*Creado por:*

*Isabel Maniega*

# MLlib

## MLlib usando Titanic dataset

In [1]:
```
from pyspark.sql import SparkSession
from pyspark .sql.types import StructType

spark = SparkSession.builder.appName("titanic").getOrCreate()
```

In [2]:
```
df = spark.read.csv("train.csv", header=True).cache()
```

In [3]:
```
df.show(10)
```

```
+-----------+--------+------+--------------------+------+----+-----+----
-+----------------+-------+-----+--------+
|PassengerId|Survived|Pclass|                Name|   Sex| Age|SibSp|Parc
h|          Ticket|   Fare|Cabin|Embarked|
+-----------+--------+------+--------------------+------+----+-----+----
-+----------------+-------+-----+--------+
|          1|       0|     3|Braund, Mr. Owen ...|  male|  22|    1|
0|       A/5 21171|   7.25| null|       S|
|          2|       1|     1|Cumings, Mrs. Joh...|female|  38|    1|
0|        PC 17599|71.2833|  C85|       C|
|          3|       1|     3|Heikkinen, Miss. ...|female|  26|    0|
0|STON/O2. 3101282|  7.925| null|       S|
|          4|       1|     1|Futrelle, Mrs. Ja...|female|  35|    1|
0|          113803|   53.1| C123|       S|
|          5|       0|     3|Allen, Mr. Willia...|  male|  35|    0|
0|          373450|   8.05| null|       S|
|          6|       0|     3|    Moran, Mr. James|  male|null|    0|
0|          330877| 8.4583| null|       Q|
|          7|       0|     1|McCarthy, Mr. Tim...|  male|  54|    0|
0|           17463|51.8625|  E46|       S|
|          8|       0|     3|Palsson, Master. ...|  male|   2|    3|
1|          349909| 21.075| null|       S|
|          9|       1|     3|Johnson, Mrs. Osc...|female|  27|    0|
2|          347742|11.1333| null|       S|
|         10|       1|     2|Nasser, Mrs. Nich...|female|  14|    1|
0|          237736|30.0708| null|       C|
+-----------+--------+------+--------------------+------+----+-----+----
-+----------------+-------+-----+--------+
only showing top 10 rows
```

In [4]:
```
df.toPandas()
```

Out[4]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22 | 1 | 0 | A/5 21171 | |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38 | 1 | 0 | PC 17599 | 71. |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26 | 0 | 0 | STON/O2. 3101282 | 7 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35 | 1 | 0 | 113803 | |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35 | 0 | 0 | 373450 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **886** | 887 | 0 | 2 | Montvila, Rev. Juozas | male | 27 | 0 | 0 | 211536 | |
| **887** | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19 | 0 | 0 | 112053 | |
| **888** | 889 | 0 | 3 | "Johnston, Miss. Catherine Helen ""Carrie""" | female | None | 1 | 2 | W./C. 6607 | 2 |
| **889** | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26 | 0 | 0 | 111369 | |
| **890** | 891 | 0 | 3 | Dooley, Mr. Patrick | male | 32 | 0 | 0 | 370376 | |

891 rows × 12 columns

In [5]:
```
df.count()
```

Out[5]: 891

In [6]:
```
df.columns
```

Out[6]:
```
['PassengerId',
 'Survived',
 'Pclass',
 'Name',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked']
```

In [7]:
```
df.dtypes
```

Out[7]:
```
[('PassengerId', 'string'),
 ('Survived', 'string'),
 ('Pclass', 'string'),
 ('Name', 'string'),
 ('Sex', 'string'),
 ('Age', 'string'),
 ('SibSp', 'string'),
 ('Parch', 'string'),
 ('Ticket', 'string'),
 ('Fare', 'string'),
 ('Cabin', 'string'),
 ('Embarked', 'string')]
```

In [8]:
```
df.describe().toPandas()
```

Out[8]:

| | summary | PassengerId | Survived | Pclass | Nam |
|---|---|---|---|---|---|
| **0** | count | 891 | 891 | 891 | 89 |
| **1** | mean | 446.0 | 0.3838383838383838 | 2.308641975308642 | Non |
| **2** | stddev | 257.3538420152301 | 0.48659245426485753 | 0.8360712409770491 | Non |
| **3** | min | 1 | 0 | 1 | "Andersson, M August Edvar (""Wennerstrom"") |
| **4** | max | 99 | 1 | 3 | van Melkebeke Mr. Philemo |

