The background of the slide features a complex network diagram with numerous nodes connected by lines, creating a web-like structure. The nodes are small squares, and the lines are thin and light blue. The overall color scheme is dark blue with lighter blue accents.

Aprendizaje de **Máquina Supervisado**

Sesión 1

Algoritmos de Clasificación

Exploraremos cinco algoritmos fundamentales: regresión logística, K-Nearest Neighbors, árboles de decisión, bosques aleatorios y máquinas de soporte vectorial (SVM).

Cada algoritmo tiene características únicas, ventajas y desventajas que los hacen adecuados para diferentes tipos de problemas. Veremos sus fundamentos matemáticos, implementación en Python y aplicaciones prácticas.



Regresión Logística

¿En qué consiste?

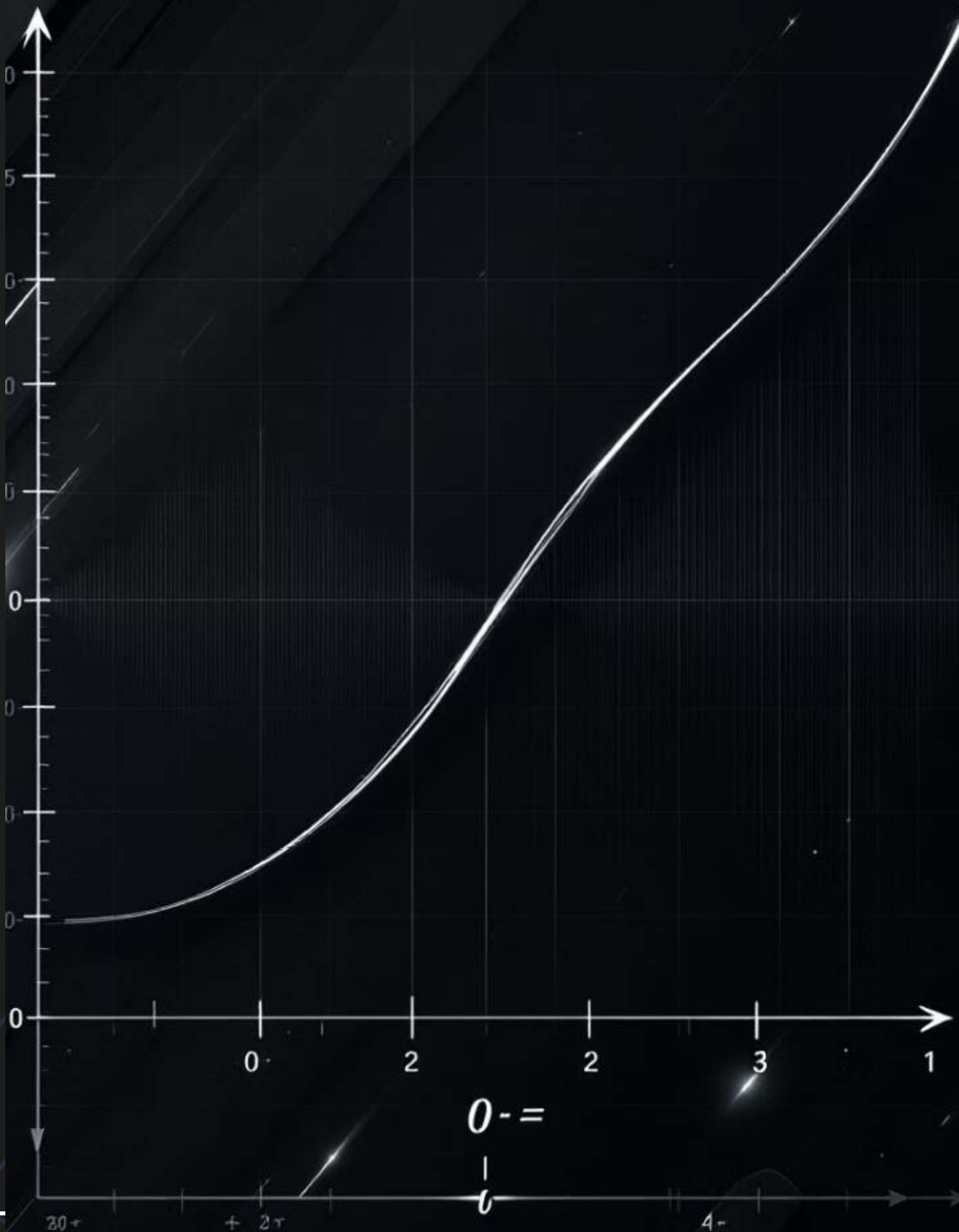
La regresión logística es un algoritmo de aprendizaje supervisado para clasificación binaria. Utiliza la función sigmoidea para transformar la salida de un modelo lineal en una probabilidad entre 0 y 1.

