

# BIG DATA PROJECT

Barbara Jean June 6, 2024

**Hotel Booking Analysis:** Problem: Process and analyze hotel booking data to identify trends, booking preferences, and cancellations.

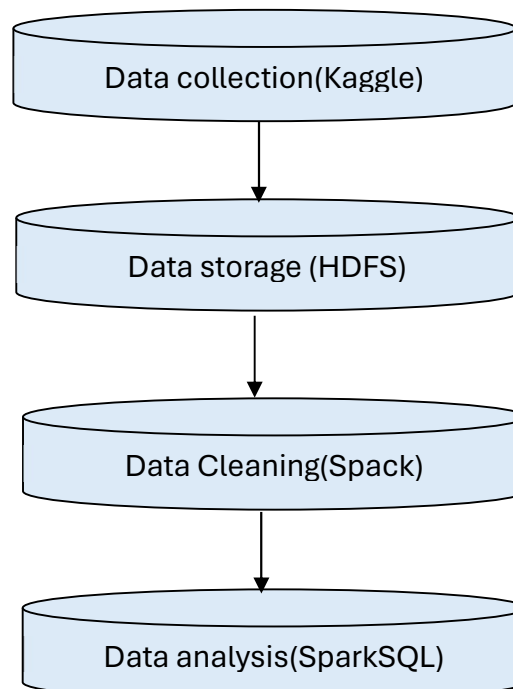
## Introduction

Large-scale websites and platforms are continuously updated in real-time as computer networks develop, producing copious amounts of data. For a long time, the hotel has been collecting vital information from online user comments. Data from booking systems, such as consumer choices, cancellations, and other booking-related information, can provide insightful information about the travel and tourism sector. It is imperative to conduct an efficient analysis and pinpoint noteworthy patterns to utilize this data appropriately. When data is used correctly, many inherent biases are eliminated, giving hotel chains that understand data analysis and automation a significant competitive edge in their decision-making. Digital information is used by hotel data analytics to help your hotel make better decisions.

### 1- Selecting Data Source

Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019. This dataset contains 119390 observations for a City Hotel and a Resort Hotel. Each observation represents a hotel booking between the 1st of July 2015 and the 31st of August 2017, including bookings that effectively arrived and canceled bookings and other booking-related information.

### 2- Selecting Technologies:



### 3- Describe the operations or transformations you performed on the data.

Data transformation is applying several operations to organize, clean, and standardize data. Standard methods of transformation consist of:

**Data Cleaning:** Cleaning up data involves addressing outliers, inconsistent data, and missing numbers. Incorrect, partial, or inaccurate source data must be eliminated or altered during the data cleansing.

**Aggregation:** Computing indicators such as rates of cancelation, trends in bookings, peak booking hours, and more. Data consistency in terms of its definition based on usage is guaranteed by aggregation. Search performance and speed are also enhanced by aggregation.

**Feature Engineering:** Creating a new feature out of an existing one, such as determining the length of stay depending on the reservation dates. Data engineers create and manage data transformations to guarantee that data arrives suitable for analysis as part of the pipelines they design and oversee that move data from source to destination.

**Statistical analysis:** Describes the process of uncovering trends, patterns, and correlations in large amounts of raw data to help make data-informed decisions.

### 4- Explain why you chose to use the components you did.

**HDFS:** The Hadoop Distributed File System is a good option for big data storage. Replication techniques are used by the Hadoop Distributed File System (HDFS), which divides data into blocks for storage on cluster nodes and controls access to the data.

**Spark:** Excellent for effectively processing big amounts of data, especially for complex transformations and analytics. Spark is an open-source framework for real-time workloads, machine learning, and interactive queries.

Two layers are suggested for processing and analysis: data sources, collection, storage, and analysis. The project displays multiple booking and information sources at the Data Sources layer. The information will be combined with Spark for big data and HDFS. Lastly, employing SparkSQL and Scala technology (Apache Spark), the experimental study examines the latent topic in a small sample of data the project retrieved based on the topic model.

