Name: **RahimaSiddiqui(2303.KHI.DEG.030)**

Peer name: **<u>Osama Abdul Razzak (2303.KHI.DEG.029)</u>**

Peer name:M Humza Moeen(2303.KHI.DEG.019)

Assignment 5.3

Read data from source to DataFrame in local Spark setup and display DataFrame schema.

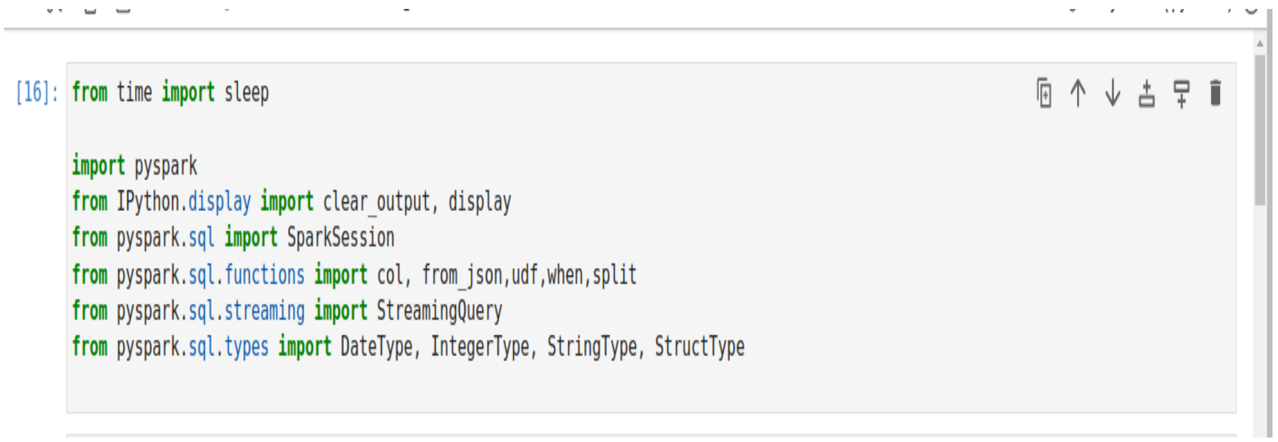
`tasks/5_data_pipelines/day_3_spark/data_assignment`

For numerical columns, calculate minimum, maximum and average values.

For categorical columns, create and apply UDF that will change the last letter of every word to "1".

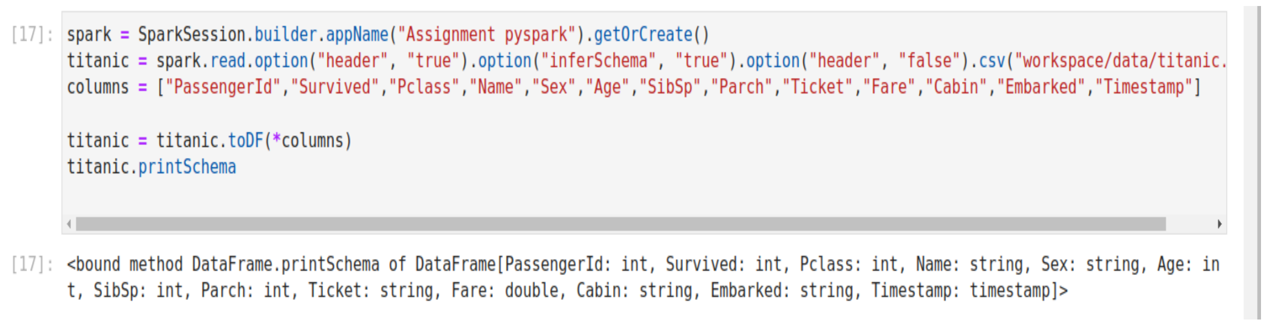
Sort DataFrame by the first column and save the results to the Parquet file.

