KP02-Copy3

November 16, 2021

[2]:
```
!pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/lib/spark/python (3.1.2)
Requirement already satisfied: py4j==0.10.9 in
/opt/conda/miniconda3/lib/python3.8/site-packages (from pyspark) (0.10.9)
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommended to
use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

[3]:
```
import os
import pandas as pd
import numpy as np

from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession, SQLContext

import pyspark.sql.functions as F
from pyspark.sql.functions import udf, col

from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
from pyspark.ml.evaluation import BinaryClassificationEvaluator

from pyspark.sql.functions import udf, col


from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.ml.tuning import ParamGridBuilder, CrossValidator,␣
↪CrossValidatorModel
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml.evaluation import RegressionEvaluator

from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
from pyspark.ml.evaluation import BinaryClassificationEvaluator

from pyspark.ml import Pipeline
```

[4]:
```
from pyspark.sql import SparkSession
spark = SparkSession \
    .builder \
    .getOrCreate()
```

[5]:
```
df = spark.read.format('csv').option("header","true").option("delimiter",",").
↪load("gs://bdamlproject/TurnoverIntentionatBenzInfotech.csv")
```

[6]:
```
for col in df.columns:
            print(col)
```

```
Tenure
PerfRating
JobSatisfaction
Engagement
OnsiteOpportunity
Awards
Flexihours
Intention
```

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

[6]:
```
361
```

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

[7]:
```
[('Tenure', 'string'),
 ('PerfRating', 'string'),
 ('JobSatisfaction', 'string'),
 ('Engagement', 'string'),
 ('OnsiteOpportunity', 'string'),
 ('Awards', 'string'),
 ('Flexihours', 'string'),
 ('Intention', 'string')]
```

[8]:
```
df1=df.toPandas()
```

[9]:
```
df1
```

[9]:

	Tenure	PerfRating	JobSatisfaction	Engagement	OnsiteOpportunity	Awards	\
0	2	4	4	1	0	1	
1	1	1	3	1	0	1	
2	1	1	3	2	1	0	
3	1	1	2	4	0	1	
4	1	1	3	1	0	1	

```
..      …         …              …         …                 …    …
356      4         3              3         3                 0    1
357      2         2              2         3                 1    0
358      4         3              2         1                 1    0
359      5         1              5         5                 1    0
360      2         1              1         1                 1    1

    Flexihours Intention
0            0         1
1            1         1
2            1         1
3            1         1
4            1         1
..         …       …
356          1         1
357          1         1
358          1         0
359          0         0
360          1         1

[361 rows x 8 columns]
```

[10]:
```
df1.describe()
```

[10]:
```
        Tenure PerfRating JobSatisfaction Engagement OnsiteOpportunity Awards  \
count      361        361              361        361                361    361
unique       8          5                5          5                  2      2
top          2          1                4          4                  1      0
freq        61         90               85         86                194    182

       Flexihours Intention
count         361       361
unique          2         2
top             0         0
freq          196       181
```

[11]:
```
df1.isna().sum()
```

[11]:
```
Tenure              0
PerfRating          0
JobSatisfaction     0
Engagement          0
OnsiteOpportunity   0
Awards              0
Flexihours          0
Intention           0
dtype: int64
```

[1]:
```
df= df.withColumn("Tenure",df["Tenure"].cast(IntegerType()))
df= df.withColumn("PerfRating",df["PerfRating"].cast(IntegerType()))
df= df.withColumn("JobSatisfaction",df["JobSatisfaction"].cast(IntegerType()))
df= df.withColumn("Engagement",df["Engagement"].cast(IntegerType()))
df= df.withColumn("OnsiteOpportunity",df["OnsiteOpportunity"].
 ↪cast(IntegerType()))
df= df.withColumn("Awards",df["Awards"].cast(IntegerType()))
df= df.withColumn("Flexihours",df["Flexihours"].cast(IntegerType()))
df= df.withColumn("Intention",df["Intention"].cast(IntegerType()))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_8947/2029380153.py in <module>
----> 1 df= df.withColumn("Tenure",df["Tenure"].cast(IntegerType()))
      2 df= df.withColumn("PerfRating",df["PerfRating"].cast(IntegerType()))
      3 df= df.withColumn("JobSatisfaction",df["JobSatisfaction"].
 ↪cast(IntegerType()))
      4 df= df.withColumn("Engagement",df["Engagement"].cast(IntegerType()))
      5 df= df.withColumn("OnsiteOpportunity",df["OnsiteOpportunity"].
 ↪cast(IntegerType()))

