Machine Learning

Objetivos:

- Conocer en qué consiste el **Aprendizaje Automático** o **Machine Learning (M.L)**

- Obtener una vista panorámica sobre todo el contexto que rodea a M.L: Manejo de los datos, métodos y algoritmos de aprendizaje.

- Introducirse, de forma particular, en la librerías de M.L, **Scikit-Learn** para **Python 3**, realizando una práctica sobre un conjunto de datos de ejemplo.

Primera Parte:

Machine Learning (M.L)

El aprendizaje automático

Scikit-Learn

Contenidos :

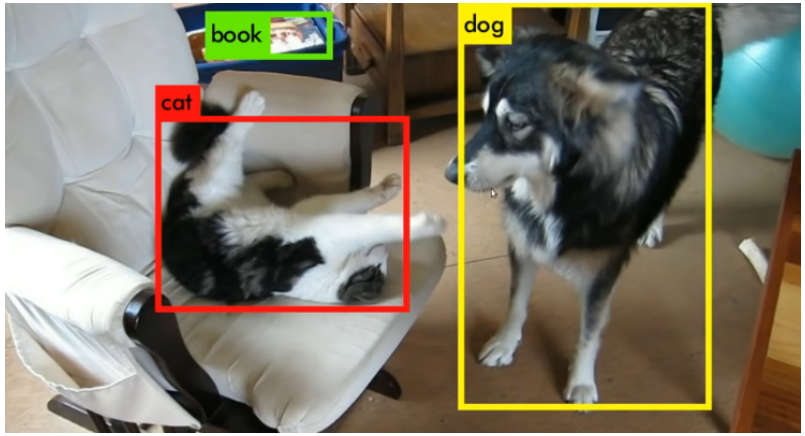
- Machine Learning. Introducción
- La Inteligencia Artificial (I.A) hoy día.
    - Usos increíbles
    - Contexto histórico
        - Hitos
        - Un poco de Historia
- Conceptos
    - Contexto de la Inteligencia Artificial
    - Algunas definiciones.
        - Inteligencia Artificial
        - Machine Learning
        - Redes Neuronales
        - Deep Learning
- Justificación del uso de M.L
- Mitos del M.L
- ¿Qué necesitamos para comenzar?
- Librerías de Machine Learning en Python
- IDES para Machine Learning con Python
- Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir
- El preprocesamiento de los datos.
- Incertidumbre del modelo.
- ¿Cómo preprocesamos los datos con Python?
- Paradigmas del aprendizaje en M.L
    1. Aprendizaje Supervisado.
    2. Aprendizaje No Supervisado.
    3. Aprendizaje por Refuerzo
- Los errores en Machine Learning
    - Underfitting o Subajuste.
    - Overfitting o Sobreajuste
    - Métodos para evitar el Sobreajuste

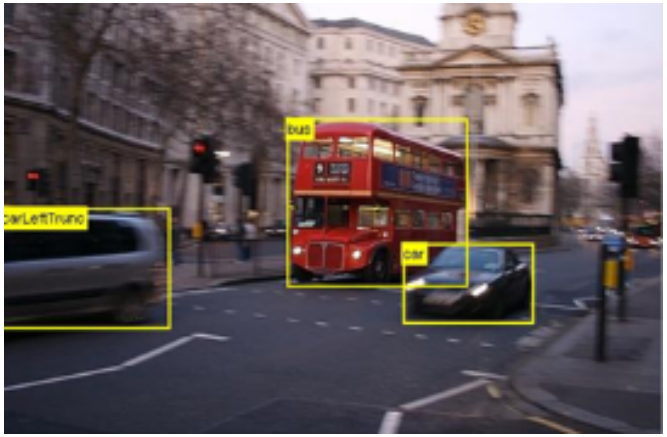
La Inteligencia Artificial (I.A) hoy día.

Usos increíbles:

- Reconocimiento de objetos en los fotogramas de un video en tiempo real. https://www.youtube.com/watch?v=z8RVC7VmY8k

Charla TED sobre YOLO (You Only Look Once).
https://www.youtube.com/watch?v=Cgxsv1riJhI





La Inteligencia Artificial (I.A) hoy día.

Usos increíbles:

- China ha implantado 170.000.000 de cámaras con reconocimiento facial.

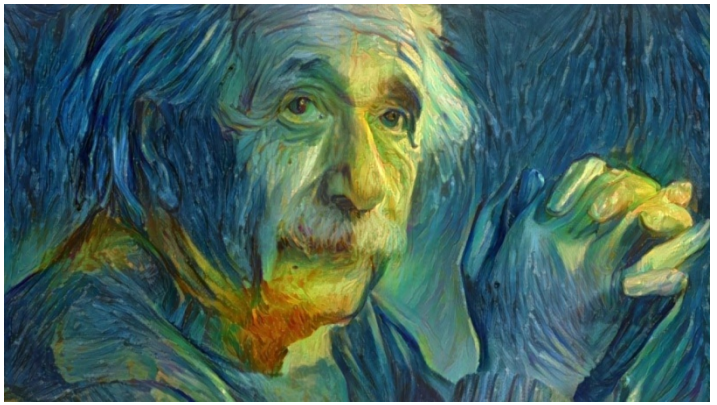
https://retina.elpais.com/retina/2018/04/25/tendencias/1524640135_207540.html

https://www.lavanguardia.com/tecnologia/20171107/432679958989/china-sistema-deteccion-personas.html

https://www.abc.es/sociedad/abci-gran-hermano-chino-todo-camaras-reconocen-caras-segundos-201905190159_noticia.html

- Autopiloto. Transporte autónomo y pilotaje de Drones.
- Generar arte, mediante transferencia de estilo.

https://www.youtube.com/watch?time_continue=4&v=olj6rktnr40



La Inteligencia Artificial (I.A) hoy día.

Otros usos:

- Detección precoz de situaciones (casos de ***Churn***, cuando un cliente va a abandonar la empresa)
- Estimaciones temporales (llamadas a un Call center)
- Sistemas de recomendación (Amazon, o el famoso concurso abierto que hizo Netflix, en 2006, con 1 millos de dólares para el ganador que mejorase su sistema al menos un 10%)
- Campañas de marketing
- Ciudades Inteligentes o la **IoT**. Ej: mediante uso de cámaras, hacer una predicción sobre una aglomeración de personas
- Traductores de idiomas. Han mejorado mucho en poco tiempo
- Filtros anti Spam

La Inteligencia Artificial (I.A) hoy día.
Algunos hitos históricos:

- **1950** Alan Turing, publica el artículo “Computing Machenery and Inteligence”. Donde nace “El test de turing”
- **1951** Marvin Minsky, uno de los padres fundadores de la I.A, crea SNARC (Stochastic Neural Analog Reinforcement Calculator), la primera red neuronal artificial que imitaba a una rata navegando por un laberinto.
- **1952** Arthur Samuel escribe el primer programa de ordenador que jugaba a las damas y era capaz de aprender de sus errores.
- **1959** Frank Rosenblatt diseña el Perceptron, para reconocimiento de caracteres (OCR), modelando de esta forma un sistema biológico, la neurona, que reconoce patrones.
- **1979** En la universidad de Stanford, los estudiantes diseñan un carro que se mueve por una habitación de forma autónoma evitando los obstáculos (tardó 5 h https://www.youtube.com/watch?v=ypE64ZLwC5w&t=)
- **1981** Gerald Dejong crea el concepto de Aprendizaje Basado en la Experiencia, donde un ordenador analiza los datos de entrenamiento creando una regla general y descartando los datos irrelevantes.
- **1985** Terry Sejnowski inventa NetTalk. Un software que aprende a pronunciar palabras de la misma manera que un niño.
- **1990’s** Primeros programas que analizan grandes cantidades de datos y sacan sus propias conclusiones o aprenden de los resultados.
- **1996** Deep Blue de IBM vence al campeón del mundo de ajedrez Gary Kasparov, en una partida.
- **2006** Geoffrey Hinton presenta el concepto de Deep Learning para explicar los nuevos algoritmos.
- **2010** El periférico Kinect de Microsoft es capaz de reconocer 20 características del cuerpo humano, 30 veces por segundo.