In [9]:
```
# Realizamos la transformación de las columnas de string a numeros:

from pyspark.sql.functions import col

dataset = df.select(col("Survived").cast("float"),
                    col("Pclass").cast("float"),
                    col("Sex"),
                    col("Age").cast("float"),
                    col("Fare").cast("float"),
                    col("Embarked"))
dataset.show()
```

```
+--------+------+------+----+-------+--------+
|Survived|Pclass|   Sex| Age|   Fare|Embarked|
+--------+------+------+----+-------+--------+
|     0.0|   3.0|  male|22.0|   7.25|       S|
|     1.0|   1.0|female|38.0|71.2833|       C|
|     1.0|   3.0|female|26.0|  7.925|       S|
|     1.0|   1.0|female|35.0|   53.1|       S|
|     0.0|   3.0|  male|35.0|   8.05|       S|
|     0.0|   3.0|  male|null| 8.4583|       Q|
|     0.0|   1.0|  male|54.0|51.8625|       S|
|     0.0|   3.0|  male| 2.0| 21.075|       S|
|     1.0|   3.0|female|27.0|11.1333|       S|
|     1.0|   2.0|female|14.0|30.0708|       C|
|     1.0|   3.0|female| 4.0|   16.7|       S|
|     1.0|   1.0|female|58.0|  26.55|       S|
|     0.0|   3.0|  male|20.0|   8.05|       S|
|     0.0|   3.0|  male|39.0| 31.275|       S|
|     0.0|   3.0|female|14.0| 7.8542|       S|
|     1.0|   2.0|female|55.0|   16.0|       S|
|     0.0|   3.0|  male| 2.0| 29.125|       Q|
|     1.0|   2.0|  male|null|   13.0|       S|
|     0.0|   3.0|female|31.0|   18.0|       S|
|     1.0|   3.0|female|null|  7.225|       C|
+--------+------+------+----+-------+--------+
only showing top 20 rows
```

In [10]:
```
from pyspark.sql.functions import isnull, when, count
```

In [11]:
```
# Mostraremos la información en las columnas en las que falten datos:

dataset.select([count(when(isnull(c),c)).alias(c)
                for c in dataset.columns]).show()
```

```
+--------+------+---+---+----+--------+
|Survived|Pclass|Sex|Age|Fare|Embarked|
+--------+------+---+---+----+--------+
|       0|     0|  0|177|   0|       2|
+--------+------+---+---+----+--------+
```

In [12]:
```
dataset = dataset.replace("?", None).dropna(how="any")
```

In [13]:
```
# Modificación de las columnas en formato string a número.

from pyspark.ml.feature import StringIndexer

dataset = StringIndexer(inputCol="Sex",
                        outputCol="Gender",
                        handleInvalid="keep")\
        .fit(dataset).transform(dataset)

dataset = StringIndexer(inputCol="Embarked",
                        outputCol="Boarded",
                        handleInvalid="keep") \
        .fit(dataset).transform(dataset)

dataset.show()
```

```
+--------+------+------+----+-------+--------+------+-------+
|Survived|Pclass|   Sex| Age|   Fare|Embarked|Gender|Boarded|
+--------+------+------+----+-------+--------+------+-------+
|     0.0|   3.0|  male|22.0|   7.25|       S|   0.0|    0.0|
|     1.0|   1.0|female|38.0|71.2833|       C|   1.0|    1.0|
|     1.0|   3.0|female|26.0|  7.925|       S|   1.0|    0.0|
|     1.0|   1.0|female|35.0|   53.1|       S|   1.0|    0.0|
|     0.0|   3.0|  male|35.0|   8.05|       S|   0.0|    0.0|
|     0.0|   1.0|  male|54.0|51.8625|       S|   0.0|    0.0|
|     0.0|   3.0|  male| 2.0| 21.075|       S|   0.0|    0.0|
|     1.0|   3.0|female|27.0|11.1333|       S|   1.0|    0.0|
|     1.0|   2.0|female|14.0|30.0708|       C|   1.0|    1.0|
|     1.0|   3.0|female| 4.0|   16.7|       S|   1.0|    0.0|
|     1.0|   1.0|female|58.0|  26.55|       S|   1.0|    0.0|
|     0.0|   3.0|  male|20.0|   8.05|       S|   0.0|    0.0|
|     0.0|   3.0|  male|39.0| 31.275|       S|   0.0|    0.0|
|     0.0|   3.0|female|14.0| 7.8542|       S|   1.0|    0.0|
|     1.0|   2.0|female|55.0|   16.0|       S|   1.0|    0.0|
|     0.0|   3.0|  male| 2.0| 29.125|       Q|   0.0|    2.0|
|     0.0|   3.0|female|31.0|   18.0|       S|   1.0|    0.0|
|     0.0|   2.0|  male|35.0|   26.0|       S|   0.0|    0.0|
|     1.0|   2.0|  male|34.0|   13.0|       S|   0.0|    0.0|
|     1.0|   3.0|female|15.0| 8.0292|       Q|   1.0|    2.0|
+--------+------+------+----+-------+--------+------+-------+
only showing top 20 rows
```

