



Machine Learning y Data Analytics

HW 1

Ian Amighini
Julieta Brey
Lorenzo Natri
Camila Sobrino

Profesor Titular: Sergio Pernice

Consigna

Pídanle a su IA favorita que genere un programa Python que genere tres vectores 3D: v_1 , v_2 y v_3 al azar (vectores cuyas coordenadas son números elegidos entre -1 y 1 al azar con una distribución constante, pídanle que les explique qué significa lo que acabo de escribir si no saben el significado).

Pídanle que genere otro programa Python que grafique en un gráfico 3D a los tres vectores junto con el plano correspondiente al span de los primeros dos vectores v_1 y v_2 . Observe el gráfico y note que v_3 no pertenece al span de v_1 y v_2 . Explique por qué.

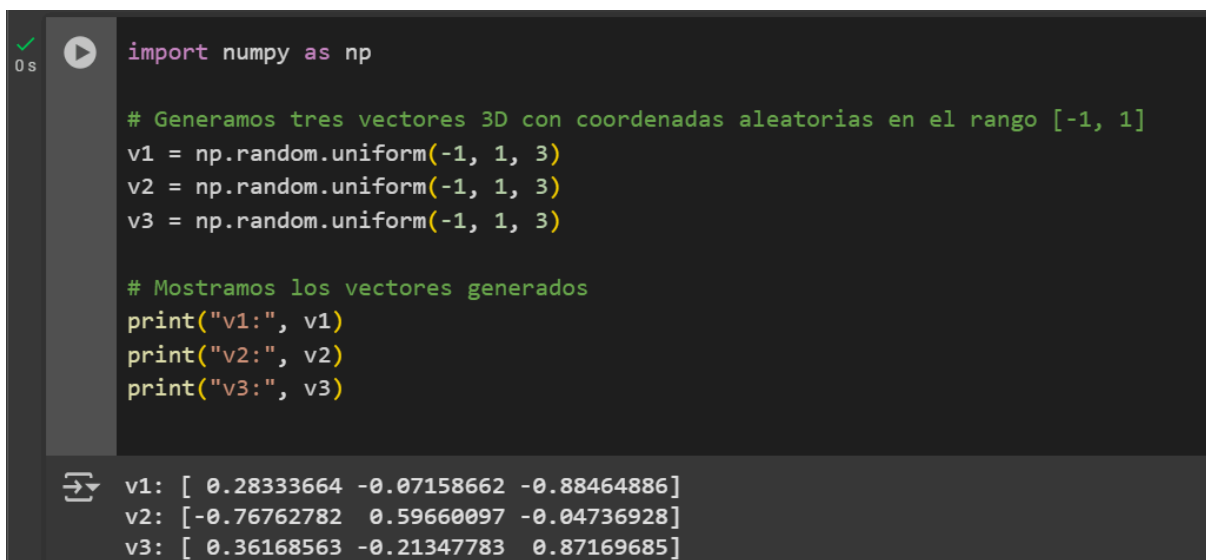
Corran el programa en su plataforma Python favorita. Si pueden trate de especificarle a la IA la plataforma Python y que el gráfico se pueda girar con el mouse.

1) Generación de vectores

En este trabajo, se generaron tres vectores tridimensionales (v_1 , v_2 y v_3) con coordenadas aleatorias en el intervalo $[-1, 1]$. Estas coordenadas deben ser elegidas al azar con una distribución constante, lo que significa que cada número dentro del intervalo debe tener la misma probabilidad de ser seleccionado.

Distribución Uniforme: cualquier número dentro del rango dado tiene la misma probabilidad de ser seleccionado, sin sesgos hacia valores específicos.

Para asegurar que los valores de los vectores cumplan con este criterio, se utilizó la función `np.random.uniform(-1, 1, 3)` de la biblioteca NumPy. Esta función genera tres números aleatorios en el intervalo $[-1, 1]$ siguiendo una distribución uniforme.



```
import numpy as np

# Generamos tres vectores 3D con coordenadas aleatorias en el rango [-1, 1]
v1 = np.random.uniform(-1, 1, 3)
v2 = np.random.uniform(-1, 1, 3)
v3 = np.random.uniform(-1, 1, 3)

# Mostramos los vectores generados
print("v1:", v1)
print("v2:", v2)
print("v3:", v3)
```

v1: [0.28333664 -0.07158662 -0.88464886]
v2: [-0.76762782 0.59660097 -0.04736928]
v3: [0.36168563 -0.21347783 0.87169685]

2) Representación gráfica de vectores 3D y el span de dos vectores

Se generaron y visualizaron tres vectores tridimensionales aleatorios en un espacio 3D, junto con el **plano** definido por el **span** de los dos primeros vectores (v_1 y v_2). A través de

esta representación gráfica, se buscó analizar la relación entre los vectores y *demostrar que el tercer vector (v_3) no pertenece a dicho plano en la mayoría de los casos.*

Concepto de Span

El *span* de dos vectores v_1 y v_2 en \mathbb{R}^3 es el conjunto de todas las combinaciones lineales posibles de esos vectores:

$$P = \{\alpha v_1 + \beta v_2 \mid \alpha, \beta \in \mathbb{R}\}$$

Geométricamente, esto representa un plano que pasa por el origen del espacio tridimensional. Para que el tercer vector v_3 pertenezca a este plano, debe existir una combinación lineal de la forma:

$$v_3 = \alpha v_1 + \beta v_2$$

Si tal combinación no existe, significa que v_3 se encuentra fuera del plano generado por v_1 y v_2 .

