DROPPED CONNECTION PREDICTIONS ON TELECOMM NETWORKS

PRESENTED BY:

FLORENCE JOY ARAN

INTRODUCTION

- **The Challenge of Dropped Connections**
  - **Problem:** Abruptly ended calls or data sessions in 5G networks.
  - **Impact:** Degraded user experience, deeper network performance issues, direct impact on telecom revenue and brand reputation.
  - **Solution:** Predictive modeling to anticipate and prevent dropped connections.

OBJECTIVES

- **Our Mission: Ensuring Seamless Connectivity**
- **Binary Classification:** Develop a model to predict if a mobile connection will be:
  - 0: Stable
  - 1: Dropped
- **Feature Importance:** Identify and interpret key factors contributing to connection drops (e.g., signal strength, jitter, congestion).
- **Model Assessment:** Train, validate, and evaluate the model using metrics like F1-score and confusion matrix.

BUSINESS UNDERSTANDING

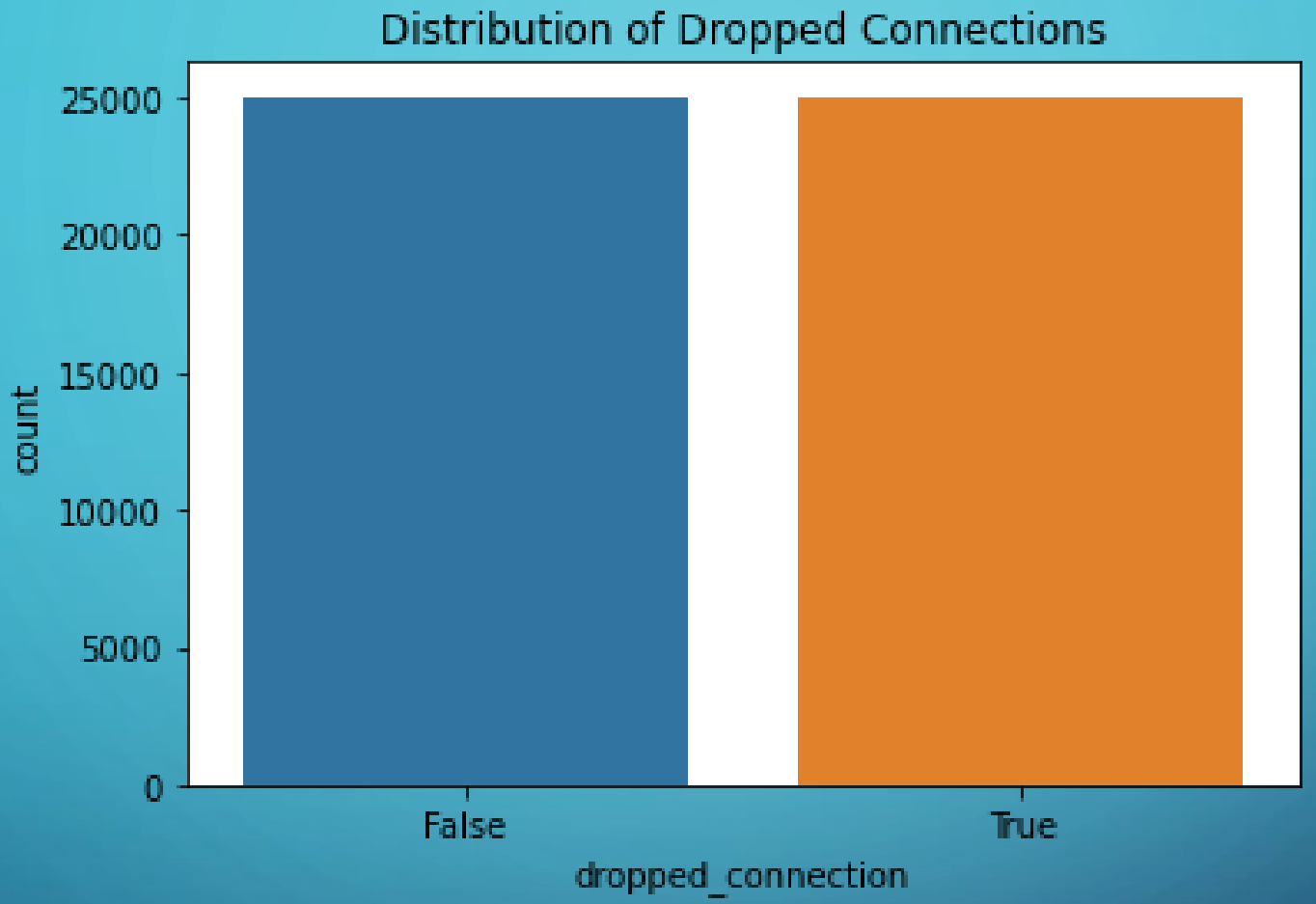
- **From Reactive to Proactive:** Shift from fixing issues after they occur to predicting and preventing them.
- **Improved Service Reliability:** Enhance customer satisfaction by minimizing service interruptions.
- **Reduced Churn:** Prevent customers from switching providers due to poor service.
- **Optimized Infrastructure:** Make data-driven decisions for efficient resource allocation (load balancing, signal boosting, client rerouting).