Función Sigmoidea

Transforma cualquier valor real en un rango entre 0 y 1. Cuando z es grande y positivo, $\sigma(z)$ se acerca a 1; cuando z es grande y negativo, $\sigma(z)$ se acerca a 0; y cuando $z = 0$, $\sigma(z) = 0.5$.

Ventajas y Desventajas

Entre sus ventajas destacan su simplicidad, eficiencia computacional e interpretabilidad. Sus desventajas incluyen limitación a relaciones lineales, sensibilidad a outliers y posible overfitting con demasiadas variables independientes.



K Nearest Neighbors (K-NN)

Fundamentos

K-NN es un método de aprendizaje supervisado utilizado tanto para clasificación como regresión. Es un algoritmo basado en instancias que no aprende un modelo explícito, sino que utiliza los datos directamente para hacer predicciones, asignando la etiqueta más común entre los k vecinos más cercanos.

Escalamiento de Datos

Es crucial escalar los datos para K-NN, ya que se basa en medidas de distancia. Si las variables tienen escalas diferentes, la variable con valores más grandes dominará la distancia. Los métodos comunes son la normalización (Min-Max Scaling) y la estandarización (Z-score Scaling).

Selección del Valor K

El valor de k afecta directamente la precisión del modelo. Un k pequeño puede causar overfitting y sensibilidad al ruido, mientras que un k grande puede causar underfitting. La elección óptima suele determinarse mediante validación cruzada.

Distancia Euclidiana en K-NN

Concepto

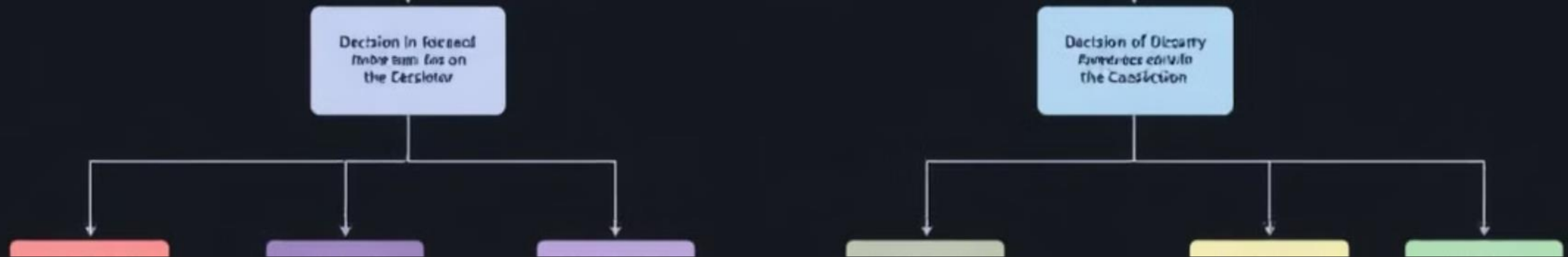
La distancia euclidiana es una medida entre dos puntos en un espacio multidimensional. Es la distancia "en línea recta" entre dos puntos y es fundamental para el algoritmo K-NN.

