Laboratory Manual-13

for

Fundamentals of Big Data Lab

Course Instructor: Dr Iqra Safdar
Lab Instructors: Mr. Muhammad Mazarib; Mr Muhammad Aiss Shahid
Section: BDS-4B
Date: 08-May-2023
Semester: Spring 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Introduction to DataFrames in PySpark

There are limitations of RDDs. So, dataframes overcome that limitations of Rdds. In this manual you go through how to use data frames in pyspark.
For detailed information, go through the documentation of Pyspark.

How to create DataFrames?

There are multiple ways to create DataFrames in Apache Spark:

- DataFrames can be created using an existing RDD
- You can create a DataFrame by loading a CSV file directly
- You can programmatically specify a schema to create a DataFrame

```
# spark is an existing SparkSession
df = spark.read.csv("examples/src/main/resources/people.csv")
# Displays the content of the DataFrame to stdout
df.show()
# +----+-------+
# | age|   name|
# +----+-------+
# |null|Michael|
# |  30|   Andy|
# |  19| Justin|
# +----+-------+
```

Some basic operations of dataframes

In Python, it's possible to access a DataFrame's columns either by attribute (df.age) or by indexing (df['age']). While the former is convenient for interactive data exploration, users are highly encouraged to use the latter form, which is future proof and won't break with column names that are also attributes on the DataFrame class.

```
# spark, df are from the previous example
# Print the schema in a tree format
df.printSchema()
# root
# |-- age: long (nullable = true)
# |-- name: string (nullable = true)

# Select only the "name" column
df.select("name").show()
# +-------+
# |   name|
# +-------+
# |Michael|
# |   Andy|
# | Justin|
# +-------+

# Select everybody, but increment the age by 1
```

```
df.select(df['name'], df['age'] + 1).show()
# +-------+---------+
# |   name|(age + 1)|
# +-------+---------+
# |Michael|     null|
# |   Andy|       31|
# | Justin|       20|
# +-------+---------+

# Select people older than 21
df.filter(df['age'] > 21).show()
# +---+----+
# |age|name|
# +---+----+
# | 30|Andy|
# +---+----+

# Count people by age
df.groupBy("age").count().show()
# +----+-----+
# | age|count|
# +----+-----+
# |  19|    1|
# |null|    1|
# |  30|    1|
# +----+-----+
```

Running SQL Queries Programmatically

```
# Register the DataFrame as a SQL temporary view
df.createOrReplaceTempView("people")

sqlDF = spark.sql("SELECT * FROM people")
sqlDF.show()
# +----+-------+
# | age|   name|
# +----+-------+
# |null|Michael|
# |  30|   Andy|
# |  19| Justin|
# +----+-------+
```

How to read CSV file and its other operations follow the link: https://spark.apache.org/docs/latest/sql-data-sources-csv.html

Machine Learning Library (MLlib) Guide

MLlib is Spark's machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. At a high level, it provides tools such as:

- ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
- Persistence: saving and load algorithms, models, and Pipelines
- Utilities: linear algebra, statistics, data handling, etc.

Link for all the machine learning approaches in Pyspark

https://spark.apache.org/docs/latest/ml-guide.html

The given below is the example of Machine Learning algorithm in Pyspark

- Finding correlation

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.stat import Correlation

data = [(Vectors.sparse(4, [(0, 1.0), (3, -2.0)]),),
        (Vectors.dense([4.0, 5.0, 0.0, 3.0]),),
        (Vectors.dense([6.0, 7.0, 0.0, 8.0]),),
        (Vectors.sparse(4, [(0, 9.0), (3, 1.0)]),)]
df = spark.createDataFrame(data, ["features"])

r1 = Correlation.corr(df, "features").head()


print("Pearson correlation matrix:\n" + str(r1[0]))

r2 = Correlation.corr(df, "features", "spearman").head()


print("Spearman correlation matrix:\n" + str(r2[0]))
```

It is list of all python apis: https://spark.apache.org/docs/latest/api/python/reference/index.html

This Link gives you insights how machine learning algorithm has been implemented using machine learning libraries from pyspark.

Lab Tasks

You are provided dataset “Movies.csv” that contains information about different aspects of movies explore the dataset and do lab task 1 to 5 using pyspark data frames.

1. Find the title, year, and director of action films that won an award.
2. For each award-winning actor, find the movies he acted it. Print the names of the movies and the director of the movie.
3. Find the top 10 most popular movies that did not win an award.
4. Find the 10 least popular movies that were released before 1980.
5. Sort the movie’s release before 1990 by the title.
6. Explore and preprocess the “wine” data set which was given you in previous lab, use spark dataframes and analyse it. Find the outliers or noise in the data and find the correlation between different features.
7. Use PySpark built-in K-means and bisecting K-means clustering algorithms for clustering.