NameError: name 'df' is not defined
```

[ ]:
```
from pyspark.ml.stat import Correlation
from pyspark.ml.feature import VectorAssembler
```

[43]:
```
from pyspark.sql.types import IntegerType
```

[44]:
```
from pyspark.sql.functions import when
from pyspark.sql.functions import lit
```

[45]:
```
df.columns
```

[45]:
```
['Tenure',
 'PerfRating',
 'JobSatisfaction',
 'Engagement',
 'OnsiteOpportunity',
 'Awards',
 'Flexihours',
 'Intention']
```

[46]:
```
features = ['Tenure',
 'PerfRating',
 'JobSatisfaction',
 'Engagement',
```

```
 'OnsiteOpportunity',
 'Awards',
 'Flexihours']
from pyspark.ml.feature import VectorAssembler, StringIndexer, VectorIndexer,␣
 ↪MinMaxScaler,Normalizer

from pyspark.ml.classification import LogisticRegression

Vect = VectorAssembler(inputCols=['Tenure',
 'PerfRating',
 'JobSatisfaction',
 'Engagement',
 'OnsiteOpportunity',
 'Awards',
 'Flexihours'], outputCol="features")
```

[47]:
```
catIdx = VectorIndexer(inputCol = Vect.getOutputCol(), outputCol =␣
 ↪"idxCatFeatures")
```

[48]:
```
train, test = df.randomSplit([0.8,0.2])
```

[49]:
```
train.count()
```

[49]: 278

[50]:
```
test.count()
```

[50]: 83

[51]:
```
lr =␣
 ↪LogisticRegression(labelCol='Intention',featuresCol="features",maxIter=10,regParam=0.
 ↪3)
```

[52]:
```
pipeline1 = Pipeline(stages=[Vect, catIdx, lr])
```

[53]:
```
pipelineModel = pipeline1.fit(train)
```

```
21/11/13 01:19:35 WARN com.github.fommil.netlib.BLAS: Failed to load
implementation from: com.github.fommil.netlib.NativeSystemBLAS
21/11/13 01:19:35 WARN com.github.fommil.netlib.BLAS: Failed to load
implementation from: com.github.fommil.netlib.NativeRefBLAS
```

[54]:
```
prediction1 = pipelineModel.transform(test)
```

[56]:
```
predicted=prediction1.select("Intention","prediction")
predicted.show(100, truncate=False)
```

Intention	prediction
1	1.0
1	1.0
1	1.0
0	1.0
1	1.0
0	0.0
1	1.0
0	1.0
0	1.0
1	1.0
1	1.0
1	1.0
1	1.0
1	1.0
1	1.0
1	1.0
1	1.0
0	1.0
1	0.0
1	0.0
0	0.0
1	0.0
1	1.0
1	1.0
0	1.0
1	1.0
0	1.0
1	1.0
1	1.0
1	1.0
1	1.0
1	1.0
1	0.0
1	1.0
1	1.0
1	1.0
0	1.0
1	1.0
1	1.0
1	1.0
0	0.0
0	0.0
1	1.0
0	0.0
0	0.0

