Módulo 3: Desarrollo de software aplicado a la ciencia de datos

Intro (30 min)

Acceso al material

Temario del módulo

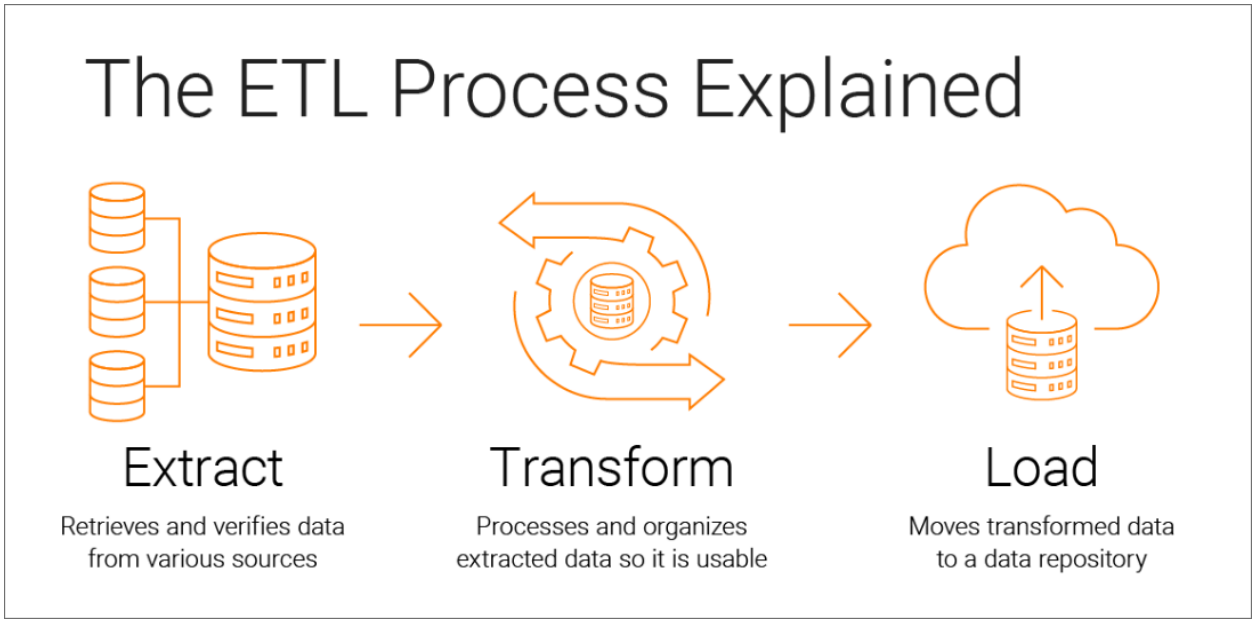
ETLs

Bibliotecas de python

SQL

¿Qué es ETL?

- No es un programa, es un **proceso**.
- ETL se puede aplicar con diferentes herramientas, cada una tiene ventajas y desventajas.
- Muchas veces el proceso



fuente

Ejemplo de como se ve un ETL muy simplificado en Python:

Simple ETL with Pandas

3.1 Programación de álgebra lineal uso de bibliotecas vectorizadas (numpy) 2 horas

El álgebra lineal es parte primaria de la ciencia de datos, ya que muchos de los algoritmos de aprendizaje máquina usan matrices para almacenar los datos.

Numpy es una biblioteca de cómputo numérico que nos permite usar arreglos y matrices en Python. Además contiene una sintaxis simplificada para hacer operaciones de álgebra lineal y aplicar transformaciones a las matrices en pocas líneas.

Puedes descargarlo e instalarlo en https://numpy.org/
O usarlo en los ambientes de google colab.

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques tutorial numpy

Termina el laboratorio de tarea y trae tus dudas para la siguiente clase.

3.2 Bibliotecas para la manipulación y visualización de los datos uso de bibliotecas para el manejo de set de datos (pandas) 2 horas

Trabajar con los sets de datos y prepararlos para hacer análisis o ser usados en algoritmos de machine learning es sumamente laborioso y tardado. Por ellos se han creado una gran gama de librerías enfocadas a simplificar y hacer más rápida esta labor. Pandas es la librería más popular de Python y una de las más relevantes en la industria para el manejo de datos.

Trabaja con objetos Data Frames que son objetos que contienen hashes que a su vez contienen arreglos de numpy cuando usan datos numéricos.

Puedes descargarlo e instalarlo en
https://pandas.pydata.org/
O usarlo en los ambientes de google colab.

Series y Dataframes:
Partes de un dataframe

Documentación básica de un data frame:
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques tutorial pandas basico

3.3 ETL: Extracción de datos desde el sistema origen

Proceso de extracción

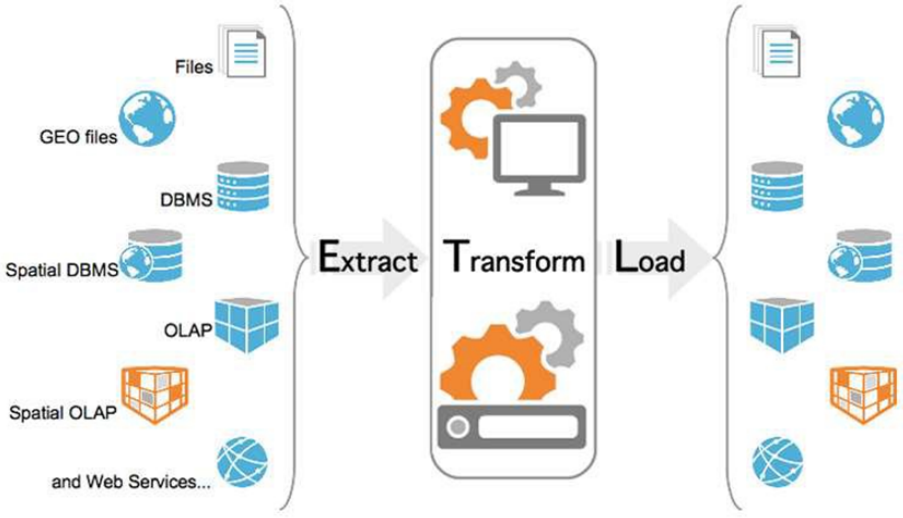
La primera parte del proceso ETL consiste en extraer los datos desde el sistema de origen. Esta fase consta de tres pasos:

