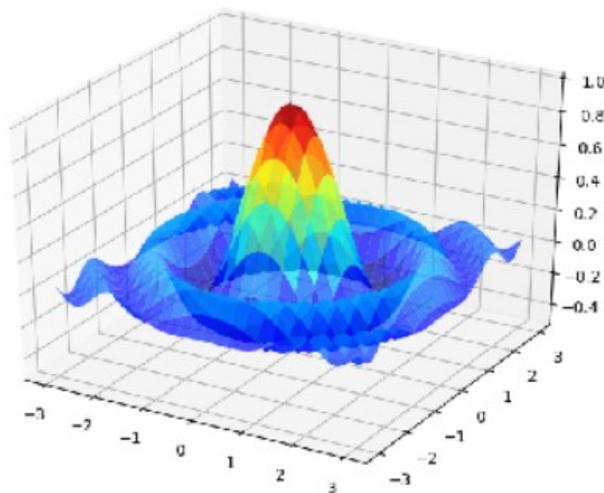
- Pandas: Representación y tratamiento de datos siguiendo el formato DataFrame de R



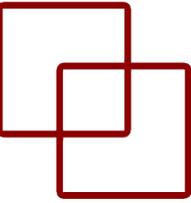


2. Visualización de datos

- Tanto R (*ggplot2*) como Matlab son populares (entre otras cosas) por su capacidad de generación de gráficos.
- En Python disponemos de potentes librerías para la generación de gráficos: *matplotlib* para la generación más detallada (y a más bajo nivel) y *seaborn*, entre otros.



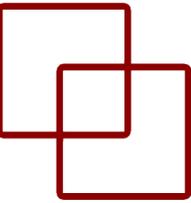
3. Machine learning y Big Data



- Machine Learning aplicado a problemas tradicionales



3. Machine learning y Big Data

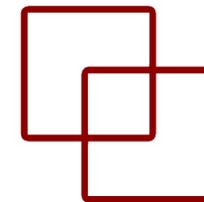


- Machine Learning aplicado a problemas tradicionales



- Machine Learning aplicado a problemas de Big Data



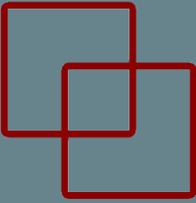


4. Ecosistema SciPy

- “The SciPy ecosystem, a collection of open source software for scientific computing in Python.”
- La computación científica en Python se construye a partir de un pequeño conjunto de paquetes.
 - Numpy
 - Matplotlib
 - Pandas
 - ...

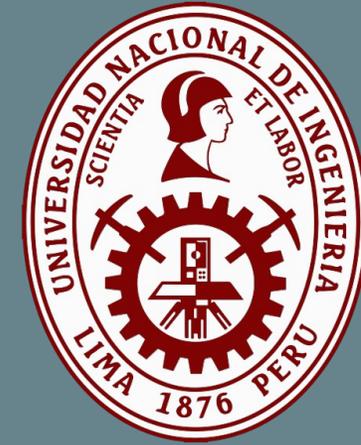


SciPy



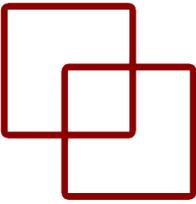
Smart City

LAB CTIC UNI



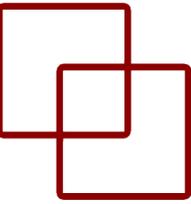
Gestión de entornos virtuales

1. Distribuciones alternativas de Python



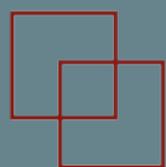
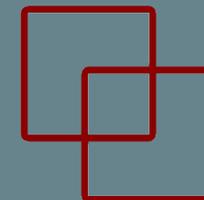
- Aunque Python es interpretado, existen ciertas librerías que se traducen a código compilado, como Numpy.
- Hay distribuciones que se encargan de mantener un correcto versionado de los paquetes, precompilados con respecto a sistemas operativos, arquitecturas hardware, etc.
- Uno de los más populares en el stack científico y en particular orientado a data science y machine learning es **Anaconda**.
 - Intel ha anunciado recientemente una distribución propia sobre Anaconda, optimizado para la ejecución en procesadores Intel.