```
|0         |1.0        |
|1         |1.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|1         |1.0        |
|0         |0.0        |
|1         |1.0        |
|0         |0.0        |
|0         |0.0        |
|0         |1.0        |
|0         |0.0        |
|0         |0.0        |
|0         |1.0        |
|1         |1.0        |
|0         |0.0        |
|1         |0.0        |
|1         |1.0        |
|1         |0.0        |
|0         |0.0        |
|0         |0.0        |
|1         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|1         |0.0        |
|0         |0.0        |
|1         |1.0        |
|1         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
|0         |0.0        |
+----------+-----------+
```

[58]:
```
tp = float(predicted.filter("prediction == 1.0 AND Intention== 1").count())
fp = float(predicted.filter("prediction == 1.0 AND Intention == 0").count())
tn = float(predicted.filter("prediction == 0.0 AND Intention== 0").count())
fn = float(predicted.filter("prediction == 0.0 AND Intention == 1").count())
pr = tp / (tp + fp)
re = tp / (tp + fn)
```

```
metrics = spark.createDataFrame([
 ("TP", tp),
 ("FP", fp),
 ("TN", tn),
 ("FN", fn),
 ("Precision", pr),
 ("Recall", re),
 ("F1", 2*pr*re/(re+pr))],["metric", "value"])
metrics.show()
```

```
[Stage 46:>                                                         (0 + 1) / 1]
```

```
+---------+------------------+
|   metric|             value|
+---------+------------------+
|       TP|              34.0|
|       FP|              10.0|
|       TN|              30.0|
|       FN|               9.0|
|Precision|0.7727272727272727|
|   Recall|0.7906976744186046|
|       F1|0.7816091954022988|
+---------+------------------+
```

[59]:
```
evaluator = BinaryClassificationEvaluator(labelCol="AttritionMod",␣
 ↪rawPredictionCol="rawPrediction", metricName="areaUnderROC")
aur = evaluator.evaluate(prediction)
print ("AUR = ", aur)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_5439/3420554418.py in <module>
      1 evaluator = BinaryClassificationEvaluator(labelCol="AttritionMod",␣
 ↪rawPredictionCol="rawPrediction", metricName="areaUnderROC")
----> 2 aur = evaluator.evaluate(prediction)
      3 print ("AUR = ", aur)

NameError: name 'prediction' is not defined
```

[61]:
```
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=featureCols, outputCol="features")

assembled_df = assembler.transform(df_final)
assembled_df.show(10, truncate=False)
```

```
+---+---------------+---------------------+-------------+--------+---------
-----+-------------+-----------------+----------------+-----------------+-----
-----------+--------------------+--------------+-------------+-----------+-
--------------+---------+-----------------------------------------------------
------------------+
|Age|DistanceFromHome|EnvironmentSatisfaction|JobInvolvement|JobLevel|JobSatisfa
ction|MonthlyIncome|NumCompaniesWorked|PercentSalaryHike|PerformanceRating|Total
WorkingYears|TrainingTimesLastYear|WorkLifeBalance|YearsAtCompany|AttritionMod|M
aritalStatusMod|GenderMod|features
|
+---+---------------+---------------------+-------------+--------+---------
-----+-------------+-----------------+----------------+-----------------+-----
-----------+--------------------+--------------+-------------+-----------+-
--------------+---------+-----------------------------------------------------
------------------+
|41 |1              |2                    |3            |2       |4
|5993         |8                |11               |3               |8
|0                    |1             |6              |1           |0
|0
|[41.0,1.0,2.0,3.0,2.0,4.0,5993.0,8.0,11.0,3.0,8.0,0.0,1.0,6.0,0.0,0.0]  |
|49 |8              |3                    |2            |2       |2
|5130         |1                |23               |4               |10
|3                    |3             |10             |0           |0
|1
|[49.0,8.0,3.0,2.0,2.0,2.0,5130.0,1.0,23.0,4.0,10.0,3.0,3.0,10.0,0.0,1.0]|
|37 |2              |4                    |2            |1       |3
|2090         |6                |15               |3               |7
|3                    |3             |0              |1           |0
|1
|[37.0,2.0,4.0,2.0,1.0,3.0,2090.0,6.0,15.0,3.0,7.0,3.0,3.0,0.0,0.0,1.0]  |
|33 |3              |4                    |3            |1       |3
|2909         |1                |11               |3               |8
|3                    |3             |8              |0           |0
|0
|[33.0,3.0,4.0,3.0,1.0,3.0,2909.0,1.0,11.0,3.0,8.0,3.0,3.0,8.0,0.0,0.0]  |
|27 |2              |1                    |3            |1       |2
|3468         |9                |12               |3               |6
|3                    |3             |2              |0           |0
|1
|[27.0,2.0,1.0,3.0,1.0,2.0,3468.0,9.0,12.0,3.0,6.0,3.0,3.0,2.0,0.0,1.0]  |
|32 |2              |4                    |3            |1       |4
|3068         |0                |13               |3               |8
|2                    |2             |7              |0           |0
|1
|[32.0,2.0,4.0,3.0,1.0,4.0,3068.0,0.0,13.0,3.0,8.0,2.0,2.0,7.0,0.0,1.0]  |
|59 |3              |3                    |4            |1       |1
|2670         |4                |20               |4               |12
|3                    |2             |1              |0           |0
```

