Use Spark To Preprocess Data In Ganymede HPC And IntelliJ IDEA Respectively

Runtao Xue - UT Dallas
rxx180001@utdallas.edu

About the Code

The theme of this project is to use Spark to perform data preprocessing in two different development environments, which are the Ganymede HPC development environment and the localized IntelliJ IDEA development environment. For the Ganymede HPC command-line development environment, the main components of the code are entering Spark-Shell, importing data, observing data structures, clearing null values, correcting data types, and exporting to json data files. For the IntelliJ IDEA compiler development environment, the main components of the code are: first build the required scala development environment, then update the build.sbt file to obtain the relevant spark toolkit, then build a local Spark Session, and finally perform similar data Pre-processing operation and export json data file. In addition, the data used in this project is the survey data of airline passenger satisfaction, the data source has been marked in the final reference.

Command-Line

```
// --STARTING SPARK SHELL--
spark-shell --conf spark.ui.port = 4057
```

```
[rxx180001@x2go ~]$ spark-shell --conf spark.ui.port=4057
20/04/21 17:05:30 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://x2go.localdomain:4057
Spark context available as 'sc' (master = local[*], app id = local-1587506737993).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.4
      /_/

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_222)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

```
// --SETTING DATA PATH--
val data_path = "./scratch/train.csv"
// --LOADING DATA--
val df_raw = spark.read.format("csv").option("header","true").load(data_path)
```

```
scala> val df_raw = spark.read.format("csv").option("header","true").load("./scratch/train.csv")
df_raw: org.apache.spark.sql.DataFrame = [_c0: string, id: string ... 23 more fields]
```

```
println("Number of Observation:"+df_raw.count())
```

```
scala> println("Number of Observation:"+df_raw.count())
Number of Observation:103904
```

```
df_raw.show(10)
```

```
scala> df_raw.show(10)
20/04/21 17:26:03 WARN CSVDataSource: CSV header does not conform to the schema.
 Header: , id, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Inflight wifi service, Departure/Arrival time convenient, Ease of Online booking, Gate location, Food and drink, Online boarding, Seat comfort, Inflight entertainment, On-board service, Leg room service, Baggage handling, Checkin service, Inflight service, Cleanliness, Departure Delay in Minutes, Arrival Delay in Minutes, satisfaction
 Schema: _c0, id, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Inflight wifi service, Departure/Arrival time convenient, Ease of Online booking, Gate location, Food and drink, Online boarding, Seat comfort, Inflight entertainment, On-board service, Leg room service, Baggage handling, Checkin service, Inflight service, Cleanliness, Departure Delay in Minutes, Arrival Delay in Minutes, satisfaction
Expected: _c0 but found:
CSV file: file:///home/rxx180001/scratch/train.csv
+---+------+------+-----------------+---+---------------+--------+---------------+---------+--------------------+
|_c0|    id|Gender|    Customer Type|Age| Type of Travel|   Class|Flight Distance| ... ... |        satisfaction|
+---+------+------+-----------------+---+---------------+--------+---------------+---------+--------------------+
|  0| 70172|  Male|   Loyal Customer| 13|Personal Travel|Eco Plus|            460|         |neutral or dissat...|
|  1|  5047|  Male|disloyal Customer| 25|Business travel|Business|            235|         |neutral or dissat...|
|  2|110028|Female|   Loyal Customer| 26|Business travel|Business|           1142| ... ... |           satisfied|
|  3| 24026|Female|   Loyal Customer| 25|Business travel|Business|            562| ... ... |neutral or dissat...|
|  4|119299|  Male|   Loyal Customer| 61|Business travel|Business|            214|         |           satisfied|
|  5|111157|Female|   Loyal Customer| 26|Personal Travel|     Eco|           1180|         |neutral or dissat...|
|  6| 82113|  Male|   Loyal Customer| 47|Personal Travel|     Eco|           1276| ... ... |neutral or dissat...|
|  7| 96462|Female|   Loyal Customer| 52|Business travel|Business|           2035| ... ... |           satisfied|
|  8| 79485|Female|   Loyal Customer| 41|Business travel|Business|            853|         |neutral or dissat...|
|  9| 65725|  Male|disloyal Customer| 20|Business travel|     Eco|           1061|         |neutral or dissat...|
+---+------+------+-----------------+---+---------------+--------+---------------+---------+--------------------+
only showing top 10 rows
```

