**1.Respuesta 1 Pregunta 3 pf-m3**

Entrenar modelos de machine learning implica alimentar algoritmos con datos para que puedan aprender patrones y realizar predicciones o tomar decisiones automatizadas

El proceso de entrenamiento del machine learning consta de varias etapas, que van desde la preparación y limpieza de los datos hasta la evaluación y ajuste del modelo. Es fundamental tener en cuenta aspectos como el overfitting, underfitting y la selección adecuada de métricas de evaluación para garantizar la eficacia del modelo.

Diagrama

Descripción generada automáticamente

Nos vamos a centrar en el paso de entrenamiento del modelo.

**Entrenamiento del modelo**

Una vez que hemos seleccionado el modelo más adecuado para nuestro problema, es hora de entrenarlo con el conjunto de datos de entrenamiento que hemos preparado en la sección anterior. Es importante evaluar (en una siguiente fase) el desempeño del modelo utilizando métricas apropiadas y técnicas como la validación cruzada para garantizar su generalización a nuevos datos.

**División del conjunto de datos en entrenamiento y prueba**

Antes de empezar a entrenar el modelo, **debemos dividir nuestro conjunto de datos en dos partes:** el conjunto de datos de entrenamiento y el conjunto de datos de prueba. La idea es utilizar el conjunto de datos de entrenamiento para ajustar los parámetros del modelo y luego utilizar el conjunto de datos de prueba para evaluar su rendimiento en datos que no ha visto antes.

En Scikit-learn, podemos utilizar la función *train\_test\_split* del módulo *sklearn.model\_selection* para realizar esta división. Esta función recibe como entrada los datos que queremos dividir y el porcentaje que queremos utilizar para el conjunto de datos de prueba. Por ejemplo, si queremos utilizar el 30% de los datos para prueba, podemos llamar a la función de la siguiente manera:

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

En este ejemplo, *X* e *y* son los datos que queremos dividir y *test\_size* es el porcentaje que queremos utilizar para prueba (en este caso, el 30%). La función devuelve cuatro conjuntos de datos: *X\_train* y *y\_train* corresponden al conjunto de datos de entrenamiento, mientras que *X\_test* e *y\_test* corresponden al conjunto de datos de prueba.

**Entrenamiento del modelo con el conjunto de datos de entrenamiento**

Una vez que tenemos los conjuntos de datos de entrenamiento y prueba, podemos entrenar nuestro modelo utilizando los datos de entrenamiento. Para ello, simplemente tenemos que llamar al método *fit* del objeto que representa nuestro modelo. Por ejemplo, si estamos trabajando con un modelo de regresión lineal, podemos entrenarlo de la siguiente manera:

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

model.fit(X\_train, y\_train)

En este ejemplo, hemos creado un objeto *model* que representa nuestro modelo de regresión lineal y lo hemos entrenado con los conjuntos de datos de entrenamiento *X\_train* e *y\_train*. Después de ejecutar esta línea de código, nuestro modelo estará ajustado a los datos de entrenamiento y podremos utilizarlo para hacer predicciones en nuevos datos.

La Regresión Lineal, uno de los algoritmos básicos del Machine Learning.

La Regresión Lineal es un procedimiento que permite encontrar la línea recta que mejor representa la relación entre dos [variables.es](http://variables.es) decir que un incremento en la variable *x* genera un incremento proporcional en la variable *y*.

El objetivo de la Regresión Lineal es entonces encontrar la línea recta que mejor se ajusta a los datos. Para ello utilizamos la ecuación 𝑦=𝑤𝑥+𝑏, donde *w* representa la pendiente (inclinación) de la línea recta y *b* es la intersección con el eje *y*. Así, el objetivo de la Regresión Lineal es encontrar los valores más adecuados de *w* y *b* que representen de la mejor forma posible la relación entre las variables *x* y *y*.

Para seguir entendiendo el desarrollo de la Regresión Lineal, por favor remitirse a este [link](https://www.codificandobits.com/blog/regresion-lineal/), que lo explica perfectamente.

Otros links de referencia:

1. [Guía completa sobre cómo entrenar el machine learning - Triunfa Emprendiendo](https://triunfaemprendiendo.com/como-se-entrena-el-machine-learning/)
2. [Entrenar un modelo de Machine Learning con Scikit-learn | OpenWebinars](https://openwebinars.net/blog/como-entrenar-un-modelo-de-machine-learning-con-scikit-learn/)
3. <https://www.codificandobits.com/blog/regresion-lineal/>
4. <https://www.themachinelearners.com/regresion-lineal/>