### 5- Include screenshots of the data usage across all the components you used.

Load the hotel\_bookings.csv into HDFS

```
barbara@bigdata1: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase
bash-5.0# hdfs dfs -ls /
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/program/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/program/tez/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/program/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2024-05-31 05:48:54,994 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 8 items
drwxr-xr-x - root supergroup          0 2024-05-31 04:39 /data
drwxr-xr-x - root supergroup          0 2024-05-31 04:28 /hbase
-rw-r--r-- 1 root supergroup    263149 2024-05-31 05:31 /hotel_bookings.csv
drwxr-xr-x - root supergroup          0 2024-05-30 07:27 /log
drwxr-xr-x - root supergroup          0 2024-05-30 07:28 /spark-jars
drwxr-xr-x - root supergroup          0 2024-05-30 07:29 /tez
drwxr-xr-x - root supergroup          0 2024-05-31 04:29 /tmp
drwxrwx--- - root supergroup          0 2024-05-30 07:28 /user
bash-5.0#
```

Using SparkSQL with Scala: Spark SQL lets you query structured data inside Spark programs, using either SQL.

Read and create a temporary view for hotel\_booking dataset

```
val df = spark.read.format("csv").option("header",
"true").load("/data/hotel_bookings2.csv")
```

```
df.createOrReplaceTempView("df")
```

```
barbara@bigdata1: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase/data
-rw-r--r-- 1 root supergroup          747 2024-05-31 23:48 /data/grades.csv
-rw-r--r-- 1 root supergroup    263149 2024-05-31 22:50 /data/hotel_bookings.csv
-rw-r--r-- 1 root supergroup    16846346 2024-06-01 00:26 /data/hotel_bookings2.csv
bash-5.0# docker-compose exec master bash
bash: docker-compose: command not found
bash-5.0# spark-shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/program/spark/jars/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/program/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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://localhost:4040
Spark context available as 'sc' (master = yarn, app id = application_1717194929363_0006).
Spark session available as 'spark'.
Welcome to

  ____      _
 / ___|  __| | | |
 \___ \  | | | | | |
  ___) | | | | | | |
 |_____|_|_|_|_|_|_|

version 3.0.0

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val df = spark.read.format("csv").option("header", "true").load("/data/hotel_bookings2.csv")
df: org.apache.spark.sql.DataFrame = [hotel: string, is_canceled: string ... 30 more fields]

scala>
```

```

barbara@bigdata1: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase/data
df: org.apache.spark.sql.DataFrame = [hotel: string, is_canceled: string ... 30 more fields]

scala> df.createOrReplaceTempView("df")
188074 [main] WARN org.apache.spark.sql.catalyst.util.package - Truncated the string repres
behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.

scala> spark.sql("SHOW TABLES").show()
216319 [main] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.strict.mana
216319 [main] WARN org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.create.as.1
216440 [main] WARN org.apache.spark.sql.hive.client.HiveClientImpl - Detected HiveConf hive
'mr' to disable useless hive logic

+-----+-----+
|database|tableName|isTemporary|
+-----+-----+
|         |df        |true       |
+-----+-----+

scala>

```

```
spark.sql("SELECT * FROM df").show()
```

```

scala> spark.conf.set("spark.debug.maxToStringFields", 100)

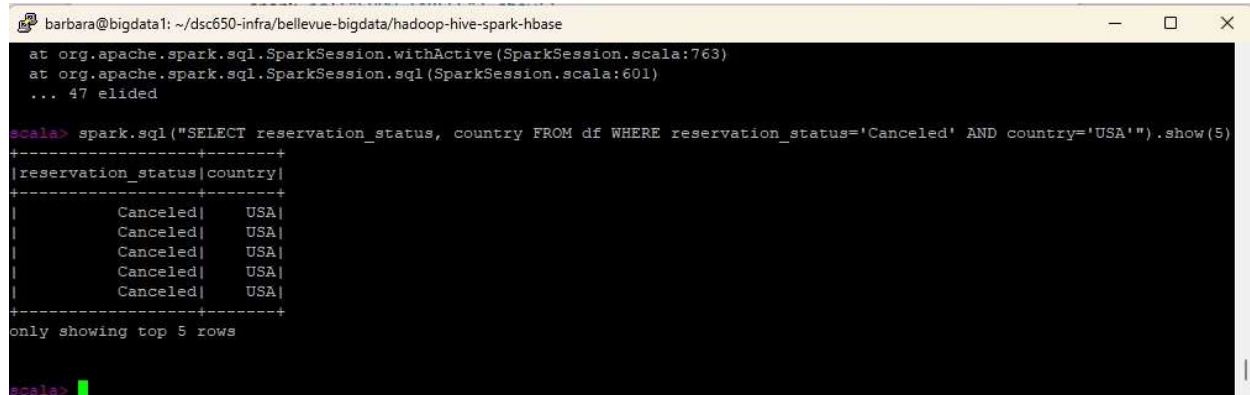
scala> spark.sql("SELECT * FROM df").show()
-----
| hotel|is_canceled|lead_time|arrival_date_year|arrival_date_month|arrival_date_week_number|arrival_date_day_of_month|stays_in_week_end_nights|stays_in_week_nights|adults|children|babies|meal|country|market_segment|distribution_channel|is_repeated_guest|previous_cancellations|previous_bookings_not_canceled|reserved_room_type|assigned_room_type|booking_changes|deposit_type|agent|company|days_in_waiting_list|customer_type|adr|required_car_parking_spaces|total_of_special_requests|reservation_status|reservation_status_date|
-----
|Resort|Hotel|0|342|2015|July|27|1|0|0|BB|PRT|Direct|Direct|0|0|No Deposit|NULL|NULL|7/1/2015|
|0|Transient|0|0|0|BB|C|0|Check-Out|7/1/2015|
|Resort|Hotel|0|737|2015|July|27|1|0|0|BB|PRT|Direct|Direct|0|0|No Deposit|NULL|NULL|7/1/2015|
|0|Transient|0|0|0|BB|C|0|Check-Out|7/1/2015|
|Resort|Hotel|0|7|2015|July|27|1|0|0|BB|GBR|Direct|Direct|0|0|No Deposit|NULL|NULL|7/2/2015|
|0|Transient|75|0|0|A|C|0|Check-Out|7/2/2015|