```
|0
|[59.0,3.0,3.0,4.0,1.0,1.0,2670.0,4.0,20.0,4.0,12.0,3.0,2.0,1.0,0.0,0.0] |
|30 |24              |4                      |3             |1       |3
|2693         |1                 |22               |4                |1
|2                    |3              |1              |0           |0
|1
|[30.0,24.0,4.0,3.0,1.0,3.0,2693.0,1.0,22.0,4.0,1.0,2.0,3.0,1.0,0.0,1.0] |
|38 |23              |4                      |2             |3       |3
|9526         |0                 |21               |4                |10
|2                    |3              |9              |0           |0
|1
|[38.0,23.0,4.0,2.0,3.0,3.0,9526.0,0.0,21.0,4.0,10.0,2.0,3.0,9.0,0.0,1.0]|
|36 |27              |3                      |3             |2       |3
|5237         |6                 |13               |3                |17
|3                    |2              |7              |0           |0
|1
|[36.0,27.0,3.0,3.0,2.0,3.0,5237.0,6.0,13.0,3.0,17.0,3.0,2.0,7.0,0.0,1.0]|
+---+----------------+----------------------+-------------+--------+----------
-----+-------------+-----------------+----------------+-----------------+-----
-----------+---------------------+--------------+-------------+-----------+-
---------------+---------+-----------------------------------------------------
------------------+
only showing top 10 rows
```

[62]:
```
normalizer = Normalizer(inputCol="features",outputCol="features_norm")
```

[63]:
```
lr =␣
↪LogisticRegression(featuresCol="features_norm",labelCol="AttritionMod",maxIter=10,regParam=
↪3,elasticNetParam=0.2)
```

[64]:
```
pipeline = Pipeline(stages=[assembler,normalizer,lr])
```

[65]:
```
piplineModel = pipeline.fit(train)
```

[66]:
```
prediction = piplineModel.transform(test)
```

[67]:
```
prediction.select("AttritionMod","prediction").show(100)
```

```
+------------+----------+
|AttritionMod|prediction|
+------------+----------+
|           0|       0.0|
|           1|       0.0|
|           1|       0.0|
|           1|       0.0|
|           0|       0.0|
|           0|       0.0|
```

0	0.0
0	0.0
1	0.0
1	0.0
0	0.0
1	0.0
1	0.0
0	0.0
0	0.0
1	0.0
0	0.0
0	0.0
1	0.0
1	0.0
0	0.0
0	0.0
1	0.0
0	0.0
0	0.0
0	0.0
0	0.0
0	0.0
1	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
1	0.0
0	0.0
1	0.0
0	0.0
0	0.0
0	0.0
0	0.0
0	0.0
0	0.0
1	0.0
0	0.0

