Assignment - 8

In [1]:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('logregconsult').getOrCreate()

from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator
```

In [2]:

```
data = spark.read.csv("gs://bigdata-roja/notebooks/jupyter/customer_churn.csv", inferSch
data.printSchema()
```

```
root
 |-- Names: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- Total_Purchase: double (nullable = true)
 |-- Account_Manager: integer (nullable = true)
 |-- Years: double (nullable = true)
 |-- Num_Sites: double (nullable = true)
 |-- Onboard_date: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- Churn: integer (nullable = true)
```

In [3]:

```
data.printSchema()
```

```
root
 |-- Names: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- Total_Purchase: double (nullable = true)
 |-- Account_Manager: integer (nullable = true)
 |-- Years: double (nullable = true)
 |-- Num_Sites: double (nullable = true)
 |-- Onboard_date: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- Churn: integer (nullable = true)
```

In [4]:

```
data.describe().show()
```

```
+-------+-------------+-----------------+-----------------+------------------+----------
-------+------------------+-------------------+--------------------+--------------------
+-------------------+
|summary|        Names|              Age|   Total_Purchase|   Account_Manager|          
  Years|         Num_Sites|       Onboard_date|            Location|             Company
|              Churn|
+-------+-------------+-----------------+-----------------+------------------+----------
-------+------------------+-------------------+--------------------+--------------------
+-------------------+
|  count|          900|              900|              900|               900|          
    900|               900|                900|                 900|                 900
|                900|
|   mean|         null|41.81666666666667|10062.82403333334|0.4811111111111111| 5.2731555
5555555| 8.587777777777777|               null|                null|                null
|0.16666666666666666|
| stddev|         null|6.127560416916251|2408.644531858096|0.4999208935073339|1.27444901
3194616|1.7648355920350969|               null|                null|                null
| 0.3728852122772358|
|    min|   Aaron King|                22.0|             100.0|                  0|
   1.0|               3.0|2006-01-02 04:16:13|00103 Jeffrey Cre...|     Abbott-Thompson
|                 0|
|    max|Zachary Walsh|                65.0|          18026.01|                  1|
  9.15|              14.0|2016-12-28 04:07:38|Unit 9800 Box 287...|Zuniga, Clark and...
|                 1|
+-------+-------------+------------------+------------------+-------------------+----------
-------+------------------+--------------------+---------------------+--------------------
+-------------------+
```

In [5]:

```
data.columns
```

Out[5]:

```
['Names',
 'Age',
 'Total_Purchase',
 'Account_Manager',
 'Years',
 'Num_Sites',
 'Onboard_date',
 'Location',
 'Company',
 'Churn']
```

In [6]:

```
from pyspark.ml.feature import VectorAssembler
```

In [7]:

```
assembler = VectorAssembler(inputCols=['Age','Total_Purchase','Account_Manager','Years',
```

In [8]:

```
output=assembler.transform(data)
```

In [9]:

```
final_data=output.select('features','churn')
```

In [10]:

```
training_churn,test_churn=final_data.randomSplit([0.7,0.3])
```

In [11]:

```
logreg_churn = LogisticRegression( labelCol='churn' )
fitted_churn_model = logreg_churn.fit( training_churn )
```

```
22/03/31 16:34:02 WARN BLAS: Failed to load implementation from: com.github.fommil.netli
b.NativeSystemBLAS
22/03/31 16:34:02 WARN BLAS: Failed to load implementation from: com.github.fommil.netli
b.NativeRefBLAS
```

In [12]:

```
training_sum = fitted_churn_model.summary
training_sum.predictions.describe().show()
```

```
+-------+------------------+------------------+
|summary|             churn|        prediction|
+-------+------------------+------------------+
|  count|               621|               621|
|   mean|0.1610305958132045|0.1143317230273752|
| stddev|0.3678554686783657|0.3184702540043179|
|    min|               0.0|               0.0|
|    max|               1.0|               1.0|
+-------+------------------+------------------+
```

In [13]:

```
predictions_and_labels=fitted_churn_model.evaluate(test_churn)
predictions_and_labels.predictions.show()
```