1. **Análisis previo de las necesidades.** Normalmente, hay una primera etapa de análisis donde se evalúan las necesidades concretas de la organización en cuanto a movimiento y transformación de datos. Por lo general, no se comienza extrayendo toda la información, sino que se empieza con el tratamiento de unos datos concretos según una necesidad puntual y luego se realizan ampliaciones a medida que se van necesitando.
2. **Identificación de archivos.** Antes de extraer los datos, se identifica de qué tipo son y en qué formato se encuentran los sistemas fuente. Por ejemplo, si se trata de bases de datos relacionales o no relacionales, archivos planos, archivos complejos, archivos VSAM, WebServices, archivos en Excel, etc. Lo habitual es que los datos de los sistemas de origen provengan de formatos distintos, que tendrán que ser fusionados.
3. **Extracción de los datos.** En función de las necesidades detectadas, se procede a la extracción en sí de dichos datos.

Modos de extracción

Existen tres modos distintos de extracción. El tipo de necesidad de la organización es lo que, normalmente, determinará la elección de una u otra forma.

1. **Full Extract o extracción total**: Esta modalidad consiste en extraer la totalidad de datos. En este caso, se barren tablas completas que pueden llegar a tener millones de registros.
2. **Incremental Extract o extracción incremental**: Se va procesando por lotes únicamente lo que fue modificado o agregado. También puede haber filas que se borren por estar duplicadas, tratarse de datos erróneos, etc.
3. **Update Notification o notificación de actualizaciones**: En este caso, solo se van extrayendo los datos a medida que se produce una actualización (por ejemplo, un insert)



Debemos tomar siempre en consideración los siguientes puntos:

1. Extraer los datos desde la fuente de datos.
2. Analizar los datos extraídos .
3. Verificar que los datos extraídos cumplan con los requerimientos del proyecto así como la estructura.
4. Convertir los datos a una formato preparado para iniciar el proceso de transformación.

Actividad: extracción de datos con pandas.

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques, intenta después replicarlo por tu cuenta pero con el ejemplo del Titanic tutorial pandas extraccion (Sola lleva a acabo la parte de extracción y detente antes de limpiez de datos).

3.4 ETL: Aplicación de transformaciones, limpieza e imputación de los datos (2 horas)

Proceso de Transformación

En esta etapa se deben aplicar las reglas del negocio así como la naturaleza del tipo de datos que se requiere para posteriormente aplicar modelos analíticos y visualizaciones.

Consideraciones:

1. Datos NULOS
2. Unidades de Medida (litros, Km, Cm, etc).
3. Traducir códigos (H-1 , M-2)
4. Obtener valores calculados (Month(fecha))
5. Relacionar diferentes fuentes de datos
6. Agrupar datos (región , temporada)
7. Generar campos ID
8. Dividir columnas (Nombre: Nombre, Apellido Paterno, Apellido Materno).
9. Definir tratamiento de excepciones para datos erróneos.
10. Datos NO-ESTRUCTURADOS

Limpieza de los datos

Consiste en asegurar la calidad de los datos que serán cargados posteriormente en el DATA-SET. Para lograr lo anterior, necesitamos que los datos cumplan los requisitos siguientes:

1. Correctos
2. Inequívocos
3. Consistentes
4. Completos

- **Correctos**: Que los valores de datos describan los objetos asociados con veracidad y fidelidad.
- **Inequívocos**: Los valores y las descripciones de los datos pueden tomar sólo un sentido.
- **Consistentes**: Sea un solo valor.
- **Completos**: Dentro de este punto, existen dos aspectos:

    - El primero, es asegurar que los valores individuales y descripciones de los datos se definen (no nulos) para cada instancia.
    - El segundo, es asegurarse de que el número total de registros se ha completado o de que no se pierde de alguna manera por completo los registros en algún lugar de su flujo de información. (Total de registros esté correcto)

La limpieza de datos también es conocida como limpieza de datos (data cleansing) o depuración de datos (data scrubbing), y se refiere a la detección y eliminación o corrección de registros incorrectos o inconsistencias de datos para mejorar la calidad de los mismos.

¿Porqué es importante la limpieza de datos?

1. El conjunto de datos que obtenemos del mundo real contienen valores no válidos: espacios en blanco, NA, NaN, valores atípicos, etc.
2. La mayoría de veces los métodos que se utilizan para cargar los datos asumen que todos los valores de entrada son numéricos o categóricos.
3. Algunos métodos excluyen cualquier caso en el que falta una entrada, y otros generan un error.
4. Todas las variables en un modelo deben ser significativas y estar relacionadas con el fenómeno que intentamos predecir.
5. La mayoría de los métodos basados en la distancia (es decir, k-means, regresión lineal, redes neuronales) darán una predicción errónea si las características no se distribuyen de forma normal: Distribución gaussiana o distribución de Laplace-Gauss con media cero y varianza unitaria.
6. Si la relación no es lineal, puede ser necesario aplicar alguna técnica de transformación polinomial, u otros tipos de transformaciones no lineales.

