DTC:

import pickle

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.metrics import accuracy\_score

from sklearn.datasets import make\_classification

# --- Replace below with your dataset ---

X, y = make\_classification(n\_samples=1000, n\_features=10, n\_classes=2, random\_state=42)

# ----------------------------------------

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = DecisionTreeClassifier()

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

y\_pred = model.predict(X\_test)

print(f"Decision Tree Accuracy: {accuracy\_score(y\_test, y\_pred):.4f}")

with open('decision\_tree\_model.pkl', 'wb') as f:

pickle.dump(model, f)

print("Decision Tree model saved as decision\_tree\_model.pkl")

DTR:

import pickle

from sklearn.tree import DecisionTreeRegressor

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

from sklearn.metrics import mean\_squared\_error

from sklearn.datasets import make\_regression

# --- Replace below with your dataset ---

X, y = make\_regression(n\_samples=1000, n\_features=10, noise=0.1, random\_state=42)

# ----------------------------------------

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = DecisionTreeRegressor()

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

y\_pred = model.predict(X\_test)

print(f"Decision Tree Regression Mean Squared Error: {mean\_squared\_error(y\_test, y\_pred):.4f}")

with open('decision\_tree\_regressor\_model.pkl', 'wb') as f:

pickle.dump(model, f)

print("Decision Tree Regressor model saved as decision\_tree\_regressor\_model.pkl")