```
|            0|       0.0|
|            0|       0.0|
|            1|       0.0|
|            0|       0.0|
|            0|       0.0|
|            0|       0.0|
|            0|       0.0|
|            1|       0.0|
|            1|       0.0|
|            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|
|            1|       0.0|
|            1|       0.0|
|            1|       0.0|
|            0|       0.0|
|            0|       0.0|
|            1|       0.0|
|            0|       0.0|
|            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|
+-------------+----------+
only showing top 100 rows
```

[68]:
```
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=featureCols, outputCol="features")

assembled_df = assembler.transform(df_final)
assembled_df.show(10, truncate=False)
```

```
+---+---------------+---------------------+-------------+--------+----------
-----+-------------+-----------------+----------------+-----------------+-----
-----------+--------------------+--------------+-------------+-----------+-
--------------+---------+-----------------------------------------------------
------------------+
|Age|DistanceFromHome|EnvironmentSatisfaction|JobInvolvement|JobLevel|JobSatisfa
ction|MonthlyIncome|NumCompaniesWorked|PercentSalaryHike|PerformanceRating|Total
WorkingYears|TrainingTimesLastYear|WorkLifeBalance|YearsAtCompany|AttritionMod|M
aritalStatusMod|GenderMod|features
|
+---+---------------+---------------------+-------------+--------+----------
-----+-------------+-----------------+----------------+-----------------+-----
-----------+--------------------+--------------+-------------+-----------+-
--------------+---------+-----------------------------------------------------
------------------+
|41 |1              |2                    |3            |2       |4
|5993         |8                |11               |3                |8
|0                    |1             |6              |1           |0
|0
|[41.0,1.0,2.0,3.0,2.0,4.0,5993.0,8.0,11.0,3.0,8.0,0.0,1.0,6.0,0.0,0.0]  |
|49 |8              |3                    |2            |2       |2
|5130         |1                |23               |4                |10
|3                    |3             |10             |0           |0
|1
|[49.0,8.0,3.0,2.0,2.0,2.0,5130.0,1.0,23.0,4.0,10.0,3.0,3.0,10.0,0.0,1.0]|
|37 |2              |4                    |2            |1       |3
|2090         |6                |15               |3                |7
|3                    |3             |0              |1           |0
|1
|[37.0,2.0,4.0,2.0,1.0,3.0,2090.0,6.0,15.0,3.0,7.0,3.0,3.0,0.0,0.0,1.0]  |
|33 |3              |4                    |3            |1       |3
|2909         |1                |11               |3                |8
|3                    |3             |8              |0           |0
|0
|[33.0,3.0,4.0,3.0,1.0,3.0,2909.0,1.0,11.0,3.0,8.0,3.0,3.0,8.0,0.0,0.0]  |
|27 |2              |1                    |3            |1       |2
|3468         |9                |12               |3                |6
|3                    |3             |2              |0           |0
|1
|[27.0,2.0,1.0,3.0,1.0,2.0,3468.0,9.0,12.0,3.0,6.0,3.0,3.0,2.0,0.0,1.0]  |
```