La Inteligencia Artificial (I.A) hoy día.
Algunos hitos históricos:

- **2011** El ordenador Watson de IBM vence a dos concursantes en la tercera ronda del concurso estadounidense Jeopardy.
- **2011** Se crea GoogleBrain por Jeff Dean (Google) y Andrew Ng (Unv. Stanford). Una red neuronal gigantesca destinada a detectar patrones en videos y imágenes
- **2012** Laboratorios Google X. Generan un algoritmo que navega por los videos de *youtube* identificando gatos.
- **2014** Eugene Goostman, un programa de *chat bot*, pasa el **Test de Turing** con el 30% de lo jueces que lo evaluaron (Pensaron que era un niño ucraniano de 13 años)
- **2014** Facebook desarrollada DeepFace, para reconocer individuos en las fotos.
- **2015** Amazon lanza su propia plataforma de Machine Learning
- **2015** Azure de Microsoft crea el “Distributed Machine Learning Toolkit”. Aprendizaje Automático en múltiples computadoras.
- **2015** Google entrena un chat bot de I.A, para dar soporte técnico y responder preguntas generales, incluidas de moralidad.
- **2015** OpenAI. (Uno de sus fundadores es Elon Musk) Nace como compañía sin fines de lucro, para promover la I.A con el fin de beneficiar a la humanidad.
- **2015** Carta abierta firmada por 3000 investigadores con respaldo de Stephen Hawking, Elon Musk y Steve Wozniak, advirtiendo del peligro de las armas automáticas que atacan objetivos sin intervención humana.
- **2016** AlphaGo de “Google DeepMing” vence 5 de 5 partidas de **Go**. Juego de mesa más complejo que el ajedrez
- **2017** OpenAI, entrena chat bot que inventan su propio lenguaje y cooperan entre ellos para lograr su objetivo de forma efectiva.
- **2017** Facebook crea otros agentes capaces de negociar y mentir.
- **2017** OpenAI derrota con un algoritmo, a los mejores jugadores online del juego Dota 2

Un poco de Historia

- **80's:** Enfoque Lógico y SBC. Son un grupo tradicional que añadieron lógica difusa. Los sistemas expertos con lógica difusa cogieron mucha fuerza. Pero la incertidumbre frenó a la lógica
- **90's** Enfoque Probabilístico. Gente más matemática que usaban Redes Bayesianas . Algunos resultados son mejores que la lógica. Ej: filtro anti-spam de las redes bayesianas o los traductores de idiomas. Búsqueda en Grafos. Procesos de decisión de Markov (MDP)
- **00's** Enfoque Geométrico. Basados en distancias como: K-NN (supervisado), K-Media (No supervisado), SVM (máquinas de vectores soporte.) Árboles de Decisión (a medio camino entre los Grafos y el enfoque Geométrico), se pueden incluir en el Enfoque Probabilístico.
- **10's** Enfoque conexionista. **(*Redes Neuronales*)** Ya comenzaron en los años 50, pero se estancaron en varias ocasiones por problemas diversos. Aparecen trabajos entre 2006 y 2012, sobre ***Deep Learning***. La llegada del *Data Science*, el avance en computo y potencia, además de los datos (mucha cantidad y barata de almacenar), ayuda a desarrollo, junto con ciertos algoritmos que han aparecido, como campo bien abonado y favorecedor de las Redes Neuronales. Nuevas BD no SQL (no relacionales)

Contexto de la Inteligencia Artificial

Algunas definiciones. Inteligencia Artificial

- **I.A** = (1955) Es la subdisciplina del campo de la Informática, que busca la creación de máquinas que puedan ***imitar*** comportamientos inteligentes (conducir, analizar patrones, reconocer voces, jugar a juegos)

  - DÉBIL. Conjunto muy limitado de tareas. (Todas las I.As actuales)
  - FUERTE. I.As que pueden usarse con una gran cantidad de problemas o dominios diferentes. (Las I.As de Hollywood)

  Subcategorías o áreas de estudio de la I.A: Robótica, Visión, Voz, N.L.P…

"Las máquinas podrán hacer cualquier cosa que hagan las personas, porque las personas no son más que máquinas"

- Marvin Minsky. (1927-2016. uno de los padres de la I.A. fundador de laboratorio de I.A del MIT.)

Algunas definiciones. Machine Learning

- La capacidad de aprendizaje es una de las cosas que nos definen como agentes inteligentes. ***El Machine Learning. o Aprendizaje Automático es una categoría, dentro de la I.A que tiene por finalidad dotar a las máquinas de la capacidad de aprendizaje (1959)***

  ¿Qué es mejor?

  - Programar los movimientos del robot VS programar al robot para que aprenda a moverse.

  - ¿Qué elementos conforman una cara? Vs ¿qué es una cara?

- *Andrew Ng, Profesor de la Universidad de Stanford, la define como la ciencia de hacer que las computadoras actúen sin estar explícitamente programadas.*
- *Los expertos de Nvidia, lo definen como la práctica de usar algoritmos para analizar datos, aprender de ellos y luego hacer una determinación o predicción sobre algo en el mundo.*
- *McKinsey & Company, una de las consultoras más grande en este área, indica que el Aprendizaje Automático o Machine Learning se basa en algoritmos que pueden aprender de los datos sin depender de la programación basada en reglas.*

Algunas definiciones. Ciencia de datos

- Drew Conway creó un simpático diagrama de Venn en el que interrelaciona diversos campos.

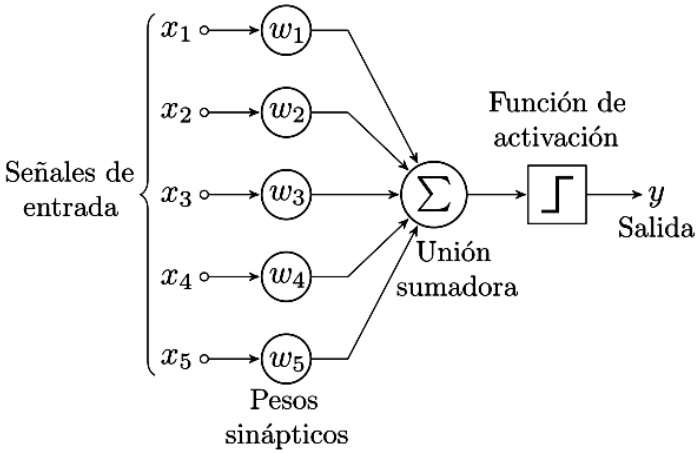
Algunas definiciones. *Redes Neuronales*

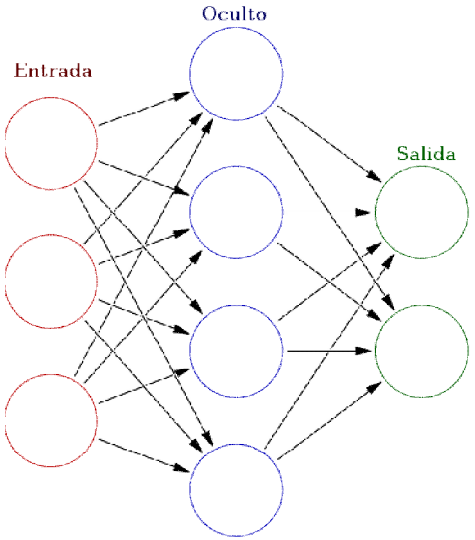
El nombre, como podéis imaginar, viene de la idea de imitar el funcionamiento de las neuronas de los organismos vivos

Está basada en la neurona artificial, el perceptrón de Frank Rosenblatt (1958). Un conjunto de ellas se conectan entre sí para transmitirse información. La información de entrada atraviesa la red neuronal, donde se somete a diversas operaciones, produciendo unos valores de salida.

La respuesta de cada neurona a la estimulación depende de una serie de valores o pesos, que cada neurona actualiza buscando reducir el valor de la función de pérdida. Este proceso se realiza mediante la propagación hacia atrás.

Las redes neuronales se han utilizado para resolver una amplia variedad de tareas, como la visión por computador y el reconocimiento de voz.





Algunas definiciones. *Deep Learning*

- *"Oye Cortana, define Deep Learning ..."*

- *Esto es lo que he encontrado:"es un conjunto de algoritmos de aprendizaje automático (en inglés, machine learning) que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial."*

En una definición mucho más simplista que la de Cortana: Deep Learning es la versión vitaminada de las redes neuronales.

Consiste en aprender de forma jerarquizada por niveles, Primero se empieza con capas que aprenden conceptos concretos (tornillo , rueda,...) y en las capas posteriores se usa la información de los niveles anteriores para aprender conceptos más abstractos (coche, moto…). El número de capas no tiene límite y la complejidad y abstracción aumenta, de ahí el nombre con el que se les ha bautizado.

