from pyspark.sql import SparkSession

from pyspark.sql.functions import col, monotonically\_increasing\_id, udf, lit

from pyspark.sql.types import StringType

import random

import string

spark = SparkSession.builder.appName("DataModification\_DifferentAndSimilar").getOrCreate()

csv\_filename = "/content/Generated\_Data.csv"

df = spark.read.option("header", "true").csv(csv\_filename)

def introduce\_typo(word):

    if not word or len(word) < 2:

        return word

    word = list(word)

    idx = random.randint(0, len(word) - 2)

    word[idx], word[idx + 1] = word[idx + 1], word[idx]

    return "".join(word)

def generate\_different(word):

    return ''.join(random.choices(string.ascii\_letters + string.digits, k=len(word)))

introduce\_typo\_udf = udf(introduce\_typo, StringType())

generate\_different\_udf = udf(generate\_different, StringType())

df = df.withColumn("row\_id", monotonically\_increasing\_id())

mod\_df = df.filter(col("Category").isin(["Alpha", "Alphanumeric"])).orderBy("row\_id")

total\_rows = mod\_df.count()

num\_per\_category = total\_rows // 3  # Divide into 3 equal parts

different\_df = mod\_df.limit(num\_per\_category)

different\_df = different\_df.withColumn("Modified\_Value", generate\_different\_udf(col("Value")))

different\_df = different\_df.withColumn("Similarity\_Label", lit("Different"))

remaining\_df = mod\_df.subtract(different\_df.select("row\_id", "Value", "Category"))

similar\_df = remaining\_df.limit(num\_per\_category)

similar\_df = similar\_df.withColumn("Modified\_Value", introduce\_typo\_udf(col("Value")))

similar\_df = similar\_df.withColumn("Similarity\_Label", lit("Similar"))

same\_df = remaining\_df.subtract(similar\_df.select("row\_id", "Value", "Category"))

same\_df = same\_df.withColumn("Modified\_Value", col("Value"))

same\_df = same\_df.withColumn("Similarity\_Label", lit("Same"))

final\_mod\_df = different\_df.union(similar\_df).union(same\_df)

final\_df = df.join(final\_mod\_df.select("row\_id", "Modified\_Value", "Similarity\_Label"), on="row\_id", how="left").drop("row\_id")

modified\_csv\_filename = "/content/processed\_data\_different\_similarys.csv"

final\_df.write.option("header", "true").csv(modified\_csv\_filename)

final\_df.select("Value", "Modified\_Value", "Category", "Similarity\_Label").show(truncate=False)