Aplicación

En K-NN, esta distancia determina cuáles son los k vecinos más cercanos a un punto de prueba, influyendo directamente en la clasificación o regresión final.

Fórmula

Para dos puntos $P=(p_1, p_2, \dots, p_n)$ y $Q=(q_1, q_2, \dots, q_n)$, la distancia euclidiana se calcula como la raíz cuadrada de la suma de los cuadrados de las diferencias entre las coordenadas correspondientes.



Árboles de Decisión

Estructura

1

Un árbol de decisión es un algoritmo de aprendizaje supervisado que divide recursivamente el conjunto de datos en subconjuntos más pequeños y homogéneos. Consta de nodos raíz (conjunto completo), nodos internos (decisiones) y nodos hoja (resultados).

2

Hiperparámetros

Los principales hiperparámetros incluyen max_depth (profundidad máxima), min_samples_split (muestras mínimas para dividir), min_samples_leaf (muestras mínimas en hojas), criterion (medida de impureza) y max_features (características máximas por división).

Medidas de Impureza

3

Las medidas de impureza evalúan la homogeneidad de las clases en un nodo. Las principales son el Índice de Gini (probabilidad de clasificación incorrecta), Entropía (desorden o incertidumbre) y Error Cuadrático Medio (para regresión).

Ventajas y desventajas árboles de Decisión

Ventajas

- Interpretabilidad: fáciles de entender y visualizar
- No requieren escalamiento de características
- Manejan datos mixtos (numéricos y categóricos)
- No paramétricos: no hacen suposiciones sobre la distribución

Desventajas

- Propensos a overfitting: pueden crear árboles muy complejos
- Inestabilidad: pequeños cambios generan árboles diferentes
- Sesgo hacia clases dominantes en problemas desbalanceados
- Limitados para capturar relaciones complejas no lineales



Bosques Aleatorios

1

Concepto de Bagging

Bosques aleatorios es un algoritmo de ensamblado que combina múltiples árboles de decisión. Utiliza Bootstrap Aggregating (Bagging), creando subconjuntos de datos mediante muestreo con reemplazo y entrenando un árbol en cada subconjunto.

2

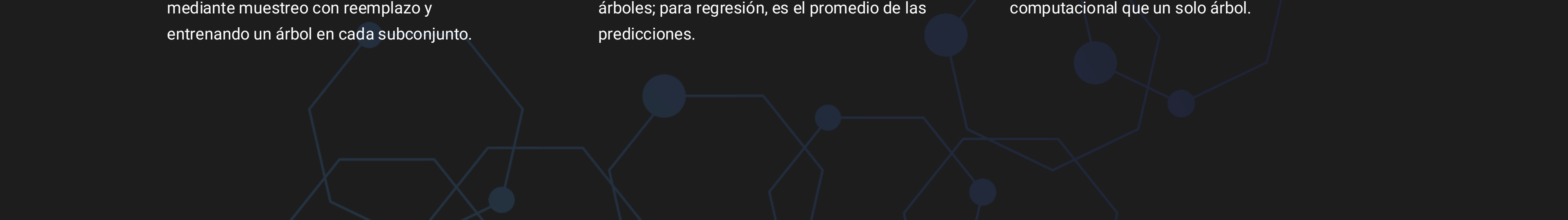
Entrenamiento

Cada árbol se entrena con un subconjunto aleatorio de datos y características, añadiendo diversidad. Para clasificación, la predicción final es la clase más frecuente entre todos los árboles; para regresión, es el promedio de las predicciones.

3

Ventajas y Desventajas

Ventajas: reducción del overfitting, robustez ante ruido y outliers, y medición de importancia de características. Desventajas: menor interpretabilidad y mayor costo computacional que un solo árbol.



