Session 23

Assignment 1

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 23 – Assignment 1 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 23 – Assignment 1** |
|  |  |
| **Start Date:** | 25/10/2017 |
|  |  |
|  |  |



# 

# Contents

[Contents 2](#_Toc496760924)

[Change History 3](#_Toc496760925)

[1. Problem Statement 4](#_Toc496760926)

[2. Datasets 4](#_Toc496760927)

[3. Solutions 4](#_Toc496760928)

[4. Results 6](#_Toc496760929)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 26/10/2017 | Duncan Burgess |  | All | Initial release. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Problem Statement

**Aviation data analysis**

The U.S. Department of Transportation’s (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, canceled, and diverted flights appears in DOT’s monthly Air Travel Consumer Report, published about 30 days after the month’s end, as well as in summary tables posted on this website. Summary statistics and raw data are made available to the public at the time the Air Travel Consumer Report is released.

A sample of the data can be seen below

**Delayed\_Flights.csv Datasets**

There are 29 columns in this dataset. Some of them have been mentioned below:

* Year: 1987 – 2008
* Month: 1 – 12
* FlightNum: Flight number
* Canceled: Was the flight canceled?
* CancelleationCode: The reason for cancellation.

# ****Datasets****

**Delayed\_Flights.csv**

,Year,Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,AirTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay

0,2008,1,3,4,2003.0,1955,2211.0,2225,WN,335,N712SW,128.0,150.0,116.0,-14.0,8.0,IAD,TPA,810,4.0,8.0,0,N,0,,,,,

1,2008,1,3,4,754.0,735,1002.0,1000,WN,3231,N772SW,128.0,145.0,113.0,2.0,19.0,IAD,TPA,810,5.0,10.0,0,N,0,,,,,

2,2008,1,3,4,628.0,620,804.0,750,WN,448,N428WN,96.0,90.0,76.0,14.0,8.0,IND,BWI,515,3.0,17.0,0,N,0,,,,,

4,2008,1,3,4,1829.0,1755,1959.0,1925,WN,3920,N464WN,90.0,90.0,77.0,34.0,34.0,IND,BWI,515,3.0,10.0,0,N,0,2.0,0.0,0.0,0.0,32.0

5,2008,1,3,4,1940.0,1915,2121.0,2110,WN,378,N726SW,101.0,115.0,87.0,11.0,25.0,IND,JAX,688,4.0,10.0,0,N,0,,,,,

6,2008,1,3,4,1937.0,1830,2037.0,1940,WN,509,N763SW,240.0,250.0,230.0,57.0,67.0,IND,LAS,1591,3.0,7.0,0,N,0,10.0,0.0,0.0,0.0,47.0

# Solutions

**Code written**

**package** com.duncb.spark

**import** org.apache.spark.\_

**import** org.apache.spark.SparkContext.\_

**import** org.apache.log4j.\_

**import** org.apache.spark.sql.\_

**import** org.apache.spark.sql.functions.\_

**object** flights {

**def** main(args: Array[*String*]) {

// Create a SparkContext using every core of the local machine

**val** sc = **new** SparkContext("local[\*]", "flights")

**val** sqlContext = **new** org.apache.spark.sql.SQLContext(sc)

**val** spark = SparkSession

.builder

.appName("SparkSQL")

.master("local")

.config("spark.sql.warehouse.dir", "file:///C:/temp") // Necessary to work around a Windows bug in Spark 2.0.0; omit if you're not on Windows.

.getOrCreate()

Logger.getLogger("org").setLevel(Level.ERROR)

println(“Find out the top 5 most visited destinations”)

**val** delayed\_flights = spark.sparkContext.textFile("file:////N:/Datasets/DelayedFlights.csv")

**val** mapping = delayed\_flights.map(x => x.split(",")).map(x => (x(18), 1)).filter(x =>

x.\_1 != **null**).reduceByKey(\_ + \_).map(x => (x.\_2, x.\_1)).sortByKey(**false**).map(x => (x.\_2, x.\_1)).take(5)

//sortByKey(false) -->> order by key desc

mapping.foreach(println)

println(“Which month has seen the most number of cancellations due to bad weather”)

