**📌 Objective:**

To create a classification model that predicts the probability of customer churn using a large dataset with:

* **167,000 rows**
* **215+ features (X0–X215)**
* **Binary target: Target\_ChurnFlag**

**🧠** Domain Note: In churn prediction, false negatives (predicting a customer will stay when they actually churn) are far more critical than false positives, as missing real churners directly impacts customer retention and revenue. This insight shaped the model evaluation and final choice.   
- I did not use RandomForest Classifier due to RAM Limitations

**✅ Approach 1: Keep Everything Untouched (Final Preferred Approach)**

• No missing value imputation  
• No outlier removal  
• No duplicate removal  
• Cleaned column types, label encoded categorical variables, and trained a LightGBM model

**🔍 *Outcome*:**

• Accuracy: 96%  
• Confusion Matrix:  
  • Low false positives and false negatives  
• Recall (Churn Class 1): Excellent  
• AUC Score: ~0.99  
• Model Generalization: Strong across test set  
• This approach used the natural signals in data, even from missing values and outliers

**💡 *Insight:***

In real-world churn prediction, missing values and outliers often provide hidden signals (e.g., unengaged users, drop-off behavior) that models like LightGBM can use effectively**.**

**A blue squares with white text

AI-generated content may be incorrect.**

**⚠️ Approach 2: Drop Missing Values, Outliers & Duplicates**

**•** Dropped:  
  • All columns with high missingness  
  • Top 10 columns with extreme outlier counts  
  • Duplicates  
• Reduced dataset from 215+ to ~190 features

***🔍 Outcome:***

• Accuracy dropped compared to the previous approach  
• False negatives increased significantly  
  (i.e., churners predicted as non-churners — a major concern in churn modeling)  
• Recall for Class 1 (churn): dropped

***🚩 Red Flag:***

High false negatives in churn prediction means risky business decisions.  
Missing a real churner is worse than falsely identifying one.

A blue squares with white text

AI-generated content may be incorrect.

**🧪 Approach 3: Handle Missing Values via Imputation**

**• Imputed:**  • Categorical → using mode  
  • Numeric → using median  
  • Dates → using median date  
• Kept all columns but filled all missing entries

***🔍 Outcome:***

• Model achieved unrealistically perfect scores:  
  • Accuracy: ~100%  
  • Confusion Matrix: No misclassifications  
• Clear overfitting occurred — the model likely memorized the training data

***⚠️ Likely Reasons:***

• Leakage during preprocessing  
• Overtraining without proper validation  
• Preserved all signals but introduced false patterns

A blue squares with white text

AI-generated content may be incorrect.

**🏆 Final Choice: Return to Approach 1 (Original Best-Performing)**

**✅ Why?**

• Gave the most balanced, generalizable, and interpretable results  
• Treated LightGBM as an effective learner that handles missing values directly  
• Avoided bias from artificial imputation  
• Preserved important signals in missing values and extreme entries

**🛠️ Added Measures:**

• Performed stratified train-test split  
• Used early stopping and regularization  
• Verified via confusion matrix, ROC AUC, and recall

**🔍 Conclusion**

| **Approach** | **Accuracy** | **FN Rate** | **Overfitting** | **Final Verdict** |
| --- | --- | --- | --- | --- |
| 1. Untouched | ✅ High | ✅ Low | ❌ No | ✅ Best |
| 2. Dropped | ⚠️ Medium | ❌ High | ❌ No | ❌ Not Preferred |
| 3. Imputed | ✅ Too High | ✅ Zero | ✅ Yes | ❌ Overfit |

**🖥️ Deployment: Streamlit Web App**

To make the churn prediction system accessible and interactive, a **Streamlit-based web application** was developed.

**✅ App Features:**

* User-friendly form interface to input customer details
* Focuses on **top 15 most important features** for simplicity
* Backend automatically fills the rest of the 215+ features with defaults
* Handles data consistency by ensuring:
  + All model-required features are present
  + Categorical columns are cast to the correct types (category)
* Displays churn probability in real-time using the trained LightGBM model

**⚙️ Technical Highlights:**

* **Model Inference:** Uses the exact feature\_columns.pkl and categorical\_columns.pkl saved during training
* **Input Handling:** Missing features are defaulted to zero (neutral baseline)
* **Error Handling:** Captures user input errors and parsing failures gracefully

**📊 User Workflow:**

1. User enters values for 15 key features like X19, X85, X104, etc.
2. The app constructs a full feature vector of size 215+ with appropriate types
3. Model outputs a churn probability (e.g., **18.75%**) in under 1 second

**🚀 Deployment-Ready:**

* Compatible with **Streamlit Cloud**, **Render**, or internal demo servers
* Can be extended to include CSV uploads, batch prediction, or SHAP-based feature explanations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.