```
|32 |2                |4                        |3              |1        |4
|3068          |0                  |13                 |3                  |8
|2                    |2               |7               |0            |0
|1
|[32.0,2.0,4.0,3.0,1.0,4.0,3068.0,0.0,13.0,3.0,8.0,2.0,2.0,7.0,0.0,1.0]  |
|59 |3                |3                        |4              |1        |1
|2670          |4                  |20                 |4                  |12
|3                    |2               |1               |0            |0
|0
|[59.0,3.0,3.0,4.0,1.0,1.0,2670.0,4.0,20.0,4.0,12.0,3.0,2.0,1.0,0.0,0.0] |
|30 |24               |4                        |3              |1        |3
|2693          |1                  |22                 |4                  |1
|2                    |3               |1               |0            |0
|1
|[30.0,24.0,4.0,3.0,1.0,3.0,2693.0,1.0,22.0,4.0,1.0,2.0,3.0,1.0,0.0,1.0] |
|38 |23               |4                        |2              |3        |3
|9526          |0                  |21                 |4                  |10
|2                    |3               |9               |0            |0
|1
|[38.0,23.0,4.0,2.0,3.0,3.0,9526.0,0.0,21.0,4.0,10.0,2.0,3.0,9.0,0.0,1.0]|
|36 |27               |3                        |3              |2        |3
|5237          |6                  |13                 |3                  |17
|3                    |2               |7               |0            |0
|1
|[36.0,27.0,3.0,3.0,2.0,3.0,5237.0,6.0,13.0,3.0,17.0,3.0,2.0,7.0,0.0,1.0]|
+---+----------------+----------------------+--------------+--------+----------
-----+-------------+-----------------+----------------+-----------------+-----
-----------+--------------------+---------------+--------------+------------+-
---------------+---------+-----------------------------------------------------
------------------+
only showing top 10 rows
```

[69]:
```
standardScaler = StandardScaler(inputCol="features",␣
 ↪outputCol="features_scaled")
scaled_df = standardScaler.fit(assembled_df).transform(assembled_df)

scaled_df.select("features", "features_scaled").show(10, truncate=False)
train_data, test_data = scaled_df.randomSplit([0.8,0.2])
```

```
+-------------------------------------------------------------------------+------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------+
|features
|features_scaled
|
```

```
+--------------------------------------------------------------------------+------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
-----------------------------------------+
|[41.0,1.0,2.0,3.0,2.0,4.0,5993.0,8.0,11.0,3.0,8.0,0.0,1.0,6.0,0.0,0.0]  |[4.488
048578282528,0.12335225387518614,1.8296885387233528,4.216081821875181,1.80678282
70748993,3.6269792461350723,1.2729513621525657,3.2025505034441926,3.005515626752
296,8.31431377233402,1.0281743317865935,0.0,1.4154765923196735,0.979347974707337
7,0.0,0.0]                                |
|[49.0,8.0,3.0,2.0,2.0,2.0,5130.0,1.0,23.0,4.0,10.0,3.0,3.0,10.0,0.0,1.0]|[5.363
765374044972,0.9868180310014891,2.744532808085029,2.810721214583454,1.8067828270
748993,1.8134896230675361,1.089644666751654,0.40031881293052407,6.28425994684571
,11.085751696445358,1.2852179147332419,2.3268970467566765,4.246429776959021,1.63
22466245122293,0.0,2.0405470343872216]      |
|[37.0,2.0,4.0,2.0,1.0,3.0,2090.0,6.0,15.0,3.0,7.0,3.0,3.0,0.0,0.0,1.0]  |[4.050
190180401305,0.24670450775037228,3.6593770774467056,2.810721214583454,0.90339141
35374497,2.7202344346013043,0.4439293086765997,2.401912877583144,4.0984304001167
68,8.31431377233402,0.8996525403132692,2.3268970467566765,4.246429776959021,0.0,
0.0,2.0405470343872216]                  |
|[33.0,3.0,4.0,3.0,1.0,3.0,2909.0,1.0,11.0,3.0,8.0,3.0,3.0,8.0,0.0,0.0]  |[3.612
331782520083,0.37005676162555845,3.6593770774467056,4.216081821875181,0.90339141
35374497,2.7202344346013043,0.6178901238948462,0.40031881293052407,3.00551562675
2296,8.31431377233402,1.0281743317865935,2.3268970467566765,4.246429776959021,1.
3057972996097835,0.0,0.0]                |
|[27.0,2.0,1.0,3.0,1.0,2.0,3468.0,9.0,12.0,3.0,6.0,3.0,3.0,2.0,0.0,1.0]  |[2.955
54418569825,0.24670450775037228,0.9148442693616764,4.216081821875181,0.903391413
5374497,1.8134896230675361,0.7366252834882525,3.6028693163747167,3.2787443200934
137,8.31431377233402,0.771130748839945,2.3268970467566765,4.246429776959021,0.32
64493249024459,0.0,2.0405470343872216]      |
|[32.0,2.0,4.0,3.0,1.0,4.0,3068.0,0.0,13.0,3.0,8.0,2.0,2.0,7.0,0.0,1.0]  |[3.502
8671830497777,0.24670450775037228,3.6593770774467056,4.216081821875181,0.9033914
135374497,3.6269792461350723,0.6516627363731139,0.0,3.5519730134345315,8.3143137
7233402,1.0281743317865935,1.5512646978377842,2.830953184639347,1.14257263715856
05,0.0,2.0405470343872216]               |
|[59.0,3.0,3.0,4.0,1.0,1.0,2670.0,4.0,20.0,4.0,12.0,3.0,2.0,1.0,0.0,0.0] |[6.458
411368748028,0.37005676162555845,2.744532808085029,5.621442429166908,0.903391413
5374497,0.9067448115337681,0.5671250019935509,1.6012752517220963,5.4645738668223
57,11.085751696445358,1.54226149767989,2.3268970467566765,2.830953184639347,0.16
322466245122294,0.0,0.0]                 |
|[30.0,24.0,4.0,3.0,1.0,3.0,2693.0,1.0,22.0,4.0,1.0,2.0,3.0,1.0,0.0,1.0] |[3.283
9379841091665,2.9604540930044676,3.6593770774467056,4.216081821875181,0.90339141
35374497,2.7202344346013043,0.5720103484526713,0.40031881293052407,6.01103125350
4592,11.085751696445358,0.12852179147332418,1.5512646978377842,4.246429776959021
,0.16322466245122294,0.0,2.0405470343872216]|
|[38.0,23.0,4.0,2.0,3.0,3.0,9526.0,0.0,21.0,4.0,10.0,2.0,3.0,9.0,0.0,1.0]|[4.159
6547798716115,2.8371018391292813,3.6593770774467056,2.810721214583454,2.71017424
0612349,2.7202344346013043,2.023383059547028,0.0,5.737802560163474,11.0857516964
```