**val** cancelled = delayed\_flights.map(x => x.split(",")).filter(x => ((x(22).equals("1"))&&

(x(23).equals("B")))).map(x => (x(2),1)).reduceByKey(\_+\_).map(x =>

(x.\_2,x.\_1)).sortByKey(**false**).map(x => (x.\_2,x.\_1)).take(1)

cancelled.foreach(println)

println(“Top ten origins with the highest AVG departure delay”)

**val** header = delayed\_flights.first()

**val** filter\_delayed\_flights = delayed\_flights.filter(row => row!= header)

**val** avg = filter\_delayed\_flights.map(x => x.split(",")).map(x => (x(17), x(16).toDouble)).mapValues((\_,

1)).reduceByKey((x, y) => (x.\_1 + y.\_1, x.\_2 + y.\_2)).mapValues{ **case** (sum, count) => (1.0 \*

sum)/count}.map(x => (x.\_2,x.\_1)).sortByKey(**false**).map(x => (x.\_2,x.\_1)).take(10)

avg.foreach(println)

println(“Which route (origin & destination) has seen the maximum diversion”)

**val** diversion = delayed\_flights.map(x => x.split(",")).filter(x => ((x(24).equals("1")))).map(x =>

((x(17)+","+x(18)),1)).reduceByKey(\_+\_).map(x => (x.\_2,x.\_1)).sortByKey(**false**).map(x =>

(x.\_2,x.\_1)).take(10)

diversion.foreach(println)

}

}

# Results

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/N:/Spark2/jars/slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

1. Find out the top 5 most visited destinations

[Stage 0:> (0 + 2) / 8]

[Stage 0:=======> (1 + 2) / 8]

[Stage 0:==============> (2 + 2) / 8]

[Stage 0:======================> (3 + 2) / 8]

[Stage 0:=============================> (4 + 2) / 8]

[Stage 0:=============================> (4 + 3) / 8]

[Stage 0:====================================> (5 + 2) / 8]

[Stage 0:============================================> (6 + 2) / 8]

(ORD,108984)

(ATL,106898)

(DFW,70657)

(DEN,63003)

(LAX,59969)

Which month has seen the most number of cancellations due to bad weather

[Stage 5:> (0 + 2) / 8]

[Stage 5:==============> (2 + 2) / 8]

[Stage 5:==============> (2 + 3) / 8]

[Stage 5:=============================> (4 + 2) / 8]

[Stage 5:====================================> (5 + 2) / 8]

[Stage 5:============================================> (6 + 2) / 8]

(12,250)

Top ten origins with the highest AVG departure delay

[Stage 14:> (0 + 2) / 8]

[Stage 14:=======> (1 + 2) / 8]

[Stage 14:=====================> (3 + 2) / 8]

[Stage 14:=============================> (4 + 2) / 8]

[Stage 14:===========================================> (6 + 2) / 8]

[Stage 14:==================================================> (7 + 1) / 8]

(CMX,116.1470588235294)

(PLN,93.76190476190476)

(SPI,83.84873949579831)

(ALO,82.2258064516129)

(MQT,79.55665024630542)

(ACY,79.3103448275862)

(MOT,78.66165413533835)

(HHH,76.53005464480874)

(EGE,74.12891986062718)

(BGM,73.15533980582525)

Which route (origin & destination) has seen the maximum diversion

[Stage 19:=======> (1 + 2) / 8]

[Stage 19:==============> (2 + 2) / 8]

[Stage 19:=====================> (3 + 2) / 8]

[Stage 19:=============================> (4 + 2) / 8]

[Stage 19:====================================> (5 + 2) / 8]

[Stage 19:===========================================> (6 + 2) / 8]

[Stage 19:==================================================> (7 + 1) / 8]

(ORD,LGA,39)

(DAL,HOU,35)

(DFW,LGA,33)

(ATL,LGA,32)

(SLC,SUN,31)

(ORD,SNA,31)

(MIA,LGA,31)

(BUR,JFK,29)

(HRL,HOU,28)

(BUR,DFW,25)