Matemáticamente, esto se debe a que el *span* de dos vectores en un espacio 3D es un plano, que tiene dimensión 2, mientras que el espacio completo tiene dimensión 3. La probabilidad de que un punto aleatorio caiga exactamente en un plano específico dentro de un espacio 3D es prácticamente cero, ya que los planos tienen volumen cero en un espacio tridimensional.

```
# Creamos trazos para los vectores (líneas desde el origen hasta cada vector)
trace_v1 = go.Scatter3d(
    x=[0, v1[0]], y=[0, v1[1]], z=[0, v1[2]],
    mode='lines+markers',
    line=dict(color='red', width=6),
    marker=dict(size=4),
    name='v1'
)

trace_v2 = go.Scatter3d(
    x=[0, v2[0]], y=[0, v2[1]], z=[0, v2[2]],
    mode='lines+markers',
    line=dict(color='green', width=6),
    marker=dict(size=4),
    name='v2'
)

trace_v3 = go.Scatter3d(
    x=[0, v3[0]], y=[0, v3[1]], z=[0, v3[2]],
    mode='lines+markers',
    line=dict(color='blue', width=6),
    marker=dict(size=4),
    name='v3'
)
```

```

# Generamos la malla para el plano definido por el span de v1 y v2
s = np.linspace(-1, 1, 10)
t = np.linspace(-1, 1, 10)
S, T = np.meshgrid(s, t)
X_plane = S * v1[0] + T * v2[0]
Y_plane = S * v1[1] + T * v2[1]
Z_plane = S * v1[2] + T * v2[2]

plano = go.Surface(
    x=X_plane,
    y=Y_plane,
    z=Z_plane,
    colorscale='YlOrRd',
    opacity=0.5,
    name='Span de v1 y v2'
)

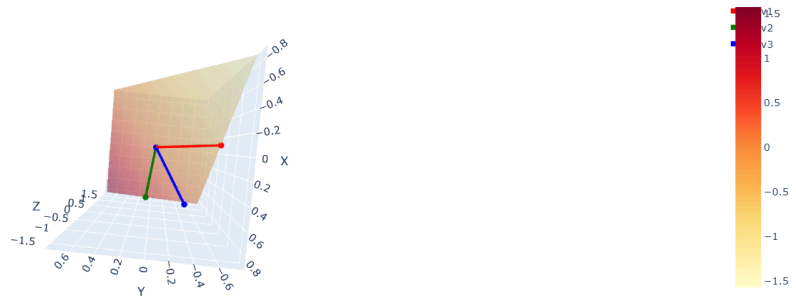
```

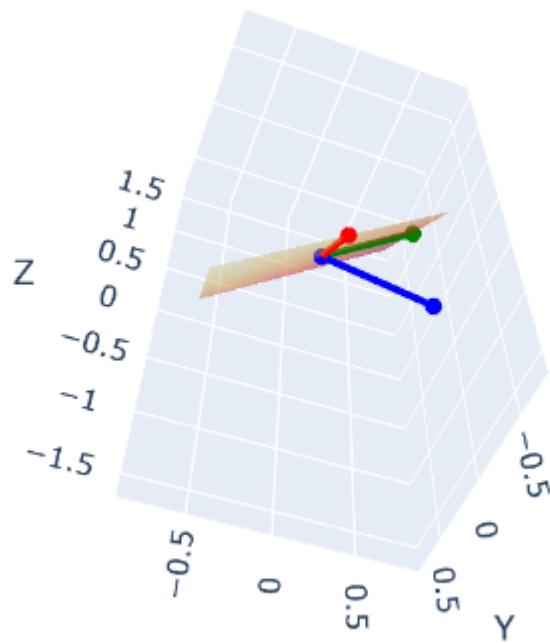
```

# Creamos la figura e incluimos los trazos
fig = go.Figure(data=[trace_v1, trace_v2, trace_v3, plano])
fig.update_layout(
    title='Vectores y plano del span de v1 y v2',
    scene=dict(
        xaxis_title='X',
        yaxis_title='Y',
        zaxis_title='Z'
    )
)

```

Vectores y plano del span de v1 y v2





Plataforma de Python con la imagen

<https://colab.research.google.com/drive/18gllrMnPtkG10-LdX3nGnV2StGqF7pXO?authuser=0#scrollTo=YzAwQagLdyK7>