Support Vector Machine (SVM)

Concepto Básico

Una Máquina de Soporte de Vectores (SVM) es un algoritmo de aprendizaje supervisado que busca encontrar un hiperplano óptimo que separe las clases maximizando el margen entre ellas. Los puntos más cercanos al hiperplano se denominan vectores de soporte.

Funcionamiento

Para datos no linealmente separables, SVM utiliza kernels para transformar los datos a un espacio de mayor dimensión. Resuelve un problema de optimización para encontrar el hiperplano óptimo, formulado como minimización de una función objetivo con restricciones.

Aplicaciones

SVM es efectivo para clasificación binaria, puede extenderse a problemas multiclase mediante técnicas como One-vs-One o One-vs-Rest, y también puede usarse para regresión (SVR) para predecir valores continuos.

Tipos de Kernel en SVM



Kernel Lineal

No realiza transformación, asumiendo que los datos son linealmente separables en el espacio original. Su fórmula es $K(x,y) = x \cdot y$. Se utiliza cuando los datos ya presentan separabilidad lineal.



Kernel Polinómico

Transforma los datos usando una función polinómica: $K(x,y) = (x \cdot y + c)^d$, donde c es una constante y d el grado del polinomio. Útil para datos con relaciones no lineales moderadamente complejas.



Kernel Radial (RBF)

Utiliza una función gaussiana: $K(x,y) = \exp(-\gamma \|x-y\|^2)$, donde γ controla el alcance de influencia de cada punto. Es el kernel más utilizado para relaciones no lineales complejas.



Kernel Sigmoidoide

Similar a la función de activación de una red neuronal: $K(x,y) = \tanh(\gamma x \cdot y + c)$. Menos común pero útil en ciertos casos específicos.

Ejercicio Guiado: Preparación de datos

Importamos Librerías

Cargamos Datos

Dividimos datos

```
1  from sklearn.datasets import load_iris
2  import pandas as pd
3  from sklearn.model_selection import train_test_split
4  from sklearn.linear_model import LogisticRegression
5  from sklearn.metrics import accuracy_score
6  from sklearn.tree import DecisionTreeClassifier
7  from sklearn.ensemble import RandomForestClassifier
8  from sklearn.svm import SVC
9
10 # Cargar el conjunto de datos Iris
11 data = load_iris()
12 X = pd.DataFrame(data.data, columns=data.feature_names) # Características
13 y = pd.Series(data.target) # Etiquetas
14
15 # Mostrar las primeras filas de los datos
16 print("Características (X):")
17 print(X.head())
18
19 print("\nEtiquetas (y):")
20 print(y.head())
21
22 # Información sobre el conjunto de datos
23 # print("\nInformación del conjunto de datos:")
24 # print(data.DESCR)
25
26 # Dividir los datos en entrenamiento y prueba
27 X_train, X_test, y_train, y_test = train_test_split(
28     X, y, test_size=0.25, random_state=42
29 )
30
31 print(f"\nTamaño del conjunto de entrenamiento: {X_train.shape}")
32 print(f"Tamaño del conjunto de prueba: {X_test.shape}")
```