First, we import these libraries that we are used in this assignment.


```
[16]: from time import sleep

      import pyspark
      from IPython.display import clear_output, display
      from pyspark.sql import SparkSession
      from pyspark.sql.functions import col, from_json,udf,when,split
      from pyspark.sql.streaming import StreamingQuery
      from pyspark.sql.types import DateType, IntegerType, StringType, StructType
```

Then we import dataset and naming the columns according to dataset.

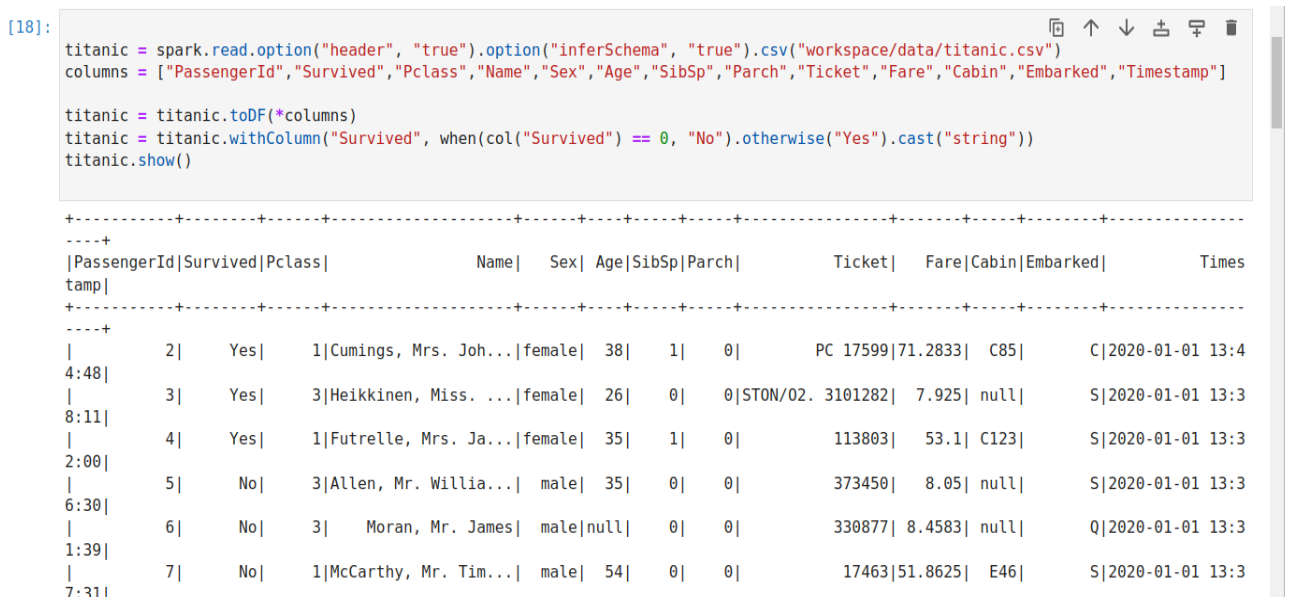


```
[17]: spark = SparkSession.builder.appName("Assignment pyspark").getOrCreate()
      titanic = spark.read.option("header", "true").option("inferSchema", "true").option("header", "false").csv("workspace/data/titanic.
      columns = ["PassengerId","Survived","Pclass","Name","Sex","Age","SibSp","Parch","Ticket","Fare","Cabin","Embarked","Timestamp"]

      titanic = titanic.toDF(*columns)
      titanic.printSchema
```

```
[17]: <bound method DataFrame.printSchema of DataFrame[PassengerId: int, Survived: int, Pclass: int, Name: string, Sex: string, Age: in
      t, SibSp: int, Parch: int, Ticket: string, Fare: double, Cabin: string, Embarked: string, Timestamp: timestamp]>
```

Then we modifying the schema and changing the data type of survived to string and converting the binary values into string so that we can categorize it.

