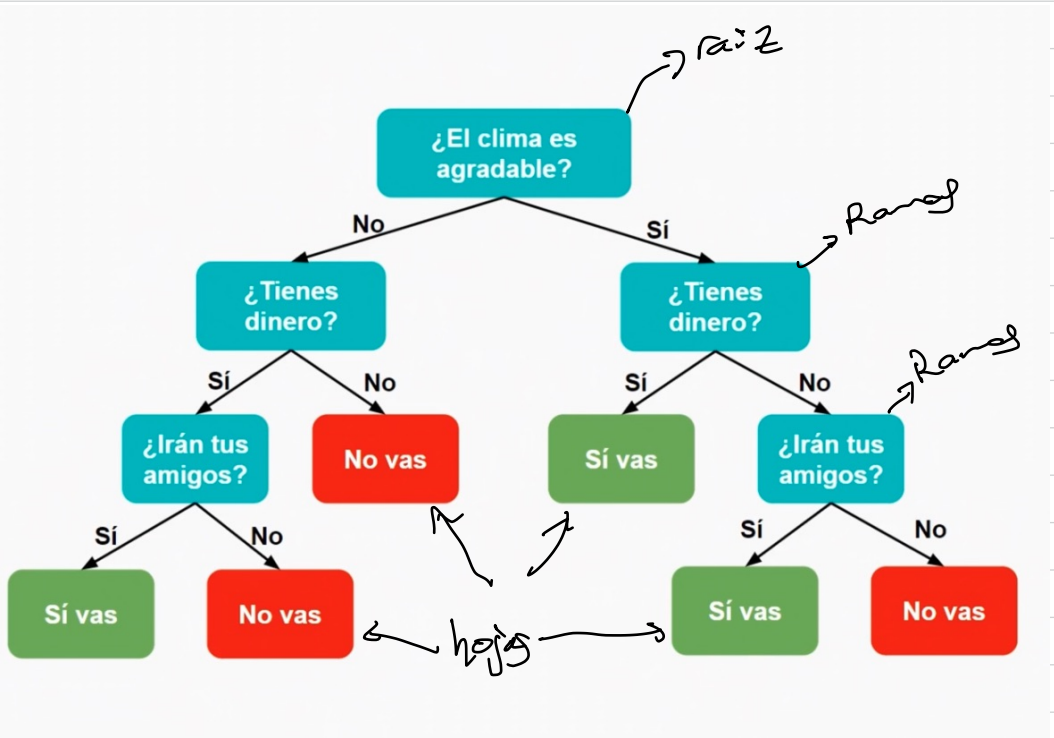


Los Árboles de decisión, son algoritmos de ML, que se pueden usar como clasificación y regresión

Me voy de fiesta?



Crear el objeto

```
from sklearn.tree import DecisionTreeClassifier

# hay que tener cuidado, con este parametro, para no hacer que se sibreajuste

modelo = DecisionTreeClassifier(max_depth = 50 ) # que tan profundo puede ser el arbol hacia abajo

modelo.fit(X_train,y_train)

predicciones = modelo.predict(X_test)
```

Evaluar el puntaje de exactitud para comparar con el modelo anterior

```
#evaluar como le fué en el entrenamiento

from sklearn.metrics import accuracy_score

accuracy_score(y_test, predicciones)
```

Podemos de forma simple buscar la mejor profundidad

```
) #evaluar multiples profundidades
from sklearn.tree import DecisionTreeClassifier

resultados = []
for i in range (1,15):
    modelo = DecisionTreeClassifier(max_depth = i)
    modelo.fit(X_train,y_train)
    predicciones = modelo.predict(X_test)
    exactitud = accuracy_score(y_test, predicciones)
    print(f"resultado para {i}: {exactitud}")
    resultados.append(exactitud)
```

Podemos Graficar el arbol de decisi3n

```
# se puede graficar el arbol para ver las decisiones
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

plt.figure(figsize=(100,80))
plot_tree(
    modelo,
    feature_names= X_train.columns,
    class_names=["Muri3", "Vivi3"], # se debe colocar en orden ascendentes los valores numericos
    filled=True, # para rellenar los recuadros que va a dibujar
    label="none" # para evitar que muestra informaci3n innecesaria
)
```