Se utilizan como técnica potente para el proceso de análisis conocido como BIG DATA.

Un ejemplo de lo que se puede hacer con estos algoritmos de DEEP LEARNING : Coloreas una foto en blanco y negro, determinando los elementos que contiene dicha foto.

Scikit-Learn

Contenidos :

- Machine Learning. Introducción
- La Inteligencia Artificial (I.A) hoy día.
  - Usos increíbles
  - Contexto histórico
    - Hitos
    - Un poco de Historia
- Conceptos
  - Contexto de la Inteligencia Artificial
  - Algunas definiciones.
    - Inteligencia Artificial
    - Machine Learning
    - Redes Neuronales
    - Deep Learning
- Justificación del uso de M.L
- Mitos del M.L
- ¿Qué necesitamos para comenzar?
- Librerías de Machine Learning en Python
- IDES para Machine Learning con Python
- Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir
- El preprocesamiento de los datos.
- Incertidumbre del modelo.
- ¿Cómo preprocesamos los datos con Python?
- Paradigmas del aprendizaje en M.L
  1. Aprendizaje Supervisado.
  2. Aprendizaje No Supervisado.
  3. Aprendizaje por Refuerzo
- Los errores en Machine Learning
  - Underfitting o Subajuste.
  - Overfitting o Sobreajuste
  - Métodos para evitar el Sobreajuste

Justificación del uso de M.L

Diversos algoritmos de búsqueda pueden tomar mucho tiempo en resolverse y cuanto mayor sea el campo de búsqueda, crecerán los tiempos de respuesta, tomando más tiempo de lo que un ser humano puede llegar a vivir.

Para poder resolver este tipo de problemas surgen soluciones, de tipo heurísticas, que intentan atajos hacia el camino correcto para resolver el problema. Éstas pueden obtener buenos resultados en tiempos menores de procesamiento, pero muchas veces pueden llegar a fallar.

Los algoritmos de M.L intentan utilizar menos recursos para «entrenar» grandes volúmenes de datos e ir aprendiendo por sí mismos.

Podemos subdividir el ML en 2 grandes categorías:

- Aprendizaje Supervisado
- Aprendizaje No Supervisado.

Lo que M.L puede o no puede hacer. Mitos

El Aprendizaje Automático se encuentra en las primeras páginas de los periódicos y es objeto de acalorados debates:

*¡los algoritmos de aprendizaje conducen automóviles, traducen el habla y ganan en **Jeopardy,** Amazon Google y Facebook nos estudian y espían para ofrecernos anuncios !*

1. Machine Learning es nuevo. MITO

Machine Learning ha existido desde hace bastante tiempo y ya se usaba para computadoras que jugaban ajedrez, filtros de spam.

2. Las supercomputadoras son lo que permitió Machine Learning

Esto es **VERDAD.** Hace unos años, las máquinas podían aprender y jugar al ajedrez, ya que se compone de un número limitado, aunque gigantesco, de posibilidades y combinaciones, pero los juegos como Go, solo se hicieron accesible para Machine Learning con componentes nuevos y potentes. Todo esto se logro gracias a las “supercomputadoras” ya que permiten que los sistemas procesen rápidamente muchos más datos y esa es parte de la revolución actual de Machine Learning.

Lo que M.L puede o no puede hacer. Mitos

3. Machine Learning es una nueva forma de programación. VERDAD

Mientras que la programación tradicional detalla todas las interacciones e hipótesis para apoyar el comportamiento del sistema, Machine Learning se centra en un algoritmo que faculta al sistema para aprender por sí mismo.

4. Machine Learning es costoso. MITO

Aunque este mito ciertamente tenía alguna base en la realidad en un momento dado, a medida que se desarrollaba el uso de sistemas de Machine Learning, el precio dependía de conocimientos y herramientas que eran prohibitivamente costosos para muchas empresas, pero, al igual que con cualquier tecnología, con el tiempo los costos han disminuido a medida que gana aceptación, y nuevas herramientas entran al mercado.

Lo que M.L puede o no puede hacer. Mitos

5. Machine Learning se puede utilizar para cualquier aplicación. MITO

Machine Learning solo se puede aplicar a tareas en las que existe una gran cantidad de conjuntos de datos de entrada o que se puedan capturar potencialmente.

Pero, a su vez, la idea de que cualquier información puede ser introducida en el sistema, y la máquina producirá automáticamente datos útiles, sin importar que tipo de datos sean, esto es simplemente incorrecto.

Aunque los sistemas basados en Machine Learning a menudo pueden encontrar patrones e ideas que anteriormente estaban ocultos, eso no significa que uno pueda alimentarlos con datos basura sin ningún tipo de configuración o preparación y esperar tener una visión útil que se presente instantáneamente, Machine Learning no funciona de esa manera.

Lo que M.L puede o no puede hacer. Mitos

6. Los datos son fundamentales en Machine Learning. VERDAD

Con computadoras cada vez más potentes, las máquinas ahora pueden procesar más datos, lo que amplía los límites de sus probabilidades cognitivas. La entrada más importante para una herramienta de Machine Learning es la información, no solo cualquier dato, sino los datos correctos.

En lugar de ingerir cualquier cosa, un sistema de Machine Learning necesita información y contenido cuidadosamente seleccionado y de alta calidad, los datos incorrectos proporcionan malos resultados, sin importar el sistema. Un algoritmo es un programa y los programas necesitan buenos datos. Cuando un sistema utiliza Machine Learning, el programa llega a una respuesta a través de aproximaciones continuas y aprende la mejor manera de llegar a esa respuesta haciendo ajustes a la forma en que procesa esos datos, entonces tener los datos correctos es más importante que el algoritmo.

Lo que M.L puede o no puede hacer. Mitos

7. Con más datos se toma mejores decisiones. MITO

Con nuevas y potentes tecnologías, las máquinas pueden procesar gran cantidad de datos, pero las mejores decisiones se basan en los datos más calificados, no es necesario recargar la computadora con demasiada información sin valor, sino más bien concentrarse en el valor agregado.

8. Las computadoras pueden aprender como los humanos. MITO.

Las máquinas ni siquiera están cerca de la forma en que aprenden los chimpancés. Si comparamos el proceso de aprendizaje de una máquina con el de un niño, se hace evidente que Machine Learning todavía está en su infancia, ya que las máquinas, requieren orientación y apoyo en cada paso del aprendizaje.

Por su parte Deep Learning, es poderoso pero está muy lejos de alcanzar la complejidad del cerebro humano o imitar las capacidades humanas, la síntesis creativa de diversos conceptos y fuentes de información del pensamiento humano.

Lo que M.L puede o no puede hacer. Mitos

9. Las computadoras pueden tomar decisiones como los humanos. MITO

Las computadoras solo pueden deducir la hipótesis más lógica de los datos recopilados, analizan los datos, enumeran todas las hipótesis posibles y dan una puntuación de probabilidad a cada uno de ellos, de acuerdo con la experiencia y el conocimiento, entonces, el resultado es más una hipótesis de probabilidades que una decisión razonada.

10. Machine Learning reemplazará a los humanos. MITO

Esta es de las mayores afirmaciones que se hace dentro de Machine Learning ya que se asocia con el desempleo y el miedo al futuro. Ciertamente las computadoras pueden hacer muchas tareas limitadas muy bien, pero todavía no tienen sentido común, tan valioso en los humanos. Es cierto que algunos trabajos no cualificados están en transición a las máquinas, el valor agregado se coloca cada vez más en la ingeniería, donde los seres humanos siempre serán necesarios, como ejemplo tenemos a Alemania, este es el tercer país más robotizado del mundo, y su tasa de desempleo ha disminuido en un 37%.

¿Qué necesitamos para comenzar?

- **Algebra lineal 35%**

  El algebra lineal aparece en todas partes, desde transformación de coordenadas, reducción de dimensiones, algoritmos de regresión lineal, solución de sistemas lineales de ecuaciones, entre muchas otras. Los datos se representan mediante ecuaciones lineales, que se presentan en forma de matrices y vectores, por lo que son estas representaciones las que se trabajan más dentro de esta área. Entender algebra lineal te ayudará a tomar mejores decisiones durante el desarrollo de los modelos de M.L

- **Probabilidad y estadísticas. 25%**

  M.L y las estadísticas no son campos muy diferentes, por su parte la teoría de la probabilidad es un marco matemático para representar afirmaciones inciertas, proporciona un medio para cuantificar la incertidumbre, así como los axiomas para derivar nuevos estados de incertidumbre. Algunas de las teorías estadísticas y de probabilidad fundamentales para M.L son: teorema de Bayes, variables aleatorias, varianza y expectativa, distribuciones estándar.