DATA UNDERSTANDING

- **The Dataset: Real-time 5G Network Telemetry**
- **Source:** Real-time measurements from mobile devices and network logs.
- **Key Features:**
  - signal_strength_dbm: Power level of received signal (more negative = weaker).
  - download_speed_megabits_per_second: Data rate for downloads.
  - upload_speed_megabits_per_second: Data rate for uploads.
  - latency_milliseconds: Delay in data transmission.
  - jitter_milliseconds: Variability in latency.
- **Target Variable:** dropped_connection (True/False or 1/0).

EXPLORATORY DATA ANALYSIS

- **Uncovering Patterns in Connection Drops**
- **Signal Strength:** Dropped connections are more frequent under poor signal conditions (e.g., below -90 dBm).
- **Latency & Jitter:** High latency and increased jitter are associated with higher dropped connection rates.
- **Handover Count:** Sessions with multiple handovers show increased instability and higher chances of drops



The graph shows the distribution of dropped connections in the dataset. The class seems balanced.

MODELING

Building the Predictive Model

- **Libraries:** pandas, numpy, matplotlib, seaborn, sklearn, imblearn.
- **Data Preparation:**
  - Encoding of categorical features (Network Type, Device Model, Carrier, Band, Network Congestion Level, VoNR Enabled, Location).
  - Scaling of numerical features.
- **Model Selection:**
  - Initial experiments included Logistic Regression and Decision Tree Classifier.
  - Pipeline used for combining preprocessing steps and model training.

MODEL EVALUATION AND RESULTS

Performance and Limitations

- **Key Metrics:** F1-score and Confusion Matrix were used for evaluation.
- **Observations:**
  - Models (Logistic Regression, Decision Tree) showed limited predictive power.
  - F1-scores were low, indicating poor performance, especially for the minority class (dropped connections).
  - Confusion matrices revealed a bias towards the majority class (non-dropped connections).

- The root causes of the poor performance might be;
  - Potentially overlapping feature distributions.
  - The current feature set and data quality might be insufficient for reliable prediction.

NEXT STEPS & FUTURE WORK

- **Feature Engineering:**
  - Develop new features: time-of-day indicators, signal strength variability, interactions between latency and jitter.
  - Uncover hidden patterns crucial for dropped connection prediction.
- **Robust Model Exploration:**
  - Experiment with ensemble-based classifiers like Random Forest or XGBoost.
  - These models can capture complex relationships more effectively than linear or shallow models.
- **Advanced Validation & Tuning:**
  - Apply stratified cross-validation to ensure consistent performance across all data subsets.
  - Focus on probability threshold tuning to optimize performance for the minority class.