Proceso para la limpieza de datos:

- Eliminar variables irrelevantes,
- Verificar el tipo de valores perdidos: probablemente debamos eliminar columnas o filas.
- Identificar variables categóricas y valores posibles
- Ver los límites superior e inferior para datos numéricos.

Ejemplo: Bases de datos del Titanic

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	NaN	S
	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C
	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	NaN	S
	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	NaN	S
	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.0750	NaN	S
	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333	NaN	S
	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708	NaN	C

```
VARIABLE DESCRIPTIONS:
survival        Survival
                (0 = No; 1 = Yes)
pclass          Passenger Class
                (1 = 1st; 2 = 2nd; 3 = 3rd)
name            Name
sex             Sex
age             Age
sibsp           Number of Siblings/Spouses Aboard
parch           Number of Parents/Children Aboard
ticket          Ticket Number
fare            Passenger Fare
cabin           Cabin
embarked        Port of Embarkation
                (C = Cherbourg; Q = Queenstown; S = Southampton)
```

```
Sibling:  Brother, Sister, Stepbrother, or Stepsister o
Spouse:   Husband or Wife of Passenger Aboard Titanic (
Parent:   Mother or Father of Passenger Aboard Titanic
Child:    Son, Daughter, Stepson, or Stepdaughter of Pa
```

```
SPECIAL NOTES:
Pclass is a proxy for socio-economic status (SES)
 1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower

Age is in Years; Fractional if Age less than One (1)
 If the Age is Estimated, it is in the form xx.5
```

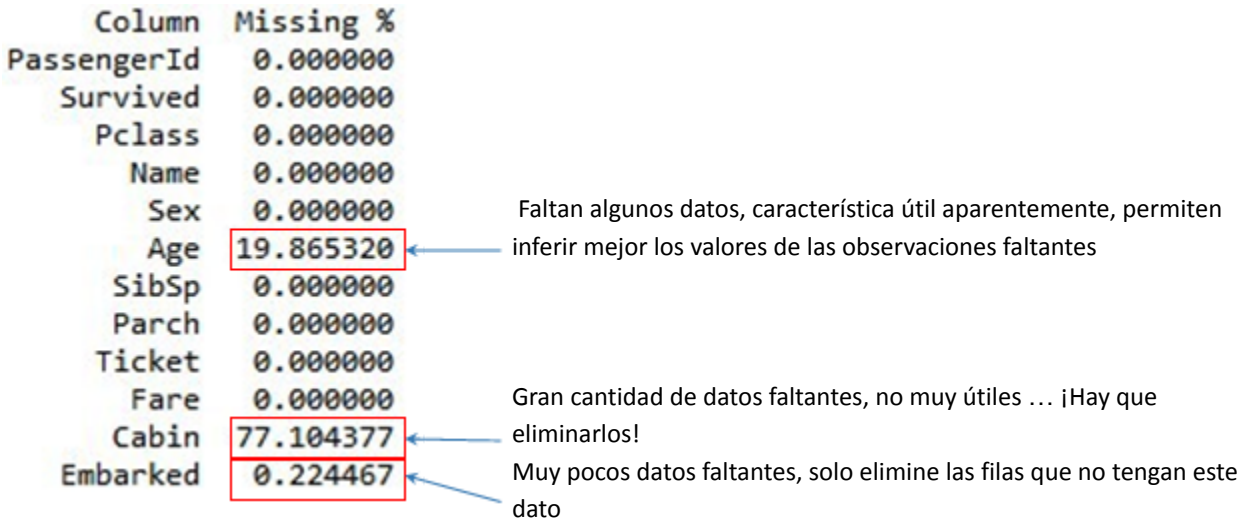
¿Qué columnas debemos utilizar?

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	NaN	S
6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708	NaN	C

Datos irrelevantes (para el análisis actual)

PassengerId	Survived	Pclass	Sex	Age	SibSp	Fare	Cabin	Embarked
1	0	3	male	22	1	7.2500	NaN	S
2	1	1	female	38	1	71.2833	C85	C
3	1	3	female	26	0	7.9250	NaN	S
4	1	1	female	35	1	53.1000	C123	S
5	0	3	male	35	0	8.0500	NaN	S
6	0	3	male	NaN	0	8.4583	NaN	Q
7	0	1	male	54	0	51.8625	E46	S
8	0	3	male	2	3	21.0750	NaN	S
9	1	3	female	27	0	11.1333	NaN	S
10	1	2	female	14	1	30.0708	NaN	C

Valores faltantes



Proceso de imputación

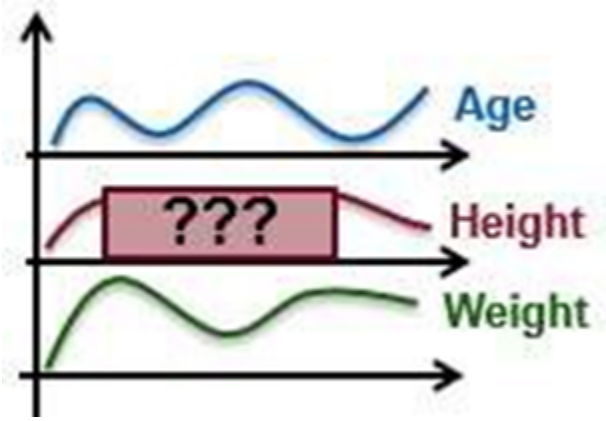
Imputación de datos, conocida también como Data Imputation en inglés, no es más que un conjunto de métodos para manejar los datos faltantes en una base de datos.

