In [1]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('myPySparkProject').getOrCreate()
```

In [2]:
```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import (VectorAssembler,VectorIndexer,OneHotEncoder
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StandardScaler
from pyspark.ml import Pipeline
from pyspark.sql.functions import *
from pyspark.ml.evaluation import BinaryClassificationEvaluator
```

In [3]:
```
p = "gs://11514794_bucket1/Training_Data/customer_churn.csv"
traindata = spark.read.csv(r,inferSchema = True,header = True)
traindata.describe().show()
```

```
+-------+-------------+-----------------+-----------------+--------------
----+-----------------+------------------+--------------------+----------
----------+-------------------+
|summary|        Names|              Age|   Total_Purchase|   Account_Man
ager|            Years|         Num_Sites|            Location|
Company|              Churn|
+-------+-------------+-----------------+-----------------+--------------
----+-----------------+------------------+--------------------+----------
----------+-------------------+
|  count|          900|              900|              900|
900|              900|               900|                 900|
900|                900|
|   mean|         null|41.81666666666667|10062.82403333334|0.481111111111
1111| 5.27315555555555| 8.587777777777777|                null|
null|0.16666666666666666|
| stddev|         null|6.127560416916251|2408.644531858096|0.499920893507
3339|1.274449013194616|1.7648355920350969|                null|
null| 0.3728852122772358|
|    min|   Aaron King|             22.0|            100.0|
0|              1.0|               3.0|00103 Jeffrey Cre...|     Abbott-T
hompson|                  0|
|    max|Zachary Walsh|             65.0|         18026.01|
1|             9.15|              14.0|Unit 9800 Box 287...|Zuniga, Clark
and...|                  1|
+-------+-------------+-----------------+-----------------+--------------
----+-----------------+------------------+--------------------+----------
----------+-------------------+
```

In [4]:
```
traindata.columns
```

Out[4]:
```
['Names',
 'Age',
 'Total_Purchase',
 'Account_Manager',
 'Years',
 'Num_Sites',
 'Onboard_date',
 'Location',
 'Company',
 'Churn']
```

In [5]:
```
traindata.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: timestamp (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- Churn: integer (nullable = true)
```

In [6]:
```
useful_columns = traindata.select(['Names',
 'Age',
 'Total_Purchase',
 'Account_Manager',
 'Years',
 'Num_Sites',
 'Onboard_date',
 'Location',
 'Company',
 'Churn'])

my_final_ColumnData = useful_columns.na.drop()
```

In [7]:
```
Assembler = VectorAssembler(inputCols=['Age',
 'Total_Purchase',
 'Account_Manager',
 'Years',
 'Num_Sites'
],outputCol='features')
```

In [8]:
```
Output = Assembler.transform(my_final_ColumnData)
```

In [9]:
```
final_output =  Output.select("features","Churn")
```

In [10]:
```
final_output.show()
```

```
+--------------------+-----+
|            features|Churn|
+--------------------+-----+
|[42.0,11066.8,0.0...|    1|
|[41.0,11916.22,0....|    1|
|[38.0,12884.75,0....|    1|
|[42.0,8010.76,0.0...|    1|
|[37.0,9191.58,0.0...|    1|
|[48.0,10356.02,0....|    1|
|[44.0,11331.58,1....|    1|
|[32.0,9885.12,1.0...|    1|
|[43.0,14062.6,1.0...|    1|
|[40.0,8066.94,1.0...|    1|
|[30.0,11575.37,1....|    1|
|[45.0,8771.02,1.0...|    1|
|[45.0,8988.67,1.0...|    1|
|[40.0,8283.32,1.0...|    1|
|[41.0,6569.87,1.0...|    1|
|[38.0,10494.82,1....|    1|
|[45.0,8213.41,1.0...|    1|
|[43.0,11226.88,0....|    1|
|[53.0,5515.09,0.0...|    1|
|[46.0,8046.4,1.0,...|    1|
+--------------------+-----+
only showing top 20 rows
```

In [14]:
```
Train_Customers_Data , Test_Customers_Data = final_output.randomSplit([0.7,
```

In [15]:
```
log_reg_Customers = LogisticRegression(labelCol='Churn')
```

In [16]:
```
fit_Customermodel = log_reg_Customers.fit(Train_Customers_Data)
```

In [17]:
```
fit_Customermodel.summary
```

Out[17]:
```
<pyspark.ml.classification.BinaryLogisticRegressionTrainingSummary at 0x7
fdcc0748450>
```

In [18]:
```
fit_Customermodel_Summary = fit_Customermodel.summary
```

In [19]:
```
fit_Customermodel_Summary.predictions.show()
```

