|  |  |
| --- | --- |
| **Nombre:** | Luis de la Garza González |
| **Matrícula:** | al03101869 |
| **Nombre del curso:** | Machine Learning. |
| **Trabajo:** | Actividad 02.- Las Matrices y sus Aplicaciones. |
| **Nombre del profesor:** | Mtro. Igor García Atutxa |
| **Fecha:** | 21 de septiembre de 2025 |

**Contenido**

[**Objetivo** 3](#_Toc205390891)

[**Objetivo** 5](#_Toc205390892)

[**Desarrollo** 5](#_Toc205390893)

[**Conclusiones** 6](#_Toc205390894)

[**Liga al código** 7](#_Toc205390895)

# **Objetivo**

Explorar conceptos fundamentales de álgebra lineal aplicados a vectores, normas y proyecciones, utilizando Python y Jupyter Notebook.

# **Instrucciones:**

**1. Investigación sobre normas**

* Investiga los diferentes tipos de normas que existen.
* Identifica sus principales aplicaciones.
* Elabora una **infografía** para compartir lo aprendido.

**2. Utilizando un cuaderno de Jupyter Notebook y el lenguaje de programación Python, realiza las siguientes operaciones:**

* Genera dos vectores en **R**2 y represéntalos en un plano utilizando la librería Matplotlib.
* Calcula el producto punto entre los dos vectores generados.
* Calcula la longitud de ambos vectores.
* ****Encuentra el ángulo que forman los dos vectores entre sí y genera un nuevo vector que sea ortogonal a cualquiera de los dos primeros.

**3. Sea el vector base** que genera el subespacio unidimensional

Encuentra:

* El valor de la coordenada λ lambda
* La proyección πu(x) ∈ U
* La matriz de proyección Pπ para proyectar por el subespacio

Utiliza un cuaderno de Jupyter Notebook para elaborar tu respuesta.

**4. Operaciones con matrices cuadradas**

Usando Python en un nuevo cuaderno de Jupyter Notebook:

* Genera una matriz cuadrada con ( n > 3 ) y calcula su determinante y su traza.
* Calcula el rango de la matriz generada y los valores de los coeficientes del polinomio característico.
* Encuentra las raíces del polinomio característico de la matriz generada

# **Objetivo**

Explorar conceptos fundamentales de álgebra lineal aplicados a vectores, normas y proyecciones, utilizando Python y Jupyter Notebook.

# **Desarrollo**

**1. Investigación sobre normas**

1. Investiga los diferentes tipos de normas que existen.
2. Identifica sus principales aplicaciones.

En álgebra lineal, las normas son funciones que asignan un número real no negativo a un vector, proporcionando una medida de su "tamaño" o "longitud". Existen varios tipos de normas, cada una con sus propias características y aplicaciones:

| **Norma** | **Principales Aplicaciones** |
| --- | --- |
| Norma Euclidiana (Norma (L\_2)): También conocida como la norma (L\_2) o norma (2), es la más común y se define como la raíz cuadrada de la suma de los cuadrados de las componentes del vector. Para un vector , se calcula como: | Ampliamente utilizada en la geometría y en la teoría de la optimización. Es fundamental en el cálculo de distancias en el espacio euclidiano, lo que la hace esencial en algoritmos de aprendizaje automático, como el k-means clustering y los métodos de regresión. También se utiliza en la física para calcular magnitudes como la velocidad y la aceleración. |
| Norma Manhattan (Norma (L\_1)): También conocida como la norma (L\_1) o norma de la "suma absoluta", se define como la suma de los valores absolutos de las componentes del vector:  ] | se utiliza en problemas de optimización donde se desea minimizar la suma de las diferencias absolutas. Es común en la teoría de redes y en la optimización de rutas, como en el problema del viajante. También se utiliza en el aprendizaje automático, especialmente en técnicas de regularización como la regresión Lasso, que ayuda a seleccionar características importantes en modelos predictivos. |
| Norma Infinita (Norma (L\_\infty)): También conocida como la norma del "máximo", se define como el valor absoluto máximo de las componentes del vector:  ] | se utiliza en problemas de optimización donde se desea minimizar el error máximo. Es común en el análisis de estabilidad de sistemas y en el diseño de controladores robustos. También se aplica en la teoría de juegos y en la programación lineal, donde se busca minimizar el peor caso posible. |
| Norma (L\_p): Es una generalización de las normas (L\_1), (L\_2) y (L\_\infty). Para un valor (p \geq 1), la norma (L\_p) se define como:  Cuando (p = 2), se obtiene la norma Euclidiana, y cuando (p = 1), se obtiene la norma Manhattan. | Las normas (L\_p) son una generalización de las normas (L\_1), (L\_2) y (L\_\infty), y se utilizan en diversas aplicaciones de análisis funcional y teoría de espacios de Banach. En el procesamiento de señales, las normas (L\_p) se utilizan para medir la energía de una señal y en la compresión de datos. También se aplican en la teoría de la probabilidad y en el análisis de datos. |
| Norma de Frobenius: Es una norma utilizada para matrices y se define como la raíz cuadrada de la suma de los cuadrados de todos los elementos de la matriz. Para una matriz (A), se calcula como:  ] | Se utiliza principalmente en el análisis de matrices. Es útil en la descomposición en valores singulares (SVD) y en la factorización de matrices, que son técnicas fundamentales en el procesamiento de imágenes y en la reducción de dimensionalidad. También se utiliza en la teoría de control y en la optimización de sistemas dinámicos. |

1. Elabora una infografía para compartir lo aprendido.

**2. Utilizando un cuaderno de Jupyter Notebook y el lenguaje de programación Python, realiza las siguientes operaciones:**

* Genera dos vectores en **R**2 y represéntalos en un plano utilizando la librería Matplotlib.

**Gráfico de líneas

El contenido generado por IA puede ser incorrecto.**

* Calcula el producto punto entre los dos vectores generados.

Para calcular el producto punto entre dos vectores, utilizamos la siguiente fórmula:

el producto punto se calcula de la siguiente manera:

[

]

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* Calcula la longitud de ambos vectores.

Para calcular la **longitud** (o norma) de un vector en , usamos la fórmula:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.∣∣v⃗∣∣ = ( x2 + y2  )1/2

* Encuentra el ángulo que forman los dos vectores entre sí y genera un nuevo vector que sea ortogonal a cualquiera de los dos primeros.

**Angulo que forman los dos vectores entre sí:**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* **Nuevo vector que sea ortogonal a cualquiera de los dos primeros:**

Una forma rápida de encontrar un vector ortogonal en es rotarlo 90°:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Verificación: el producto punto de dos vectores ortogonales debe ser cero:

**3. Sea el vector base** que genera el subespacio unidimensional

Encuentra:

* El valor de la coordenada λ lambda

Interpretación geométrica

El valor de 𝜆 indica cuánto se escala el vector base para llegar al vector 𝑥⃗. Si ambos componentes coinciden, entonces

𝑥⃗ pertenece al subespacio generado por 𝑣⃗

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Utiliza un cuaderno de Jupyter Notebook para elaborar tu respuesta.

* La proyección πu(x) ∈ U

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

* La matriz de proyección Pπ para proyectar por el subespacio

La **matriz de proyección ortogonal** Pπ permite proyectar cualquier vector sobre el subespacio generado por un vector base , es una herramienta clave para entender cómo se descompone un vector en componentes alineadas y ortogonales al subespacio.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Para encontrar el ángulo que forman los dos vectores , utilizamos la fórmula del producto punto y la magnitud de los vectores. La fórmula es:

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.Donde ) es el ángulo entre los dos vectores.

1. **Calcula los autovalores y autovectores de la siguiente matriz**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Realiza su autodescomposición y grafica los resultados obtenidos en un plano con Matplotlib.

**Texto

El contenido generado por IA puede ser incorrecto.**

**Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.**

**Interpretación:**

Las flechas rojas y azules representan los autovectores, que mantienen su dirección bajo la acción de 𝐴.

