PARALLEL DISTRIBUTED PROCESSING (HADOOP)

Assignment 3

Abdullah Ghayumi
634072@student.inholland.nl

Assignment 3

Details

Name: Abdullah Ghayumi

Student number: 634072

GitHub URL: https://github.com/Aras53/Hadoop

Table of Contents

Details 1

Problem a 2

Solution a 3

Juyper Notebook Code 3

Result 3

Problem b 4

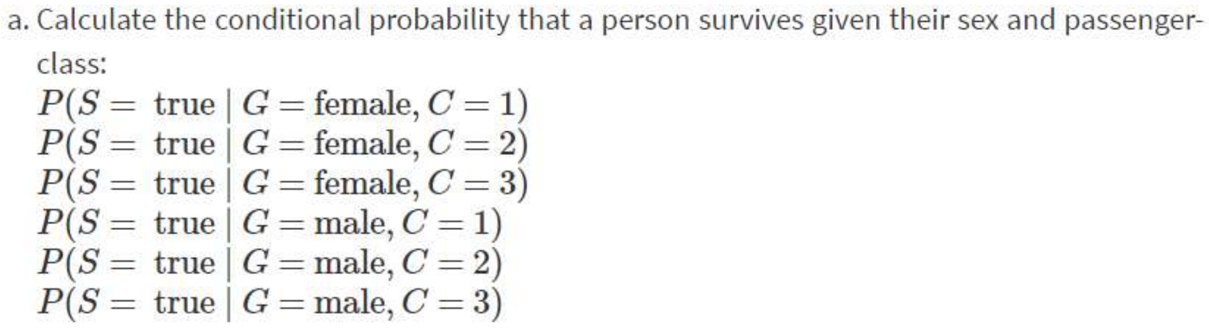
Solution b 4

Problem c 5

Solution c 5

Result 5

Problem a

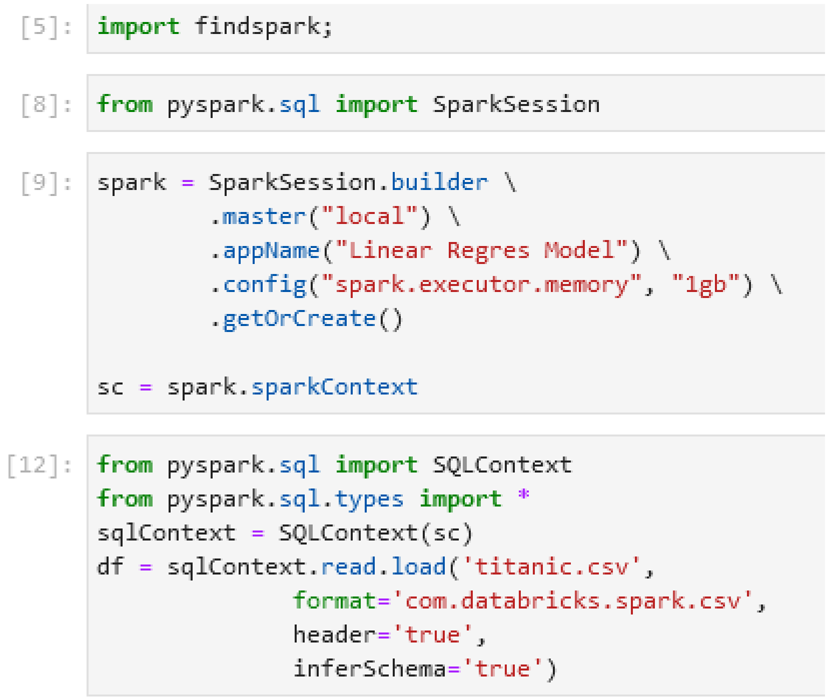


a. Calculate the conditional probability that a person survives given their sex and passenger-class:

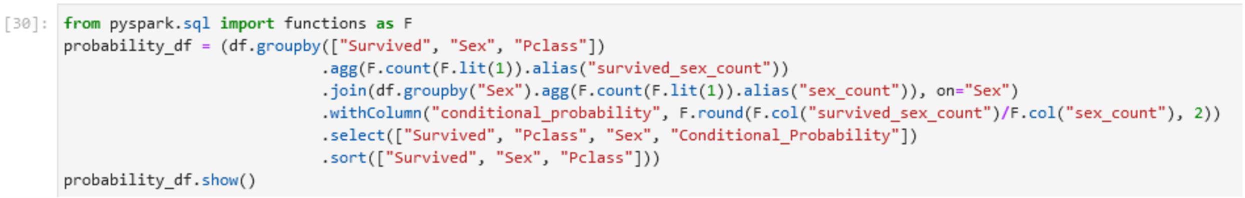
$P(S = \text{ true} \mid G = \text{female}, C = 1)$
$P(S = \text{ true} \mid G = \text{female}, C = 2)$
$P(S = \text{ true} \mid G = \text{female}, C = 3)$
$P(S = \text{ true} \mid G = \text{male}, C = 1)$
$P(S = \text{ true} \mid G = \text{male}, C = 2)$
$P(S = \text{ true} \mid G = \text{male}, C = 3)$

