

```

import argparse
from model_pipeline import prepare_data, train_model, evaluate_model, save_model, load_model, plot_confusion_matrix

def main():
    # Configuration des arguments CLI
    parser = argparse.ArgumentParser(description="Pipeline de classification Machine Learning")
    parser.add_argument("--data", required=True, help='C:/Users/ROYAUME MEDIAS/Downloads/ML_Project_Files/archive (2)/merged_data.csv')
    parser.add_argument("--target", required=True, help='Churn')
    parser.add_argument("--model", required=True, help="model_pipeline.py")
    parser.add_argument("--action", required=True, choices=["train", "evaluate"], help="Action à effectuer (train/evaluate)")
    parser.add_argument("--test_size", type=float, default=0.2)
    parser.add_argument("--random_state", type=int, default=42, help="Graine aléatoire pour la reproductibilité")
    args = parser.parse_args()

    # Chargement et préparation des données
    print("Préparation des données...")
    X_train, X_test, y_train, y_test = prepare_data(
        filepath=args.data,
        target_column=args.target,
        test_size=args.test_size,
        random_state=args.random_state
    )

    if args.action == "train":
        # Entraînement du modèle
        print("Entraînement du modèle...")
        model = train_model(X_train, y_train, random_state=args.random_state)

        # Sauvegarde du modèle
        print(f"Sauvegarde du modèle dans : {args.model}")
        save_model(model, args.model)

    elif args.action == "evaluate":
        # Chargement du modèle
        print(f"Chargement du modèle depuis : {args.model}")
        model = load_model(args.model)

        # Évaluation du modèle
        print("Évaluation du modèle...")
        accuracy, report, conf_matrix = evaluate_model(model, X_test, y_test)

        print(f"Précision : {accuracy * 100:.2f}%")
        print("\nRapport de classification :")
        print(report)

        # Affichage de la matrice de confusion
        plot_confusion_matrix(conf_matrix, labels=["Classe 0", "Classe 1"], title="Matrice de Confusion")

if __name__ == "__main__":
    main()

```