```
+--------------------+-----+--------------------+--------------------+---
-------+
|            features|Churn|       rawPrediction|         probability|pre
diction|
+--------------------+-----+--------------------+--------------------+---
-------+
|[22.0,11254.38,1....|  0.0|[4.96050524236784...|[0.99303940400139...|
0.0|
|[27.0,8628.8,1.0,...|  0.0|[5.90597469051668...|[0.99728426899703...|
0.0|
|[28.0,8670.98,0.0...|  0.0|[8.19872388612018...|[0.99972507132363...|
0.0|
|[28.0,9090.43,1.0...|  0.0|[1.70655875854888...|[0.84638940648796...|
0.0|
|[28.0,11128.95,1....|  0.0|[4.43696894638154...|[0.98830660639038...|
0.0|
|[28.0,11204.23,0....|  0.0|[1.81406789469421...|[0.85985279779525...|
0.0|
|[28.0,11245.38,0....|  0.0|[3.69189769600253...|[0.97568147324479...|
0.0|
|[29.0,5900.78,1.0...|  0.0|[4.59258813963566...|[0.98997490494809...|
0.0|
|[29.0,8688.17,1.0...|  1.0|[2.96650688314298...|[0.95103787662661...|
0.0|
|[29.0,9378.24,0.0...|  0.0|[4.92859535749842...|[0.99281533194024...|
0.0|
|[29.0,10203.18,1....|  0.0|[4.01265512959848...|[0.98223595571025...|
0.0|
|[29.0,11274.46,1....|  0.0|[4.78549371666795...|[0.99171914494550...|
0.0|
|[29.0,13240.01,1....|  0.0|[7.03746026219900...|[0.99912241630757...|
0.0|
|[29.0,13255.05,1....|  0.0|[4.30395241860729...|[0.98666518428846...|
0.0|
|[30.0,6744.87,0.0...|  0.0|[3.71004655959537...|[0.97610839642025...|
0.0|
|[30.0,7960.64,1.0...|  1.0|[3.57503617850392...|[0.97274900827954...|
0.0|
|[30.0,8403.78,1.0...|  0.0|[6.44804766880066...|[0.99841889297270...|
0.0|
|[30.0,8677.28,1.0...|  0.0|[4.41655478245109...|[0.98806831982257...|
0.0|
|[30.0,8874.83,0.0...|  0.0|[3.23928343658757...|[0.96228611290728...|
0.0|
|[30.0,10183.98,1....|  0.0|[3.10250401931621...|[0.95699591531282...|
0.0|
+--------------------+-----+--------------------+--------------------+---
-------+
only showing top 20 rows
```

In [20]:
```
fit_Customermodel_Summary.predictions.describe().show()
```

```
+-------+-------------------+-------------------+
|summary|              Churn|         prediction|
+-------+-------------------+-------------------+
|  count|                624|                624|
|   mean|0.18269230769230768|0.14102564102564102|
| stddev| 0.3867240627102176|0.34832721924783666|
|    min|                0.0|                0.0|
|    max|                1.0|                1.0|
+-------+-------------------+-------------------+
```

In [22]:
```
results = fit_Customermodel.transform(Test_Customers_Data)
```

In [23]:
```
Customers_eval = BinaryClassificationEvaluator(rawPredictionCol='prediction
                                               labelCol='Churn')
```

In [24]:
```
results.select('Churn','prediction').show()
```

```
+-----+----------+
|Churn|prediction|
+-----+----------+
|    0|       0.0|
|    1|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
|    0|       0.0|
+-----+----------+
only showing top 20 rows
```

In [25]:
```
Acurate_Value = Customers_eval.evaluate(results)
```

In [26]:
```
Acurate_Value
```

Out[26]:
```
0.7569444444444445
```

Prediction of New Dataset

In [28]:
```
New_dataset = "gs://11514794_bucket1/Training_Data/new_customers.csv"
New_traindata = spark.read.csv(New_dataset,inferSchema = True,header = True
New_traindata.describe().show()
```

```
+-------+-------------+------------------+-----------------+-------------
-----+-----------------+------------------+--------------------+---------
-------+
|summary|        Names|               Age|   Total_Purchase|   Account_Ma
nager|            Years|         Num_Sites|            Location|
Company|
+-------+-------------+------------------+-----------------+-------------
-----+-----------------+------------------+--------------------+---------
-------+
|  count|            6|                 6|                6|
6|                6|                 6|                   6|
6|
|   mean|         null|35.166666666666664|7607.156666666667|0.83333333333
33334|6.808333333333334|12.333333333333334|                null|
null|
| stddev|         null| 15.71517313511584|4346.008232825459| 0.4082482904
63863|3.708737880555414|3.3862466931200785|                null|
null|
|    min|Andrew Mccall|              22.0|            100.0|
0|              1.0|               8.0|085 Austin Views ...|Barron-Robert
son|
|    max| Taylor Young|              65.0|         13147.71|
1|             10.0|              15.0|Unit 0789 Box 073...|        Wood
LLC|
+-------+-------------+------------------+-----------------+-------------
-----+-----------------+------------------+--------------------+---------
-------+
```

In [29]:
```
Final_Model = log_reg_Customers.fit(final_output)
```

In [30]:
```
New_traindata.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: timestamp (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
```

In [31]:
```
Customers_Valid = Assembler.transform(New_traindata)
```

In [32]:
```
Customers_Valid.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: timestamp (nullable = true)
 |-- Location: string (nullable = true)
 |-- Company: string (nullable = true)
 |-- features: vector (nullable = true)
```

In [33]:
```
result = Final_Model.transform(Customers_Valid)
```

In [37]:
```
result.select("Names","prediction").show()
```

```
+--------------+----------+
|         Names|prediction|
+--------------+----------+
| Andrew Mccall|       0.0|
|Michele Wright|       1.0|
|  Jeremy Chang|       1.0|
|Megan Ferguson|       1.0|
|  Taylor Young|       0.0|
| Jessica Drake|       1.0|
+--------------+----------+
```