Las flechas verdes y moradas muestran cómo la base estándar se transforma, evidenciando el efecto de la matriz sobre el espacio.

Esta visualización confirma que los autovectores son direcciones privilegiadas donde la transformación solo escala, sin rotar.

**4. Operaciones con matrices cuadradas**

Usando Python en un nuevo cuaderno de Jupyter Notebook:

* Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

  El contenido generado por IA puede ser incorrecto.Genera una matriz cuadrada con ( n > 3 ) y calcula su determinante y su traza.
* Calcula el rango de la matriz generada y los valores de los coeficientes del polinomio característico.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.El rango de una matriz es el número máximo de filas (o columnas) linealmente independientes. Para encontrar el rango, podemos reducir la matriz a su forma escalonada mediante eliminación gaussiana.

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

**El polinomio característico se define como:**

donde (I) es la matriz identidad y (\lambda) es un escalar.

Para obtener sus coeficientes:

Texto

El contenido generado por IA puede ser incorrecto.

* **Interfaz de usuario gráfica, Texto

  El contenido generado por IA puede ser incorrecto.**Encuentra las raíces del polinomio característico de la matriz generada

Esto indica que la matriz tiene una mezcla de modos reales y oscilatorios. En contextos como sistemas dinámicos o machine learning, esto puede reflejar componentes estables y rotacionales.

1. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   El contenido generado por IA puede ser incorrecto.**Calcula los autovalores y autovectores de la siguiente matriz**

Realiza su autodescomposición y grafica los resultados obtenidos en un plano con Matplotlib.

**Texto

El contenido generado por IA puede ser incorrecto.**

En la siguiente página, graficamos los resultados obtenidos en un plano con Matplotlib.

**Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.**

Las flechas rojas y azules representan los autovectores, que mantienen su dirección bajo la acción de 𝐴.

Las flechas verdes y moradas muestran cómo la base estándar se transforma, evidenciando el efecto de la matriz sobre el espacio.

Esta visualización confirma que los autovectores son direcciones privilegiadas donde la transformación solo escala, sin rotar.

# **Conclusiones**

* Se calcularon los **autovalores** λ1≈5.56\lambda\_1 \approx 5.56, λ2≈1.44\lambda\_2 \approx 1.44 y sus respectivos **autovectores**:

Estos vectores representan las direcciones invariantes bajo la transformación lineal definida por la matriz:

A=[5122]A = \begin{bmatrix} 5 & 1 \\ 2 & 2 \end{bmatrix}

* Se visualizó la acción de AA sobre la base estándar y los autovectores, mostrando cómo se **estira y rota** el espacio en función de los autovalores.
* Se corrigió un error en la función plt.quiver, ajustando la dimensión de los vectores para que coincidan con el número de flechas a graficar.
* Se reforzaron conceptos clave como:
  + Producto punto y ángulo entre vectores.
  + Proyección ortogonal sobre subespacios.
  + Construcción de la matriz de proyección PπP\_\pi.
  + Interpretación geométrica del **determinante** y la **traza**.
* Se extendió el análisis a una matriz 4×44 \times 4, observando autovalores reales y complejos, y su impacto en la dinámica de la transformación.

Esta actividad integró teoría algebraica, visualización geométrica y programación en Python. El uso de numpy y matplotlib permitió validar los conceptos de forma visual y clara, fortaleciendo la comprensión de las transformaciones lineales y su aplicación en contextos como el Análisis de Componentes Principales (PCA) y machine learning.

PCA es una técnica de reducción de dimensionalidad que se utiliza para transformar un conjunto de datos de alta dimensión en un conjunto de datos de menor dimensión, preservando la mayor cantidad de variabilidad posible. Esto se logra encontrando nuevas variables (componentes principales) que son combinaciones lineales de las variables originales. PCA es útil en el preprocesamiento de datos, la visualización de datos y la eliminación de ruido.

# **Liga al código**