Observamos que nos ha creado dos nuevas columnas con los valores de:

- Hombre: 0
- Mujer: 1
- S: 0
- C: 1
- Q: 2

In [14]:
```
dataset.dtypes
```

Out[14]:
```
[('Survived', 'float'),
 ('Pclass', 'float'),
 ('Sex', 'string'),
 ('Age', 'float'),
 ('Fare', 'float'),
 ('Embarked', 'string'),
 ('Gender', 'double'),
 ('Boarded', 'double')]
```

In [15]:
```
dataset = dataset.drop("Sex")
dataset = dataset.drop("Embarked")
dataset.show()
```

```
+--------+------+----+-------+------+-------+
|Survived|Pclass| Age|   Fare|Gender|Boarded|
+--------+------+----+-------+------+-------+
|     0.0|   3.0|22.0|   7.25|   0.0|    0.0|
|     1.0|   1.0|38.0|71.2833|   1.0|    1.0|
|     1.0|   3.0|26.0|  7.925|   1.0|    0.0|
|     1.0|   1.0|35.0|   53.1|   1.0|    0.0|
|     0.0|   3.0|35.0|   8.05|   0.0|    0.0|
|     0.0|   1.0|54.0|51.8625|   0.0|    0.0|
|     0.0|   3.0| 2.0| 21.075|   0.0|    0.0|
|     1.0|   3.0|27.0|11.1333|   1.0|    0.0|
|     1.0|   2.0|14.0|30.0708|   1.0|    1.0|
|     1.0|   3.0| 4.0|   16.7|   1.0|    0.0|
|     1.0|   1.0|58.0|  26.55|   1.0|    0.0|
|     0.0|   3.0|20.0|   8.05|   0.0|    0.0|
|     0.0|   3.0|39.0| 31.275|   0.0|    0.0|
|     0.0|   3.0|14.0| 7.8542|   1.0|    0.0|
|     1.0|   2.0|55.0|   16.0|   1.0|    0.0|
|     0.0|   3.0| 2.0| 29.125|   0.0|    2.0|
|     0.0|   3.0|31.0|   18.0|   1.0|    0.0|
|     0.0|   2.0|35.0|   26.0|   0.0|    0.0|
|     1.0|   2.0|34.0|   13.0|   0.0|    0.0|
|     1.0|   3.0|15.0| 8.0292|   1.0|    2.0|
+--------+------+----+-------+------+-------+
only showing top 20 rows
```

# Realizamos un vetor con la información correspondiente a la X:

In [16]:
```
required_features = ['Pclass', 'Age',
                     'Fare', 'Gender',
                     'Boarded']
```

In [17]:
```
from pyspark.ml.feature import VectorAssembler
```

In [18]:
```
assembler = VectorAssembler(inputCols=required_features,
                            outputCol="feature")
transformed_data = assembler.transform(dataset)
```

In [19]:
```
transformed_data.show(5)
```

```
+--------+------+----+-------+------+-------+--------------------+
|Survived|Pclass| Age|   Fare|Gender|Boarded|             feature|
+--------+------+----+-------+------+-------+--------------------+
|     0.0|   3.0|22.0|   7.25|   0.0|    0.0|[3.0,22.0,7.25,0....|
|     1.0|   1.0|38.0|71.2833|   1.0|    1.0|[1.0,38.0,71.2833...|
|     1.0|   3.0|26.0|  7.925|   1.0|    0.0|[3.0,26.0,7.92500...|
|     1.0|   1.0|35.0|   53.1|   1.0|    0.0|[1.0,35.0,53.0999...|
|     0.0|   3.0|35.0|   8.05|   0.0|    0.0|[3.0,35.0,8.05000...|
+--------+------+----+-------+------+-------+--------------------+
only showing top 5 rows
```

# Realizamos los modelos...

### Algoritmo de Decission Tree Classifier

In [20]:
```
# Crearemos los dataset de entrenamiento y test:

training_data, test_data = transformed_data.randomSplit([0.8, 0.2])
```

