KP02-Copy1

November 14, 2021

[1]:
```
!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
```

[2]:
```
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
```

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

```
[4]: df = spark.read.format('csv').option("header","true").option("delimiter",",").
      ↪load("gs://bdamlproject/WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

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

```
Age
Attrition
BusinessTravel
DailyRate
Department
DistanceFromHome
Education
EducationField
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
Gender
HourlyRate
JobInvolvement
JobLevel
JobRole
JobSatisfaction
MaritalStatus
MonthlyIncome
MonthlyRate
NumCompaniesWorked
Over18
OverTime
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
```

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

[6]:
```
1470
```

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

[7]:
```
[('Age', 'string'),
 ('Attrition', 'string'),
 ('BusinessTravel', 'string'),
 ('DailyRate', 'string'),
 ('Department', 'string'),
 ('DistanceFromHome', 'string'),
 ('Education', 'string'),
 ('EducationField', 'string'),
 ('EmployeeCount', 'string'),
 ('EmployeeNumber', 'string'),
 ('EnvironmentSatisfaction', 'string'),
 ('Gender', 'string'),
 ('HourlyRate', 'string'),
 ('JobInvolvement', 'string'),
 ('JobLevel', 'string'),
 ('JobRole', 'string'),
 ('JobSatisfaction', 'string'),
 ('MaritalStatus', 'string'),
 ('MonthlyIncome', 'string'),
 ('MonthlyRate', 'string'),
 ('NumCompaniesWorked', 'string'),
 ('Over18', 'string'),
 ('OverTime', 'string'),
 ('PercentSalaryHike', 'string'),
 ('PerformanceRating', 'string'),
 ('RelationshipSatisfaction', 'string'),
 ('StandardHours', 'string'),
 ('StockOptionLevel', 'string'),
 ('TotalWorkingYears', 'string'),
 ('TrainingTimesLastYear', 'string'),
 ('WorkLifeBalance', 'string'),
 ('YearsAtCompany', 'string'),
 ('YearsInCurrentRole', 'string'),
 ('YearsSinceLastPromotion', 'string'),
 ('YearsWithCurrManager', 'string')]
```

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

```
21/11/13 01:39:17 WARN org.apache.spark.sql.catalyst.util.package: Truncated the
string representation of a plan since it was too large. This behavior can be
```

```
adjusted by setting 'spark.sql.debug.maxToStringFields'.
```

[9]:
```
df1
```

[9]:
```
      Age Attrition     BusinessTravel DailyRate              Department  \
0      41       Yes      Travel_Rarely      1102                   Sales
1      49        No  Travel_Frequently       279  Research & Development
2      37       Yes      Travel_Rarely      1373  Research & Development
3      33        No  Travel_Frequently      1392  Research & Development
4      27        No      Travel_Rarely       591  Research & Development
…     ..       …               …         …                      …
1465   36        No  Travel_Frequently       884  Research & Development
1466   39        No      Travel_Rarely       613  Research & Development
1467   27        No      Travel_Rarely       155  Research & Development
1468   49        No  Travel_Frequently      1023                   Sales
1469   34        No      Travel_Rarely       628  Research & Development

     DistanceFromHome Education EducationField EmployeeCount EmployeeNumber  \
0                   1         2  Life Sciences             1              1
1                   8         1  Life Sciences             1              2
2                   2         2          Other             1              4
3                   3         4  Life Sciences             1              5
4                   2         1        Medical             1              7
…                 …         …            …             …              …
1465               23         2        Medical             1           2061
1466                6         1        Medical             1           2062
1467                4         3  Life Sciences             1           2064
1468                2         3        Medical             1           2065
1469                8         3        Medical             1           2068

      … RelationshipSatisfaction StandardHours StockOptionLevel  \
0     …                        1            80                0
1     …                        4            80                1
2     …                        2            80                0
3     …                        3            80                0
4     …                        4            80                1
…   …                      …             …                …
1465  …                        3            80                1
1466  …                        1            80                1
1467  …                        2            80                1
1468  …                        4            80                0
1469  …                        1            80                0

     TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany  \
0                    8                     0               1              6
1                   10                     3               3             10
2                    7                     3               3              0
```

```
3                    8                   3            3            8
4                    6                   3            3            2
…                 …                   …            …            …
1465                17                   3            3            5
1466                 9                   5            3            7
1467                 6                   0            3            6
1468                17                   3            2            9
1469                 6                   3            4            4

     YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                     4                        0                     5
1                     7                        1                     7
2                     0                        0                     0
3                     7                        3                     0
4                     2                        2                     2
…                   …                      …                   …
1465                  2                        0                     3
1466                  7                        1                     7
1467                  2                        0                     3
1468                  6                        0                     8
1469                  3                        1                     2

[1470 rows x 35 columns]
```

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

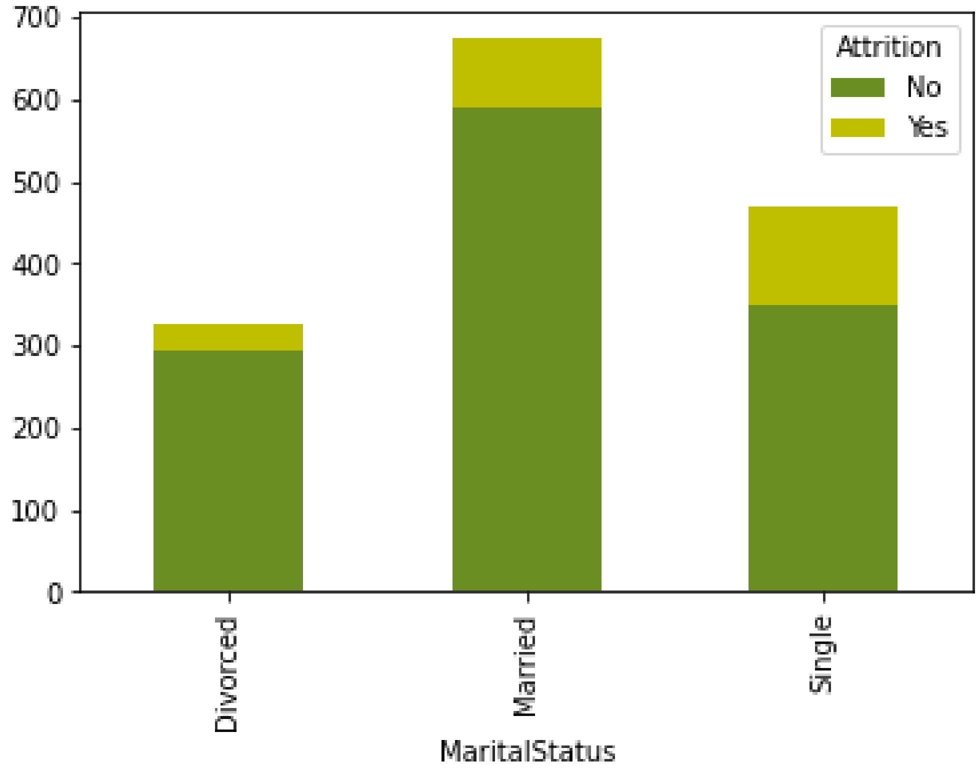
[10]:
```
         Age Attrition BusinessTravel DailyRate              Department  \
count   1470      1470           1470      1470                    1470
unique    43         2              3       886                       3
top       35        No  Travel_Rarely       691  Research & Development
freq      78      1233           1043         6                     961

       DistanceFromHome Education EducationField EmployeeCount EmployeeNumber  \
count              1470      1470           1470          1470           1470
unique               29         5              6             1           1470
top                   2         3  Life Sciences             1            190
freq                211       572            606          1470              1

        … RelationshipSatisfaction StandardHours StockOptionLevel  \
count   …                     1470          1470             1470
unique  …                        4             1                4
top     …                        3            80                0
freq    …                      459          1470              631

       TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany  \
count               1470                  1470            1470           1470
unique                40                     7               4             37
```

```
top                     10                     2               3              5
freq                   202                   547             893            196

       YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
count                1470                     1470                  1470
unique                 19                       16                    18
top                     2                        0                     2
freq                  372                      581                   344

[4 rows x 35 columns]
```

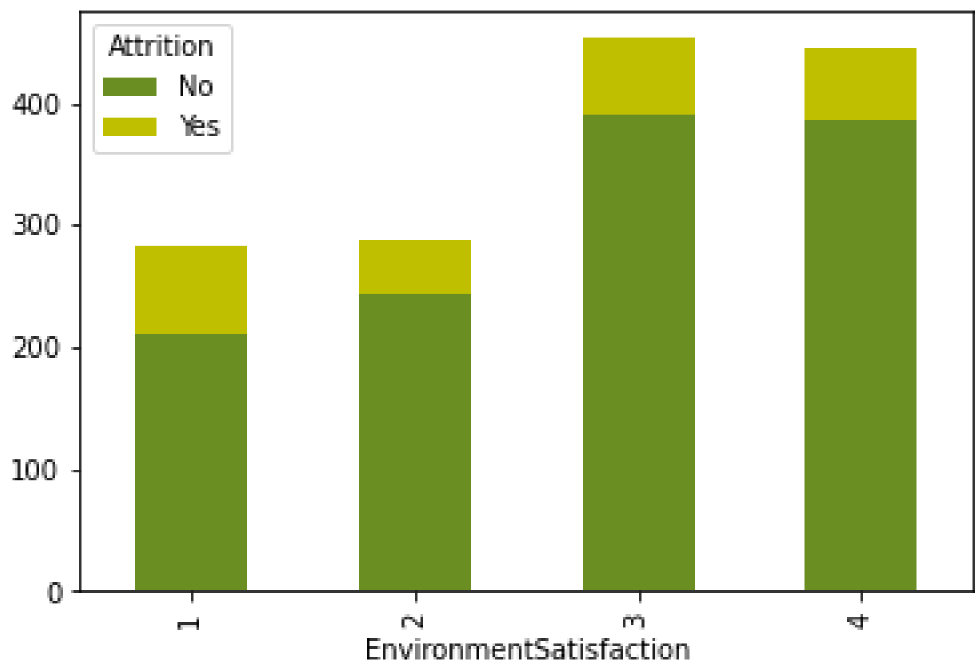
[11]:
```
df_plot_Mar = df1.groupby(['Attrition', 'MaritalStatus']).size().reset_index().
 ↪pivot(columns = 'Attrition', index = 'MaritalStatus', values = 0)
df_plot_Env = df1.groupby(['Attrition', 'EnvironmentSatisfaction']).size().
 ↪reset_index().pivot(columns = 'Attrition', index =␣
 ↪'EnvironmentSatisfaction', values = 0)
df_plot_Inv = df1.groupby(['Attrition', 'JobInvolvement']).size().reset_index().
 ↪pivot(columns = 'Attrition', index = 'JobInvolvement', values = 0)
df_plot_Bln = df1.groupby(['Attrition', 'WorkLifeBalance']).size().
 ↪reset_index().pivot(columns = 'Attrition', index = 'WorkLifeBalance', values␣