Los objetivos del Data Imputation son principalmente:

1. Minimizar el sesgo entre los datos
2. Maximizar el uso de la información disponible
3. Obtener estimaciones apropiadas de incertidumbre

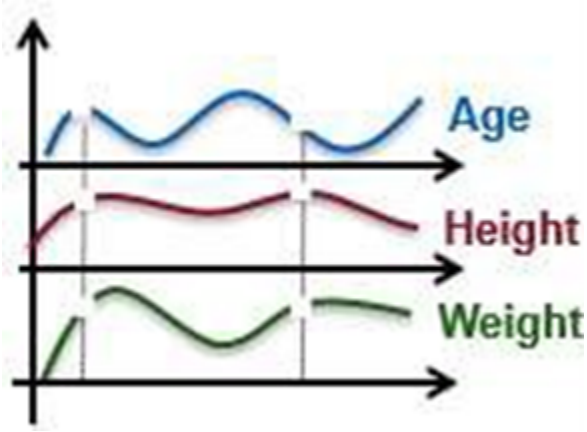
Casos de falta de datos

Fully missing (I)



Available
-Time stamps
Missing
-Intervals of records for specific variables (like Height)

Fully missing (II)



Available
-Time stamps
Missing
-Records for all variables at known times

Partially missing

Available
-Time stamps
Missing
-Random records of variables like age, height, weight

Nota: Los datos pueden mostrar una combinación de estos tipos de datos faltantes

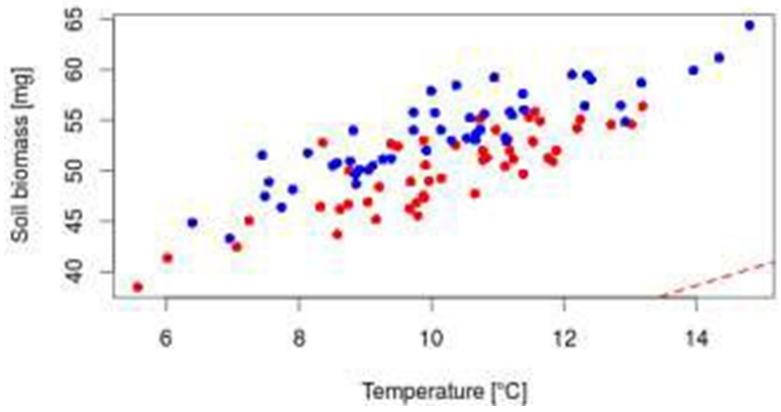
Técnicas de imputación:

Las técnicas más comunes a aplicar en el campo del Data Imputation son:

- Análisis completo de casos (eliminación de Listwise):
  - Eliminar casos con datos faltantes.
    - Pérdida de casos / datos, parcialidad. (consecuencia)

- Imputación media:
  - Reemplazar los valores perdidos con la media general estimada.
    - Reducción artificial de la variabilidad. (consecuencia)

- Imputación única (imputación de regresión):
  - Reemplaza los valores perdidos con puntajes pronosticados de una ecuación de regresión o una función de transferencia.
    - Disminución de la variabilidad. (consecuencia)



Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques, intenta después replicarlo por tu cuenta pero con el ejemplo del Titanic Tutorial Pandas

Pandas como herramienta de ETL:
Limpiar los datos: https://pandas.pydata.org/docs/user_guide/missing_data.html#cleaning-filling-missing-data
Imputar datos: https://pandas.pydata.org/docs/user_guide/missing_data.html#filling-with-a-pandasobject

Termina el laboratorio de tarea y trae tus dudas para la siguiente clase.

3.5 ETL: Carga de los datos procesados a su destino

Una vez que ya tenemos terminamos con nuestras transformaciones, es momento de cargar los datos a donde se vaya a utilizar, ya sea una base SQL o alguna otra tecnología. Hay varia formas de como hacer esto, normalmente aquí se conectan con servicios como AWS, servidores en linea que tengan la capacidad de almacenar y procesar cantidades masivas de datos.
Hay varias alternativas de como migrar la información, pandas sugiere:
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_sql.html

3.6 Biblioteca para estadística avanzada y aprendizaje máquina scipy, scikit-learn 30 min

Todos los algoritmos y métodos de estadística avanzada o usar algoritmos de machine learning se pueden implementar en los lenguajes de programación de propósito general, como python, C++, Java, entre otros. Sin embargo dado que estos métodos suelen hacer millones de operaciones deben de implementarse de manera adecuada usando todos los trucos y mañas posible de cada lenguaje. Esto no

solo es muy complejo, sino que toma días implementar las rutinas básicas. Este problema recurrente en todos los lenguajes usados para ciencia datos dio origen a las bibliotecas avanzadas para estadística y machine learning. En este caso vamos a ver 2 bibliotecas en Python la primera es SciPy que está enfocada a métodos de estadística y el segundo es SciKit-learn que está enfocado a métodos de machine learning.

3.6.1 uso de bibliotecas para estadística (scipy) 30 min

SciPy es una biblioteca que contiene un compendio de algoritmos matemáticos para operaciones complejas, como clusterización, estadística avanzada, métodos de optimización, entre muchos otros. La intención de SciPy es poder dotar a Python de una funcionalidad similar a Matlab, Octave o R que son frameworks comúnmente usados en ingeniería, estadística y ciencia de datos. SciPy es una extensión de NumPy por lo que todos los algoritmos de SciPy funcionan sobre arreglos de NumPy.

Puedes descargarlo e instalarlo en https://docs.scipy.org/
O usarlo en los ambientes de google colab.