In [21]:
```
from pyspark.ml.classification import DecisionTreeClassifier

dt = DecisionTreeClassifier(labelCol="Survived",
                            featuresCol="feature",
                            maxDepth=5)
```

In [22]:
```
model = dt.fit(training_data)
```

In [23]:
```
predictions = model.transform(test_data)
```

In [24]:
```
predictions.show()
```

```
+--------+------+----+-------+------+-------+-------------------+------
-------+--------------------+----------+
|Survived|Pclass| Age|   Fare|Gender|Boarded|            feature|rawPre
diction|         probability|prediction|
+--------+------+----+-------+------+-------+-------------------+------
-------+--------------------+----------+
|     0.0|   1.0| 2.0| 151.55|   1.0|    0.0|[1.0,2.0,151.5500...|   [2.
0,97.0]|[0.02020202020202...|       1.0|
|     0.0|   1.0|19.0|  263.0|   0.0|    0.0|[1.0,19.0,263.0,0...|   [1
2.0,7.0]|[0.63157894736842...|       0.0|
|     0.0|   1.0|28.0|   47.1|   0.0|    0.0|[1.0,28.0,47.0999...|   [1
2.0,7.0]|[0.63157894736842...|       0.0|
|     0.0|   1.0|31.0|50.4958|   0.0|    0.0|[1.0,31.0,50.4958...|   [1
2.0,7.0]|[0.63157894736842...|       0.0|
|     0.0|   1.0|33.0|    5.0|   0.0|    0.0|[1.0,33.0,5.0,0.0...|   [2.
0,10.0]|[0.16666666666666...|       1.0|
|     0.0|   1.0|40.0|27.7208|   0.0|    1.0|[1.0,40.0,27.7208...|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|45.5|   28.5|   0.0|    0.0|[1.0,45.5,28.5,0....|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|47.0|   52.0|   0.0|    0.0|[1.0,47.0,52.0,0....|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|51.0|61.3792|   0.0|    1.0|[1.0,51.0,61.3791...|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|58.0|113.275|   0.0|    1.0|[1.0,58.0,113.275...|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|61.0|32.3208|   0.0|    0.0|[1.0,61.0,32.3208...|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|61.0|   33.5|   0.0|    0.0|[1.0,61.0,33.5,0....|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   1.0|65.0|61.9792|   0.0|    1.0|[1.0,65.0,61.9791...|  [35.
0,14.0]|[0.71428571428571...|       0.0|
|     0.0|   2.0|16.0|   10.5|   0.0|    0.0|[2.0,16.0,10.5,0....| [226.
0,29.0]|[0.88627450980392...|       0.0|
|     0.0|   2.0|18.0|   11.5|   0.0|    0.0|[2.0,18.0,11.5,0....| [226.
0,29.0]|[0.88627450980392...|       0.0|
|     0.0|   2.0|19.0|  36.75|   0.0|    0.0|[2.0,19.0,36.75,0...| [226.
0,29.0]|[0.88627450980392...|       0.0|
|     0.0|   2.0|24.0|   10.5|   0.0|    0.0|[2.0,24.0,10.5,0....| [226.
0,29.0]|[0.88627450980392...|       0.0|
|     0.0|   2.0|24.0|   13.0|   1.0|    0.0|[2.0,24.0,13.0,1....|   [2.
0,97.0]|[0.02020202020202...|       1.0|
|     0.0|   2.0|25.0|   13.0|   0.0|    0.0|[2.0,25.0,13.0,0....| [226.
0,29.0]|[0.88627450980392...|       0.0|
|     0.0|   2.0|25.0|   26.0|   0.0|    0.0|[2.0,25.0,26.0,0....| [226.
0,29.0]|[0.88627450980392...|       0.0|
+--------+------+----+-------+------+-------+-------------------+------
-------+--------------------+----------+
only showing top 20 rows
```

In [25]:
```
# Evaluamos el modelo:

from pyspark.ml.evaluation import MulticlassClassificationEvaluator

evaluator = MulticlassClassificationEvaluator(labelCol="Survived",
                                              predictionCol="prediction",
                                              metricName="accuracy")
```

In [26]:
```
accuracy = evaluator.evaluate(predictions)
```

```
print("Test Accuracy: ", accuracy*100)
```

```
Test Accuracy:  78.62595419847328
```

**Algortimo de Gradient-boosted tree Classifier**

In [27]:
```
from pyspark.ml.classification import GBTClassifier

gbt = GBTClassifier(labelCol="Survived",
                    featuresCol="feature",
                    maxIter=5)
```

