Big Data, Machine Learning & Business Intelligence

Por: Carlos Carreño

ccarreno@cienciadedatos.es

Unidad 1 Inteligencia Artificial y Machine Learning

- ¿Qué es la inteligencia artificial?
- ¿Qué es Machine Learning?
- Actualidad de la inteligencia artificial Machine Learning
- Algoritmos de Machine Learning
  - Aprendizaje supervisado
  - Aprendizaje no supervisado
  - Aprendizaje por refuerzo
- Lenguajes de Machine Learning
- Certificación Internacional
- Machine Learning
  - Dimensionalidad y reducción

Que es la Inteligencia Artificial?

- La inteligencia artificial (**IA**) se refiere al estudio, al desarrollo y a la aplicación de **técnicas informáticas** que les permiten a las **computadoras** adquirir ciertas **habilidades propias de la inteligencia humana**. Por ejemplo:
  - ❑Entender las situaciones y los contextos.
  - ❑Identificar objetos y reconocer sus significados.
  - ❑Analizar y resolver problemas.
  - ❑Aprender a realizar nuevas tareas.
  - ❑Comprender el lenguaje natural (Natural Language Processing).
  - ❑Reconocer imágenes (Computer Vision).

Actualidad de la IA

- En los 80s la IA se implementaba en sistemas basado en Reglas.
- Una regla es especificar que si ocurre una o mas condiciones ejecuta una o mas acciones.

"Si ocurre X cosa entonces debes hacer Z cosa"

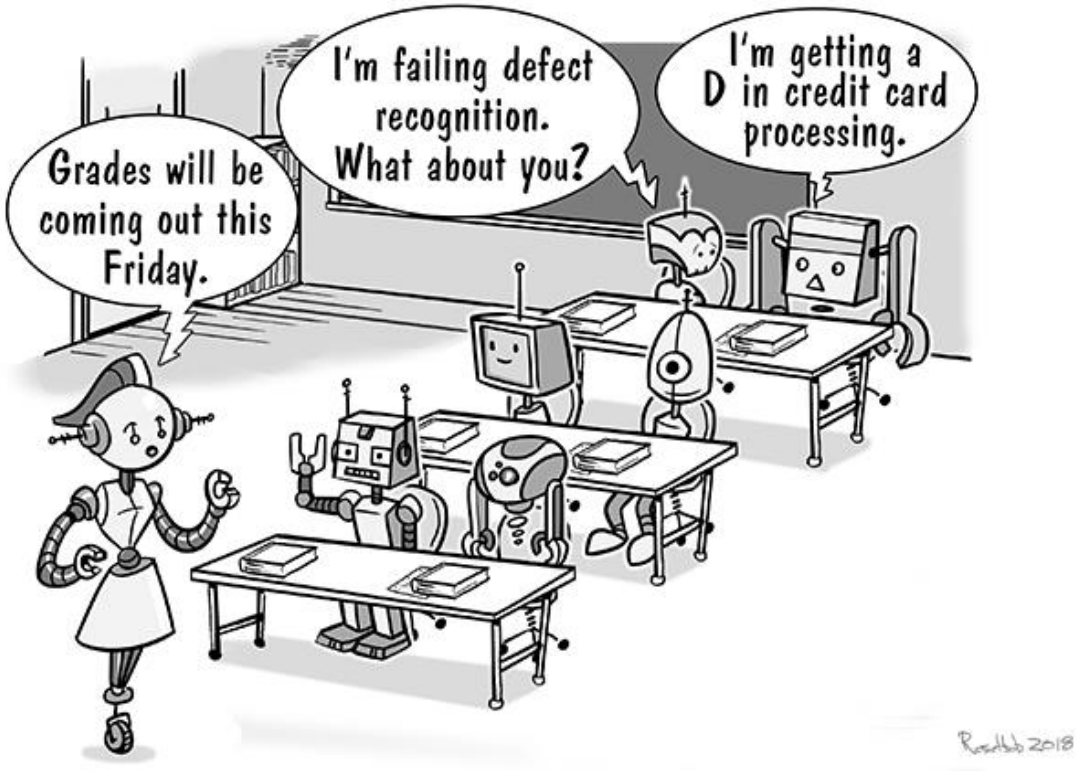
- Al surgir el **machine learning**, esa técnica basada en reglas para desarrollar inteligencia artificial se abandonó. Esto se debe a que el ML es capaz de aportarle a las computadoras una capacidad real de aprendizaje, mucho más adaptada al concepto de "**inteligencia artificial**".