```
+--------------------+-----+--------------------+--------------------+----------+
|            features|churn|       rawPrediction|         probability|prediction|
+--------------------+-----+--------------------+--------------------+----------+
|[25.0,9672.03,0.0...|    0|[4.43315863376195...|[0.98826248988013...|       0.0|
|[28.0,8670.98,0.0...|    0|[7.30986688401807...|[0.99933154104802...|       0.0|
|[29.0,5900.78,1.0...|    0|[3.63349447141014...|[0.97425655075026...|       0.0|
|[29.0,13240.01,1....|    0|[6.31286800319051...|[0.99819045380943...|       0.0|
|[30.0,6744.87,0.0...|    0|[3.24301906944412...|[0.96242145093633...|       0.0|
|[30.0,10744.14,1....|    1|[1.58676297931137...|[0.83016018941307...|       0.0|
|[30.0,10960.52,1....|    0|[2.18727933157522...|[0.89910136041175...|       0.0|
|[30.0,11575.37,1....|    1|[3.68720345621111...|[0.97556984331829...|       0.0|
|[31.0,5387.75,0.0...|    0|[2.35639495463551...|[0.91344119213218...|       0.0|
|[31.0,7073.61,0.0...|    0|[2.89001066704590...|[0.94735041361396...|       0.0|
|[31.0,8688.21,0.0...|    0|[6.22215436935792...|[0.99801896801632...|       0.0|
|[32.0,6367.22,1.0...|    0|[2.56634412240949...|[0.92866388321110...|       0.0|
|[32.0,9036.27,0.0...|    0|[-0.1636660045440...|[0.45917458923493...|       1.0|
|[32.0,10716.75,0....|    0|[4.23747097929389...|[0.98576158567008...|       0.0|
|[33.0,7492.9,0.0,...|    0|[4.55731091755606...|[0.98961867237317...|       0.0|
|[33.0,8556.73,0.0...|    0|[3.53131628625153...|[0.97156579821635...|       0.0|
|[33.0,10306.21,1....|    0|[1.83523493164256...|[0.86238417592137...|       0.0|
|[33.0,11370.28,1....|    0|[6.21320911442575...|[0.99800120326389...|       0.0|
|[34.0,5447.16,1.0...|    0|[2.95799455670423...|[0.95063997627165...|       0.0|
|[34.0,7324.32,0.0...|    0|[1.11126310298428...|[0.75236451787980...|       0.0|
+--------------------+-----+--------------------+--------------------+----------+
only showing top 20 rows
```

In [14]:

```
evaluator = BinaryClassificationEvaluator(rawPredictionCol='prediction',labelCol='churn'
```

In [15]:

```
au=evaluator.evaluate(predictions_and_labels.predictions)
print(au)
```

```
0.7981659388646288
```

In [16]:

```
logreg_model_final = logreg_churn.fit(final_data)
#new customer data
new_cust = spark.read.csv("gs://bigdata-roja/notebooks/jupyter/new_customers.csv",inferS
new_cust.printSchema()
```

```
root
 |-- Names: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- Total_Purchase: double (nullable = true)
 |-- Account_Manager: integer (nullable = true)
 |-- Years: double (nullable = true)
 |-- Num_Sites: double (nullable = true)
 |-- Onboard_date: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
```

In [17]:

```
new_cust_t = assembler.transform(new_cust)
new_cust_t.printSchema()
```

```
root
 |-- Names: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- Total_Purchase: double (nullable = true)
 |-- Account_Manager: integer (nullable = true)
 |-- Years: double (nullable = true)
 |-- Num_Sites: double (nullable = true)
 |-- Onboard_date: string (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- features: vector (nullable = true)
```

In [18]:

```
final_results = logreg_model_final.transform(new_cust_t)
final_results.select('Company','prediction').show(25)
```

```
+----------------+----------+
|         Company|prediction|
+----------------+----------+
|        King Ltd|       0.0|
|   Cannon-Benson|       1.0|
|Barron-Robertson|       1.0|
|   Sexton-Golden|       1.0|
|        Wood LLC|       0.0|
|   Parks-Robbins|       1.0|
+----------------+----------+
```

In [ ]:

```
```