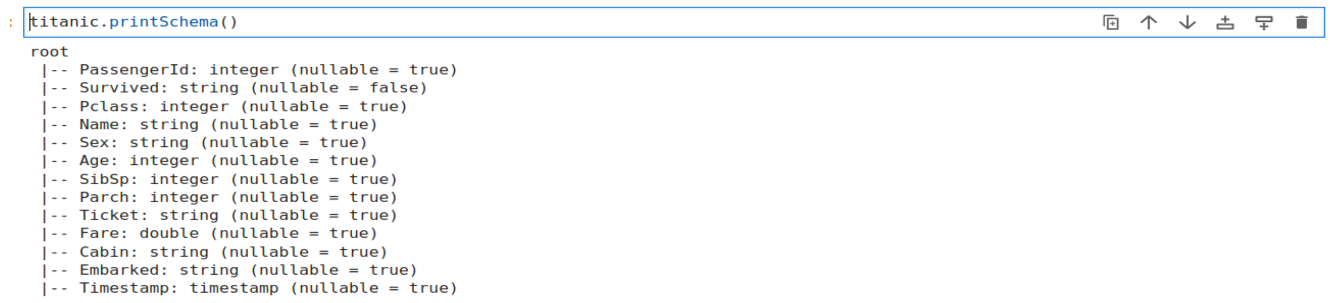


```
[18]:
titanic = spark.read.option("header", "true").option("inferSchema", "true").csv("workspace/data/titanic.csv")
columns = ["PassengerId","Survived","Pclass","Name","Sex","Age","SibSp","Parch","Ticket","Fare","Cabin","Embarked","Timestamp"]

titanic = titanic.toDF(*columns)
titanic = titanic.withColumn("Survived", when(col("Survived") == 0, "No").otherwise("Yes").cast("string"))
titanic.show()
```

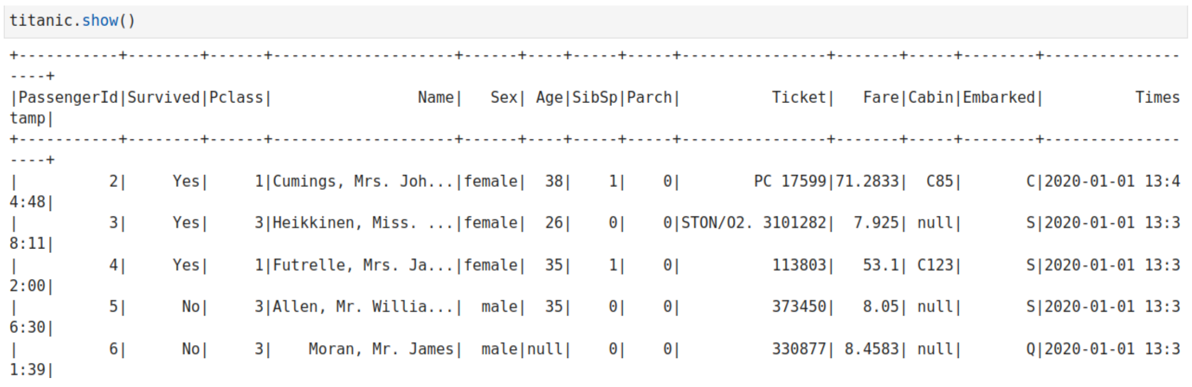
```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|          Times
tamp|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|          2|     Yes|     1|Cumings, Mrs. Joh...|female|  38|    1|    0|        PC 17599|71.2833|  C85|       C|2020-01-01 13:4
4:48|
|          3|     Yes|     3|Heikkinen, Miss. ...|female|  26|    0|    0|STON/O2. 3101282|  7.925| null|       S|2020-01-01 13:3
8:11|
|          4|     Yes|     1|Futrelle, Mrs. Ja...|female|  35|    1|    0|          113803|   53.1| C123|       S|2020-01-01 13:3
2:00|
|          5|      No|     3|Allen, Mr. Willia...|  male|  35|    0|    0|          373450|   8.05| null|       S|2020-01-01 13:3
6:30|
|          6|      No|     3|    Moran, Mr. James|  male|null|    0|    0|          330877| 8.4583| null|       Q|2020-01-01 13:3
1:39|
|          7|      No|     1|McCarthy, Mr. Tim...|  male|  54|    0|    0|           17463|51.8625|  E46|       S|2020-01-01 13:3
7:31|
```