```
// --DEAL WITH NULL VALUES--
df_raw.describe().show()
```

```
scala>df_raw.describe().show()
20/04/21 17:38:47 WARN CSVDataSource: CSV header does not conform to the schema.
 Header: , id, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Inflight wifi service, ... ... Baggage handling, Checkin service, Inflight service, Cleanliness, Departure Delay in Minutes, Arrival Delay in Minutes, satisfaction
 Schema: _c0, id, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Inflight wifi service, ... ...Baggage handling, Checkin service, Inflight service, Cleanliness, Departure Delay in Minutes, Arrival Delay in Minutes, satisfaction
Expected: _c0 but found:
CSV file: file:///home/rxx180001/scratch/train.csv
+-------+-----------------+-----------------+------+---------------+---------+----------------------+--------------------+
|summary|              _c0|               id|Gender|    Customer Type| ... ... |Arrival Delay in Minutes|        satisfaction|
+-------+-----------------+-----------------+------+---------------+---------+----------------------+--------------------+
|  count|           103904|           103904|103904|          103904|         |                103594|              103904|
|   mean|         51951.5|64924.21050200185|  null|            null| ... ... |     15.178678301832152|                null|
| stddev|29994.64552215945| 37463.8122515513|  null|            null| ... ... |      38.6986820209665|                null|
|    min|                0|                1|Female|   Loyal Customer|         |                   0.0|neutral or dissat...|
|    max|            99999|            99999|  Male|disloyal Customer|         |                  99.0|           satisfied|
+-------+-----------------+-----------------+------+---------------+---------+----------------------+--------------------+
```

From this one , we know there are some missing values in the data.

```
val df = df_raw.na.drop("any")
```

```
scala> val df = df_raw.na.drop("any")
df: org.apache.spark.sql.DataFrame = [_c0: string, id: string ... 23 more fields]
```

```
df.describe().show()
```

```
scala>df_raw.describe().show()
20/04/21 17:48:23 WARN CSVDataSource: CSV header does not conform to the schema.
 Header: , id, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Inflight wifi service, ... ... Baggage handling, Checkin service, Inflight service, Cleanliness, Departure Delay in Minutes, Arrival Delay in Minutes, satisfaction
 Schema: _c0, id, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Inflight wifi service, ... ...Baggage handling, Checkin service, Inflight service, Cleanliness, Departure Delay in Minutes, Arrival Delay in Minutes, satisfaction
Expected: _c0 but found:
CSV file: file:///home/rxx180001/scratch/train.csv
+-------+-----------------+-----------------+------+---------------+---------+----------------------+--------------------+
|summary|              _c0|               id|Gender|    Customer Type| ... ... |Arrival Delay in Minutes|        satisfaction|
+-------+-----------------+-----------------+------+---------------+---------+----------------------+--------------------+
|  count|           103594|           103594|103594|          103594|         |                103594|              103594|
|   mean|         51951.5|64924.21050200185|  null|            null| ... ... |     15.178678301832152|                null|
| stddev|29994.64552215945| 37463.8122515513|  null|            null| ... ... |      38.6986820209665|                null|
|    min|                0|                1|Female|   Loyal Customer|         |                   0.0|neutral or dissat...|
|    max|            99999|            99999|  Male|disloyal Customer|         |                  99.0|           satisfied|
+-------+-----------------+-----------------+------+---------------+---------+----------------------+--------------------+
```

