Name: Maria Gloria Raquel Obono

NUID: 002667315

HOMEWORK (SPARK 2)

1. Exploratory Data Analysis- Follow up on the previous spark assignment 1 and explain a few statistics.

Survival Analysis:

The dataset allows us to analyze the survival rates of passengers based on various factors such as age, gender, ticket class, family relations, and more. We found that there is a significant variation in survival rates based on different factors, including ticket class, age, and gender. For example, 1st-class passengers had a higher chance of survival compared to 3rd-class passengers.

Age and Ticket Fare Analysis:

We grouped passengers into 10-year age intervals to analyze the relationship between age and ticket fares.The analysis showed that the average ticket fare varies across different age groups. This can help identify if certain age groups paid higher or lower fares.

Identifying Specific Passengers:

We filtered the dataset to find passengers who could possibly be Rose DeWitt Bukater and Jack Dawson based on specific characteristics. This analysis can help identify if there are any passengers who closely match the descriptions of these two fictional characters from the Titanic movie.

Ticket Class Analysis:

We calculated the average ticket fare for each ticket class (Pclass). This analysis provides insights into the cost of tickets for different classes, which can be an indicator of the socio-economic status of passengers.

Some key statistics and insights that could be derived from these analyses include:

- The Titanic dataset contains information on passengers from different socio-economic backgrounds, and this diversity has an impact on survival rates and ticket fares.
- Passengers in the 1st class generally had higher survival rates and paid higher average ticket fares compared to those in the 3rd class.
- The age and gender of passengers also played a role in survival rates. For example, women and children had higher chances of survival.
- The dataset contains passengers who closely match the descriptions of Rose DeWitt Bukater and Jack Dawson from the movie "Titanic," adding a fictional and human element to the analysis.

2. Feature Engineering - Create new attributes that may be derived from the existing attributes. This may include removing certain columns in the dataset.

```
// Feature Engineering
val trainDataWithFeatures = dataCleaned
//val trainDataWithFeatures = trainData
  // Create a new feature 'FamilySize' by summing 'SibSp' and 'Parch'
  .withColumn( colName = "FamilySize", col( colName = "SibSp") + col( colName = "Parch"))
  // Extract the title from the 'Name' column
  .withColumn( colName = "Title", regexp_extract(col( colName = "Name"), exp = "(Mrs\\.|Mr\\.|Miss\\.|Master\\.|Rev\\.|Dr\\.|Major\\.|Capt\\.|Col\\.|Lady\\.

  // Use StringIndexer to convert categorical 'Sex' and 'Embarked' columns to numerical
  .transform { df =>
    val sexIndexer = new StringIndexer()
      .setInputCol("Sex")
      .setOutputCol("SexIndex")
      .setHandleInvalid("skip") // Skip rows with invalid values
      .fit(df)

    val embarkedIndexer = new StringIndexer()
      .setInputCol("Embarked")
      .setOutputCol("EmbarkedIndex")
      .setHandleInvalid("skip") // Skip rows with invalid values
      .fit(df)

    val titleIndexer = new StringIndexer()
      .setInputCol("Title")
      .setOutputCol("TitleIndex")
```

PassengerId	Pclass	Age	FamilySize	TitleIndex	SexIndex	Fare	EmbarkedIndex	Survived
1	3	22	1.0	0.0	0.0	7	0.0	0
2	1	38	1.0	2.0	1.0	71	1.0	1
3	3	26	0.0	1.0	1.0	7	0.0	1
4	1	35	1.0	2.0	1.0	53	0.0	1
5	3	35	0.0	0.0	0.0	8	0.0	0
6	3	0	0.0	0.0	0.0	8	2.0	0
7	1	54	0.0	0.0	0.0	51	0.0	0
8	3	2	4.0	3.0	0.0	21	0.0	0
9	3	27	2.0	2.0	1.0	11	0.0	1
10	2	14	1.0	2.0	1.0	30	1.0	1
11	3	4	2.0	1.0	1.0	16	0.0	1
12	1	58	0.0	1.0	1.0	26	0.0	1
13	3	20	0.0	0.0	0.0	8	0.0	0
14	3	39	6.0	0.0	0.0	31	0.0	0
15	3	14	0.0	1.0	1.0	7	0.0	0
16	2	55	0.0	2.0	1.0	16	0.0	1
17	3	2	5.0	3.0	0.0	29	2.0	0
18	2	0	0.0	0.0	0.0	13	0.0	1
19	3	31	1.0	2.0	1.0	18	0.0	0
20	3	0	0.0	2.0	1.0	7	1.0	1

