# Churn Prediction for KKBox

*a music platform*

**Group 1:**
Chew En Chin (Emma)
Nguyen Thi Ngoc Linh
Le Trung Kien
Mohammad Farhan Ahsan
Nguyen Phuong Hoa

**01**

# Introduction

# Background & Business Problem

**Background: 2017**

KKBOX is Asia's leading music streaming service operating on user subscription revenue. The company previously used a one-size-fits-all ***retention strategy*** towards all customers to maintain their subscriber userbase.

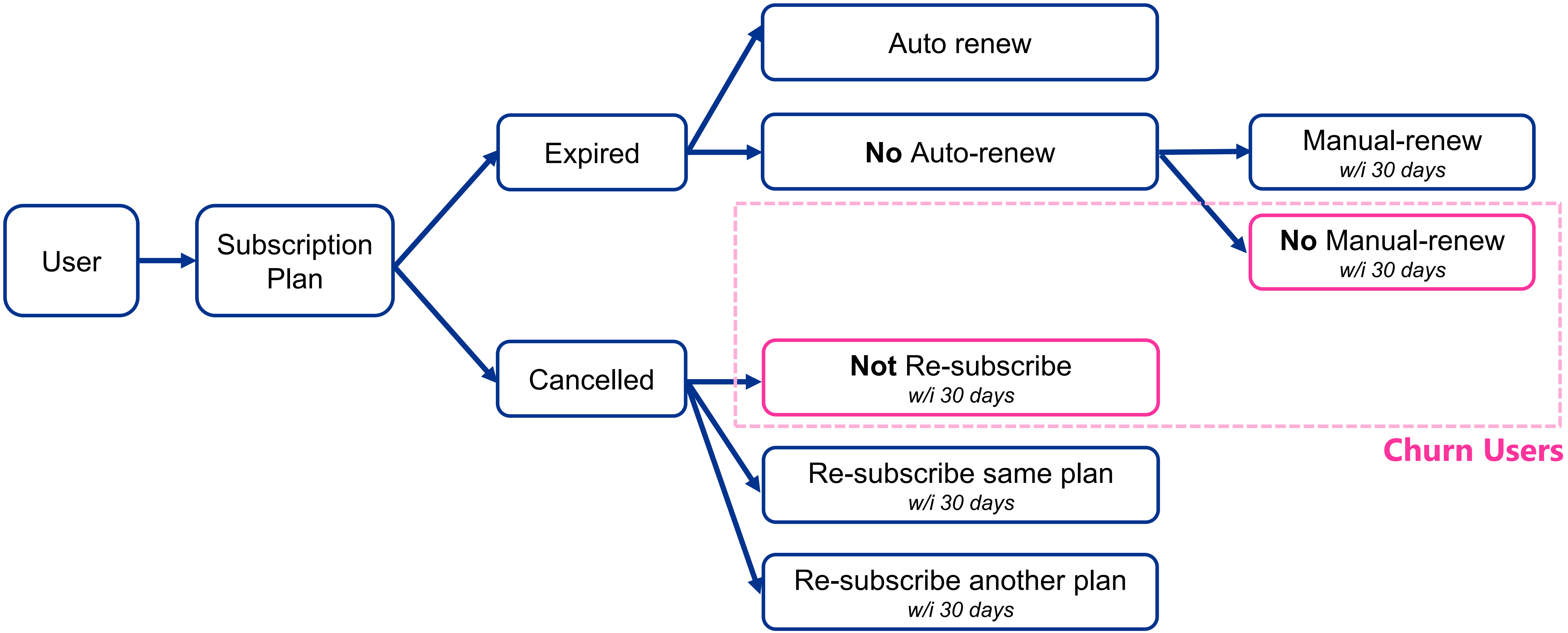
**Business Problem:**

As competition in the music streaming space intensifies, and as userbase grows KKBOX needs more **scalable methods to identify at-risk users** on the day when their subscriptions expire.

**Business Metrics:**

The business aims to reduce **Churn Rate %**, **User Retention Cost** and improve **Customer Lifetime Value (CLV)** with the usage of a ML model.

# In-App User Journey




**Definition of Churn**

A user who fails to resubscribe within 30 days after their subscription has expired is considered churned.

**02**

# Dataset

# Dataset

This dataset comes from a Kaggle based on KKBOX's Churn Prediction Challenge (WSDM 2018), using multi-source historical data to predict user churn.
KKBOX is Asia's leading music streaming service.

| Dataset | Description | Time Period | Key Columns |
|---|---|---|---|
| **user_logs.csv** | Daily user listening behavior (song completions, total seconds, unique songs). | 2015-01-01 → 2017-02-28 | `msno, date, num_25, num_50, num_75, num_985, num_100, num_unq, total_secs` |
| **user_logs_v2.csv** | Continuation of user_logs for March 2017. | 2017-03-01 → 2017-03-31 | `Same columns as above` |
| **transactions.csv** | Subscription payments, plan details, renewals, cancellations. | 2015-01-01 → 2017-02-28 | `msno, payment_method_id, payment_plan_days, plan_list_price, actual_amount_paid, is_auto_renew, transaction_date, membership_expire_date, is_cancel` |
| **transactions_v2.csv** | Transaction continuation (fewer rows). | 2015-01-01 → 2017-03-31 | `Same columns as above` |
| **members_v3.csv** | User demographics and registration metadata. | — | `msno, city, bd, gender, registered_via, registration_init_time` |

03

# Model Selection

# Model Selection — Algorithms

**Task**
Classification: Predict if a user will churn (1) or stay (0)
Target: is_churn (30 days post-subscription)

**Models**
- Logistic Regression – interpretable baseline
- Decision Tree – visual, easy to explain
- XGBoost – high accuracy, SHAP-based interpretability
- Random Forest – stable, handles skewed data

**Model Selection Rationale**
Explainability: Models must provide insights into key churn drivers.
Scalability: Efficient handling of millions of user logs and transaction records.
Performance: Tree-based ensembles deliver strong predictive power without sacrificing interpretability.

**Dataset Size & Complexity**
Over 6 million users, 1.4 million transactions, 300~ million user log data
Dataset includes categorical, numerical, temporal features
Imbalanced classes will be handled with class weighting or SMOTE

04

# Data Pipeline Architecture

# Design Principles

### 📚 Batch-First Architecture

Daily batch processing (not real-time)- minutes to hours acceptable for 10K-1M users

### 🌎 Scalable Data Processing

Handle multi-GB CSV data from multiple sources with automated cleaning and feature engineering

### 🤖 Full Pipeline Automation

Orchestrate end-to-end workflow from ingestion to prediction with scheduled daily execution

### 👣 Experiment Tracking

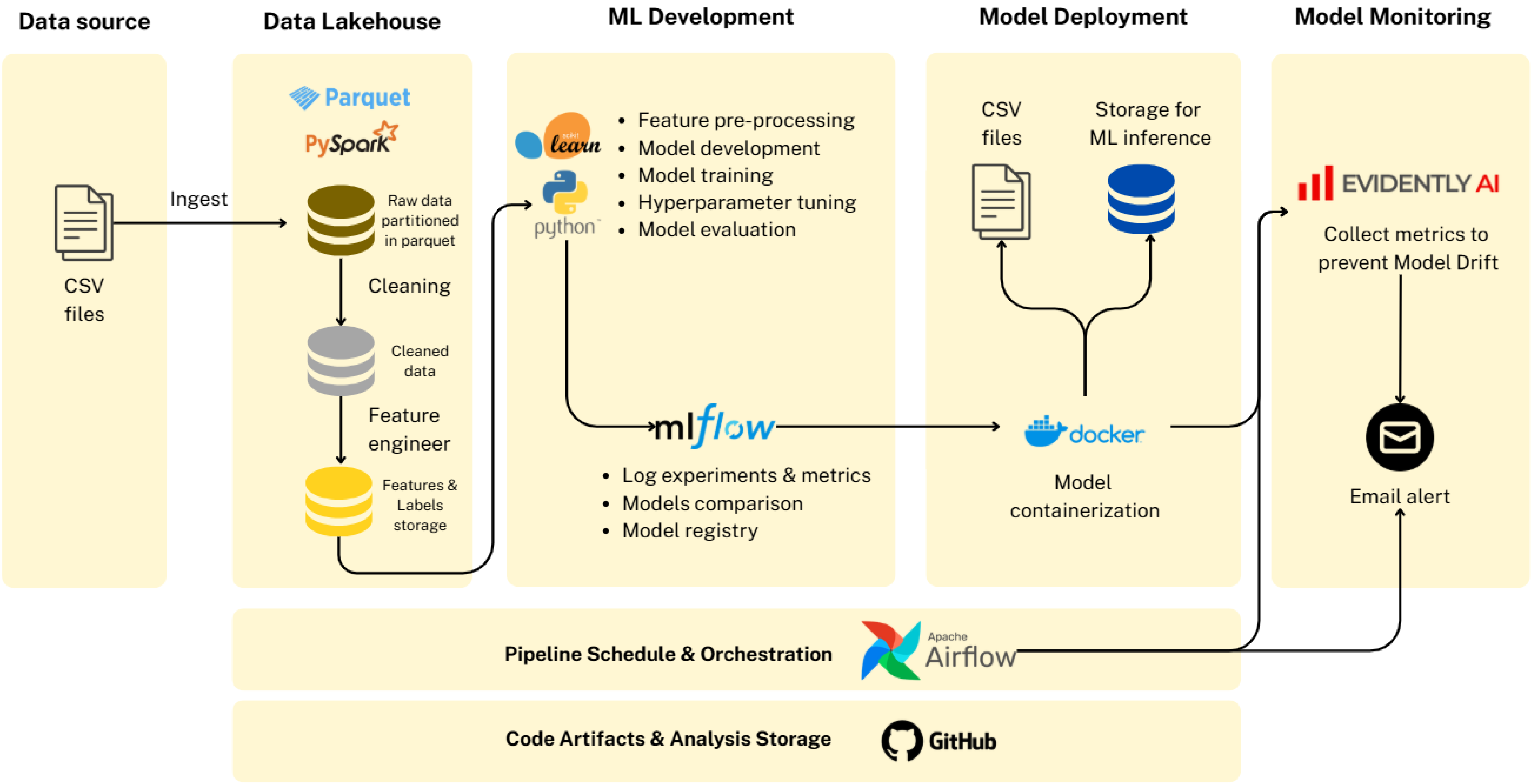
Version control for code, models, and configs; systematic comparison of models and hyperparameters

### 🚵 Monitoring & Retraining

Detect drift and performance degradation; auto-retrain when thresholds exceeded using 30-day feedback

### 💵 Cost-Efficient Reliability

Reproducible deployments with open-source tools; alerting for critical issues


Data source
Data Lakehouse
ML Development
Model Deployment
Model Monitoring
CSV files
Ingest
Parquet
PySpark
Raw data partitioned in parquet
Cleaning
Cleaned data
Feature engineer
Features & Labels storage
learn
python
Feature pre-processing
Model development
Model training
Hyperparameter tuning
Model evaluation
mlflow
Log experiments & metrics
Models comparison
Model registry
CSV files
Storage for ML inference
docker
Model containerization
EVIDENTLY AI
Collect metrics to prevent Model Drift
Email alert
Pipeline Schedule & Orchestration
Apache Airflow
Code Artifacts & Analysis Storage
GitHub

# Thank you!

All questions and comments are welcome

**Group 1**