```
// --DEAL WITH DATA TYPES--
df.printSchema()
```

```
scala> df.printSchema()
root
 |-- _c0: string (nullable = true)
 |-- id: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Customer Type: string (nullable = true)
 |-- Age: string (nullable = true)
 |-- Type of Travel: string (nullable = true)
 |-- Class: string (nullable = true)
 |-- Flight Distance: string (nullable = true)
 |-- Inflight wifi service: string (nullable = true)
 |-- Departure/Arrival time convenient: string (nullable = true)
 |-- Ease of Online booking: string (nullable = true)
 |-- Gate location: string (nullable = true)
 |-- Food and drink: string (nullable = true)
 |-- Online boarding: string (nullable = true)
 |-- Seat comfort: string (nullable = true)
 |-- Inflight entertainment: string (nullable = true)
 |-- On-board service: string (nullable = true)
 |-- Leg room service: string (nullable = true)
 |-- Baggage handling: string (nullable = true)
 |-- Checkin service: string (nullable = true)
 |-- Inflight service: string (nullable = true)
 |-- Cleanliness: string (nullable = true)
 |-- Departure Delay in Minutes: string (nullable = true)
 |-- Arrival Delay in Minutes: string (nullable = true)
 |-- satisfaction: string (nullable = true)
```

By checking the data, we can easily find that these data types are wrong, and we need to correct them.

```
// --CORRECT DATA TYPES--
val df_typeready = df.withColumn("Age",col("Age").cast("Int"))
    .withColumn("Flight Distance",col("Flight Distance").cast("Double"))
    .withColumn("Inflight wifi service",col("Inflight wifi service").cast("Int"))
    .withColumn("Departure/Arrival time convenient",col("Departure/Arrival time convenient").cast("Int"))
    .withColumn("Ease of Online booking",col("Ease of Online booking").cast("Int"))
    .withColumn("Gate location",col("Gate location").cast("Int"))
    .withColumn("Food and drink",col("Food and drink").cast("Int"))
    .withColumn("Online boarding",col("Online boarding").cast("Int"))
    .withColumn("Seat comfort",col("Seat comfort").cast("Int"))
    .withColumn("Inflight entertainment",col("Inflight entertainment").cast("Int"))
    .withColumn("On-board service",col("On-board service").cast("Int"))
    .withColumn("Leg room service",col("Leg room service").cast("Int"))
    .withColumn("Baggage handling",col("Baggage handling").cast("Int"))
    .withColumn("Checkin service",col("Checkin service").cast("Int"))
    .withColumn("Inflight service",col("Inflight service").cast("Int"))
    .withColumn("Cleanliness",col("Cleanliness").cast("Int"))
    .withColumn("Departure Delay in Minutes",col("Departure Delay in Minutes").cast("Double"))
    .withColumn("Arrival Delay in Minutes",col("Arrival Delay in Minutes").cast("Double"))
// --DROPING IRRELEVANT COLUMNS--
val df_ready = df_typeready.drop("_c0").drop("id")
// --WRITE DATA INTO JSON FILE--
df_ready.write.json("./scratch/airline_data_json")
```

```
scala> df_ready.write.json("./scratch/airline_data_json")
[rxx180001@x2go ~]$ cd ./scratch/airline_data_json/
[rxx180001@x2go airline_data_json]$ ls -l
total 55584
-rw-r--r-- 1 rxx180001 jsomBigData 19619890 Apr 21 19:45 part-00000-135cf068-08db-48cd-8ced-ae0b0cc8fa67-c000.json
-rw-r--r-- 1 rxx180001 jsomBigData 19558755 Apr 21 19:45 part-00001-135cf068-08db-48cd-8ced-ae0b0cc8fa67-c000.json
-rw-r--r-- 1 rxx180001 jsomBigData 17723404 Apr 21 19:45 part-00002-135cf068-08db-48cd-8ced-ae0b0cc8fa67-c000.json
-rw-r--r-- 1 rxx180001 jsomBigData        0 Apr 21 19:45 _SUCCESS
```

IntelliJ IDEA

IntelliJ Spark Processing Code Listing

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.{SparkConf, SparkContext}

object SatisfactionAnalysis {
  def main(args:Array[String]): Unit ={
    //Create a SparkConfig object and SparkContext to initialize Spark
    val conf = new SparkConf()
    conf.setMaster("local")
    conf.setAppName("SatisfactionAnalysis")
    val sc = new SparkContext(conf)
    val spark = SparkSession.builder().master("local").config(conf).getOrCreate()

    //Read in data
    val data_path = "D:\\SparkCode\\src\\main\\resource\\train.csv"
    val df_raw = spark.read.format(source = "csv").option("header","true").load(data_path)

    //Data description and cleaning missing values
    df_raw.show(numRows = 10)
    println("Number of Observation:"+df_raw.count())
    df_raw.describe().show()
    val df = df_raw.na.drop(how = "any")
    df.describe().show()

    //Correction of data types and drop irrelevant columns
    df.printSchema()
    ... ...
```