In [28]:
```
model = gbt.fit(training_data)
```

In [29]:
```
predictions = model.transform(test_data)
```

In [30]:
```
predictions.show()
```

```
+--------+------+----+-------+------+-------+--------------------+------
--------------+--------------------+----------+
|Survived|Pclass| Age|   Fare|Gender|Boarded|             feature|
rawPrediction|         probability|prediction|
+--------+------+----+-------+------+-------+--------------------+------
--------------+--------------------+----------+
|     0.0|   1.0| 2.0| 151.55|   1.0|    0.0|[1.0,2.0,151.5500...|[-1.12
88233001191...|[0.09469192265227...|       1.0|
|     0.0|   1.0|19.0|  263.0|   0.0|    0.0|[1.0,19.0,263.0,0...|[0.363
59022815136...|[0.67418623843183...|       0.0|
|     0.0|   1.0|28.0|   47.1|   0.0|    0.0|[1.0,28.0,47.0999...|[0.655
44229008781...|[0.78766114050320...|       0.0|
|     0.0|   1.0|31.0|50.4958|   0.0|    0.0|[1.0,31.0,50.4958...|[0.655
44229008781...|[0.78766114050320...|       0.0|
|     0.0|   1.0|33.0|    5.0|   0.0|    0.0|[1.0,33.0,5.0,0.0...|[-0.64
90811448625...|[0.21447446304673...|       1.0|
|     0.0|   1.0|40.0|27.7208|   0.0|    1.0|[1.0,40.0,27.7208...|[0.438
25944010983...|[0.70610032853383...|       0.0|
|     0.0|   1.0|45.5|   28.5|   0.0|    0.0|[1.0,45.5,28.5,0....|[0.381
22188544048...|[0.68188406759584...|       0.0|
|     0.0|   1.0|47.0|   52.0|   0.0|    0.0|[1.0,47.0,52.0,0....|[0.714
08380913205...|[0.80661564890119...|       0.0|
|     0.0|   1.0|51.0|61.3792|   0.0|    1.0|[1.0,51.0,61.3791...|[0.137
79946413715...|[0.56846691396054...|       0.0|
|     0.0|   1.0|58.0|113.275|   0.0|    1.0|[1.0,58.0,113.275...|[0.383
99436229520...|[0.68308565397057...|       0.0|
|     0.0|   1.0|61.0|32.3208|   0.0|    0.0|[1.0,61.0,32.3208...|[0.381
22188544048...|[0.68188406759584...|       0.0|
|     0.0|   1.0|61.0|   33.5|   0.0|    0.0|[1.0,61.0,33.5,0....|[0.381
22188544048...|[0.68188406759584...|       0.0|
|     0.0|   1.0|65.0|61.9792|   0.0|    1.0|[1.0,65.0,61.9791...|[0.137
79946413715...|[0.56846691396054...|       0.0|
|     0.0|   2.0|16.0|   10.5|   0.0|    0.0|[2.0,16.0,10.5,0....|[0.873
69001093507...|[0.85162204134876...|       0.0|
|     0.0|   2.0|18.0|   11.5|   0.0|    0.0|[2.0,18.0,11.5,0....|[0.873
69001093507...|[0.85162204134876...|       0.0|
|     0.0|   2.0|19.0|  36.75|   0.0|    0.0|[2.0,19.0,36.75,0...|[0.873
69001093507...|[0.85162204134876...|       0.0|
|     0.0|   2.0|24.0|   10.5|   0.0|    0.0|[2.0,24.0,10.5,0....|[0.873
69001093507...|[0.85162204134876...|       0.0|
|     0.0|   2.0|24.0|   13.0|   1.0|    0.0|[2.0,24.0,13.0,1....|[-1.17
65057745327...|[0.08682669597733...|       1.0|
|     0.0|   2.0|25.0|   13.0|   0.0|    0.0|[2.0,25.0,13.0,0....|[0.873
69001093507...|[0.85162204134876...|       0.0|
|     0.0|   2.0|25.0|   26.0|   0.0|    0.0|[2.0,25.0,26.0,0....|[0.873
69001093507...|[0.85162204134876...|       0.0|
+--------+------+----+-------+------+-------+--------------------+------
--------------+--------------------+----------+
only showing top 20 rows
```

In [31]:
```
accuracy = evaluator.evaluate(predictions)
print("Test Accuracy: ", accuracy*100)
```

```
Test Accuracy:  77.86259541984732
```

*Creado por:*

*Isabel Maniega*