1. Executive Summary

This report documents the deployment and application implementation phase of the churn prediction project. Two key components were developed:

- Streamlit Web Application (app.py) – A user-friendly interface allowing real-time customer churn prediction.
- MLFlow Deployment Script (MLFlow_Deployment.py) – A script to log all trained models and their performance metrics to MLFlow for experiment tracking and model management.

The XGBoost model (previously identified as the best-performing model) has been successfully integrated into the Streamlit application, enabling stakeholders to make quick churn predictions with new customer data.

2. Application Implementation (app.py)

2.1. Purpose

The Streamlit application serves as an interactive tool for customer service teams or business analysts to predict customer churn based on input features.

2.2. Key Features

User Inputs: Sidebar form for entering customer details (state, account length, plan details, call minutes, charges, etc.)

Feature Engineering:

- Derived features: total_national_minutes, avg_minutes_per_call, tenure_category, has_all_plans, etc.
- Binary flags: high_service_calls, zero_vmail_messages

Model Integration:

- Loads the pre-trained XGBoost model (best.joblib)
- Uses a preprocessor (preprocessor.joblib) for consistent data transformation

Prediction Output:

- Displays churn prediction (Not Likely to Churn / Likely to Churn)
- Shows probability score when available via predict_proba

2.3. Technical Notes

- The app uses State from the original dataset (churn_cleaned.csv) for dropdown options
- Input validation is implicit via Streamlit's number/select inputs
- The preprocessing pipeline ensures inputs match the model's expected feature space

3. MLFlow Deployment (MLFlow_Deployment.py)

3.1. Purpose

To systematically log all trained models and their performance metrics into MLFlow for:

- Experiment tracking
- Model versioning
- Performance comparison across models

3.2. Implementation Details

Models Tracked: All seven models from the training phase (XGBoost, Random Forest, Decision Tree, SVC, KNN, GaussianNB, Logistic Regression)

Metrics Logged:

- Accuracy, Precision, Recall, F1-Score, AUC
- Confusion matrix (saved as a temporary PNG and logged as an artifact)

Workflow:

- Loads test data (X_test_scaled.csv, y_test.csv)
- Iterates through each model, generates predictions, computes metrics
- Logs metrics and model artifacts to MLFlow under the experiment "Customer Churn Prediction"

3.3. Output

- All models and their performance metrics are visible in the MLFlow UI
- Confusion matrices are stored as artifacts for visual comparison
- Enables reproducible model selection and audit trails

4. Integration & Deployment Readiness

4.1. Model Consistency

- The same preprocessed test set (X_test_scaled.csv) is used in both the evaluation report and MLFlow deployment
- The Streamlit app uses the same preprocessing pipeline (preprocessor.joblib) as the training phase

4.2. Deployment Status

- The Streamlit app is ready for deployment on Streamlit Cloud or a similar platform
- MLFlow tracking can be hosted locally or on a remote server for collaborative model management

5. Conclusion

Both the Streamlit application and MLFlow deployment script have been successfully implemented, completing the end-to-end machine learning pipeline from model training to deployment. The XGBoost model is now accessible via a user-friendly web interface, while all model versions and experiments are tracked systematically in MLFlow for future reference and iteration.

This report confirms that the churn prediction system is fully functional, well-documented, and ready for operational use.