DATA LOADING ,PREPROCESSING AND CLEANING

In [ ]:
```
!apt-get update -qq
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Use archive.apache.org to download a valid Spark version
!wget -q https://archive.apache.org/dist/spark/spark-3.5.1/spark-3.5.1-bin-hadoc

!tar xf spark-3.5.1-bin-hadoop3.tgz
!pip install -q findspark
```

```
W: Skipping acquire of configured file 'main/source/Sources' as repository 'https://
r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem to provide it (sources.l
entry misspelt?)
```

In [ ]:
```
!ls /content
```

```
sample_data  spark-3.5.1-bin-hadoop3  spark-3.5.1-bin-hadoop3.tgz
```

In [ ]:
```
import os, findspark

os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.1-bin-hadoop3"

findspark.init()
```

In [ ]:
```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("BigDataAnalysis").getOrCreate()
print("Spark session started!")
```

```
Spark session started!
```

In [ ]:
```
!wget https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.
```

```
--2025-09-20 11:39:40--  https://raw.githubusercontent.com/datasciencedojo/datasets/r
titanic.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133,
185.199.111.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|
connected.
HTTP request sent, awaiting response... 200 OK
Length: 60302 (59K) [text/plain]
Saving to: ‘titanic.csv.1’


titanic.csv.1         0%[                    ]       0  --.-KB/s
titanic.csv.1       100%[===================>]  58.89K  --.-KB/s    in 0.01s

2025-09-20 11:39:40 (4.21 MB/s) - ‘titanic.csv.1’ saved [60302/60302]
```

In [ ]:
```
# Titanic dataset ko Spark DataFrame mein load karna
df = spark.read.csv("/content/titanic.csv", header=True, inferSchema=True)

# Pehli 5 rows dikhana
df.show(5)
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------
+-------+-----+--------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|
Ticket|   Fare|Cabin|Embarked|
+-----------+--------+------+--------------------+------+----+-----+-----+----------
+-------+-----+--------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|22.0|    1|    0|       A/5
21171|   7.25| NULL|       S|
|          2|       1|     1|Cumings, Mrs. Joh...|female|38.0|    1|    0|        PC
71.2833|  C85|       C|
|          3|       1|     3|Heikkinen, Miss. ...|female|26.0|    0|    0|STON/O2.
3101282|  7.925| NULL|       S|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|35.0|    1|    0|
113803|   53.1| C123|       S|
|          5|       0|     3|Allen, Mr. Willia...|  male|35.0|    0|    0|
373450|   8.05| NULL|       S|
+-----------+--------+------+--------------------+------+----+-----+-----+----------
+-------+-----+--------+
only showing top 5 rows
```

In [ ]:

```
# Columns aur data types
df.printSchema()

# Total rows
print("Total Rows:", df.count())
```

```
root
 |-- PassengerId: integer (nullable = true)
 |-- Survived: integer (nullable = true)
 |-- Pclass: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- SibSp: integer (nullable = true)
 |-- Parch: integer (nullable = true)
 |-- Ticket: string (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Cabin: string (nullable = true)
 |-- Embarked: string (nullable = true)

Total Rows: 891
```

In [ ]:

```
df.describe()
```

Out[ ]:

```
DataFrame[summary: string, PassengerId: string, Survived: string, Pclass: string, N
string, Sex: string, Age: string, SibSp: string, Parch: string, Ticket: string, Far
string, Cabin: string, Embarked: string]
```

ANALYSIS

In [ ]:

```
from pyspark.sql import functions as F
#survival rate by gender
survival_by_gender = df.groupBy("Sex") \
    .agg(F.avg("Survived").alias("survival_rate")) \
    .withColumn("survival_rate", F.col("survival_rate") * 100)

survival_by_gender.show()
```

```
+------+------------------+
|   Sex|      survival_rate|
+------+------------------+
|female| 74.20382165605095|
|  male|18.890814558058924|
+------+------------------+
```

In [ ]:

```
#average age and fare by class
avg_stats = df.groupBy("Pclass") \
    .agg(F.avg("Age").alias("avg_age"),
         F.avg("Fare").alias("avg_fare"))

avg_stats.show()
```

```
+------+------------------+------------------+
|Pclass|           avg_age|          avg_fare|
+------+------------------+------------------+
|     1|38.233440860215055| 84.15468749999992|
|     3| 25.14061971830986|13.675550101832997|
|     2| 29.87763005780347| 20.66218315217391|
+------+------------------+------------------+
```

In [ ]:

```
df.createOrReplaceTempView("titanic")

result = spark.sql("""
SELECT Pclass, Sex, COUNT(*) as total_passengers,
       AVG(Survived)*100 as survival_rate
FROM titanic
GROUP BY Pclass, Sex
ORDER BY Pclass, survival_rate DESC
""")

result.show()
```

```
+------+------+----------------+------------------+
|Pclass|   Sex|total_passengers|     survival_rate|
+------+------+----------------+------------------+
|     1|female|              94| 96.80851063829788|
|     1|  male|             122|36.885245901639344|
|     2|female|              76| 92.10526315789474|
|     2|  male|             108| 15.74074074074074|
|     3|female|             144|              50.0|
|     3|  male|             347|13.544668587896252|
+------+------+----------------+------------------+
```

INSIGHTS

In [ ]:

```
#Several rate by gender
result1 = spark.sql("""
SELECT Sex, AVG(Survived)*100 as survival_rate
FROM titanic
GROUP BY Sex
""")
result1.show()
```

```
+------+------------------+
|   Sex|      survival_rate|
+------+------------------+
|female| 74.20382165605095|
|  male|18.890814558058924|
+------+------------------+
```

In [ ]:

```
#Several rate by passenger class
result2 = spark.sql("""
SELECT Pclass, AVG(Survived)*100 as survival_rate
FROM titanic
GROUP BY Pclass
ORDER BY Pclass
""")
result2.show()
```

```
+------+------------------+
|Pclass|     survival_rate|
+------+------------------+
|     1| 62.96296296296296|
|     2| 47.28260869565217|
|     3|24.236252545824847|
+------+------------------+
```

In [ ]:

```
#Average fair by  survival
result3 = spark.sql("""
SELECT Survived, AVG(Fare) as avg_fare
FROM titanic
GROUP BY Survived
""")
result3.show()
```

```
+--------+------------------+
|Survived|          avg_fare|
+--------+------------------+
|       1| 48.39540760233917|
|       0|22.117886885245877|
+--------+------------------+
```