```

```
spark.sql("SELECT reservation_status, reservation_status_date FROM df").show(5)
```

```
scala> spark.sql("SELECT reservation_status, reservation_status_date FROM df").show(5)
+-----+-----+
|reservation_status|reservation_status_date|
+-----+-----+
|Check-Out|7/1/2015|
|Check-Out|7/1/2015|
|Check-Out|7/2/2015|
|Check-Out|7/2/2015|
|Check-Out|7/3/2015|
+-----+-----+
only showing top 5 rows
```

```
spark.sql("SELECT reservation_status, country FROM df WHERE reservation_status='Canceled' AND country='USA').show(5)
```

A terminal window titled 'barbara@bigdata1: ~/dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase' shows the execution of a Spark SQL query. The prompt is 'scala>'. The query is 'spark.sql("SELECT reservation\_status, country FROM df WHERE reservation\_status='Canceled' AND country='USA').show(5)'. The output is a table with two columns: 'reservation\_status' and 'country'. All five rows show 'Canceled' and 'USA'. The terminal also shows some Scala stack trace information at the top.

```
at org.apache.spark.sql.SparkSession.withActive(SparkSession.scala:763)
at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:601)
... 47 elided

scala> spark.sql("SELECT reservation_status, country FROM df WHERE reservation_status='Canceled' AND country='USA').show(5)
+-----+
|reservation_status|country|
+-----+
| Canceled|    USA|
| Canceled|    USA|
| Canceled|    USA|
| Canceled|    USA|
| Canceled|    USA|
+-----+
only showing top 5 rows

scala>
```

## Conclusion

The main goal of this final project is to process and analyze hotel booking data to find patterns, booking preferences, and cancellations. HDFS has been used to collect and store a set of data. Generally speaking, the current data set only partially satisfies Big Data requirements. Nevertheless, as a result of the project's suggested analysis, it was examined and evaluated as a sizable data set. Adopting parallel processing and vast data storage frameworks in the significant data era is inevitable. The storage framework implements distributed storage using HDFS from Hadoop. The data cleaning and analysis module uses the Spark framework because of its heavy workload. Spark processes data significantly more quickly because of its memory-based architecture.

## References

edureka Veranda. How to copy files from HDFS to local system.

<https://www.edureka.co/community/87006/how-to-copy-files-from-hdfs-to-local-system>

Apache Spark. (n.d.). LDA – PySpark 3.3.2 documentation.

<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.clustering.LDA.html#pyspark.ml.clustering.LDA.checkpointInterval>

geeksforgeeks. How to show full-column content in a PySpark Dataframe ?.

<https://www.geeksforgeeks.org/how-to-show-full-column-content-in-a-pyspark-dataframe/#>

Kaggle.Hotel booking analysis.

<https://www.kaggle.com/code/aminizahra/hotel-booking-analysis>

Amazon Web Services. (n.d.). Data warehouse vs. Data lake vs. Data mart – Comparing cloud storage solutions – AWS.

<https://aws.amazon.com/compare/the-difference-between-a-data-warehouse-data-lake-and-data-mart/>