Xgboostregression:

import pickle

import xgboost as xgb

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 = xgb.XGBRegressor()

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

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

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

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

pickle.dump(model, f)

print("XGBoost Regressor model saved as xgboost\_regressor\_model.pkl")

XGBC:

import pickle

import xgboost as xgb

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 = xgb.XGBClassifier(use\_label\_encoder=False, eval\_metric='logloss')

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

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

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

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

pickle.dump(model, f)

print("XGBoost model saved as xgboost\_model.pkl")