Customer Campaign Response Analytics Using PySpark

Problem Statement:

A retail bank runs marketing campaigns to promote term deposits. You are a Data Engineer tasked with building a pipeline to ingest, clean, and analyse campaign data using PySpark. The goal is to extract insights about customer behaviour and prepare data for Machine Learning-based customer targeting strategies.

Dataset Overview:

- **Dataset**: bank-full.csv
- **Source**: Kaggle - Bank Marketing Dataset
- **Context**: Data collected from marketing campaigns of a Portuguese bank.
- **Size**: ~45,000 records

Features:

Column Name	Description
age	Age of the client
job	Job type (admin., technician, etc.)
marital	Marital status
education	Client education level
default	Has credit in default?
balance	Bank balance
housing	Has housing loan?
loan	Has personal loan?
contact	Contact communication type
day	Last contact day of the month
month	Last contact month
duration	Contact duration (seconds)
campaign	Number of contacts during this campaign

Column Name	Description
pdays	Days since last contact (-1 means never contacted)
previous	Number of contacts before this campaign
poutcome	Outcome of the previous campaign
y	Response to the current campaign (yes/no)

Project Objectives:

1. **Ingest** the data from CSV to Spark DataFrame.
2. **Clean** and **preprocess** data (nulls, data types, filtering).
3. **Transform** data using PySpark SQL and functions (groupBy, joins, etc.).
4. **Analyze** customer traits influencing deposit subscription.
5. **Store** final curated dataset in Delta format (Databricks) or Parquet (Colab).
6. **Bonus**: Generate SQL queries on temporary views.

Real-World Scenario:

Imagine you're part of a Data Engineering team in a bank. You need to:

- Clean raw customer campaign data.
- Make it queryable for Data Analysts.
- Help Data Scientists by supplying clean training data.
- Support Marketing by identifying patterns of campaign success.

Tasks Breakdown:

Step 1: Setup

- Databricks: Use built-in cluster, upload dataset.
- Colab: Install & configure PySpark.

Step 2: Load & Inspect

Step 3: Data Cleaning

- Drop duplicates
- Cast columns to appropriate types
- Handle unknown values

Step 4: Exploratory Analysis (EDA)

- Which job type has the most subscriptions?
- Does age or balance correlate with subscription?

Step 5: Data Transformation

- Create age group buckets
- Encode categorical features (label/one-hot)
- Join with reference tables (if available)

Step 6: Save Processed Data

Outcome:

- A clean, transformed dataset ready for analysis or ML modeling.
- Skills gained: data ingestion, Spark SQL, transformation, saving data in efficient formats.
- Real-world exposure to working with campaign datasets in banking.

What you need to share (In ZIP):

- Colab-ready scripts/ A Databricks notebook (.dbc or .ipynb)
- Presentation
- Cleaned Data (If saved)