Solution a

Juyper Notebook Code

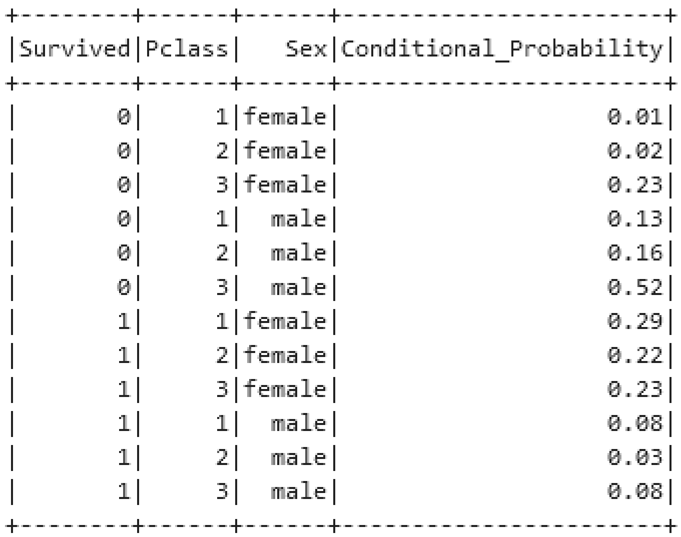


```
[5]:  import findspark;

[8]:  from pyspark.sql import SparkSession

[9]:  spark = SparkSession.builder \
              .master("local") \
              .appName("Linear Regres Model") \
              .config("spark.executor.memory", "1gb") \
              .getOrCreate()

      sc = spark.sparkContext

[12]: from pyspark.sql import SQLContext
      from pyspark.sql.types import *
      sqlContext = SQLContext(sc)
      df = sqlContext.read.load('titanic.csv',
                    format='com.databricks.spark.csv',
                    header='true',
                    inferSchema='true')
```



```
[30]: from pyspark.sql import functions as F
      probability_df = (df.groupby(["Survived", "Sex", "Pclass"])
                                      .agg(F.count(F.lit(1)).alias("survived_sex_count"))
                                      .join(df.groupby("Sex").agg(F.count(F.lit(1)).alias("sex_count")), on="Sex")
                                      .withColumn("conditional_probability", F.round(F.col("survived_sex_count")/F.col("sex_count"), 2))
                                      .select(["Survived", "Pclass", "Sex", "Conditional_Probability"])
                                      .sort(["Survived", "Sex", "Pclass"]))
      probability_df.show()
```

Result



```
+--------+------+------+-----------------------+
|Survived|Pclass|   Sex|Conditional_Probability|
+--------+------+------+-----------------------+
|       0|     1|female|                   0.01|
|       0|     2|female|                   0.02|
|       0|     3|female|                   0.23|
|       0|     1|  male|                   0.13|
|       0|     2|  male|                   0.16|
|       0|     3|  male|                   0.52|
|       1|     1|female|                   0.29|
|       1|     2|female|                   0.22|
|       1|     3|female|                   0.23|
|       1|     1|  male|                   0.08|
|       1|     2|  male|                   0.03|
|       1|     3|  male|                   0.08|
+--------+------+------+-----------------------+
```

Problem b

b. What is the probability that a child who is in third class and is 10 years old or younger survives? Since the number of data points that satisfy the condition is small use the "bayesian" approach and represent your probability as a beta distribution. Calculate a belief distribution for:

$S = \text{ true} \mid A \leq 10, C = 3$

You can express your answer as a parameterized distribution.

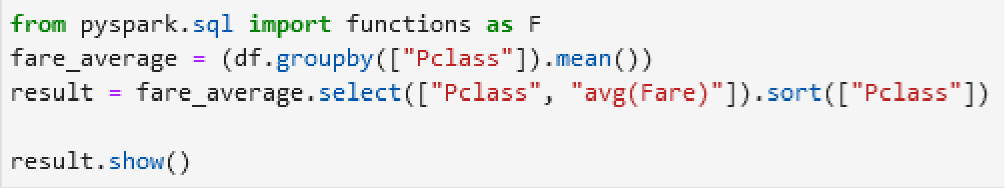
Solution b

Problem c

c. How much did people pay to be on the ship? Calculate the expectation of fare conditioned on passenger-class:

$$E[X \mid C = 1]$$
$$E[X \mid C = 2]$$
$$E[X \mid C = 3]$$

Solution c



```
from pyspark.sql import functions as F
fare_average = (df.groupby(["Pclass"]).mean())
result = fare_average.select(["Pclass", "avg(Fare)"]).sort(["Pclass"])

result.show()
```

Result

```
+------+------------------+
|Pclass|         avg(Fare)|
+------+------------------+
|     1| 84.15468749999992|
|     2| 20.66218315217391|
|     3|13.707707392197129|
+------+------------------+
```