Ejercicio Guiado: Entrenar y Evaluar

Regresión Logística

Árbol de decisión

Bosque Aleatorio

SVM

```
35 # Crear y entrenar el modelo de regresión logística
36 modelo_lr = LogisticRegression(max_iter=200, random_state=42)
37 modelo_lr.fit(X_train, y_train)
38 # Predecir y evaluar
39 y_pred_lr = modelo_lr.predict(X_test)
40 print("\nExactitud de Regresión Logística:", accuracy_score(y_test, y_pred_lr))
41
42 # Crear y entrenar el modelo de árbol de decisión
43 modelo_dt = DecisionTreeClassifier(max_depth=3, random_state=42)
44 modelo_dt.fit(X_train, y_train)
45 # Predecir y evaluar
46 y_pred_dt = modelo_dt.predict(X_test)
47 print("Exactitud de Árbol de Decisión:", accuracy_score(y_test, y_pred_dt))
48
49 # Crear y entrenar el modelo de bosque aleatorio
50 modelo_rf = RandomForestClassifier(n_estimators=100, random_state=42)
51 modelo_rf.fit(X_train, y_train)
52 # Predecir y evaluar
53 y_pred_rf = modelo_rf.predict(X_test)
54 print("Exactitud de Bosque Aleatorio:", accuracy_score(y_test, y_pred_rf))
55
56 # Crear y entrenar el modelo SVM
57 modelo_svm = SVC(kernel="rbf", gamma="scale", random_state=42)
58 modelo_svm.fit(X_train, y_train)
59 # Predecir y evaluar
60 y_pred_svm = modelo_svm.predict(X_test)
61 print("Exactitud de SVM:", accuracy_score(y_test, y_pred_svm))
62
```

Ejercicio Guiado: Comparación de modelos

Comparación

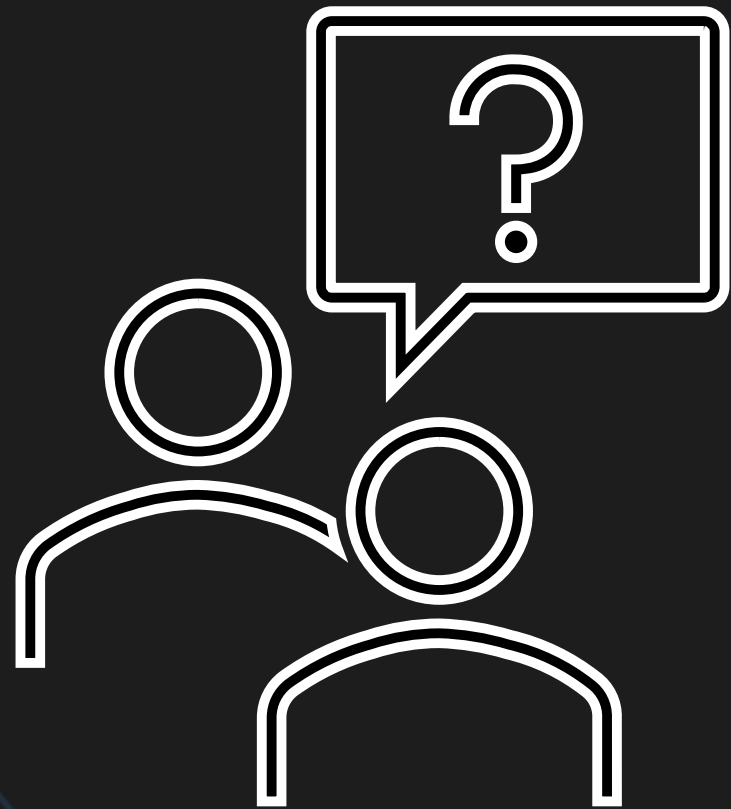
```
63 print("\nComparación de Exactitud:")
64 print(f"- Regresión Logística: {accuracy_score(y_test, y_pred_lr):.4f}")
65 print(f"- Árbol de Decisión: {accuracy_score(y_test, y_pred_dt):.4f}")
66 print(f"- Bosque Aleatorio: {accuracy_score(y_test, y_pred_rf):.4f}")
67 print(f"- SVM: {accuracy_score(y_test, y_pred_svm):.4f}")
```