Que es el Machine Learning?

- El Machine Learning es una técnica que permite a las computadoras aprender, ese proceso de aprendizaje se basa en exponer a la computadora a muchos datos para que pueda procesarlos, analizarlos y aprender de ellos.
- Como seria la IA para jugar Ajedrez de un sistema basado en reglas respecto a uno basado en Machine Learning (Aprendizaje Automático)?

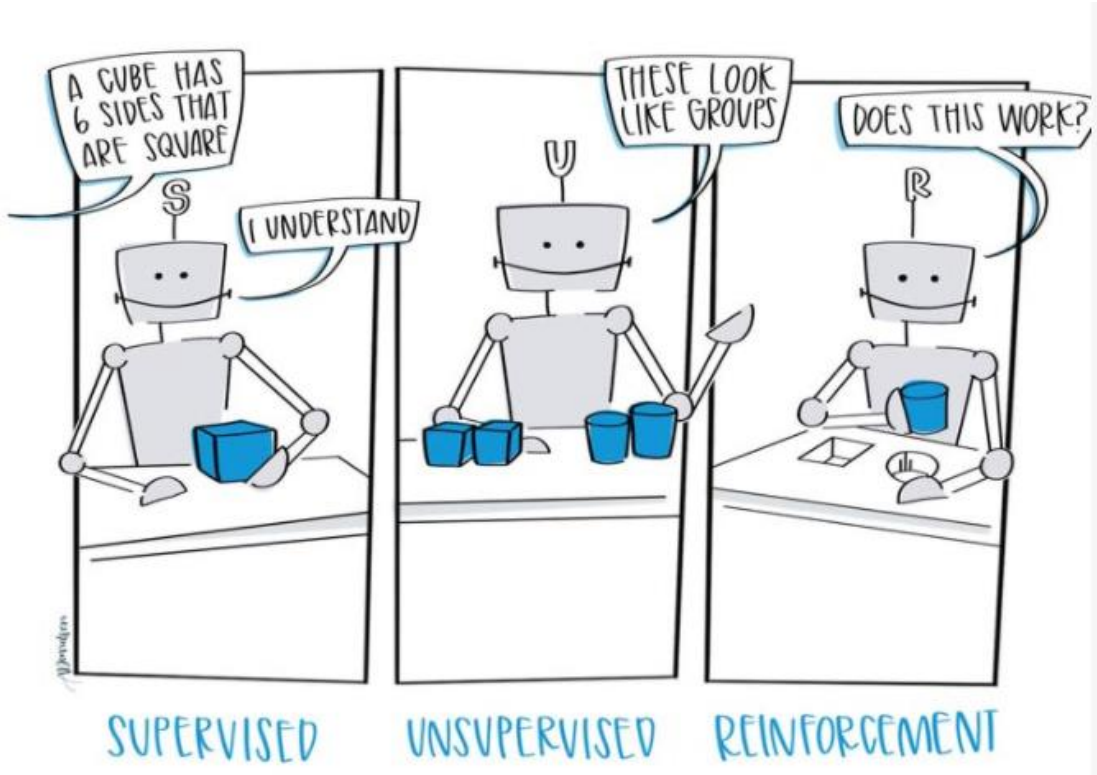
Diferencia entre IA y Machine Learning

- la **diferencia entre inteligencia artificial y machine learning** es que la IA es la capacidad de las computadoras de mostrar un comportamiento “inteligente”. Mientras que ML es una técnica que se utiliza para crear y mejorar dicho comportamiento. Esto mediante entrenamientos automáticos basados en la exposición a datos.



