PDP Assignment 3 – (Py)Spark

Name: Sjors Grooff

Student number: 634293

Email: 634293@student.inholland.nl

Github url: https://github.com/groofy98/PDP/blob/main/Assignment%203/pdp_assignment_3.py

Date: 9-9-2021

The assignment

I managed to do 3a and 3c as I didn't understand what was asked in 3b. I combined the exercises in one script because they use the same data.

The script for 3A and 3C

```
/ # @Author: Sjors Grooff

# Date: 09-09-2021
# Command: spark-submit pdp_assignment_3.py

# Import necessary libraries
from pyspark.sql import SparkSession
from pyspark.sql.types import FloatType, IntegerType
import pyspark.sql.functions as func

if __name__ == "__main__":
    # Create a SparkSession (the config bit is only for Windows!)
    spark = SparkSession.builder.appName("PDP_Assignment_3").getOrCreate()
    # Remove most of the logging
    spark.sparkContext.setLogLevel('WARN')

    # Headers: Survived,Pclass,Name,Sex,Age,Siblings/Spouses Aboard,Parents/Ch
ildren Aboard,Fare
    # Load the raw data and use the headers from the csv file
    df = spark.read.option("header", "true").csv(
        "hdfs:///user/maria_dev/titanic.csv")

    df = df.withColumn('fare', df['fare'].cast(FloatType()))
    df = df.withColumn('Survived', df['Survived'].cast(IntegerType()))

    # Assignment 3a. Calculate survivability per sex and class
    df.groupBy(['Sex', 'Pclass']) \
        .agg(func.sum("Survived").alias('Survived'), func.count("Survived").al
ias('Total')) \
```

```
        .withColumn('%', func.round((func.col('Survived')/func.col('Total'))*1
00,2)) \
        .sort(['Sex', 'Pclass']) \
        .show()

    # Assignment 3c.
    fareExpectation = df.groupBy('pClass').avg('fare').orderBy('pClass', ascen
ding = True)
    fareExpectation.show()

    # Stop the session
    spark.stop()
```

How it works

We first of start by importing the necessary libraries. We don't have a class and only one main method that is executed. It isn't pretty but it works. In the method we first create a Sparksession and set the logging level to warn to get rid most of the info messages. Then we read the csv into a dataframe using the headers. After that we convert the 'fare' and 'survived' columns to float and integers so they can be used for calculations.

We select the data for 3a as follows:

1. We group the data by Sex(gender) and PClass (Class)
2. Use an aggregate function to calculate the survialchance column
    a. This is done by counting the survivors and total passengers by class and gender and then calculating the percentage
3. Then we show the resulting dataframe

The data for 3c was somewhat easier:

1. First we only group the dataframe by class
2. Then we calculate the average of the fare column
3. Lastly we sort by class ascending
4. And ofcourse we show the data

Then we stop the Spark session

The result of 3a

The survival chance by gender and class:

Sex	Pclass	Survived	Total	%
female	1	91	94	96.81
female	2	70	76	92.11
female	3	72	144	50.0
male	1	45	122	36.89
male	2	17	108	15.74
male	3	47	343	13.7

The result of 3c

The average fare by class:

pClass	avg(fare)
1	84.15468752825701
2	20.66218318109927
3	13.707707501045244

How to run it

The machine needs to have Spark, Python and PySpark installed. The titanic.csv file must be added to the hdfs storage. Then the script can be run with the following command: spark-submit pdp_assignment_3.py