



Evaluación 1

Marcelo Paz
Inteligencia Artificial

13 de septiembre de 2024



Versión: 1.0.0

Resumen

En el siguiente documento se presenta un apunte hecho en ultimo minuto para plasmar parte de las ideas y reorganizar a mi gusto los contenidos.

Contenidos:

- Notebook de Jupyter.
 - Conociendo Python.
 - Algebra Lineal y probabilidad aplicada a IA.
 - Teorema Central Límite (parte teórica que guarda relación con los casos de uso del teorema).
- Presentación que contiene los conceptos de introducción a la IA.
- Libro Mathematics for Machine Learning (MML), pág. 17-27.

1. Inteligencia Artificial

Resumen:

- **Objetivo:** Desarrollar algoritmos que permitan aprender de la experiencia y mejorar su rendimiento en tareas específicas, mediante la obtención de datos.
- **Tipos:**
 - IA fuerte: Realizar tareas complejas asociadas al pensar humano (Capacidad de comprender, razonar y tomar decisiones).
 - IA débil: Realizar tareas específicas.
- Inteligencia Artificial > Machine Learning > Deep Learning.
- **Tipos de aprendizaje**
 - **Aprendizaje supervisado:** En este modelo, el sistema de IA se entrena con un conjunto de datos etiquetados, es decir, datos que ya tienen la respuesta correcta. El objetivo es que el sistema aprenda a predecir la salida correcta para nuevas entradas basándose en los ejemplos proporcionados. Es útil para tareas como la clasificación de imágenes y el reconocimiento de voz.

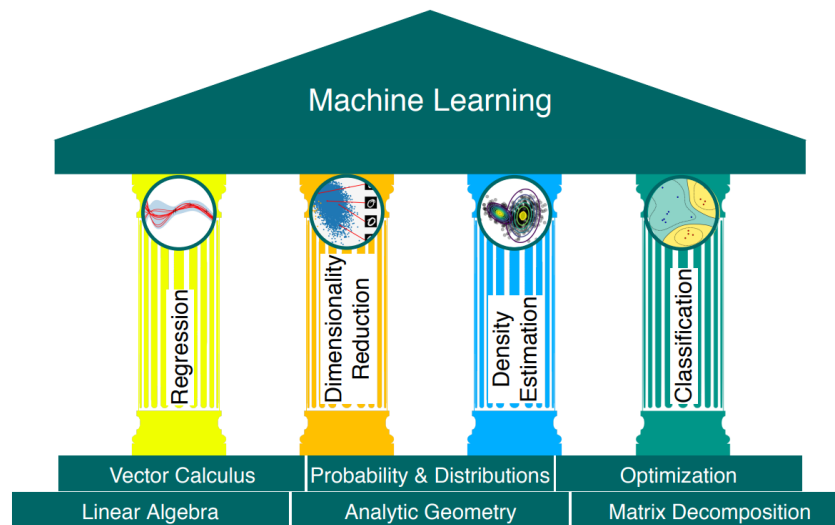


- **Aprendizaje no supervisado:** A diferencia del aprendizaje supervisado, en este modelo el sistema trabaja con datos que no están etiquetados. El objetivo es encontrar patrones o estructuras ocultas en los datos. Un ejemplo común es el análisis de agrupamiento (clustering), donde el sistema agrupa datos similares sin conocer las etiquetas de antemano.
- **Aprendizaje por refuerzo:** En este modelo, el sistema aprende a tomar decisiones mediante prueba y error, recibiendo recompensas o castigos según el resultado de sus acciones. Es especialmente útil en situaciones donde la toma de decisiones secuenciales es crucial, como en los juegos y la robótica.

1.1. Machine Learning

Resumen:

- **Objetivo:** Diseñar metodologías de propósito general para extraer patrones valiosos de los datos.
- **Definición:** Uso de algoritmos y datos para imitar la forma en que los humanos aprenden, mejorando su precisión con el tiempo.
- **Núcleo:**
 - Datos: Los datos son la base, la materia prima.
 - Objetivo: Obtener datos de calidad para que el modelo pueda aprender de ellos.
 - Modelo: Los modelos están relacionados con el proceso que genera los datos.
 - Objetivo: Diseñar buenos modelos que generalicen bien a datos aún no vistos (para predecir supongo). Para que el modelo *aprenda* debe mejorar su rendimiento después de procesar los datos.
 - Un buen modelo puede utilizarse para predecir lo que ocurriría en el mundo real sin necesidad de realizar experimentos.
 - Entrenar: Utilizar los datos disponibles para optimizar algunos parámetros del modelo con respecto a una función de utilidad que evalúa lo bien que el modelo predice los datos de entrenamiento.
 - Aprendizaje: Forma de encontrar automáticamente patrones y estructuras en los datos optimizando los parámetros del modelo.
- Algoritmo:
 1. Sistema que realiza predicciones basadas en datos. (Se denomina 'predictor').
 2. Sistema que adapta algunos parámetros internos del 'predictor' para que funcione bien con datos de entradas futuros que no se han visto. (Se denomina 'entrenamiento').
- Se puede pensar en los datos como vectores, se puede pensar en un vector:
 1. Como una matriz de números (un punto de vista informático).
 2. Como una flecha con una dirección y magnitud (un punto de vista físico).
 3. Como un objeto que obedece a la suma y la escala (un punto de vista matemático).



■ Pilares:

1. *Regresión*: predecir un valor continuo basado en una o más variables independientes.
2. *Reducción de dimensionalidad*: reducir el número de variables en un conjunto de datos, manteniendo la mayor cantidad de información posible.
3. *Estimación de densidad*: proceso de construir una función que describe la distribución de probabilidad de una variable aleatoria.
4. *Clasificación*: asignar una etiqueta a una observación basada en uno o más atributos.

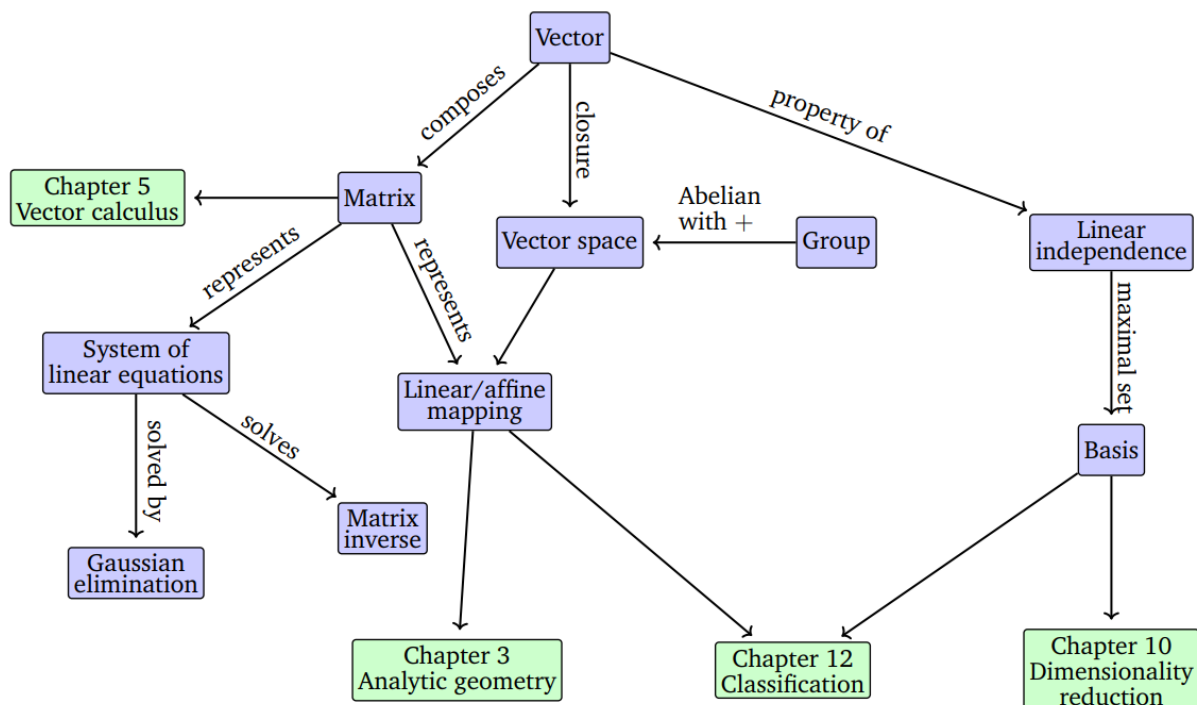
■ Bases de los pilares:

- Nivel 1:
 - *Álgebra Lineal*: Representamos los datos numéricos como vectores y representamos una tabla de dichos datos como una matriz. El estudio de vectores y matrices se denomina álgebra lineal
 - *Geometría Analítica*: Dado que dos vectores representan dos objetos en el mundo real, queremos hacer afirmaciones sobre su similitud. La idea es que nuestro algoritmo de aprendizaje automático (nuestro predictor) prediga que los vectores que son similares tienen resultados similares. Para formalizar la idea de similitud entre vectores, necesitamos introducir operaciones que tomen dos vectores como entrada y devuelvan un valor numérico que represente su similitud. La construcción de la semejanza y las distancias es central para la geometría analítica.
 - *Descomposición de la matriz*: En el capítulo 4, introducimos algunos conceptos fundamentales sobre las matrices y la descomposición de matrices. Algunas operaciones en matrices son extremadamente útiles en el aprendizaje automático y permiten una interpretación intuitiva de los datos y un aprendizaje más eficiente.



- Nivel 2:
 - *Cálculo vectorial y optimización*: Para entrenar modelos de aprendizaje automático, normalmente encontramos parámetros que maximizan alguna medida de rendimiento. Muchas técnicas de optimización requieren el concepto de gradiente, que nos indica la dirección en la que debemos buscar una solución. El capítulo 5 trata sobre el cálculo vectorial y detalla el concepto de cálculo vectorial de gradientes, que posteriormente utilizaremos en el capítulo 7, donde hablamos de optimización para encontrar máximos/mínimos de funciones.
 - *Probabilidad y distribución*: A menudo consideramos que los datos son observaciones ruidosas de alguna señal subyacente verdadera. Esperamos que mediante la aplicación del Machine learning podamos identificar la señal del ruido. Esto requiere que tengamos un lenguaje para cuantificar lo que significa ruido". A menudo también nos gustaría tener predictores que nos permitan expresar algún tipo de incertidumbre, por ejemplo, para cuantificar la confianza que tenemos sobre el valor de la predicción en un punto de datos de prueba particular. La cuantificación de la incertidumbre es el ámbito de la teoría de la probabilidad.

1.2. Matemática para Machine Learning (MML)





1.3. Python en ML

■ Librerías:

- **Numpy:** Librería de Python que proporciona soporte para arreglos y matrices multidimensionales, junto con una colección de funciones matemáticas de alto nivel para operar en estos arreglos.
- **Pandas:** Librería de Python que proporciona estructuras de datos y herramientas de análisis de datos.
- **Matplotlib:** Librería de Python que proporciona una API orientada a objetos para incorporar gráficos en aplicaciones usando GTK, Qt, WxWindows o Mac OS X.
- **Seaborn:** Librería de Python que proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos y informativos (basada en matplotlib).
- **SciPy:** Librería de Python que se utiliza para realizar cálculos científicos y técnicos (numpy con esteroides).
- **Scikit-learn:** Librería de Python que se usa para construir y evaluar modelos de aprendizaje automático.

■ Colecciones de datos:

- Tuplas (): Son inmutables, es decir, no se pueden modificar después de su creación.
- Listas []: Son mutables, es decir, se pueden modificar después de su creación.
- Diccionarios { }: Son colecciones de datos que contienen pares clave-valor.

■ **Tensor:** puede ser un número, un vector, una matriz o cualquier colección de estos objetos.

- Tensor grado 0: Escalar (número).
- Tensor grado 1: Vector.
- Tensor grado 2: Matriz.
- Tensor grado 3: 3D.
- Tensor grado n: nD.

■ Álgebra Lineal:

- **Vector:** Es un conjunto de números de una sola dimensión.