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques
Tutorial SciPy

3.6.2 uso de bibliotecas para aprendizaje máquina (scikit-learn) 1 hora

Scikit learn es una de las bibliotecas más usadas en Python para entrenar modelos básicos de machine learning. No trabaja con redes profundas de deep learning, pero contempla casi todas las familias de algoritmos en su esquema básico: regresiones, árboles, ensambles, PCA, clusters, entre otros.

Puedes descargarlo e instalarlo en https://scikit-learn.org/stable/
O usarlo en los ambientes de google colab.

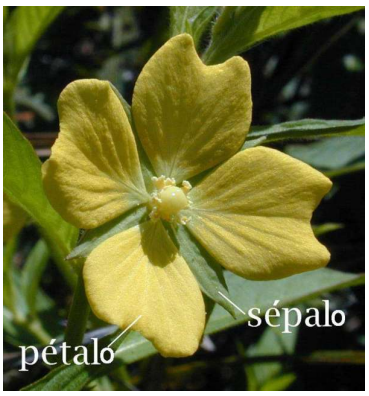
Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques
Tutorial scikit

Recordatorio, esta semana se entrega la sección de ETL en el RETO
Instrucciones:

a. **Descarguen el sets de datos del reto y analícenlo en equipo antes de empezar a limpiarlo.**
b. **Limpien los datos usando herramientas de ETL.**
c. **Expliquen y documenten cada decisión que hayan tomado sobre cómo limpiar los atributos y valores. Sean claros en sus explicaciones y concretos, respuestas ambiguas no serán tomadas en cuenta.**
d. **Suban los scripts utilizados para limpiar los valores y su documentación a su repositorio de equipo del reto.**
e. **Apliquen las transformaciones necesarias a los datos usando herramientas de ETL**
f. **Expliquen y documenten cada decisión que hayan tomado sobre cómo transformar cada variable. Sean claros en sus explicaciones y concretos, respuestas ambiguas no serán tomadas en cuenta.**
g. **Suban los scripts utilizados para transformar los valores y su documentación a su repositorio de equipo del reto.**

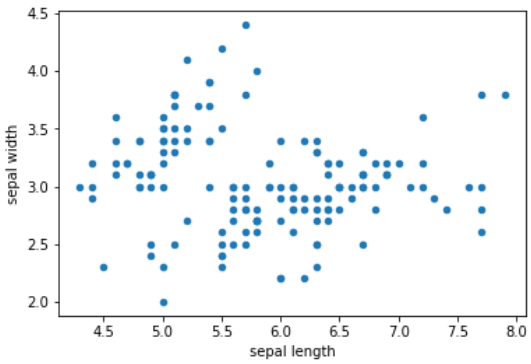
3.7 Bibliotecas para la manipulación y visualización de los datos **30 min**

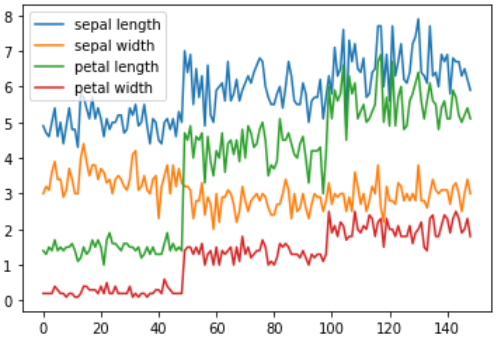
Las bibliotecas de visualización las utilizamos para poder entender los datos de manera rápida y también para poder interpretar los resultados que nos arrojan los algoritmos. El usar gráficas nos permite resaltar aspectos de los datos de manera casi instantánea.



Por ejemplo, viendo la gráfica de líneas que se presenta a la derecha, podemos ver la variable que presenta valores más altos, y también podemos ver que todas las variables siguen un patrón ascendente.

Otro ejemplo es que usando una gráfica de puntos podemos rápidamente detectar cuales son los valores extremos o menos comunes entre 2 variables. En la gráfica de la izquierda podemos observar que los sépalos en su mayoría tiene valores entre 5 y 7 de largo y entre 2 y 4 de ancho.





3.7.1. uso de bibliotecas para la visualización de datos (matplotlib) **1 hora**

Para poder visualizar los datos hay una gran gama de soluciones, en este caso la más usada en Python y con Pandas es Matplotlib. Matplotlib es un paquete gráfico muy grande, lo que numpy es para las bibliotecas de Machine Learning y estadística en Python, Matplotlib lo es para todos los paquetes de diagramas y gráficos.

Puedes descargarlo e instalarlo en https://matplotlib.org/
O usarlo en los ambientes de google colab.

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques
Tutorial visualizing

3.7.2. uso de bibliotecas para la visualización de datos (seaborn) **30 min**

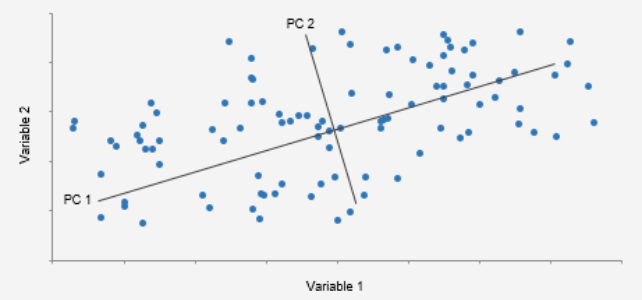
Seaborn es una biblioteca que mezcla elementos simplificados de matplotlib ya preconfigurados para tener gráficos vistosos con poco código y también mezcla algunas de las funcionalidades básicas de Pandas, de tal forma que puedas trabajar con Dataframes en Seaborn también.

Puedes descargarlo e instalarlo en https://seaborn.pydata.org/
O usarlo en los ambientes de google colab.