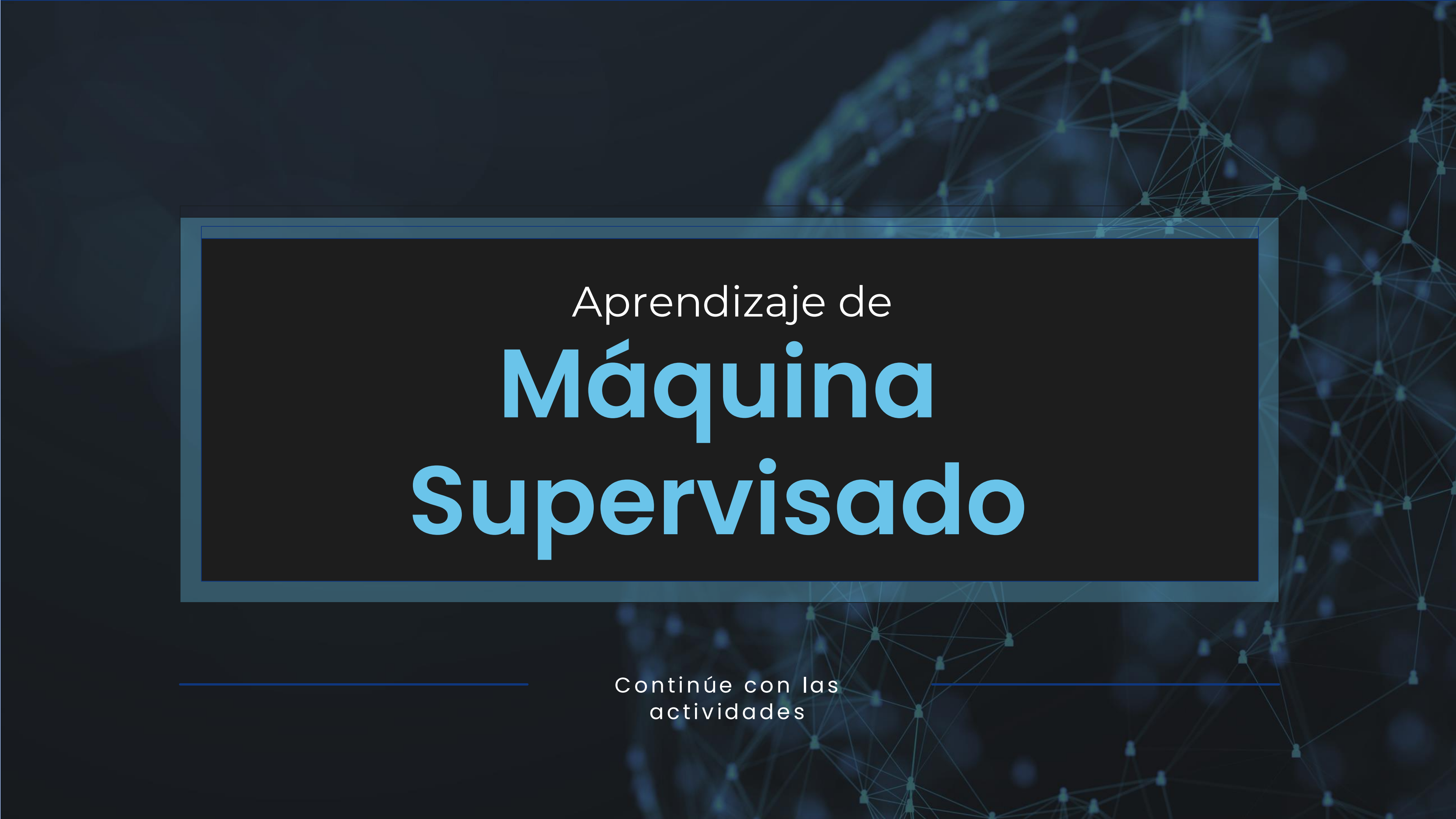
Salida

```
Comparación de Exactitud:
- Regresión Logística: 1.0000
- Árbol de Decisión: 1.0000
- Bosque Aleatorio: 1.0000
- SVM: 1.0000
```

Preguntas

Sección de preguntas



The background of the slide features a complex network diagram with numerous nodes connected by lines, creating a web-like structure. The nodes are small squares, and the lines are thin and light blue. The overall color scheme is dark blue with lighter blue accents.

Aprendizaje de **Máquina Supervisado**

Continúe con las
actividades