- **Cálculo. 15%**

  Algunos de los temas necesarios dentro de M.L incluyen cálculo diferencial e integral, derivadas parciales, funciones de valores vectoriales, gradiente direccional, entre otros.

- **Algoritmos y optimizaciones complejas 15%**

  Esto es importante para comprender la eficiencia computacional y la escalabilidad de nuestro algoritmo de M.L y para explotar la dispersión en nuestros conjuntos de datos. Se necesita conocimientos de estructuras de datos, programación dinámica, algoritmo lineales y no lineales, gráficos, entre otros.

- **Otros 10%.** Esto se compone de otros temas matemáticos no cubiertos en las cuatro áreas principales, explicadas anteriormente. En esta categoría se incluye análisis real y complejo, teoría de la información, espacios de función.

¿Qué necesitamos para comenzar?

En el tema de lenguajes de programación. Ejemplo Python.

- ***Tensorflow****: desarrollada por google, para redes neuronales. Google la hizo open source.*
- ***Keras*** *(que se extiende sobre TensorFlow), más fácil con menos código que TensorFlow, pero con toda su potencia.*
- ***Numpy*** *. Manejo de arrays con python, más rápidamente ya que usa C .*
- ***OpenCV****. Para procesar imágenes con I.A*
- ***Scikit-Learn****. Algoritmo de IA como: regresiones lineales, árboles de decisión, Kvecinos cercanos.*
- ***Pandas***, ***SciPy***, *pyBrain, Theano, pylearn2, pyEvolve, Caffe, Milk, mlpy, pyml, ...*

¿Qué necesitamos para comenzar?

En el tema de Plataformas (servidores remotos) con maquinas virtuales:

Microsoft AZURE:

- Batch AI. Cluster de computación para entrenamiento distribuido de los algoritmos

Amazon AWS:

- Amazon SageMaker, similar a de Microsoft

- Google Cloud Platform:

Cloud ML Engine , similar a los anteriores

Plataformas para no programadores: *sin líneas de código.*

- Google ML Kit (Machine learning for mobile developers)

- AZURE Machine Learning Studio

Librerías de Machine Learning en Python

- Librerías para Ciencia de datos:
  - **Pandas**: ofrece estructura de datos y herramientas para manipulación y análisis de datos de manera efectiva. Con esta librería se puede agregar y eliminar columnas fácilmente desde el DataFrame, convertir estructuras de datos en objetos y manejar datos faltantes (**NaN**).
  - **Numpy**: utiliza matrices para sus entradas y salidas, por lo que se puede realizar un procesamiento rápido de matrices.
  - **SciPy**: incluye funciones para algunos problemas matemáticos avanzados, como integrales, ecuaciones diferenciales**, entre otros.**
- Librerías para visualización:
  - **Matplotlib**: es la librería más conocida para la visualización de datos, es ideal para hacer gráficos y tramas.
  - **Seaborn**: está basado en Matplotlib, con esta librería es muy fácil generar varios diagramas como heat maps, series de tiempo. Esta es un librería de nivel superior. Sus estilos predeterminados son mucho más sofisticados que los de matplotlib
  - **Bokeh:** dirigida a visualizaciones interactiva. Es independiente de matplotlib.

Librerías de Machine Learning en Python

- Librerías para Machine Learning:
  - **Scikit-learn**: una de las más populares de Machine Learning, tiene una gran cantidad de características para la minería de datos y el análisis de datos. Contiene herramientas para modelado estadístico, incluida regresión, la clasificación, la agrupación, entre otros. Está construida en NumPy, SciPy y Matplotlib.

    Expone una interfaz concisa y consistente para los algoritmos comunes de Machine Learning, por lo que es sencillo llevar a los sistemas de producción. La librería combina código de calidad y buena documentación, facilidad de uso y alto rendimiento, y es un estándar de la industria para Machine Learning con Python.
  - **Theano**: Librerías que permiten definir, optimizar y evaluar expresiones matemáticas que involucran multidimensiones, lo que puede ser un punto de frustración para algunos desarrolladores en otras librerías. Aprovecha la GPU (unidad de procesamiento gráfico) de la computadora para realizar cálculos intensivos de datos hasta 100 veces más rápido que la CPU. Se integra estrechamente con NumPy.

Librerías de Machine Learning en Python

- Librerías para Machine Learning:
  - **Keras** : librería de código abierto considerada una de las mejores librerías de Machine Learning y está escrita en Python. Se utiliza para construir redes neuronales en un alto nivel de la interfaz. Es minimalista y directa con un alto nivel de expansión.

    En Keras es realmente fácil empezar y continúa con la creación rápida de prototipos, es altamente modular y extensible. A pesar de su facilidad, simplicidad y orientación de alto nivel, Keras sigue siendo lo suficientemente profunda y poderosa como para ser un modelo serio.
  - **TensorFlow** : Fue desarrollado por Google Brain y por consiguiente, casi todas las aplicaciones de Google usan TensorFlow para Machine Learning, si estás usando fotos de Google o búsqueda de voz de Google, indirectamente estás utilizando los modelos creados con TensorFlow. Es una librería de redes neuronales de alto nivel que ayuda a programar las arquitecturas de red al tiempo que evita los detalles de bajo nivel.

    Está escrita principalmente en C++, que incluye los enlaces de Python, por lo que no hay sacrificio con el rendimiento. Una de sus ventajas es el paralelismo. Puedes programar diferentes operaciones en diferentes procesadores como CPU o GPUs.

Librerías de Machine Learning en Python

- Librerías para la Minería de datos y el procesamiento del lenguaje natural:

  Las siguientes herramientas están diseñadas para una variedad de tareas relacionadas, desde extraer información valiosa de sitios web, hasta convertir el lenguaje natural en datos que pueden ser usados.

  - **StatsModels**: es también un módulo de Python que permite a los usuarios explorar datos, estimar estadísticas y modelos , y realizar pruebas estadísticas.
  - **Scrapy**: no es un lenguaje matemático, no realiza análisis de datos, no hace nada que crees que te gustaría hacer en M.L, sin embargo, hace una cosa realmente bien, y es rastrear la web. La web es una gran fuente de datos no estructurados, estructurados y visuales. Originalmente diseñado para ***"raspado web", Scrapy*** .También puede extraer datos de las **API.**
  - **NLTK.** Nautral Language Toolkit y, como su nombre indica, se utiliza para tareas comunes de procesamiento de lenguaje natural simbólico y estadístico. Estas librerías permiten etiquetar texto, identificar entidades con nombre y mostrar árboles de análisis, que son como diagramas de oraciones que revelan partes del habla y dependencias.

IDES para Machine Learning con Python

IDE (Integrated Development Environment) o entorno de desarrollo integrado, es una herramienta de codificación que te permite escribir, probar y depurar el código de una manera más fácil, ya que generalmente se compone de un editor de código fuente, herramientas de automatización de compilación y un depurador.

- **Spyder**: “Scientific Python Development Environment”. Escrito únicamente en Python. Este es uno de los mejores IDEs de Python para ciencia de datos y si nunca has trabajado con un IDE este podría ser tu mejor acercamiento. IDE muy simple y liviano con documentación detallada y bastante fácil de instalar. Código abierto, resaltado de sintaxis, finalización de código y exploración de variables….etc
- **PyCharm**. Muy famoso en el mundo profesional. Realizado por la compañía JetBrain. Tiene dos ediciones diferentes: Community Edition, a la que todos podemos tener acceso esencialmente gratis y la segunda es Professional Edition. Interfaz de usuario increíble y personalizable. Finalización del código, sangría automática y el formato del código entre otras muchas características.
- **Jupyter Notebook**. Aplicación web basada en la estructura servidor-cliente que permite crear y manipular documentos portátiles, o simplemente “cuadernos”. Es de código abierto, hasta 40 idiomas. Crear y comparte los documentos con ecuaciones, visualización y lo más importante, códigos en vivo…etc

IDES para Machine Learning con Python

- **Rodeo**. Diseñado expresamente para Machine Learning y análisis de datos en Python, fue desarrollado por Yhat y utiliza el núcleo de IPython.