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques
Tutorial seaborn

3.8 Programación de álgebra lineal aplicación de PCA 2 horas

Principal Component Analysis PCA por sus siglas en inglés es una técnica de reducción de dimensionalidad que toma las variables o columnas de datos numéricos en una matriz y las transforma en nuevas variables llamadas "**componentes principales**" y las acomoda de mayor a menor según su varianza.



En este video se explica paso a paso la lógica de cómo funciona PCA
https://www.youtube.com/watch?v=FgakZw6K1QQ

Actividad Sigue el siguiente código en google colab de la mano con tu profesor y ve completando los códigos que faltan para que practiques
Tutorial PCA

Termina el laboratorio de tarea y trae tus dudas para la siguiente clase.

3.9 AWS Amazon Web Services (30 min)

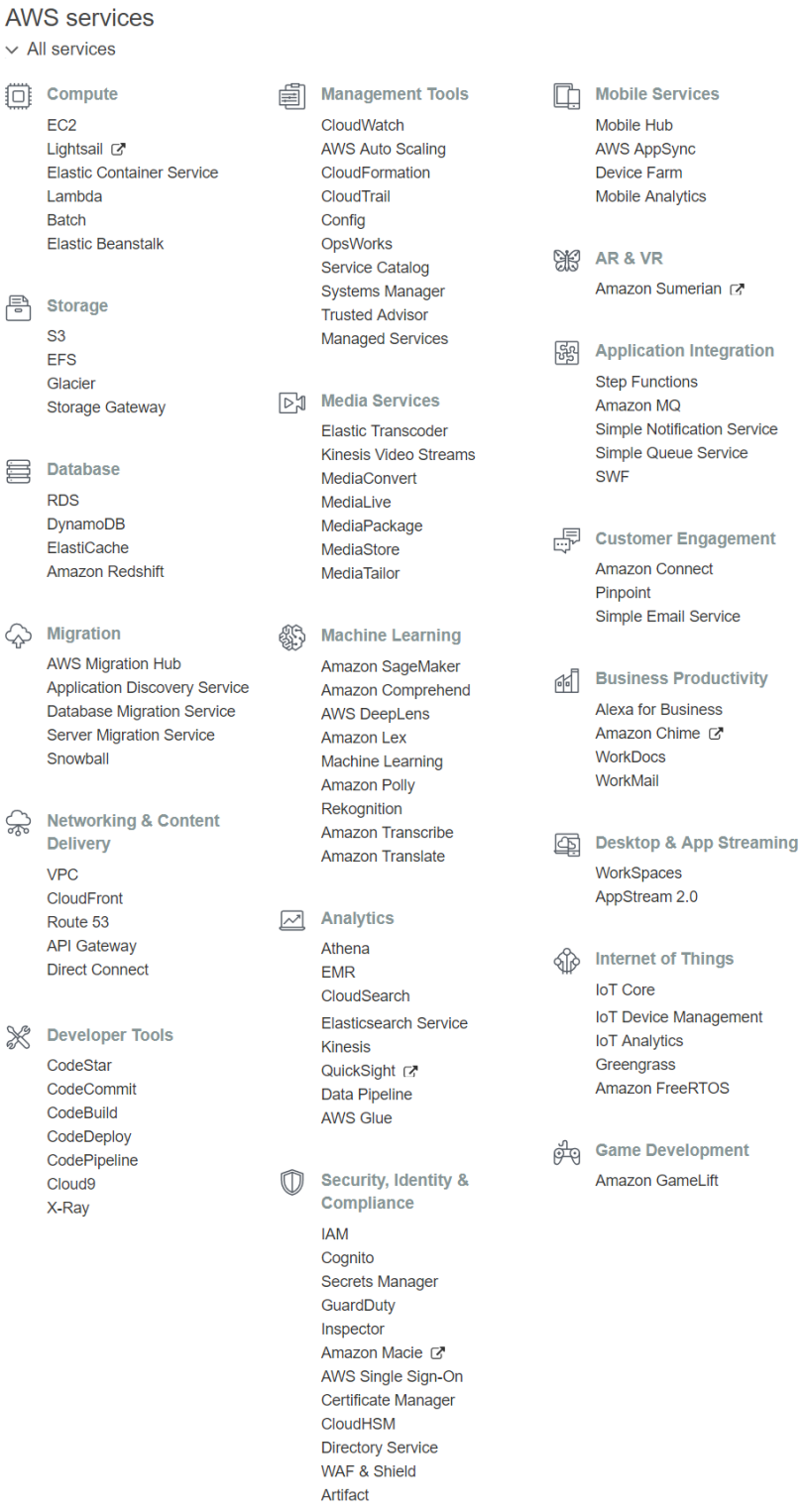
https://awsacademy.instructure.com/

Servicios para todo en la nube, muchos, muchos, muchos, servidores.

¿Que es AWS Glue?

AWS Glue es un servicio ETL (extracción, transformación y carga) completamente administrado que hace que categorizar tus datos, limpiarlos, enriquecerlos y moverlos de manera confiable entre varios almacenes de datos y flujos de datos sea simple y rentable. AWS Glue consta de un repositorio central de metadatos conocido como AWS Glue Data Catalog, un motor ETL que genera automáticamente código Python o Scala y un programador flexible que maneja la resolución de dependencias, la supervisión de trabajos y los reintentos. AWS Glue no tiene servidor, por lo que no hay infraestructura para configurar o administrar.