This is our updated schema



```
titanic.printSchema()
```

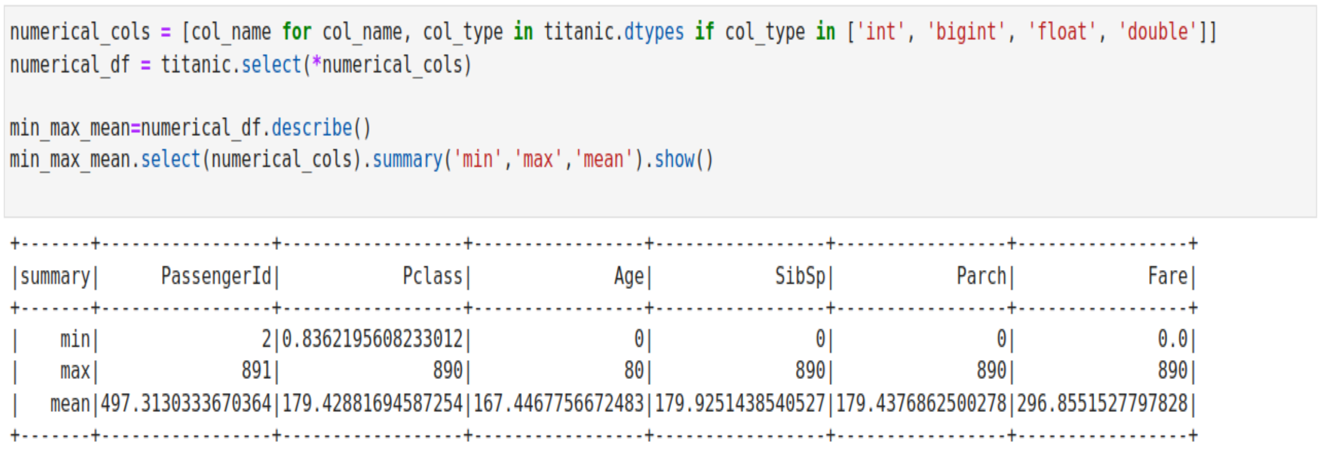
```
root
 |-- PassengerId: integer (nullable = true)
 |-- Survived: string (nullable = false)
 |-- Pclass: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- SibSp: integer (nullable = true)
 |-- Parch: integer (nullable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



```
titanic.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|          Times
tamp|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|          2|     Yes|     1|Cumings, Mrs. Joh...|female|  38|    1|    0|        PC 17599|71.2833|  C85|       C|2020-01-01 13:4
4:48|
|          3|     Yes|     3|Heikkinen, Miss. ...|female|  26|    0|    0|STON/O2. 3101282|  7.925| null|       S|2020-01-01 13:3
8:11|
|          4|     Yes|     1|Futrelle, Mrs. Ja...|female|  35|    1|    0|          113803|   53.1| C123|       S|2020-01-01 13:3
2:00|
|          5|      No|     3|Allen, Mr. Willia...|  male|  35|    0|    0|          373450|   8.05| null|       S|2020-01-01 13:3
6:30|
|          6|      No|     3|    Moran, Mr. James|  male|null|    0|    0|          330877| 8.4583| null|       Q|2020-01-01 13:3
1:39|
```

Then we are checking the minimum, maximum and average value of the numerical data

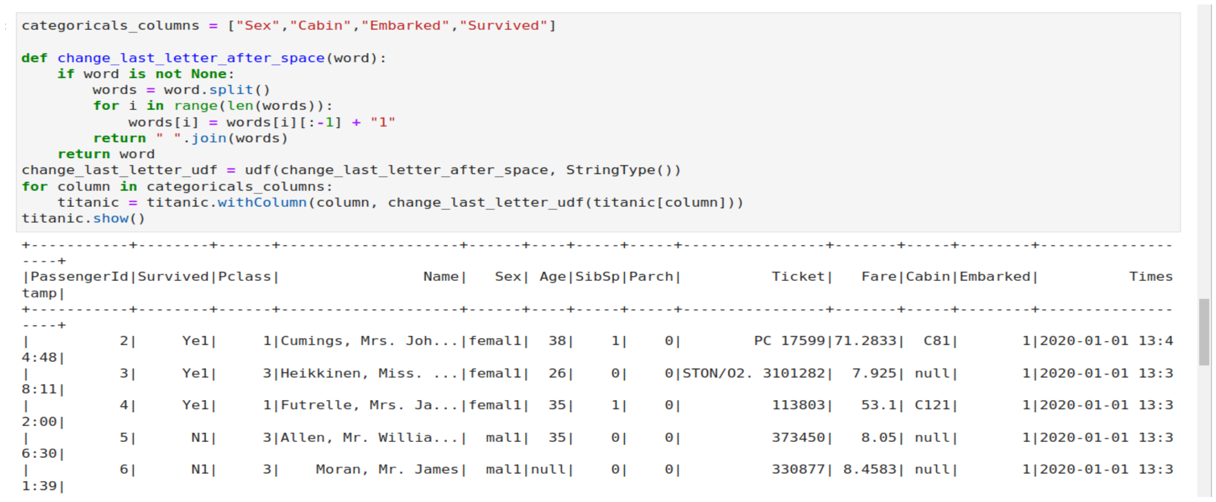