2. Anaconda



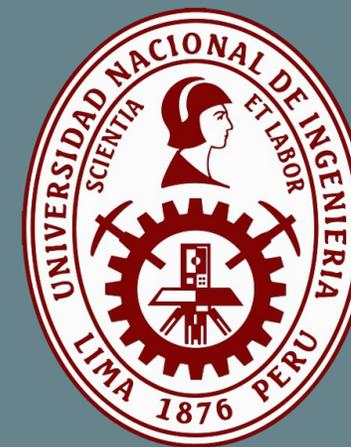
- Es *open source*, e incluye Python, una serie de paquetes orientados al stack científico y un gestor de paquetes y entornos virtuales llamado *conda*.
 - Facilita la instalación de Python y paquetes, sobre todo en Windows
 - Entornos virtuales centralizados, y referencias por nombres
 - Interfaz mejorada sobre *pip*, mostrando información de todas las dependencias, tamaño de paquetes, etc.
 - etc.
- Realmente todos trabajan sobre *Python*, *pip* y *venv* o similares



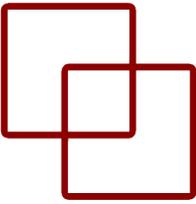


Smart City

LAB CTIC UNI



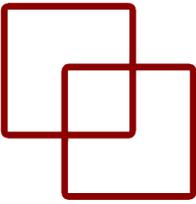
Jupyter notebook



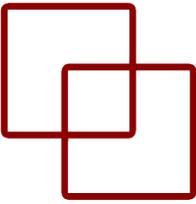
1. Contexto

- iPython es un shell interactivo de Python, con soporte para gráficos e interfaces gráficas de usuario
- Jupyter notebook es popularmente conocido como iPython notebook (nació a partir de ese proyecto)
- Jupyter: **J**ulia, **P**ython y **R** (los tres lenguajes para los que inicialmente fue diseñado). Ahora soporta **más de 40**
- Lleva iPython un paso más allá: proporciona una web interactiva sobre la que utilizar Python.

2. ¿Por que utilizar Jupyter?

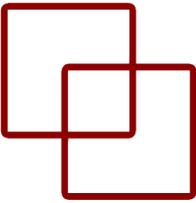


- Se despliega como un servidor web y se accede por el **navegador**.
- Es una herramienta muy cómoda para el prototipado de software, visualización de resultados, análisis exploratorio de datos, etc.
- Para el desarrollo de software más complejo, librerías, etc. existen IDEs más potentes, como PyCharm.



3. Definiciones

- La interfaz principal permite gestionar las libretas (ficheros con extensión *.ipynb*).
- Las libretas son las webs interactivas sobre las que ejecutaremos código Python
- Cada libreta tiene asociado un intérprete de Python, por lo que cada entorno es independiente
- Se diferencian dos estados dentro de las libretas:
 - **Modo de comandos:** No aparece el cursor en la celda. Se cambia a este modo pulsando escape. Sirve para ejecutar comandos de gestión de las libretas.
 - **Modo de edición:** Aparece el cursor en la celda. Se cambia a este modo pulsando intro o haciendo click en la celda. Sirve para escribir código en las celdas.



4. Paquetes en Jupyter

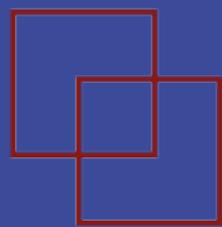
- Los paquetes o librerías son la forma en que se distribuye el código Python de terceros.
- Estos paquetes pueden descargarse e instalar en nuestro entorno de Jupyter Notebook.

```
> !pip install scikit-learn
```

- Una vez instaladas podemos importarlas

```
> import pandas as pd  
> from pandas import read_csv
```

¡GRACIAS!



Smart City

LAB CTIC UNI

Dr. Manuel Castillo-Cara
Intelligent Ubiquitous Technologies – Smart Cities (IUT-SCi)
Web: www.smartcityperu.org