 ↪= 0)
```

[12]:
```
df_plot_Mar.plot(kind = 'bar', stacked = True, color = ['#6B8E23','y'])
df_plot_Env.plot(kind = 'bar', stacked = True, color = ['#6B8E23','y'])
df_plot_Inv.plot(kind = 'bar', stacked = True, color = ['#6B8E23','y'])
df_plot_Bln.plot(kind = 'bar', stacked = True, color = ['#6B8E23','y'])
```

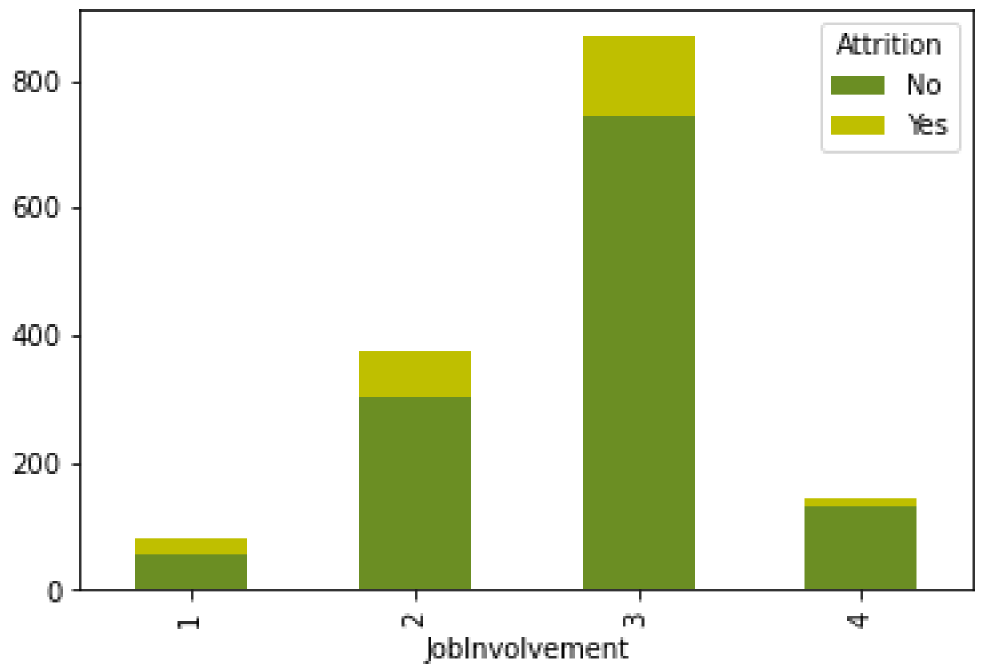
[12]:
```
<AxesSubplot:xlabel='WorkLifeBalance'>
```

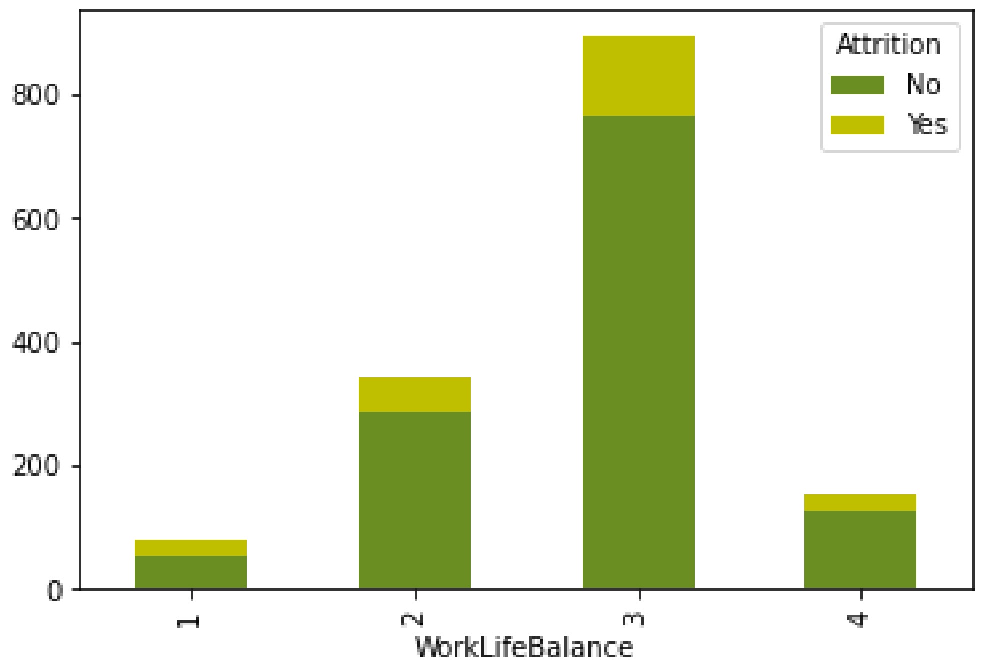


700
600
500
400
300
200
100
0
Attrition
No
Yes
Divorced
Married
Single
MaritalStatus



400
300
200
100
0
Attrition
No
Yes
1
2
3
4
EnvironmentSatisfaction





[13]:
```
df1.isna().sum()
```

[13]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

[14]:
```python
dfs = df1.groupby(['Attrition', 'Gender']).size().reset_index()
dfs
```

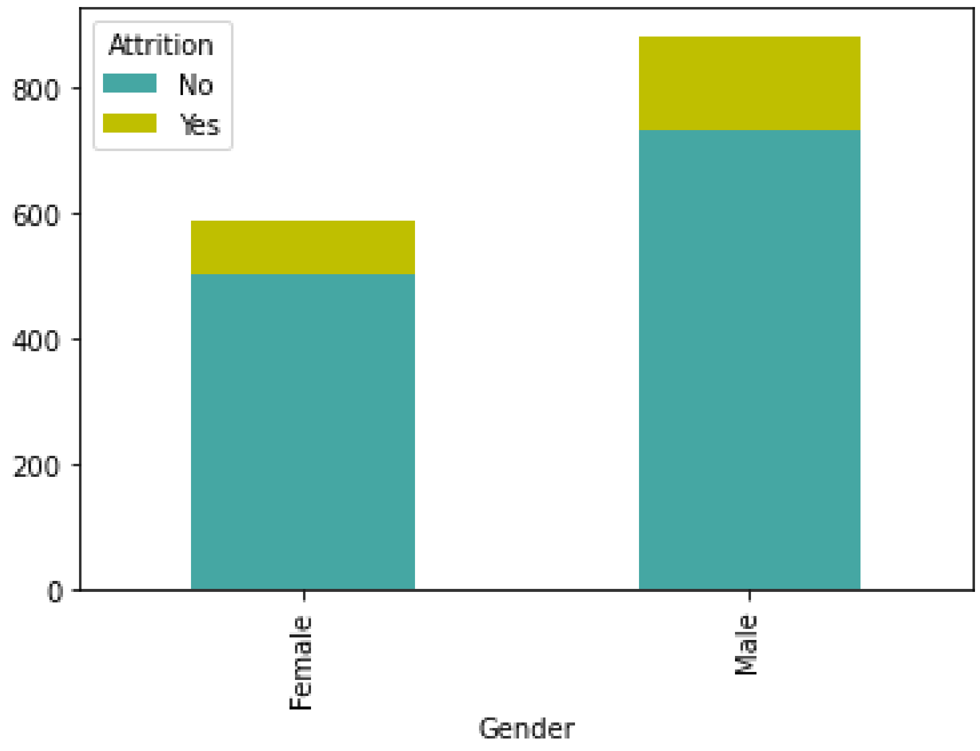
[14]:
```
  Attrition  Gender    0
0        No  Female  501
1        No    Male  732
2       Yes  Female   87
3       Yes    Male  150
```