```
numerical_cols = [col_name for col_name, col_type in titanic.dtypes if col_type in ['int', 'bigint', 'float', 'double']]
numerical_df = titanic.select(*numerical_cols)

min_max_mean=numerical_df.describe()
min_max_mean.select(numerical_cols).summary('min','max','mean').show()
```

```
+-------+----------------+------------------+----------------+----------------+----------------+----------------+
|summary|     PassengerId|            Pclass|             Age|           SibSp|           Parch|            Fare|
+-------+----------------+------------------+----------------+----------------+----------------+----------------+
|    min|               2|0.8362195608233012|               0|               0|               0|             0.0|
|    max|             891|               890|              80|             890|             890|             890|
|   mean|497.3130333670364|179.42881694587254|167.4467756672483|179.9251438540527|179.4376862500278|296.8551527797828|
+-------+----------------+------------------+----------------+----------------+----------------+----------------+
```

Then we apply udf function to change the last letter of categorical data to 1 and here is the result.

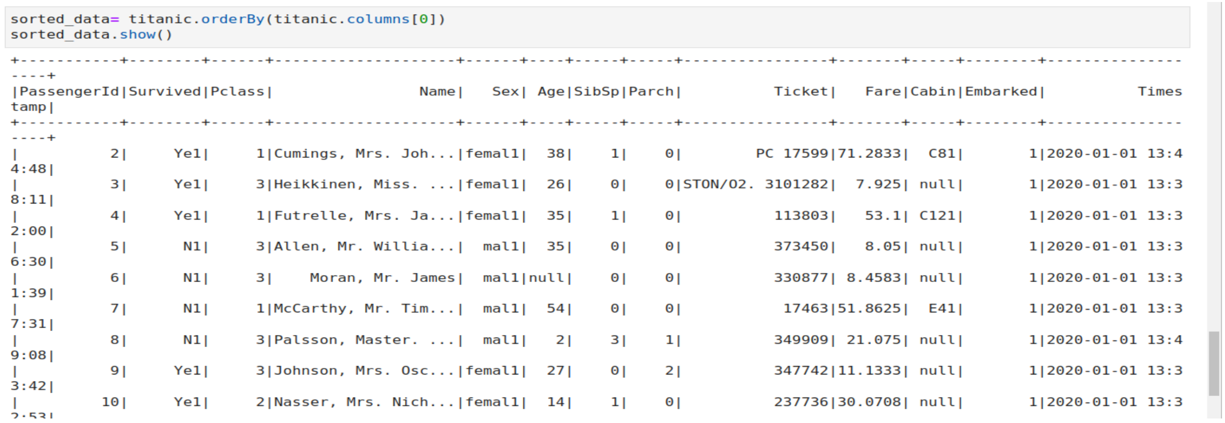