IntelliJ Spark Processing Code Listing

```
    ... ...
    val df_typeready = df.withColumn("Age",df("Age").cast("Int"))
      .withColumn("Flight Distance",df("Flight Distance").cast("Double"))
      .withColumn("Inflight wifi service",df("Inflight wifi service").cast("Int"))
      .withColumn("Departure/Arrival time convenient",df("Departure/Arrival time
convenient").cast("Int"))
      .withColumn("Ease of Online booking",df("Ease of Online booking").cast("Int"))
      .withColumn("Gate location",df("Gate location").cast("Int"))
      .withColumn("Food and drink",df("Food and drink").cast("Int"))
      .withColumn("Online boarding",df("Online boarding").cast("Int"))
      .withColumn("Seat comfort",df("Seat comfort").cast("Int"))
      .withColumn("Inflight entertainment",df("Inflight entertainment").cast("Int"))
      .withColumn("On-board service",df("On-board service").cast("Int"))
      .withColumn("Leg room service",df("Leg room service").cast("Int"))
      .withColumn("Baggage handling",df("Baggage handling").cast("Int"))
      .withColumn("Checkin service",df("Checkin service").cast("Int"))
      .withColumn("Inflight service",df("Inflight service").cast("Int"))
      .withColumn("Cleanliness",df("Cleanliness").cast("Int"))
      .withColumn("Departure Delay in Minutes",df("Departure Delay in Minutes").cast("Double"))
      .withColumn("Arrival Delay in Minutes",df("Arrival Delay in Minutes").cast("Double"))
    df_typeready.printSchema()
    val df_ready = df_typeready.drop("_c0").drop("id")
    df_ready.printSchema()

    //Write code into json
    df_ready.write.json("D:\\SparkCode\\src\\main\\resource\\json")
  }
}
```

Build.sbt Setup

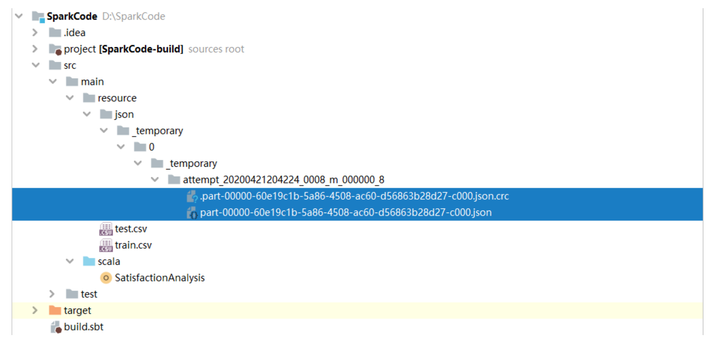
```
name := "SparkCode"

version := "0.1"

scalaVersion := "2.12.11"

// https://mvnrepository.com/artifact/org.apache.spark/spark-core
libraryDependencies += "org.apache.spark" %% "spark-core" % "2.4.3"
// https://mvnrepository.com/artifact/org.apache.spark/spark-sql
libraryDependencies += "org.apache.spark" %% "spark-sql" % "2.4.3"
```

Json File Output



Reference: 1. Airline Passenger Satisfaction | https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction 2. creating first spark project in IntelliJ with SBT | https://www.youtube.com/watch?v=FbwSbsOYqSE
3. Spark SQL, DataFrames and Datasets Guide | https://spark.apache.org/docs/2.3.0/sql-programming-guide.html#starting-point-sparksession 4. *Spark: The Definitive Guide* by Bill Chambers and Matei Zaharia (O'Reilly).