```
[15]: df_plot_Gen = df1.groupby(['Attrition', 'Gender']).size().reset_index().
       ↪pivot(columns = 'Attrition', index = 'Gender', values = 0)
      df_plot_Gen
```

```
[15]: Attrition   No  Yes
      Gender
      Female     501   87
      Male       732  150
```

```
[16]: df_plot_Gen.plot(kind = 'bar', stacked = True, color = ['#45A7A3','y'])
      plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_8047/759299616.py in <module>
      1 df_plot_Gen.plot(kind = 'bar', stacked = True, color = ['#45A7A3','y'])
----> 2 plt.show()

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



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

[18]:
```
#deleting columns that are not required
df_drop1=df.
 ↪drop('DailyRate','Department','BusinessTravel','Education','EmployeeCount','EmployeeNumber'
```

[19]:
```
df_drop1.columns
```

[19]:
```
['Age',
 'Attrition',
 'DistanceFromHome',
 'EnvironmentSatisfaction',
 'Gender',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MaritalStatus',
 'MonthlyIncome',
 'NumCompaniesWorked',
 'PercentSalaryHike',
 'PerformanceRating',
 'TotalWorkingYears',
 'TrainingTimesLastYear',
 'WorkLifeBalance',
 'YearsAtCompany']
```

