In [2]:
```
import findspark
findspark.init()
```

In [3]:
```
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("empApp").getOrCreate()
```

In [4]:
```
spark
```

Out[4]:

SparkSession - in-memory

SparkContext

Spark UI

Version	`v3.2.1`
Master	`local[*]`
AppName	`empApp`

In [5]:
```
df=spark.read.option("header","true").csv('D:\employees.csv',inferSchema=True)
df.show()
```

```
+---------+------+----------+-------------+------+------+----------------+--------------------+
|FirstName|Gender| StartDate|LastLoginTime|Salary| Bonus|SeniorManagement|                Team|
+---------+------+----------+-------------+------+------+----------------+--------------------+
|  Douglas|  Male|  8/6/1993|     12:42 PM| 97308| 6.945|            true|           Marketing|
|   Thomas|  Male| 3/31/1996|      6:53 AM| 61933|  4.17|            true|                null|
|    Maria|Female| 4/23/1993|     11:17 AM|  null|11.858|           false|             Finance|
|    Jerry|  Male|  3/4/2005|      1:00 PM|138705|  null|            true|             Finance|
|    Larry|  Male| 1/24/1998|      4:47 PM|  null| 1.389|            true|     Client Services|
|   Dennis|  Male| 4/18/1987|      1:35 AM|115163|10.125|           false|               Legal|
|     Ruby|Female|      null|         null| 65476|10.012|            true|             Product|
|     null|  null| 7/20/2015|     10:43 AM|  null|  null|            null|             Finance|
|   Angela|Female|      null|      6:29 AM| 95570|  null|            true|         Engineering|
|  Frances|Female|      null|      6:51 AM|139852| 7.524|            true|Business Development|
|    Louise|Female| 8/12/1980|      9:01 AM| 63241|15.132|            true|                null|
|     Julie|Female|10/26/1997|      3:19 PM|102508|12.637|            true|               Legal|
|  Brandon|  Male| 12/1/1980|      1:08 AM|112807|17.492|            true|     Human Resources|
|     Gary|  Male| 1/27/2008|     11:40 PM|109831| 5.831|           false|               Sales|
| Kimberly|Female| 1/14/1999|      7:13 AM| 41426|14.543|            true|             Finance|
|  Lillian|Female|  6/5/2016|      6:09 AM| 59414| 1.256|           false|             Product|
|   Jeremy|  Male| 9/21/2010|      5:56 AM| 90370| 7.369|           false|     Human Resources|
|    Shawn|  Male| 12/7/1986|      7:45 PM|111737| 6.414|           false|             Product|
|    Diana|Female|10/23/1981|     10:27 AM|132940|19.082|           false|     Client Services|
|    Donna|Female| 7/22/2010|      3:48 AM| 81014| 1.894|           false|             Product|
+---------+------+----------+-------------+------+------+----------------+--------------------+
only showing top 20 rows
```

In [6]:
```
df.count()
```

Out[6]:
```
1000
```

In [7]:
```
df1=df.na.drop(how='any')
df1
```

Out[7]:
```
DataFrame[FirstName: string, Gender: string, StartDate: string, LastLoginTime: string, Salary: int, Bonus: double, SeniorManagement: boolean, Team: string]
```

In [8]:
```
df1.count()
```

Out[8]:
```
758
```

In [9]:
```
df1.printSchema()
```

```
root
 |-- FirstName: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- StartDate: string (nullable = true)
 |-- LastLoginTime: string (nullable = true)
 |-- Salary: integer (nullable = true)
 |-- Bonus: double (nullable = true)
 |-- SeniorManagement: boolean (nullable = true)
 |-- Team: string (nullable = true)
```

In [10]:
```
df1.select('SeniorManagement').distinct().show()
```

```
+----------------+
|SeniorManagement|
+----------------+
|            true|
|           false|
+----------------+
```

In [11]:
```
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StringIndexer
```

In [12]:
```
f=df1.withColumn('SeniorManagement',df1['SeniorManagement'].cast('string'))
f.printSchema()
```

```
root
 |-- FirstName: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- StartDate: string (nullable = true)
 |-- LastLoginTime: string (nullable = true)
 |-- Salary: integer (nullable = true)
 |-- Bonus: double (nullable = true)
 |-- SeniorManagement: string (nullable = true)
 |-- Team: string (nullable = true)
```

In [49]:
```
va=VectorAssembler(inputCols=['Salary','Bonus'],outputCol='Input Features')
indexer=StringIndexer(inputCol='SeniorManagement',outputCol='SMData')
df2=indexer.fit(f).transform(f)
df3=va.transform(df2)
df3.show()
```

```
+---------+------+----------+-------------+------+------+----------------+--------------------+------+--------------------+
|FirstName|Gender| StartDate|LastLoginTime|Salary| Bonus|SeniorManagement|                Team|SMData|      Input Features|
+---------+------+----------+-------------+------+------+----------------+--------------------+------+--------------------+
|  Douglas|  Male|  8/6/1993|     12:42 PM| 97308| 6.945|            true|           Marketing|   1.0|  [97308.0,6.945]|
|   Thomas|  Male| 3/31/1996|      6:53 AM| 61933|  4.17|            true|                null|   1.0|  [61933.0,4.17]|
|    Jerry|  Male|  3/4/2005|      1:00 PM|138705|  null|            true|             Finance|   1.0|[138705.0,12.637]|
|    Larry|  Male| 1/24/1998|      4:47 PM|  null| 1.389|            true|     Client Services|   1.0|  [1389.0,1.256]|
|   Dennis|  Male| 4/18/1987|      1:35 AM|115163|10.125|           false|               Legal|   0.0|[115163.0,10.125]|
|     Ruby|Female|      null|         null| 65476|10.012|            true|             Product|   1.0|[65476.0,10.012]|
|     null|  null| 7/20/2015|     10:43 AM|  null|  null|            null|             Finance|   1.0|    [1.0,1.0]|
|   Angela|Female|      null|      6:29 AM| 95570|  null|            true|         Engineering|   1.0|  [95570.0,7.369]|
|  Frances|Female|      null|      6:51 AM|139852| 7.524|            true|Business Development|   0.0|[139852.0,16.18]|
|    Louise|Female| 8/12/1980|      9:01 AM| 63241|15.132|            true|                null|   1.0|  [63241.0,15.132]|
|     Julie|Female|10/26/1997|      3:19 PM|102508|12.637|            true|               Legal|   1.0|[102508.0,12.637]|
|  Brandon|  Male| 12/1/1980|      1:08 AM|112807|17.492|            true|     Human Resources|   1.0|[112807.0,17.492]|
|     Gary|  Male| 1/27/2008|     11:40 PM|109831| 5.831|           false|               Sales|   0.0|[109831.0,5.831]|
| Kimberly|Female| 1/14/1999|      7:13 AM| 41426|14.543|            true|             Finance|   1.0|[41426.0,14.543]|
|  Lillian|Female|  6/5/2016|      6:09 AM| 59414| 1.256|           false|             Product|   0.0|  [59414.0,1.256]|
|   Jeremy|  Male| 9/21/2010|      5:56 AM| 90370| 7.369|           false|     Human Resources|   0.0|  [90370.0,7.369]|
|    Shawn|  Male| 12/7/1986|      7:45 PM|111737| 6.414|           false|             Product|   0.0|  [111737.0,6.414]|
|    Diana|Female|10/23/1981|     10:27 AM|132940|19.082|           false|     Client Services|   0.0|[132940.0,19.082]|
|    Donna|Female| 7/22/2010|      3:48 AM| 81014| 1.894|           false|             Product|   0.0|  [81014.0,1.894]|
|  Matthew|  Male|  9/5/1995|      2:12 AM|100612|13.645|           false|           Marketing|   0.0|[100612.0,13.645]|
|     John|  Male|  7/1/1992|     10:08 PM| 97950|13.873|           false|     Client Services|   0.0|[97950.0,13.873]|
|    Craig|  Male| 2/27/2000|      7:45 AM| 37598| 7.757|            true|           Marketing|   1.0|  [37598.0,7.757]|
|    Terry|  Male|11/27/1981|      6:30 PM|124008|13.464|            true|     Client Services|   1.0|[124008.0,13.464]|
| Benjamin|  Male| 1/26/2005|     10:06 PM| 79529| 7.008|            true|               Legal|   1.0|  [79529.0,7.008]|
|Christina|Female|  8/6/2002|      1:19 PM|118780| 9.096|            true|         Engineering|   1.0|  [118780.0,9.096]|
|    Sean|Female|12/18/1993|      9:07 AM|119082| 16.18|           false|Business Development|   0.0|[119082.0,16.18]|
|    Jerry|  Male| 1/10/2004|     12:56 PM| 95734|19.096|           false|     Client Services|   0.0|[95734.0,19.096]|
|  Theresa|Female|10/10/2006|      1:12 AM| 85182|16.675|           false|               Sales|   0.0|[85182.0,16.675]|
+---------+------+----------+-------------+------+------+----------------+--------------------+------+--------------------+
only showing top 20 rows
```

In [58]:
```
finaldata=df3.select('Input Features','SMData')
finaldata.show()
```

```
+--------------------+------+
|   Input Features|SMData|
+--------------------+------+
|  [97308.0,6.945]|   1.0|
|[115163.0,10.125]|   0.0|
|[102508.0,12.637]|   1.0|
|[112807.0,17.492]|   1.0|
| [109831.0,5.831]|   0.0|
| [41426.0,14.543]|   1.0|
|  [59414.0,1.256]|   0.0|
|  [90370.0,7.369]|   0.0|
| [111737.0,6.414]|   0.0|
| [81014.0,1.894]|   0.0|
|[100612.0,13.645]|   0.0|
|[97950.0,13.873]|   0.0|
|  [37598.0,7.757]|   1.0|
|[124008.0,13.464]|   1.0|
|  [79529.0,7.008]|   1.0|
|[118780.0,9.096]|   1.0|
|[119082.0,16.18]|   0.0|
|[95734.0,19.096]|   0.0|
|[85182.0,16.675]|   0.0|
+--------------------+------+
only showing top 20 rows
```

In [51]:
```
train,test=finaldata.randomSplit([0.70,0.30])
```

In [52]:
```
from pyspark.ml.classification import DecisionTreeClassifier
```

In [53]:
```
dtcmodel=DecisionTreeClassifier(labelCol='SMData',featuresCol='Input Features')
model=dtcmodel.fit(train)
```

In [54]:
```
model
```

Out[54]:
```
DecisionTreeClassificationModel: uid=DecisionTreeClassifier_a5aa92e9b100, depth=5, numNodes=33, numClasses=2, numFeatures=2
```

In [55]:
```
prediction_res=model.transform(test)
prediction_res.show()
```

```
+--------------------+------+-------------+--------------------+----------+
|  Input Features|SMData|rawPrediction|         probability|prediction|
+--------------------+------+-------------+--------------------+----------+
| [35095.0,8.379]|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|[35381.0,11.749]|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|[35884.0,17.667]|   0.0|   [1.0,0.0]|           [1.0,0.0]|       0.0|
|[36749.0,19.754]|   0.0|   [4.0,0.0]|           [1.0,0.0]|       0.0|
| [36946.0,6.652]|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
| [37259.0,1.763]|   0.0|   [0.0,3.0]|           [0.0,1.0]|       1.0|
| [38872.0,9.302]|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|[39335.0,10.664]|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
| [39371.0,4.068]|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
| [39833.0,9.631]|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
| [40121.0,6.293]|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|[40837.0,12.182]|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
| [41190.0,3.311]|   1.0|[79.0,92.0]|[0.4619883040935...|       1.0|
| [41427.0,1.431]|   0.0|   [0.0,3.0]|           [0.0,1.0]|       1.0|
| [41643.0,4.659]|   1.0|[79.0,92.0]|[0.4619883040935...|       1.0|
| [41831.0,4.548]|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
| [42090.0,6.842]|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
| [42553.0,3.756]|   1.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|[43050.0,11.671]|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|[44486.0,17.308]|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
+--------------------+------+-------------+--------------------+----------+
only showing top 20 rows
```

In [57]:
```
prediction_res.select('SMData','rawPrediction','probability','prediction').show()
```

```
+------+-------------+--------------------+----------+
|SMData|rawPrediction|         probability|prediction|
+------+-------------+--------------------+----------+
|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   0.0|   [1.0,0.0]|           [1.0,0.0]|       0.0|
|   0.0|   [4.0,0.0]|           [1.0,0.0]|       0.0|
|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   0.0|   [0.0,3.0]|           [0.0,1.0]|       1.0|
|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   1.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   0.0|   [0.0,3.0]|           [0.0,1.0]|       1.0|
|   1.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   0.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   1.0|[79.0,92.0]|[0.4619883040935...|       1.0|
|   0.0|[134.0,113.0]|[0.5425101214574...|       0.0|
|   1.0|[134.0,113.0]|[0.5425101214574...|       0.0|
+------+-------------+--------------------+----------+
only showing top 20 rows
```

In [60]:
```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

In [64]:
```
evaluator=MulticlassClassificationEvaluator(labelCol='SMData',predictionCol='prediction')
accuracy=evaluator.evaluate(prediction_res)
print("Accuracy of model ",accuracy)
print("Error of model ",(1-accuracy))
```

```
Accuracy of model  0.5213648824123502
Error of model  0.47863511758764976
```

In [ ]:
```
```