```
45358,1.2852179147332419,1.5512646978377842,4.246429776959021,1.4690219620610065
,0.0,2.0405470343872216]                    |
|[36.0,27.0,3.0,3.0,2.0,3.0,5237.0,6.0,13.0,3.0,17.0,3.0,2.0,7.0,0.0,1.0]|[3.940
725580931,3.3305108546300257,2.744532808085029,4.216081821875181,1.8067828270748
993,2.7202344346013043,1.1123721481049536,2.401912877583144,3.5519730134345315,8
.31431377233402,2.1848704550465112,2.3268970467566765,2.830953184639347,1.142572
6371585605,0.0,2.0405470343872216]          |
+------------------------------------------------------------------------+------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
-------------------------------------------+
only showing top 10 rows
```

[70]:
```
lr =␣
 ↪LogisticRegression(featuresCol="features_scaled",labelCol="AttritionMod",maxIter=10,regPara
 ↪3,elasticNetParam=0.2)

linearModel = lr.fit(train_data)
predictions = linearModel.transform(test_data)

predictions.select("AttritionMod","prediction").show(100)
```

```
+------------+----------+
|AttritionMod|prediction|
+------------+----------+
|           1|       0.0|
|           1|       0.0|
|           0|       0.0|
|           0|       0.0|
|           0|       0.0|
|           1|       0.0|
|           0|       0.0|
|           1|       0.0|
|           0|       0.0|
|           0|       0.0|
|           0|       0.0|
|           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|
|           1|       0.0|
```

0	0.0
1	0.0
0	0.0
1	0.0
0	0.0
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
1	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
1	0.0
0	0.0
0	0.0
0	0.0
1	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
1	0.0
0	0.0
0	0.0

```
|            0|       0.0|
|            0|       0.0|
|            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|
|            1|       0.0|
|            0|       0.0|
|            1|       0.0|
|            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|
+-------------+----------+
only showing top 100 rows
```

[ ]: