# Java kurulumu

!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Spark ve Hadoop'u indir ve kurulumu yap

!wget -q https://archive.apache.org/dist/spark/spark-3.4.0/spark-3.4.0-bin-hadoop3.tgz

!tar xf spark-3.4.0-bin-hadoop3.tgz

# Ortam değişkenlerini ayarla

import os

os.environ["JAVA\_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

os.environ["SPARK\_HOME"] = "/content/spark-3.4.0-bin-hadoop3"

# findspark kurulumu

!pip install -q findspark

# Spark'ı başlat

import findspark

findspark.init()

# SparkSession oluştur

from pyspark.sql import SparkSession

spark = SparkSession.builder.master("local[\*]").appName("Colab Spark").getOrCreate()

# Spark'ın çalıştığını kontrol etmek için örnek kod

spark.range(5).show()

çalışıyor

Hadoop ve mapreduce colabta çalıştı

# Gerekli paketlerin kurulumu

!apt-get install -y openjdk-11-jdk

!pip install -q findspark

!pip install -q mrjob

# Hadoop'un indirilmesi ve çıkarılması

!wget https://downloads.apache.org/hadoop/common/hadoop-3.3.5/hadoop-3.3.5.tar.gz

!tar -xzvf hadoop-3.3.5.tar.gz

# Hadoop ayarları

import os

os.environ["HADOOP\_HOME"] = "/content/hadoop-3.3.5"

os.environ["JAVA\_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"

os.environ["PATH"] += ":/content/hadoop-3.3.5/bin:/content/hadoop-3.3.5/sbin"

# Giriş verisini oluşturma

input\_data = """text,label

"I love this product!",positive

"This is the worst thing I have ever bought.",negative

"It was okay, not great.",neutral

"Absolutely fantastic! Highly recommend.",positive

"Not what I expected.",negative

"""

with open('sentiment\_data.csv', 'w') as f:

    f.write(input\_data)

# Dosya oluşturma kontrolü

if not os.path.exists('sentiment\_data.csv'):

    print("Giriş verisi dosyası oluşturulamadı!")

    exit(1)

# MapReduce programı oluşturma

wordcount\_code = """

from mrjob.job import MRJob

class MRSentimentCount(MRJob):

    def mapper(self, \_, line):

        # İlk satırı atla

        if line.startswith("text"):

            return

        # Satırı virgüle göre ayır

        parts = line.split(',')

        # Eğer iki parçadan fazlası varsa ya da boş bir satırsa atla

        if len(parts) != 2:

            return

        \_, label = parts

        yield label, 1

    def reducer(self, label, counts):

        yield label, sum(counts)

if \_\_name\_\_ == '\_\_main\_\_':

    MRSentimentCount.run()

"""

with open('sentiment\_count.py', 'w') as f:

    f.write(wordcount\_code)

# MapReduce işini çalıştırma

try:

    !python3 sentiment\_count.py sentiment\_data.csv > output.txt

except Exception as e:

    print("MapReduce işi çalıştırılamadı!", e)

# Çıktıyı kontrol etme

try:

    !cat output.txt

except Exception as e:

    print("Çıktı dosyası okunamadı!", e)