BIG DATA ANALYTICS

REPORT TITLE
CUSTOMER LIFETIME VALUE PREDICTION

By Lyaba Farooq, Muskan Asad, and Yusra Khan

Students of Data Science

This report outlines the design of a complete Big Data and Machine Learning pipeline for predicting Customer Lifetime Value (CLV) in the Retail industry.

PHASE 1: PROBLEM IDENTIFICATION

SELECTED USE CASE

We have selected Customer Lifetime Value (CLV) from the retail analytics domain. This use case is critical because it allows the business to move beyond treating all customers equally, instead focusing efforts on those identified as the most valuable.

- **Business Goal:** The aim is to use data analysis to identify high-value customers by predicting their future spending patterns.
- **Strategic Action:** This prediction enables targeted marketing offers, prioritized sales efforts, and proactive customer service to improve retention for key customers.

PROJECT SCOPE

The scope of this project is to design a scalable pipeline that converts raw transaction history into a predicted numerical CLV score for each customer.

- The project uses past purchasing behavior to estimate the future value of a customer.

- The final output will allow the business to identify customers likely to bring the most future revenue, thus optimizing resource allocation for marketing and loyalty programs.

PHASE 2: DATA SOURCING

DATASET SELECTION

The chosen dataset is **Online Retail II**, a publicly available, real-world transactional dataset well-suited for a big data application.

- **Source and Link:** The dataset is sourced from the **UCI Machine Learning** **https://www.kaggle.com/datasets/mashlyn/online-retail-ii-uci**
- **Repository** and is available on Kaggle.
- **Data Description:** It contains transaction records from a UK-based online retail store over almost two years.
- **Metadata:** The dataset contains 1,067,371 transaction records with 8 features, making it ideal for calculating **RFM metrics** (Recency, Frequency, Monetary Value) and setting up a past-versus-future spending split for CLV calculation.

PHASE 3: PIPELINE DESIGN

We are designing a robust, scalable pipeline suitable for large, transactional data that facilitates subsequent machine learning tasks.

1. DATA INGESTION

We will employ a **batch-processing approach** using **Apache Spark** due to the static nature of the initial large CSV file.

- The raw CSV file will be loaded from **HDFS** or **AWS S3** using **PySpark** to leverage distributed reading.

- Initial validation checks and schema enforcement will be performed during this ingestion step to ensure data quality.

2. STORAGE LAYER

Storage is segmented into raw and processed zones for governance and efficiency.

- **Raw Storage:** The original, untouched CSV file will reside in a **Data Lake (HDFS or AWS S3)**, ensuring the ability to audit and reprocess the source data.
- **Processed Storage:** The cleaned, aggregated customer-level data will be stored in the **Parquet columnar format** for optimized reading speed and compression during the modeling phase.

3. PROCESSING (CLEANING + TRANSFORMATION)

The processing layer is the critical stage that creates the necessary customer features using **PySpark** for distributed **Extract, Transform, Load (ETL)** operations.

- Data cleaning involves removing cancelled invoices, filtering out missing CustomerID records, and fixing negative quantity values.
- The raw transactional data is aggregated by CustomerID to create crucial features, including recency, frequency, monetary value, average basket size, andtotal items purchased.

4. ANALYTICS & MODELING

The processed, feature-engineered data is used to train and evaluate the regression models.

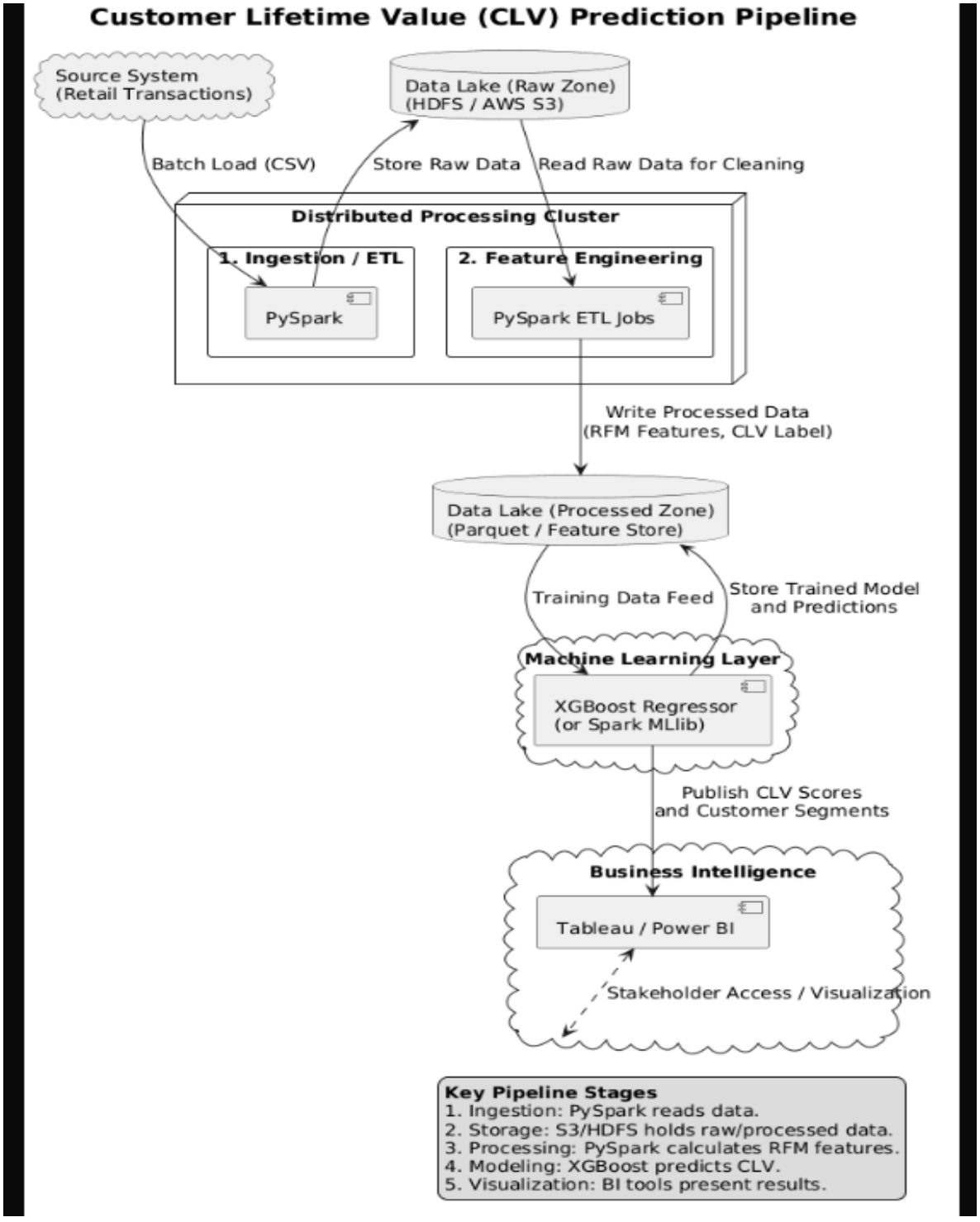
- The data is split into training and testing sets to prepare for the regression-based ML models.
- We will primarily use **XGBoost Regressor**, while **linear regression** will serve as a performance baseline.

5. VISUALIZATION LAYER

The final CLV scores and derived insights must be made actionable for business stakeholders.

- **Reporting:** Dashboards will be created using **Tableau** or **Power BI** to present CLV predictions, customer segments (high/medium/low), and revenue distributions.
- **Exploratory Analysis: Matplotlib** and **Seaborn** will be used for internal feature importance and model diagnostic visualizations.

ARCHITECTURAL DESIGN



PHASE 4: MACHINE LEARNING METHODOLOGY

PRE-PROCESSING STRATEGY

1. **Handling Missing Values:** Rows with missing **CustomerID** are removed because CLV calculation requires customer identification. Other invalid records, such as those with negative or zero quantity or unit price, are also removed to avoid distorted revenue calculations.
2. **Handling Categorical Variables:** The Country variable will be processed using **one-hot encoding** because it is a nominal (non-ordered) variable, which avoids imposing false ordering and works effectively with tree-based models.
3. **Feature Engineering:** The creation of **RFM features** (Recency, Frequency, Monetary) is the fundamental step for CLV, supplemented by features like Average Order Value and Inter-purchase Time.
4. **Scaling Strategy:** We will apply **standardization (Z-score scaling)** to all numerical features. This is crucial for improving training stability and performance, particularly for gradient boosting and any linear comparison models.

ALGORITHM RECOMMENDATION

- **Recommended Algorithm: XGBoost Regressor (Gradient Boosted Decision Tree)**.
- **Justification:**
    - **Problem Type:** CLV is a **regression** task that predicts a continuous future spending value.
    - **Data Suitability:** XGBoost excels on **structured/tabular data** common in retail analytics.
    - **Performance:** It captures **complex, non-linear patterns** in customer behavior (like seasonality and varying habits) better than simpler linear models.
    - **Robustness:** It is highly **robust to outliers and skewed distributions**, which are common in CLV, where a few customers contribute disproportionately to total revenue.

- **Scalability:** XGBoost is optimized for speed and memory, making it efficient for the large dataset volume.

DATASET ANALYSIS

The **Online Retail II** dataset exhibits key characteristics relevant to the ML approach.

- **Type:** It is **structured, multivariate, and time-series** in nature, requiring time-based feature calculation and modeling.
- **Imbalance:** The **target variable (CLV) is highly imbalanced and right-skewed**. This means a small percentage of customers generate the bulk of the revenue, which XGBoost handles effectively.
- **Dimensionality:** The raw data is low-dimensional, but it becomes **medium-dimensional** (15–25 features) after necessary feature engineering and encoding.
- **Volume:** With over 1 million rows, the dataset qualifies as large transactional data suitable for distributed processing with PySpark.

PHASE 5: IMPLEMENTATION PLAN

LIBRARY SELECTION

The implementation relies on standard Python libraries optimized for Big Data and Machine Learning.

- **Data Processing:** pyspark (for large-scale ETL and aggregation) and pandas (for smaller-scale manipulation).
- **Modeling:** xgboost (the main regressor) and scikit-learn (for scaling and splitting).
- **Visualization:** matplotlib and seaborn (for plotting feature importance and data distributions).

PSEUDO-CODE / LOGIC FLOW

The implementation follows a logical, step-by-step flow from data loading to model evaluation.

1. **LOAD** raw transaction data into a **PySpark DataFrame**.
2. **CLEAN data**: Remove canceled invoices and rows with missing CustomerID or invalid quantity/price values.
3. **CREATE customer-level features:** Group by CustomerID and compute **RFM features** and additional behavioral metrics.
4. **DEFINE TARGET (CLV):** Split the dataset into 'feature period' and 'future period' by date, and calculate the total spend in the future period as the CLV label.
5. **PRE-PROCESS:** Apply **One-Hot Encoding** to Country and apply **StandardScaler** to all numerical features.
6. **SPLIT** the feature set into **training (80%)** and **testing (20%)** sets.
7. **TRAIN the main model:** Fit the **XGBoost Regressor** on the training set.
8. **EVALUATE model:** Predict CLV on the test set and calculate evaluation metrics.
9. **INTERPRET results:** Analyze feature importance from XGBoost.
10. **DEPLOY:** Save the final trained model for production use.

EVALUATION METRICS

Since CLV is a **regression problem**, metrics that measure the error magnitude in currency terms are required.

- **Primary Metric: Root Mean Squared Error (RMSE)**.
    - **Justification:** RMSE is the most critical metric because it heavily **penalizes large prediction errors**. In CLV, underestimating or overestimating a high-value customer leads to costly business mistakes, making the minimization of larger errors a priority
- **Secondary Metrics: Mean Absolute Error (MAE) and R-squared (R^2)**
- MAE provides an easily interpretable average error in currency units, while R^2 explains the percentage of variance in customer value that the model accounts for.

CONCLUSION

This proposal details a comprehensive, scalable **Customer Lifetime Value prediction solution** utilizing a PySpark-based pipeline and the high-performance **XGBoost Regressor**. The focus on robust feature engineering (RFM) and the critical evaluation metric of **RMSE** ensures that the resulting model is highly accurate and directly supports strategic business decisions regarding customer retention and resource allocation.