```
categoricals_columns = ["Sex","Cabin","Embarked","Survived"]

def change_last_letter_after_space(word):
    if word is not None:
        words = word.split()
        for i in range(len(words)):
            words[i] = words[i][:-1] + "1"
        return " ".join(words)
    return word
change_last_letter_udf = udf(change_last_letter_after_space, StringType())
for column in categoricals_columns:
    titanic = titanic.withColumn(column, change_last_letter_udf(titanic[column]))
titanic.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|          Times
tamp|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|          2|     Ye1|     1|Cumings, Mrs. Joh...|femal1|  38|    1|    0|        PC 17599|71.2833|  C81|       1|2020-01-01 13:4
4:48|
|          3|     Ye1|     3|Heikkinen, Miss. ...|femal1|  26|    0|    0|STON/O2. 3101282|  7.925| null|       1|2020-01-01 13:3
8:11|
|          4|     Ye1|     1|Futrelle, Mrs. Ja...|femal1|  35|    1|    0|          113803|   53.1| C121|       1|2020-01-01 13:3
2:00|
|          5|      N1|     3|Allen, Mr. Willia...|  mal1|  35|    0|    0|          373450|   8.05| null|       1|2020-01-01 13:3
6:30|
|          6|      N1|     3|    Moran, Mr. James|  mal1|null|    0|    0|          330877| 8.4583| null|       1|2020-01-01 13:3
1:39|
```

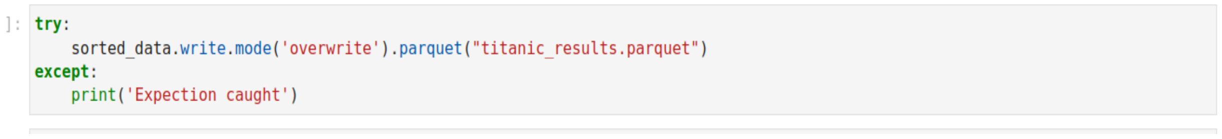
Then we sorting the data by its first column



```
sorted_data= titanic.orderBy(titanic.columns[0])
sorted_data.show()
```

```
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|          Times
tamp|
+-----------+--------+------+--------------------+------+----+-----+-----+----------------+-------+-----+--------+---------------
----+
|          2|     Ye1|     1|Cumings, Mrs. Joh...|femal1|  38|    1|    0|        PC 17599|71.2833|  C81|       1|2020-01-01 13:4
4:48|
|          3|     Ye1|     3|Heikkinen, Miss. ...|femal1|  26|    0|    0|STON/O2. 3101282|  7.925| null|       1|2020-01-01 13:3
8:11|
|          4|     Ye1|     1|Futrelle, Mrs. Ja...|femal1|  35|    1|    0|          113803|   53.1| C121|       1|2020-01-01 13:3
2:00|
|          5|      N1|     3|Allen, Mr. Willia...|  mal1|  35|    0|    0|          373450|   8.05| null|       1|2020-01-01 13:3
6:30|
|          6|      N1|     3|    Moran, Mr. James|  mal1|null|    0|    0|          330877| 8.4583| null|       1|2020-01-01 13:3
1:39|
|          7|      N1|     1|McCarthy, Mr. Tim...|  mal1|  54|    0|    0|           17463|51.8625|  E41|       1|2020-01-01 13:3
7:31|
|          8|      N1|     3|Palsson, Master. ...|  mal1|   2|    3|    1|          349909| 21.075| null|       1|2020-01-01 13:4
9:08|
|          9|     Ye1|     3|Johnson, Mrs. Osc...|femal1|  27|    0|    2|          347742|11.1333| null|       1|2020-01-01 13:3
3:42|
|         10|     Ye1|     2|Nasser, Mrs. Nich...|femal1|  14|    1|    0|          237736|30.0708| null|       1|2020-01-01 13:3
2:53|
```

Then we saving the resultant dataset in parquet.



```
]: try:
       sorted_data.write.mode('overwrite').parquet("titanic_results.parquet")
   except:
       print('Expection caught')
```