- **Geany**: Una de las mejores soluciones para IDE de peso ligero, teniendo una configuración de tamaño muy pequeño. Está escrito en C y C++. Admite resaltado de la sintaxis y la numeración de líneas. Finalización del código, cierre automático de llaves…etc

- **Atom**. IDE de código abierto desarrollado por Github. Una de las mejores ventajas de Atom es su comunidad, principalmente debido a las constantes mejoras y complementos que desarrollan para personalizar su IDE y mejorar su flujo de trabajo.

Scikit-Learn

Contenidos :

- Machine Learning. Introducción
- La Inteligencia Artificial (I.A) hoy día.
  - Usos increíbles
  - Contexto histórico
    - Hitos
    - Un poco de Historia
- Conceptos
  - Contexto de la Inteligencia Artificial
  - Algunas definiciones.
    - Inteligencia Artificial
    - Machine Learning
    - Redes Neuronales
    - Deep Learning
- Justificación del uso de M.L
- Mitos del M.L
- ¿Qué necesitamos para comenzar?
- Librerías de Machine Learning en Python
- IDES para Machine Learning con Python
- Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir
- El preprocesamiento de los datos.
- Incertidumbre del modelo.
- ¿Cómo preprocesamos los datos con Python?
- Paradigmas del aprendizaje en M.L
  1. Aprendizaje Supervisado.
  2. Aprendizaje No Supervisado.
  3. Aprendizaje por Refuerzo
- Los errores en Machine Learning
  - Underfitting o Subajuste.
  - Overfitting o Sobreajuste
  - Métodos para evitar el Sobreajuste

Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir

M.L salió hace tiempo de los laboratorios. Los gigantes de la tecnología lo llevan usando unos cuantos años. Amazon, Google Maps, Facebook, Instagram, Twitter. Todos ellos sacan beneficio de una tecnología que algunos también desean adoptar. Pero las empresas promedio se enfrentan a muchos desafíos para comenzar a utilizar Machine Learning, y en ocasiones esto se debe a que no saben cómo funciona exactamente.

Nosotros no queremos que nos pase eso y por esa razón vamos a seguir un flujo de trabajo en donde se debe cumplir cada una de las etapas.

1. **Definir el Objetivo**. Lo primero que debes realizar es seleccionar el objetivo, qué es lo que quieres lograr con M.L. Para esto se debe ser lo más objetivo posible de acuerdo las características de tu empresa o requerimiento así como también la información que puedes conseguir. A veces definir un objetivo no es están fácil o evidente, por lo que en ocasiones, es en este paso en donde se lleva gran parte de nuestro tiempo.
2. **Recolectar los datos.** Este paso es relativamente fácil, ya que probablemente estos datos ya los tengamos disponibles, pero si por el contrario no están, tendrás que ver cómo recolectarlos y sobretodo esperar un tiempo prudencial para poder obtener suficientes y utilizarlos con M.L. Recuerda que no todos los datos son útiles para manipularlos.

Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir

3. **Preparar los datos**. A este paso se le conoce como **preprocesamiento de datos**, y es fundamental para cualquier análisis de Machine Learning. En este punto limpiamos los datos, y los formateamos para que estén acorde con el algoritmo a utilizar. También podemos verificar si necesitamos más datos o si por el contrario debemos desechar algunos porque no son necesarios o no están recolectados correctamente. Este paso es muy importante y no se debe omitir.

4. **Seleccionar el algoritmo.** De acuerdo a los datos ya preprocesados, podemos definir que algoritmo es el más adecuado implementar, evaluando el objetivo que definimos en el primer paso

5. **Entrenar el modelo.** Se inicia el proceso de entrenamiento del modelo, con los datos ya preprocesados, que serán divididos en dos partes:

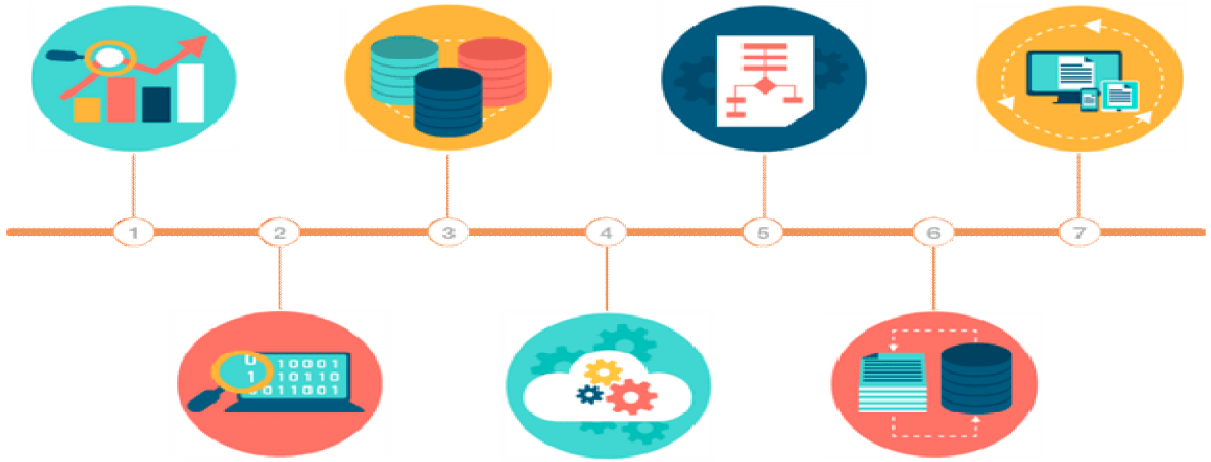
   - Un conjunto de datos usados para entrenar el modelo.

   - Otro conjunto de datos que usamos posteriormente para evaluarlo.

Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir

6. **Evaluar el modelo**. Durante la evaluación se introducen los datos que no utilizamos anteriormente y evaluamos los resultados obtenidos. En este punto es probable que el resultado obtenido no sea adecuado e incluso completamente erróneo por lo que se deberá volver al punto anterior de entrenamiento del modelo y cambiar los ajustes introducidos. Las etapas anteriores pueden repetirse hasta que se encuentren resultados satisfactorios.

7. **Realizar la predicción.** Una vez que se ha obtenido un modelo adecuado y con el que hayamos obtenido resultados satisfactorios, ya podemos implementarlo para realizar las predicciones, en este caso ya podemos introducir nuevos datos en el modelo y obtener los resultados adecuados.

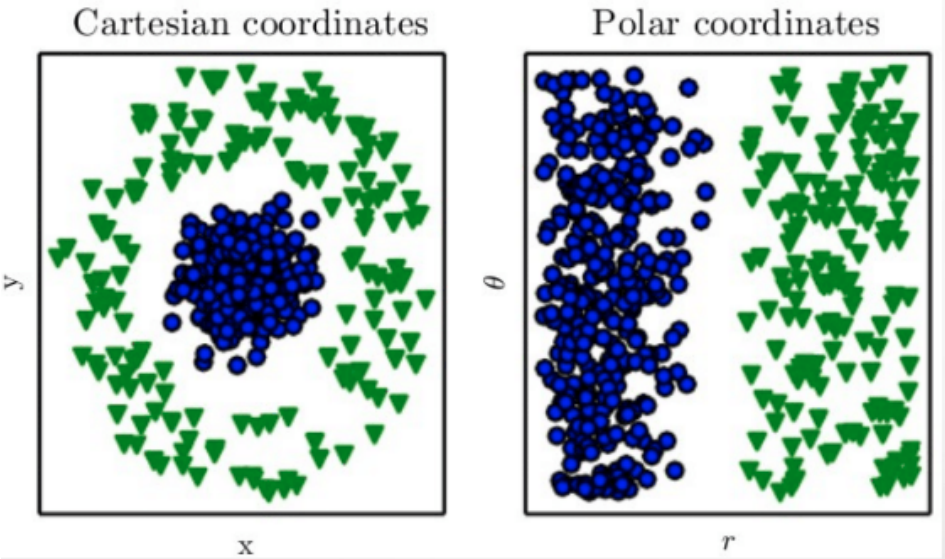


El preprocesamiento de los datos:

- Los Datos usados para entrenar al algoritmo o agente inteligente, deben ser **representativos**
- Deben usarse una **cantidad** de datos suficiente. En algunos casos, esta cantidad puede ser muy grande (Big Data).
- Debe **eliminarse el ruido** para adecuarlos al algoritmo y mejorar el rendimiento.