Algoritmos de Machine Learning

- Tipos de Machine Learning



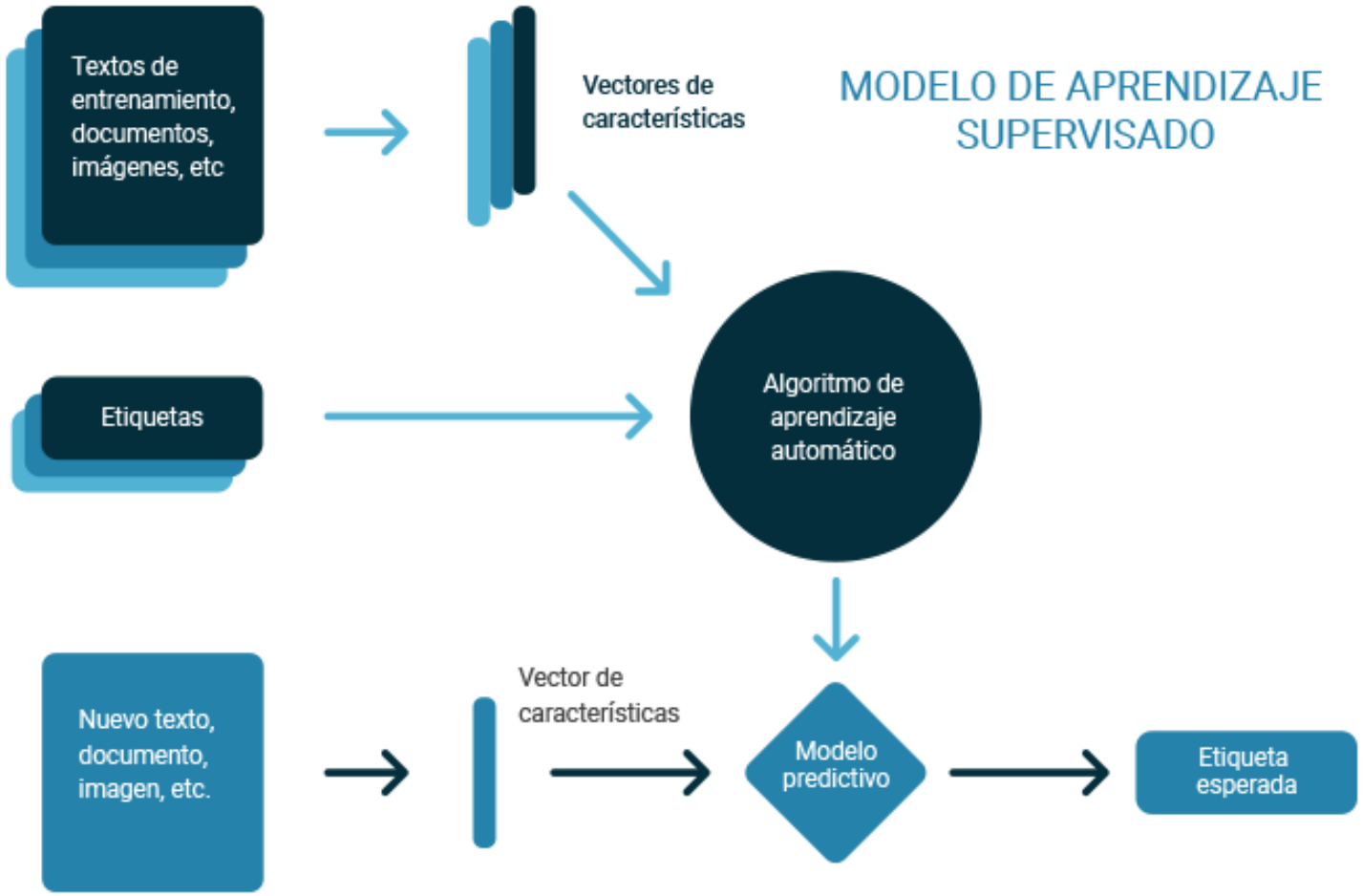
Algoritmos Supervisados

- En los algoritmos de aprendizaje supervisado se genera un **modelo predictivo**, basado en **datos de entrada y salida**.
- La palabra clave “supervisado” viene de la idea de tener un conjunto de datos previamente etiquetado y clasificado, es decir, tener un conjunto de muestra, el cual ya se sabe a qué grupo, valor o categoría pertenecen los ejemplos. Con este grupo de datos que llamamos **datos de entrenamiento**, se realiza el **ajuste al modelo inicial** planteado.
- Es de esta forma como el algoritmo va “aprendiendo” a clasificar las muestras de entrada comparando el resultado del modelo, y la etiqueta real de la muestra, realizando las compensaciones respectivas al modelo de acuerdo a cada error en la estimación del resultado.

Algoritmos Supervisados

- K vecinos más próximos (K-nearest neighbors)
- Redes neuronales artificiales (Artificial neural networks)
- Máquinas de vectores de soporte (Support vector machines)
- Clasificador Bayesiano ingenuo (Naïve Bayes classifier)
- Árboles de decisión (Decision trees)
- Regresión logística (Logistic regression)

Modelo de Aprendizaje Supervisado



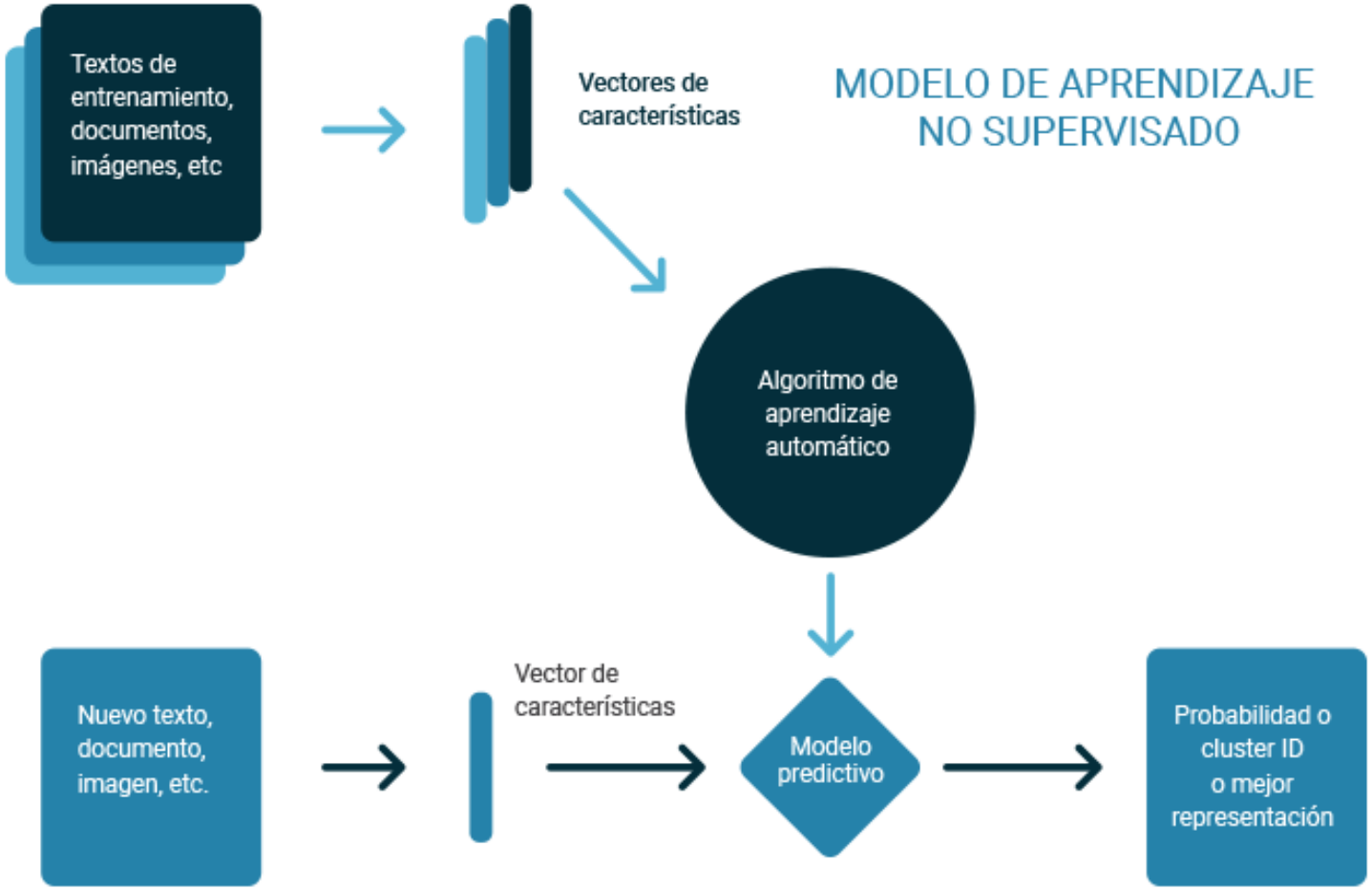
Algoritmos No Supervisados

- Los algoritmos de aprendizaje no supervisado trabajan de forma muy similar a los supervisados, con la diferencia de que éstos sólo **ajustan su modelo predictivo tomando en cuenta los datos de entrada**, sin importar los de salida.
- Es decir, a diferencia del supervisado, los datos de entrada no están clasificados ni etiquetados, y no son necesarias estas características para entrenar el modelo.

Algoritmos No Supervisados

- K-medias (K-means)
- Mezcla de Gaussianas (Gaussian mixtures)
- Agrupamiento jerárquico (Hierarchical clustering)
- Mapas auto-organizados (Self-organizing maps)

Modelo de Aprendizaje No Supervisado



Algoritmos de Aprendizaje por Refuerzo

- Los algoritmos de aprendizaje por refuerzo definen modelos y funciones enfocadas en maximizar una medida de “**recompensas**”, basados en “**acciones**” y al ambiente en el que el agente inteligente se desempeñará.
- Este algoritmo es el más apegado a la psicología conductista de los humanos, ya que es un **modelo acción-recompensa**, que busca que el algoritmo se ajuste a la mejor “recompensa” dada por el ambiente, y sus acciones por tomar están sujetas a estas recompensas.
- Este tipo de métodos pueden usarse para hacer que los robots aprendan a realizar diferentes tareas.

Algoritmos de Aprendizaje por Refuerzo

- Programación dinámica (Dynamic programming)
- Q-learning
- SARSA

Modelo de Aprendizaje por Refuerzo

Lenguajes de Machine Learning

- Lenguaje R
- Python
- Scala
- Julia

Leguaje R

- Pros
  - Excelente gama de paquetes de código abierto y de alta calidad, específicos de dominio.
  - La instalación básica viene con funciones y métodos estadísticos integrales muy completos.
  - Rtambién maneja el álgebra de matriz particularmente bien.
  - La visualización de datos es una fortaleza clave con el uso de bibliotecas como ggplot2.
- Contras
  - No es un lenguaje rápido.
  - Esfantástico para fines estadísticos y científicos de datos. Pero menos para la programación de propósito general.
  - Tiene algunas características inusuales que pueden atrapar a los programadores con experiencia en otros idiomas. Por ejemplo: indexación desde 1, utilizando, estructuras de datos no convencionales. operadores de asignación múltiple.
- Veredicto: "brillante en para lo que está diseñado".
  - Res un lenguaje poderoso que sobresale en una gran variedad de aplicaciones estadísticas y de visualización de datos, y ser de código abierto permite una comunidad muy activa de contribuyentes.

Python

- Pros
  - Es un lenguaje de programación de propósito general.
  - Python tiene una curva de aprendizaje muy rápida.
  - Posee buenos paquetes para el manejo de datos y ML.
- Contras
  - Seguridad de tipos: Python es un lenguaje de tipo dinámico.
  - Para fines específicos de análisis estadístico y de datos, la amplia gama de paquetes de R le da una ligera ventaja sobre Python. Para los lenguajes de propósito general, hay alternativas más rápidas y seguras a Python.
- Veredicto - "excelente todo terreno"
  - Python es una muy buena opción de lenguaje para la ciencia de datos, y no solo en el nivel de entrada. Gran parte del proceso de ciencia de datos gira en torno al proceso ETL (extracción-transformación-carga). Esto hace que la generalidad de Python sea ideal. Las bibliotecas como Tensorflow de Google o Keras hacen de Python un lenguaje muy útil para el aprendizaje automático.

Lenguaje SQL

- Pros
  - Muy eficiente en la consulta, actualización y manipulación de bases de datos relacionales.
  - La sintaxis declarativa hace de SQL un lenguaje a menudo muy legible.
  - SQL es muy utilizado en una amplia gama de aplicaciones, por lo que su conocimiento es muy útil.
- Contras
  - Las capacidades analíticas de SQL son bastante limitadas; más allá de agregar y sumar, contar y promediar datos, sus opciones son limitadas.
  - Para los programadores que vienen de un fondo imperativo, la sintaxis declarativa de SQL puede presentar una curva de aprendizaje lenta.
  - Hay muchas implementaciones diferentes de SQL como Postgresql, SQLite, MariaDB. Todos son lo suficientemente diferentes como para hacer que la interoperabilidad sea un dolor de cabeza.
- Veredicto - "intemporal y eficiente"
  - SQL es más útil como lenguaje de procesamiento de datos que como herramienta analítica avanzada. Sin embargo, gran parte del proceso de la ciencia de la información depende de ETL, y la longevidad y la eficiencia de SQL son la prueba de que es un lenguaje muy útil para el científico de datos moderno.

Java

- Pros
  - Ubicuidad: muchos sistemas y aplicaciones modernos se basan en un back-end de Java.
  - Fuertemente tipado: para aplicaciones de Big Data de misión crítica, esto es inestimable.
  - Java es un lenguaje compilado de propósito general y alto rendimiento.
- Contras
  - Para análisis ad hoc y aplicaciones estadísticas más dedicadas, la verbosidad de Java hace que sea una primera opción poco probable. Los lenguajes de script de escritura dinámica como R y Python se prestan a una productividad mucho mayor.
  - En comparación con los lenguajes específicos de dominio como R, no hay una gran cantidad de bibliotecas disponibles para métodos estadísticos avanzados en Java.
- Veredicto: "un serio contendiente para la ciencia de datos"
  - Hay mucho que decir para aprender Java como un lenguaje de ciencia de datos de primera elección. Muchas compañías apreciarán la capacidad de integrar sin problemas el código de producción de ciencia de datos directamente en su base de código existente, y encontrará que el rendimiento de Java y la seguridad de tipos son ventajas reales. Sin embargo, estarás sin el rango de paquetes específicos de estadísticas disponibles en otros idiomas. Dicho esto, definitivamente uno a considerar, especialmente si ya conoces uno de R y / o Python.

Scala

- Pros
  - Multi-paradigmático: imperativo y funcional.
  - Scala se compila en el bytecode de Java y se ejecuta en una JVM.
  - Scala + Spark = Computación en clúster de alto rendimiento. Ideal para grandes volúmenes.
- Contras
  - Scala tiene una curva de aprendizaje difícil.
  - La sintaxis y el sistema de tipos a menudo se describen como complejos.
- Veredicto: "perfecto, para grandes volúmenes"
  - Cuando se trata de usar la computación en clúster para trabajar con Big Data, Scala + Spark son soluciones fantásticas. Si tiene experiencia con Java y otros lenguajes de tipo estático, también apreciará estas características de Scala. Sin embargo, si su aplicación no se ocupa de los volúmenes de datos que justifican la complejidad agregada de Scala, es probable que su productividad sea mucho mayor al usar otros idiomas, como R o Python.

Scala

Julia

- Pros
  - Julia es un lenguaje compilado que ofrece un buen rendimiento. También ofrece las capacidades de simplicidad, escritura dinámica y secuencias de comandos de un lenguaje interpretado como Python.
  - Julia fue diseñado específicamente para el análisis numérico. Es capaz de programación de propósito general también.
  - Legibilidad. Muchos usuarios del lenguaje citan esto como una ventaja clave.
- Contras
  - Madurez. Como nuevo idioma, algunos usuarios de Julia han experimentado inestabilidad al usar paquetes. Pero el lenguaje central en sí es, al parecer, lo suficientemente estable para el uso de producción.
  - Los paquetes limitados son otra consecuencia de su juventud. A diferencia de R y Python de larga data, Julia no tiene la opción de paquetes (todavía).
- Veredicto - “lenguaje para el futuro"
  - El principal problema es que al ser un lenguaje recientemente desarrollado, no es tan maduro para la producción como sus principales alternativas Python y R.

Certificación Internacional

- Machine Learning with TensorFlow on Google Cloud Platform Specialization - Variable
- Machine Learning Stanford Online - USD 5040.00
- eCornell Machine Learning Certificate - USD 3600.00

Machine Learning: Dimensionality Reduction

- La **reducción de dimensionalidad** es una técnica de aprendizaje automático que reduce la cantidad de features (caracteristicas) en su conjunto de datos. Lo mejor de la reducción de dimensionalidad es que no afecta negativamente el rendimiento de su modelo de aprendizaje automático. En algunos casos, esta técnica incluso ha aumentado la precisión del modelo.
- Al reducir la cantidad de características en nuestro conjunto de datos, también estamos reduciendo el espacio de almacenamiento requerido para almacenar los datos, nuestro compilador de Python necesitará menos tiempo para revisar el conjunto de datos.

Técnicas de Reducción de Dimensionalidad

- Principle Component Analysis (PCA)
  - PCA reduce la cantidad de características en el conjunto de datos al detectar la correlación entre ellas. Cuando la correlación entre las características sea lo suficientemente fuerte, se fusionarán y formarán una sola característica.
- Linear Discriminant Analysis (LDA)
  - Similar a PCA, LDA también es un algoritmo de reducción de dimensionalidad. Pero a diferencia de PCA, LDA también encontrará las características que maximizan la separación entre múltiples clases.

- ¿por qué debería molestarme con este nuevo algoritmo que básicamente hace lo mismo que el algoritmo anterior?
  - Tanto PCA como LDA son algoritmos de reducción de dimensionalidad. Pero cuando PCA se considera un algoritmo no supervisado, LDA, por otro lado, se considera supervisado.

Laboratorio

- Lab 1 Instalación de R y R Studio
- Lab 2 Instalación de Python y Jupyter – Suite Ml Anaconda

Referencias

- https://blog.enzymeadvisinggroup.com/inteligencia-artificial-machine-learning
- https://www.ceralytics.com/3-types-of-machine-learning/
- https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2
- https://hackr.io/blog/machine-learning-certifications
- https://towardsdatascience.com/understanding-dimensionality-reduction-for-machine-learning-ad9a3811bd89