AWS Glue está diseñado para trabajar con datos semiestructurados. Introduce un componente llamado marco dinámico (dynamic frame), que puede usar en sus scripts ETL. Un marco dinámico es similar a un marco de datos de Apache Spark, que es una abstracción de datos utilizada para organizar datos en filas y columnas, excepto que cada registro se describe a sí mismo, por lo que inicialmente no se requiere ningún esquema. Con marcos dinámicos, obtiene flexibilidad de esquema y un conjunto de transformaciones avanzadas diseñadas específicamente para marcos dinámicos. Puede convertir entre marcos dinámicos y



marcos de datos de Spark, de modo que pueda aprovechar las transformaciones de AWS Glue y Spark para realizar los tipos de análisis que desee.

Puede utilizar la consola de AWS Glue para descubrir datos, transformarlos y ponerlos a disposición para realizar búsquedas y consultas. La consola llama a los servicios subyacentes para orquestar el trabajo necesario para transformar sus datos. También puede utilizar las operaciones de la API de AWS Glue para interactuar con los servicios de AWS Glue. Edite, depure y pruebe su código ETL Python o Scala Apache Spark utilizando un entorno de desarrollo que te sea familiar como por ejemplo Visual Studio Code

Manual AWS Glue

Referencia en sitio web

Registro de estudiantes en AWS:

1. Inscribir el ambiente de Lab de AWS
2. Crear cuenta cuando reciban el correo electrónico.
3. Hacer conexión de prueba.
4. Mostrar módulos

3.9.1 Carga de los datos procesados a su destino

Carga

En este último paso, los datos transformados se mueven desde el área de ensayo a un almacén de datos de destino. Por lo general, esto implica una carga inicial de todos los datos, seguida de una carga periódica de cambios de datos incrementales y, con menos frecuencia, actualizaciones completas para borrar y reemplazar datos en el almacén. Para la mayoría de las organizaciones que usan ETL, el proceso está automatizado, bien definido, continuo y por lotes. Por lo general, ETL se lleva a cabo fuera del horario laboral cuando el tráfico en los sistemas de origen y el almacén de datos está en su punto más bajo.

En caso de que quieras especializarte en AWS aquí hay un par de tutorial enfocado al uso de GLUE
Ver pequeña demo con el profesor.

Actividad 1: Seguir el siguiente tutorial para saber utilizar AWS Glue para cargar, transformar y reescribir datos en AWS S3 para que se puedan consultar y analizar de manera fácil y eficiente.

Web:
https://docs.aws.amazon.com/glue/latest/dg/aws-glue-programming-python-samples-legislators.html

Github:
https://github.com/aws-samples/aws-glue-samples/blob/master/examples/join_and_relationalize.md

Actividad 2: Este script ETL de muestra le muestra cómo aprovechar las características de Spark y AWS Glue para limpiar y transformar datos para un análisis eficiente.

Web:
https://docs.aws.amazon.com/glue/latest/dg/aws-glue-programming-python-samples-medicaid.html
Github:
https://github.com/aws-samples/aws-glue-samples/blob/master/examples/data_cleaning_and_lambda.md

Otra opciones de herramientas para ETLs Enlaces de interés:

1. https://www.ibm.com/cloud/learn/etl#toc-what-is-et-xeCDpL69
2. https://en.wikipedia.org/wiki/Extract,_transform,_load
3. https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl
4. https://aws.amazon.com/es/glue/

3.10 Bases de Database (SQL+joins+NoSQL)

3.10.1 Diseño y creación de bases de datos relacionales como fuentes de información para la creación de modelos inteligentes

Al trabajar en un proyecto de Data Science, Machine Learning, Deep Learning u otro relacionado, el elemento más importante son los datos, que se consideran la entrada principal para todos los algoritmos de Machine o Deep Learning. Mientras trabajamos en nuestros proyectos debemos concentrarnos en nuestra base de datos y debemos verificar si está centralizada o no, si incluye datos corruptos o no, porque los datos corruptos pueden causar redundancia en toda su base de datos y también puede dar los resultados inconsistentes.

Es importante conocer todos estos conceptos básicos de una base de datos antes de profundizar en ciencia de datos , y lo que también es útil para convertirse en Data Scientist o Machine Learning Engineer porque estas son cosas estándar**.**

Conceptos Básicos

Bases de datos: La base de datos se define como las conexiones reales o una conexión de información que tiene alguna asociación semántica. Semántica significa que existe una relación lógica en la base de datos. Por lo tanto, una base de datos es una combinación de datos vinculados que representan una situación del mundo real. Un sistema de base de datos está diseñado para ser creado y poblado con datos para completar un propósito específico.

Sistema administrador de bases de datos: software para almacenar y recuperar datos de forma segura y estructurada. Un sistema de administración de bases de datos (DBMS) ayuda a los usuarios y otras aplicaciones de terceros a almacenar y recuperar datos en sistemas en tiempo real.

Los sistemas de administración de bases de datos se crearon para abordar los siguientes problemas con los sistemas tradicionales de procesamiento de archivos que son compatibles con los sistemas operativos tradicionales:

– Redundancia e inconsistencia de datos
– Dificultad para acceder a los datos
– Aislamiento de datos
– Múltiples archivos y formatos
– Problemas de integridad
– Atomicidad de las actualizaciones
– Acceso simultáneo por múltiples usuarios
– Problemas de seguridad

3.10.2 Extracción de datos contenidos en múltiples tablas en bases de datos relacionales

En esta sección vamos a crear una base de datos mysql en AWS. Opcionalmente, también puedes instalar y configurar las herramientas localmente:

- Python: https://www.python.org/downloads/
- Jupyter Notebook: https://jupyter.org/install
- MySQL community server y MySQL Workbench: https://dev.mysql.com/downloads/mysql/

Para crear una base de datos en AWS sigue el siguiente proceso:

1. Entra a AWS Learner lab con tu cuenta de AWS academy

2. Lanza el laboratorio y entra a la consola de amazon
3. Entra al servicio de base de datos de AWS RBS
4. Crea una nueva base de datos en MySQL con IP pública
5. Seleccionar Free Tier para que no les cueste
6. Asigna una contraseña y deja el usuario como “admin”
7. Copia la dirección de la base de datos y el puerto

Aquí hay un video donde se muestra el proceso. https://youtu.be/-q6m80wKRAs

Una vez creada la base de datos es necesario dar permisos de acceso, para ello hay que ir a las reglas “**inbound rule**”

Agregar una nueva **inbound rule** con las siguientes características:

Mysql/Aurora
Puerto 3306
custom/personalizada
0.0.0.0 /0 anywhere

Ya con esto podrás conectarte a tu bdd desde otras plataformas.

Ahora vamos a seguir un tutorial para

1. Conexión a la base de datos en MySQL utilizando Python
2. Cargar los datos de un csv
3. Entender Comando SQLs desde python
4. Separar las variables independientes y la dependiente
5. Crear el conjunto de prueba y entrenamiento con scikit-learn
6. Aplicar un framework adicional para la construcción de modelos de regresión statsmodels

La base de datos la vas a llenar con el archivo CSV Wine data set. Descargalo para poder hacer el siguiente tutorial: Tutorial python mysql

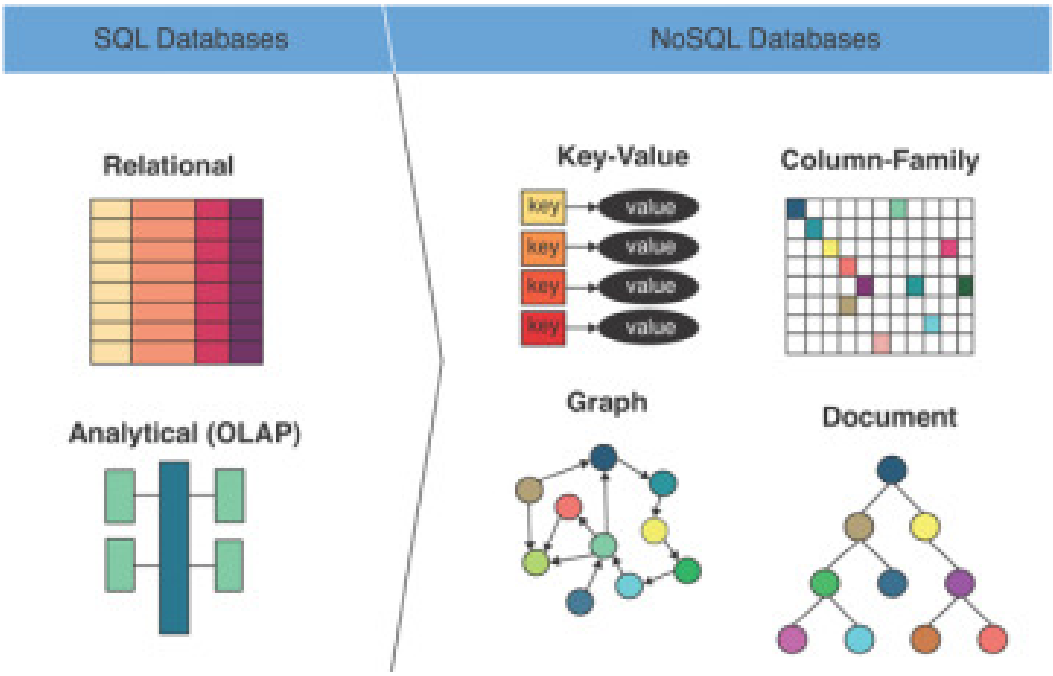
Si deseas conectarte desde la terminal con mysql puedes usar:

sudo mysql -u <usuario> -h <ip> -P <puerto> -p

Por ejemplo:

sudo mysql -u admin -h modulosql.cruoyzidkoee.us-east-1.rds.amazonaws.com -P 3306 -p

3.10.3 Bases de Database (SQL+joins+NoSQL) diseño y creación de colecciones en una base de datos NoSQL



source

Feature	Non Relational Database	Relational Database
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimised for Huge Data	Medium - Large Data Size
Performance	High	Low
Reliability	Poor	Good
Scalability	High	High (but more expensive)

source

Ejemplo de archivos en mongo:

https://www.mongodb.com/basics/json-schema-examples
https://github.com/ozlerhakan/mongodb-json-files/blob/master/datasets/books.json

Práctica con Mongo db:

Para trabajar con este módulo es necesario instalar varias herramientas localmente:

- Mongo DB Community Server: https://www.mongodb.com/try/download/community
- Mongo Compass: https://www.mongodb.com/try/download/compass

Después de instalar las herramientas el primer paso es cargar los datos dentro de un archivo CSV por MongoDB Compass. Para esto se debe crear una base de datos llamada “datasets_ml” y la colección “iris” a partir de ahí importar el archivo CSV para el llenado de la colección donde se almacenará el conjunto de datos. El archivo para importar es:
iris .csv

Una vez cargado el conjunto de datos hay que abrir Jupyter Lab para trabajar con el siguiente notebook:

Tutorial Mongo db

Dentro del tutorial se incluyen los siguientes elementos:

7. Conexión, visualización, selección o creación de base de datos
8. Visualización, selección o creación de la colección
9. Comandos MongoDB
10. MongoDB a DataFrame
11. Construcción del conjunto de datos
12. Creación de un modelo de clasificación con un árbol de decisión
13. Visualización del DT
14. Evaluación del modelo