- **La Representación del conocimiento**.

  Va a determinar fuertemente la forma de aprender. Ejemplo: uso de coordenadas cartesianas o polares para una serie de puntos que se separan por una recta.



Ojo con los problemas de sesgo en los datos (partes importantes de datos no entrenadas u omitidas). Dos ejemplos: en una presentación de un sistema de reconocimiento de voz, se olvidaron de entrenarlo con voces de mujeres….O lo que le sucedió de forma ofensiva a personas afroamericanas que fueron confundidas con monos por Google Photos.

Incertidumbre del modelo. Se debe a:

- **Ruido** en los datos, por causas externas: Aleatorio, esporádico y no intencionado, limitaciones de las medidas que acumulan el error (sensores, precisión de los aparatos). Nota: el enfoque de la lógica, no trata bien el ruido ni los errores. Es muy rígida.
- **Errores**: porcentaje de ejemplos mal clasificados.
- El pobre **rendimiento del modelo** puede deberse a que el modelo es demasiado simple para describir el objetivo, o, por el contrario, que el modelo sea demasiado complejo par expresar el objetivo. Es en este momento que se debe tener claro los conceptos de sobreajuste y subajuste o “overfitting” y “underfitting”

¿Cómo preprocesamos los datos con Python? Podemos Usar Pandas

- Los arrray de numpy son matrices en donde no se tiene las etiquetas de las columnas ni filas, por lo que utilizar en la ciencia de datos o M.L hace que sea muy difícil. Además un array de numpy es de un tipo de datos concreto.
- Por su parte, Pandas estructura los datos usando índices de filas y columnas. Los **DataFrame** son estructuras de dos dimensiones que cuentan con índices tanto en las columnas como en las filas, manipulables por el programador. Los datos son heterogéneos y el tamaño puede ser mutable.
- Objeto **DataFrame** rápido, eficiente con indexación predeterminada y personalizada.
- Herramientas para cargar datos en objetos de datos en memoria desde diferentes formatos de archivo.
- Etiquetado, corte, indexación
- Alineación de datos y manejo integrado de datos faltantes.
- Remodelación y giro de conjuntos y subconjunto de grandes conjuntos de datos.
- Las columnas de una estructura de datos se pueden eliminar o insertar.
- Agrupa por datos para agregación y transformaciones.
- Alto rendimiento de fusión y unión de datos.
- Funcionalidad de la serie de tiempo.

¿Cómo convertimos la información en conocimiento con M.L?

L os mecanismos que permiten procesar toda la información nueva que recibimos para convertirla en conocimiento, son los denominados:

Paradigmas del Aprendizaje. Conocerlos es fundamental dentro del M.L

Los Grandes paradigmas del aprendizaje se pueden clasificar:

1. **Aprendizaje Supervisado.**
2. **Aprendizaje No Supervisado.**
3. **Aprendizaje Reforzado.**

Scikit-Learn

Contenidos :

- Machine Learning. Introducción
- La Inteligencia Artificial (I.A) hoy día.
  - Usos increíbles
  - Contexto histórico
    - Hitos
    - Un poco de Historia
- Conceptos
  - Contexto de la Inteligencia Artificial
  - Algunas definiciones.
    - Inteligencia Artificial
    - Machine Learning
    - Redes Neuronales
    - Deep Learning
- Justificación del uso de M.L
- Mitos del M.L
- ¿Qué necesitamos para comenzar?
- Librerías de Machine Learning en Python
- IDES para Machine Learning con Python
- Cómo hacer ciencia de datos usando Machine Learning. Pasos a seguir
- El preprocesamiento de los datos.
- Incertidumbre del modelo.
- ¿Cómo preprocesamos los datos con Python?
- Paradigmas del aprendizaje en M.L
  1. Aprendizaje Supervisado.
  2. Aprendizaje No Supervisado.
  3. Aprendizaje por Refuerzo
- Los errores en Machine Learning
  - Underfitting o Subajuste.
  - Overfitting o Sobreajuste
  - Métodos para evitar el Sobreajuste

Paradigmas del aprendizaje en M.L

1. Aprendizaje Supervisado.

Mediante observación se generaliza un conocimiento. Es decir, el resultado que se pretende obtener, se le pasa al algoritmo, para que éste aprenda. Los modelos se construyen a partir de los algoritmos de M.L y características o atributos de los datos de entrenamiento, para que podamos predecir el valor utilizando otros valores obtenidos a partir de datos de entrada.

Los principales tipos de algoritmos de aprendizaje supervisado incluyen:

- **Regresión** (clases continuas)

- **Clasificación** (clases discretas)

- **Caracterización** (típico del enfoque lógico, o hoy día en **Deep Learning**, Ej.: devuelve una descripción de lo que está viendo en una foto.

Paradigmas del aprendizaje en M.L

Algoritmos de Aprendizaje Supervisado.

- **Regresión**
  - Lineal Regression
  - Polynomial Regression
  - Support Vector Regression (SVR)
  - Decision Tree Regression
  - Random Forest Regression

-Simple(sólo una variable independiente)
-Múltiple (dos o más variables independientes)

- **Clasificación** (clases discretas)
  - Logistic Regression
  - K-Nearest Neighbors
  - Support Vector Machine (SVM)
  - Decision Tree Classification
  - Random Forest Classification

Paradigmas del aprendizaje en M.L

1. Aprendizaje Supervisado.

Consideraciones

La implementación exitosa de algoritmos de aprendizaje supervisado, requiere gran cantidad de tiempo y la experiencia técnica de equipos especializados con el fin de construir, escalar y desplegar modelos predictivos precisos.

Dado que los modelos de aprendizaje supervisado hacen predicciones del mundo real, basado en los datos del pasado, los modelos deben ser reconstruidos periódicamente con el fin de mantener sus predicciones sin que se conviertan en obsoletas ya que en ocasiones el comportamiento de los datos puede cambiar.

Es importante tener en cuenta que solo funciona si su conjunto de datos históricos contiene valores reales para el resultado que intenta predecir.

Paradigmas del aprendizaje en M.L

2. Aprendizaje No Supervisado.

Proporciona conocimiento únicamente de los datos de entrada, sin proporcionar al sistema los resultados que se pretenden obtener.

Ejemplo: la agrupación o clustering.

Supongamos que tenemos un espacio construido que representa a hombres con gafas, otro que representa a hombres sin gafas y otro que representa a mujeres sin gafas:

Paradigmas del aprendizaje en M.L

2. Aprendizaje No Supervisado.

"El aprendizaje No Supervisado es la llave de la verdadera Inteligencia Artificial" -Yann LeCun

En el aprendizaje no supervisado, un algoritmo agrupa los datos en conjuntos que no están etiquetados, en función de algunas características comunes, subyacentes en los datos. Esta función puede ser útil para descubrir la estructura oculta de los datos y para tareas como la detección de anomalías.

Debido a que no hay etiquetas, no hay forma de evaluar el resultado. Esto es una diferencia clave frente a los algoritmos de aprendizaje supervisado

Paradigmas del aprendizaje en M.L

2. Aprendizaje No Supervisado.

El aprendizaje no supervisado se puede clasificar en dos categorías:

- Aprendizaje no supervisado paramétrico.

El modelo asume que los datos de muestra provienen de una población que sigue **una distribución** de probabilidad basada en un conjunto fijo de parámetros. Eso significa que, si conoce el promedio y la desviación estándar y que la distribución es normal, conoces la probabilidad de cualquier observación futura.

- Aprendizaje no supervisado no paramétrico

Los datos se distribuyen en grupos, donde cada grupo dice algo acerca de las categorías y clases presentes en los datos. También se los conoce como un método libre de distribución (no requieren que el modelo haga suposiciones sobre la distribución de la población)

Paradigmas del aprendizaje en M.L

2. Aprendizaje No Supervisado. AGRUPAMIENTO O CLUSTERING

El agrupamiento puede considerarse el problema de aprendizaje no supervisado más importante.

Una definición amplia de clustering podría ser, el proceso de organizar objetos en grupos cuyos miembros son similares de alguna manera.

Los algoritmos de agrupación se pueden clasificar:

• Agrupamiento exclusivo:

Los datos se agrupan de manera exclusiva, de modo que si un cierto punto de datos pertenece a un grupo definido, entonces podría no ser incluido en otro clúster.

• Superposición de clústeres:

Usa conjuntos difusos para agrupar datos, de modo que cada punto puede pertenecer a dos o más clústeres con diferentes grados de pertenencia. En este caso, los datos se asociarán con un valor de pertenencia apropiado.

•Agrupamiento jerárquico:

Se basa en la unión entre los dos clústeres más cercanos. La condición de inicio se realiza estableciendo cada punto de datos como un clúster, después de algunas iteraciones alcanza los clústeres finales deseados.

•Agrupación probabilística: utiliza un enfoque probabilístico.

Paradigmas del aprendizaje en M.L

2. Aprendizaje No Supervisado. AGRUPAMIENTO O CLUSTERING

Los algoritmos de agrupamiento más comunes incluyen:

• Agrupación de clústeres (K-Means Clustering): divide datos en clústeres distintos según la distancia al centroide de un clúster.

• Agrupamiento jerárquico (Hierarchical Clustering) : crea una jerarquía multinivel de clústeres mediante la creación de un árbol de clústeres.

• Mezclas de modelos gaussianos: clúster de modelos como una mezcla de componentes de densidad normal multivariante.

Paradigmas del aprendizaje en M.L

2. Aprendizaje No Supervisado. Consideraciones

Los algoritmos no supervisados segmentarán a los clientes en grupos grandes en lugar de tratarlos como individuos y permitir que las empresas entreguen comunicaciones altamente personalizadas, por lo que significa que el aprendizaje no supervisado es menos aplicable a contextos del mundo real.

El aprendizaje no supervisado pretende descubrir patrones previamente desconocidos en los datos, por lo que el mejor momento para utilizar el aprendizaje no supervisado es, cuando no se tienen datos sobre los resultados deseados, como determinar un mercado objetivo para un producto completamente nuevo, que tu empresa nunca haya vendido anteriormente.

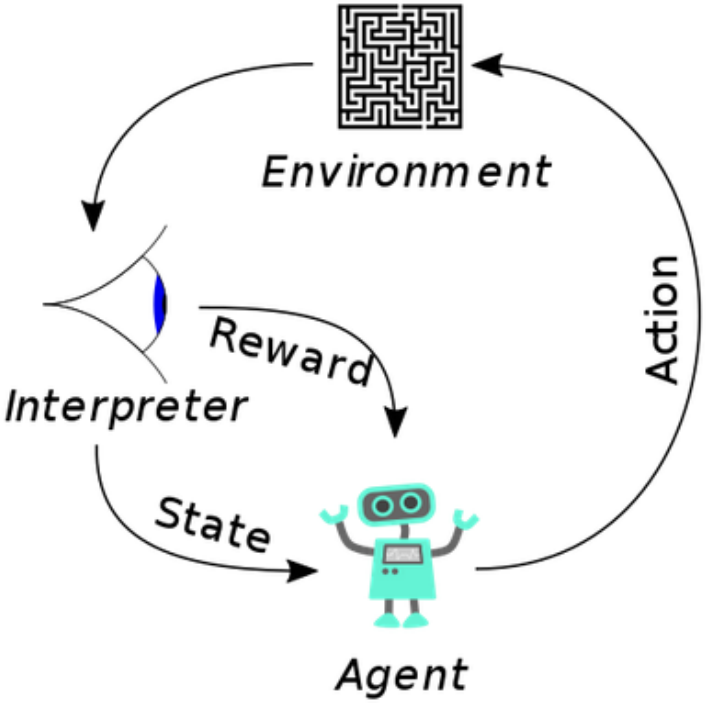
Los algoritmos de agrupamientos se pueden aplicar en muchos campos, por ejemplo:

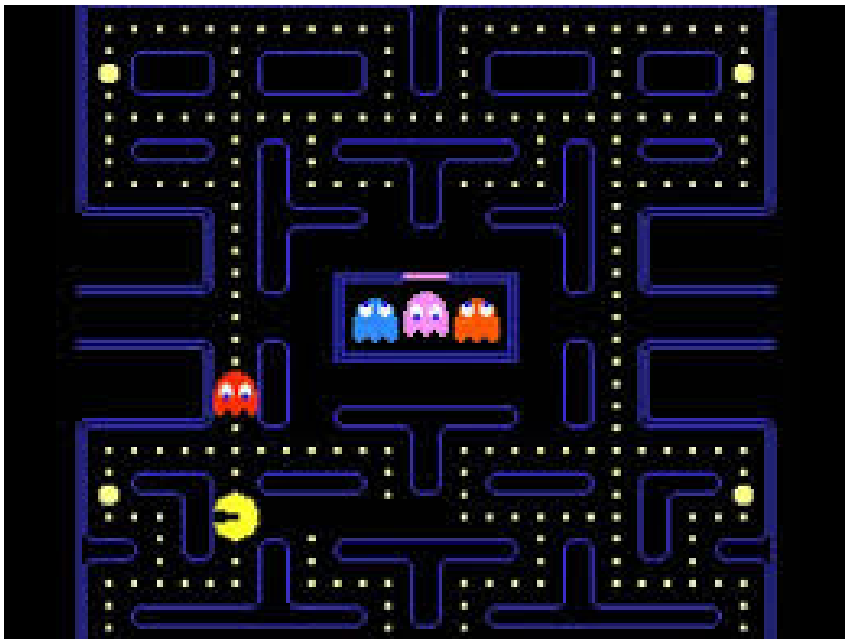
- Marketing: encontrar grupos de clientes con un comportamiento similar.
- Biología: clasificación de plantas y animales dadas sus características.
- Seguros: identificar fraudes.
- Estudios de terremotos: aglomeración de epicentros de terremotos observados para identificar zonas peligrosas.

Paradigmas del aprendizaje en M.L

3. Aprendizaje Reforzado (o Aprendizaje por Refuerzo)

Se basa en aplicar el principio a la psicología conductista a las máquinas, con el fin de que puedan aprender por sí mismas. Es decir, que las recompensas van reforzando ciertos comportamientos, porque se tiende a maximizar la ***Recompensa Acumulada Esperada*****.** Ejemplo: Pac-Man https://www.youtube.com/watch?v=QilHGSYbjDQ





Paradigmas del aprendizaje en M.L

Cuando DeepMind popularizó el aprendizaje por refuerzo

La I.A SNARC que creó Marvin Minsky en 1951 empleaba una forma simplificada de aprendizaje reforzado que prometía, pero…

….hasta 65 años después no se pudo escalar a situaciones más complejas:

En 2016, DeepMind presentó ante el mundo a AlphaGo, una IA que, tras ser entrenada durante varios meses en el análisis de miles de partidas jugadas por humanos fue capaz de batir a un campeón humano de Go

Pero, un año más tarde, DeepMind presentaba una nueva IA, AlphaGo Zero, que con menos de 3 días de entrenamiento fue capaz de ganar 100 veces seguidas a su predecesora.

¿Residía el secreto en una mayor potencia de procesamiento?

No, la clave consistió en que AlphaGo Zero, mediante **Aprendizaje por Refuerzo**, jugó millones de partidas contra sí misma, hasta que aprendió a maximizar su *recompensa acumulada esperada*.

Los errores en Machine Learning

En M.L no se puede construir un modelos 100% preciso ya que nunca pueden estar libres de errores. Comprender cómo las diferentes fuentes de error generan **bias** y **varianza** nos ayudara a mejorar el proceso de ajuste de datos, lo que resulta en modelos más precisos. Adicionalmente también evitará el error de ***Overfitting y Underfitting.***

El error de predicción, para cualquier algoritmo de Machine Learning se puede dividir en tres partes***:***

- ***Error de BIAS (Sesgo)***
- ***Error de varianza***
- ***Error irreducible.***

  El error irreducible, independientemente del algoritmo que se use, no se puede reducir. También se le conoce como ***ruido***

Los errores en Machine Learning

Error de bias

Es la diferencia entre la predicción esperada de nuestro modelo y los valores verdaderos.

• **Bajo bias:** sugiere menos suposiciones sobre la forma de la función objetivo. Los algoritmos de Machine Learning con bajo bias incluyen: árboles de decisión, Kvecinos más cercanos y máquinas de vectores de soporte.

• **Alto bias:** sugiere más suposiciones sobre la forma de la función objetivo. Por su parte, los algoritmos con alto bias se incluyen: regresión lineal, análisis discriminante lineal y regresión logística.

Los errores en Machine Learning

Error de varianza

Se refiere a cuánto cambiará, la estimación de la función objetivo, si se utiliza diferentes datos de entrenamiento. Idealmente no debería cambiar demasiado, de un conjunto de datos de entrenamiento a otro, lo que significa que el algoritmo es bueno.

• **Varianza baja:** sugiere pequeños cambios en la estimación de la función objetivo, con respecto a los cambios en el conjunto de datos de capacitación. Los algoritmos de Machine Learning con baja varianza incluyen: regresión lineal, análisis discriminante lineal y regresión logística.

• **Alta varianza:** grandes cambios en la estimación de la función objetivo, con respecto a los cambios en el conjunto de datos de capacitación. Por su parte, los algoritmos con alta varianza son: árboles de decisión, k-vecinos más cercanos y máquinas de vectores de soporte

Los errores en Machine Learning

La compensación Bias-Varianza o Trade-off

El objetivo de cualquier algoritmo supervisado de Machine Learning es lograr un **bias** bajo y una baja **varianza**, a su vez, el algoritmo debe lograr un buen rendimiento de predicción.

Los algoritmos de baja varianza (alto bias) tienden a ser menos complejos, con una estructura subyacente simple o rígida. Entrenan modelos que son consistentes, pero inexactos en promedio. Estos incluyen algoritmos paramétricos o lineales, como la regresión lineal y el ingenuo Bayes.

Los algoritmos de bajo bias (alta varianza) tienden a ser más complejos, con una estructura subyacente flexible. Entrenan modelos que son precisos en promedio pero inconsistentes. Estos incluyen algoritmos no lineales o no paramétricos, como árboles de decisión y k-vecinos más cercanos.

No hay escapatoria a la relación entre el bias y la varianza en Machine Learning, aumentar el bias disminuirá la varianza, aumentar la varianza disminuirá el bias.

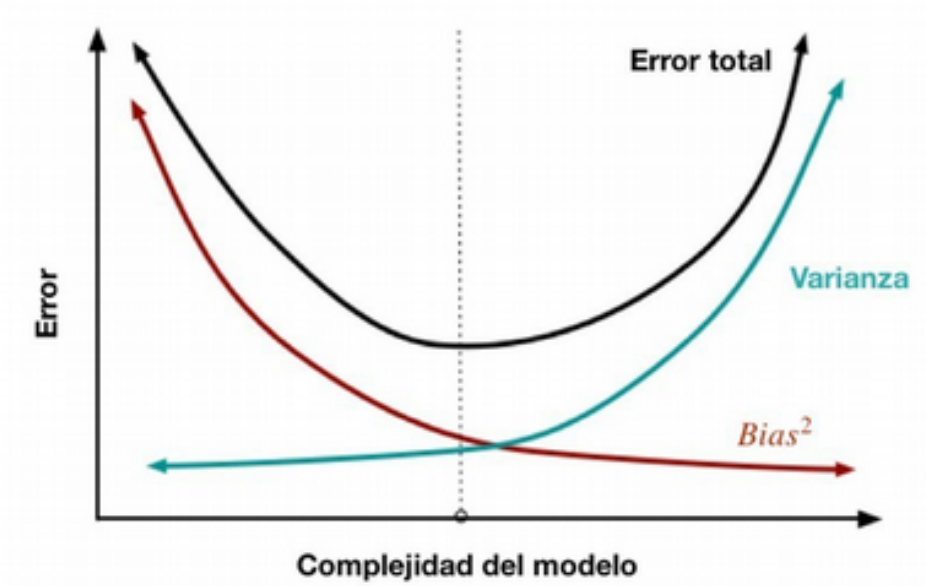
Los errores en Machine Learning

Error total

Comprender el **bias** y la **varianza** es fundamental para comprender el comportamiento de los modelos de predicción.

El punto ideal para cualquier modelo es el nivel de complejidad en el que el aumento en el **bias** es equivalente a la reducción en la **varianza**.

Para construir un buen modelo, necesitamos encontrar un buen equilibrio entre el **bias** y la **varianza** de manera que minimice el error total.



Introducción al Sobreajuste y Subajuste para Machine Learning. Overfitting y Underfitting .

- **Underfitting**

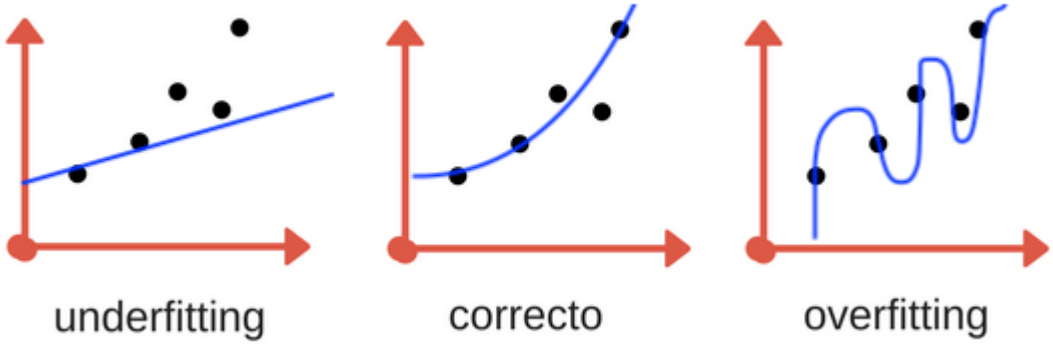
La máquina no es capaz de generalizar conocimiento

- **Overfitting**

La máquina se ha especializado en un conocimiento muy concreto.

Overfitting y Underfitting .

- El modelo del medio muestra una curva de ajuste bastante buena, cubre la mayoría de los puntos en el gráfico y también mantiene el equilibrio entre el bias o sesgo y la varianza.



- Subajuste se refiere a un modelo que no puede modelar los datos de entrenamiento no generalizar a nuevos datos, esto ocurre cuando el modelo de Machine Learning es muy simple.
- El sobreajuste se refiere a un modelo que ajusta los datos de entrenamiento demasiado bien. Esto ocurre cuando un modelo aprende el detalle, incluyendo el ruido en los datos de entrenamiento

Overfitting y Underfitting .

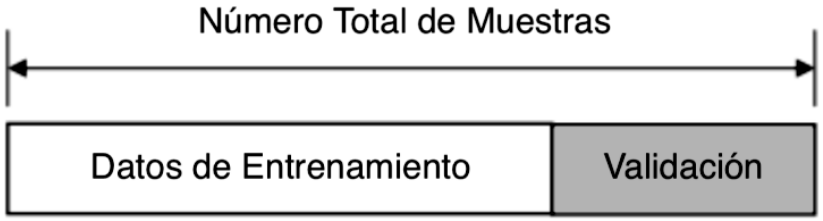
Métodos de evitar el sobreajuste de los modelos de Machine Learning

1. Usar más información para entrenamiento.

   El uso de un gran conjunto de datos de capacitación generalmente ayuda al modelo de M.L a elegir la señal de manera eficiente, sin embargo, esta técnica puede no funcionar todas las veces. Si agregamos muchos datos ruidosos y los datos relevantes son escasos, incluso tener una gran cantidad de datos totales, no ayudará al modelo a mejorar la precisión.

2. Técnica de validación cruzada.

   La validación cruzada es un estándar de oro en Machine Learning, para estimar la precisión del modelo en datos no vistos. Usar un conjunto de datos de validación cuya división se suele hacer del 80% para entrenar y 20% para validar. Una forma estándar de encontrar un error de predicción fuera de muestra es usar una validación cruzada de k iteraciones, con k = 5 por ejemplo.



$$E = \frac{1}{K}\sum_{i=1}^{K} E_i\,.$$

Overfitting y Underfitting .

Métodos para evitar el sobreajuste de los modelos de Machine Learning

3. Detección temprana.

Cuando se entrena iterativamente un modelo de M.L, observarás que hasta cierto número de iteraciones, el rendimiento del modelo mejora. Después de cierto punto, tendrá un rendimiento bajo en los conjuntos de datos de prueba. Por lo tanto, deben detenerse las iteraciones de entrenamiento, antes de que exista un ajuste excesivo.

4. Regularización

La regularización se hace para simplificar el modelo. La técnica utilizada depende del tipo de modelo. Por ejemplo, si el modelo es un árbol de decisión, la regularización podría ser podar el árbol.