only showing top 20 rows

```
Accuracy: 70.30615832028379%
```

3. Prediction - Use the train.csv to train a Machine Learning model of your choice & test it on the test.csv. You are required to predict if the records in test.csv survived or not. Note(1 = Survived, 0 = Dead)

```
// Make predictions on the test data
val testFeatures = testData
  .withColumn( colName = "Pclass", testData("Pclass").cast( to = "int"))
  .withColumn( colName = "Age", testData("Age").cast( to = "int"))
  .withColumn( colName = "Fare", testData("Fare").cast( to = "int"))
  .na.fill( value = 0.0, Seq("Age", "Fare"))
  .withColumn( colName = "FamilySize", col( colName = "SibSp") + col( colName = "Parch"))
  .withColumn( colName = "Title", regexp_extract(col( colName = "Name"), exp = "(Mrs\\.|Mr\\.|Miss\\.|Master\\.|Rev\\.|Dr\\.|Major\\.|Capt\\.|Col\\.|Lady\\.|
  .transform { df =>
    val sexIndexer = new StringIndexer()
      .setInputCol("Sex")
      .setOutputCol("SexIndex")
      .setHandleInvalid("skip")
      .fit(df)

    val embarkedIndexer = new StringIndexer()
      .setInputCol("Embarked")
      .setOutputCol("EmbarkedIndex")
      .setHandleInvalid("skip")
      .fit(df)
```

```
+-----------+------+---+----------+----------+--------+----+-------------+--------------------+--------------------+--------------------+----------+
|PassengerId|Pclass|Age|FamilySize|TitleIndex|SexIndex|Fare|EmbarkedIndex|            features|       rawPrediction|         probability|prediction|
+-----------+------+---+----------+----------+--------+----+-------------+--------------------+--------------------+--------------------+----------+
|        892|     3| 34|       0.0|       0.0|     0.0|   7|          2.0|[3.0,34.0,0.0,0.0...|[0.86956521739130...|[0.86956521739130...|       0.0|
|        893|     3| 47|       1.0|       2.0|     1.0|   7|          0.0|[3.0,47.0,1.0,1.0...|          [1.0,0.0]|          [1.0,0.0]|       0.0|
|        894|     2| 62|       0.0|       0.0|     0.0|   9|          2.0|[2.0,62.0,0.0,0.0...|[0.86956521739130...|[0.86956521739130...|       0.0|
|        895|     3| 27|       0.0|       0.0|     0.0|   8|          0.0|[3.0,27.0,0.0,0.0...|[0.86956521739130...|[0.86956521739130...|       0.0|
|        896|     3| 22|       2.0|       2.0|     1.0|  12|          0.0|[3.0,22.0,2.0,1.0...|          [1.0,0.0]|          [1.0,0.0]|       0.0|
|        897|     3| 14|       0.0|       0.0|     0.0|   9|          0.0|[3.0,14.0,0.0,0.0...|[0.86956521739130...|[0.86956521739130...|       0.0|
|        898|     3| 30|       0.0|       1.0|     1.0|   7|          2.0|[3.0,30.0,0.0,1.0...|          [1.0,0.0]|          [1.0,0.0]|       0.0|
|        899|     2| 26|       2.0|       0.0|     0.0|  29|          0.0|[2.0,26.0,2.0,0.0...|[0.23333333333333...|[0.23333333333333...|       1.0|
|        900|     3| 18|       0.0|       2.0|     1.0|   7|          1.0|[3.0,18.0,0.0,1.0...|          [0.0,1.0]|          [0.0,1.0]|       1.0|
|        901|     3| 21|       2.0|       0.0|     0.0|  24|          0.0|[3.0,21.0,2.0,0.0...|[0.23333333333333...|[0.23333333333333...|       1.0|
|        902|     3|  0|       0.0|       0.0|     0.0|   7|          0.0| (6,[0,4],[3.0,7.0])|[0.86956521739130...|[0.86956521739130...|       0.0|
|        903|     1| 46|       0.0|       0.0|     0.0|  26|          0.0|[1.0,46.0,0.0,0.0...|[0.95833333333333...|[0.95833333333333...|       0.0|
|        904|     1| 23|       1.0|       2.0|     1.0|  82|          0.0|[1.0,23.0,1.0,1.0...|          [0.0,1.0]|          [0.0,1.0]|       1.0|
|        905|     2| 63|       1.0|       0.0|     0.0|  26|          0.0|[2.0,63.0,1.0,0.0...|[0.95833333333333...|[0.95833333333333...|       0.0|
|        906|     1| 47|       1.0|       2.0|     1.0|  61|          0.0|[1.0,47.0,1.0,1.0...|          [0.0,1.0]|          [0.0,1.0]|       1.0|
|        907|     2| 24|       1.0|       2.0|     1.0|  27|          1.0|[2.0,24.0,1.0,1.0...|          [0.0,1.0]|          [0.0,1.0]|       1.0|
|        908|     2| 35|       0.0|       0.0|     0.0|  12|          2.0|[2.0,35.0,0.0,0.0...|[0.86956521739130...|[0.86956521739130...|       0.0|
|        909|     3| 21|       0.0|       0.0|     0.0|   7|          1.0|[3.0,21.0,0.0,0.0...|[0.86956521739130...|[0.86956521739130...|       0.0|
|        910|     3| 27|       1.0|       1.0|     1.0|   7|          0.0|[3.0,27.0,1.0,1.0...|          [1.0,0.0]|          [1.0,0.0]|       0.0|
|        911|     3| 45|       0.0|       2.0|     1.0|   7|          1.0|[3.0,45.0,0.0,1.0...|          [1.0,0.0]|          [1.0,0.0]|       0.0|
+-----------+------+---+----------+----------+--------+----+-------------+--------------------+--------------------+--------------------+----------+
only showing top 20 rows
```