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

- **Matriz:** Es un conjunto de números de dos dimensiones.

$$M = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$



- Lo más relevante es:
 - **Producto punto entre vectores** (Se obtiene un escalar).
 - **Multiplicación de matrices** (Se obtiene una matriz).
 - **Vector ortogonal:** Dos vectores son ortogonales si su producto punto es cero.
 - **Matriz ortogonal:** Una matriz es ortogonal si sus columnas son ortogonales entre sí y tienen una longitud de 1.

$$A^T \cdot A = I$$

- **Matriz inversa:** Una matriz tiene una inversa si su determinante es diferente de cero.
- **Matriz transpuesta:** La matriz transpuesta de una matriz es una matriz que se obtiene intercambiando filas por columnas.
- **Vector propio:** Un vector propio (o autovector) de una matriz es un vector no nulo que cambia de magnitud pero no de dirección cuando se aplica la matriz a él.
- **Descomposición en valores singulares (SVD):** SVD es una técnica importante en la reducción de dimensionalidad y se utiliza en métodos como PCA y en el tratamiento de datos en IA.
- **Sistema de ecuaciones lineales:** Un sistema de ecuaciones lineales es un conjunto de ecuaciones lineales que comparten las mismas variables.

$$Ax = b$$

Donde:

- ◊ A es una matriz de coeficientes.
- ◊ x es un vector de incógnitas.
- ◊ b es un vector de términos independientes.
- ◊ Posibles soluciones:
 1. Ninguna solución: $\text{rango}(A) \neq \text{rango}(A,b)$.
 2. Infinitas soluciones: $\text{rango}(A) = \text{rango}(A,b) < n$.
 3. Solución única: $\text{rango}(A) = \text{rango}(A,b) = n$.

■ Probabilidad:

- **Distribución normal:** Es una distribución de probabilidad continua que se utiliza para representar variables aleatorias con distribuciones simétricas alrededor de su media. Se utiliza en técnicas de regresión y modelos probabilísticos.
- **Teorema de Bayes:** Se utiliza en métodos de clasificación como el clasificador Naive Bayes y en la inferencia probabilística en redes bayesianas.



- **Teorema central del límite:** Es un teorema fundamental en estadística que establece que, a medida que el tamaño de la muestra aumenta, la distribución de la media de la muestra se aproxima a una distribución normal.
 - **Aplicaciones:**
 - ◊ **Justificación para la inferencia estadística.**
 1. **Modelos de regresión y clasificación:** En la mayoría de los algoritmos de aprendizaje supervisado, como la regresión lineal, logística y otros métodos basados en gradiente, se supone que los errores (residuos) tienen una distribución normal. El TCL proporciona una justificación teórica para esta suposición cuando el tamaño de la muestra es grande.
 2. **Intervalos de confianza y prueba de hipótesis:** En Machine Learning y Deep Learning, se realizan pruebas estadísticas para validar los modelos, y el TCL permite que los intervalos de confianza y las pruebas de hipótesis sean válidas para la mayoría de las estadísticas.
 - ◊ **Entrenamiento de redes neuronales y optimización.**
 1. **Inicialización de Pesos:** al entrenar redes neuronales profundas, los pesos se inicializan típicamente utilizando distribuciones normales. El TCL asegura que, con un gran número de nodos y conexiones, las sumas de muchas variables aleatorias (como activaciones) tienden a ser normalmente distribuidas.
Esto facilita el cálculo de las actualizaciones de los pesos utilizando métodos basados en gradientes.
 2. **Propagación de Errores:** en el entrenamiento de redes neuronales, los gradientes calculados a través del algoritmo de retropropagación a menudo se comportan como variables aleatorias. El TCL sugiere que la distribución de estos gradientes se aproximará a una distribución normal a medida que el tamaño de la red y la cantidad de muestras de entrenamiento aumentan, lo que estabiliza el proceso de optimización.
 - ◊ **Validación de modelos y evaluación de rendimiento.**
 1. **Bootstrapping y Métodos de Muestreo:** el TCL se utiliza para justificar el uso de técnicas de remuestreo como el bootstrapping en Machine Learning. Estas técnicas permiten estimar el error y la varianza de los modelos, proporcionando distribuciones normales aproximadas de los estimadores cuando se remuestran múltiples veces.
 2. **Curvas ROC y AUC:** en la evaluación del rendimiento del modelo, las áreas bajo las curvas ROC (AUC) a menudo se aproximan a una distribución normal debido al teorema central del límite, lo que permite comparar diferentes modelos estadísticamente.



◇ Reducción de dimensionalidad y preprocesamiento de datos.

1. PCA (Análisis de Componentes Principales): cuando se reduce la dimensionalidad de los datos mediante técnicas como PCA, el TCL asegura que, para un gran número de muestras, la suma de los componentes principales tiende a una distribución normal.

Esto ayuda en la selección y retención de características más relevantes y facilita el análisis y la visualización.