[20]:
```
df_drop1.dtypes
```

[20]:
```
[('Age', 'string'),
 ('Attrition', 'string'),
 ('DistanceFromHome', 'string'),
 ('EnvironmentSatisfaction', 'string'),
 ('Gender', 'string'),
 ('JobInvolvement', 'string'),
 ('JobLevel', 'string'),
 ('JobSatisfaction', 'string'),
 ('MaritalStatus', 'string'),
 ('MonthlyIncome', 'string'),
 ('NumCompaniesWorked', 'string'),
 ('PercentSalaryHike', 'string'),
 ('PerformanceRating', 'string'),
 ('TotalWorkingYears', 'string'),
 ('TrainingTimesLastYear', 'string'),
 ('WorkLifeBalance', 'string'),
 ('YearsAtCompany', 'string')]
```

[21]:
```
from pyspark.sql.types import IntegerType
```

[22]:
```python
df_drop1= df_drop1.withColumn("Age",df["Age"].cast(IntegerType()))
df_drop1= df_drop1.withColumn("DistanceFromHome",df["DistanceFromHome"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.
 ↪withColumn("EnvironmentSatisfaction",df["EnvironmentSatisfaction"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("JobInvolvement",df["JobInvolvement"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("JobLevel",df["JobLevel"].cast(IntegerType()))
df_drop1= df_drop1.withColumn("JobSatisfaction",df["JobSatisfaction"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("MonthlyIncome",df["MonthlyIncome"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("NumCompaniesWorked",df["NumCompaniesWorked"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("PercentSalaryHike",df["PercentSalaryHike"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("PerformanceRating",df["PerformanceRating"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("TotalWorkingYears",df["TotalWorkingYears"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.
 ↪withColumn("TrainingTimesLastYear",df["TrainingTimesLastYear"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("WorkLifeBalance",df["WorkLifeBalance"].
 ↪cast(IntegerType()))
df_drop1= df_drop1.withColumn("YearsAtCompany",df["YearsAtCompany"].
 ↪cast(IntegerType()))
```

[23]:
```python
from pyspark.sql.functions import when
from pyspark.sql.functions import lit
```

[24]:
```python
df_drop1=df_drop1.withColumn("AttritionMod",when((df_drop1.Attrition=='Yes'),␣
 ↪lit(1)) .otherwise(lit(0)))
```

[25]:
```python
df_drop1.describe()
```

[25]:
```
DataFrame[summary: string, Age: string, Attrition: string, DistanceFromHome:
string, EnvironmentSatisfaction: string, Gender: string, JobInvolvement: string,
JobLevel: string, JobSatisfaction: string, MaritalStatus: string, MonthlyIncome:
string, NumCompaniesWorked: string, PercentSalaryHike: string,
PerformanceRating: string, TotalWorkingYears: string, TrainingTimesLastYear:
string, WorkLifeBalance: string, YearsAtCompany: string, AttritionMod: string]
```

[26]:
```
df_drop1.head(5)
```

[26]:
```
[Row(Age=41, Attrition='Yes', DistanceFromHome=1, EnvironmentSatisfaction=2,
Gender='Female', JobInvolvement=3, JobLevel=2, JobSatisfaction=4,
MaritalStatus='Single', MonthlyIncome=5993, NumCompaniesWorked=8,
PercentSalaryHike=11, PerformanceRating=3, TotalWorkingYears=8,
TrainingTimesLastYear=0, WorkLifeBalance=1, YearsAtCompany=6, AttritionMod=1),
 Row(Age=49, Attrition='No', DistanceFromHome=8, EnvironmentSatisfaction=3,
Gender='Male', JobInvolvement=2, JobLevel=2, JobSatisfaction=2,
MaritalStatus='Married', MonthlyIncome=5130, NumCompaniesWorked=1,
PercentSalaryHike=23, PerformanceRating=4, TotalWorkingYears=10,
TrainingTimesLastYear=3, WorkLifeBalance=3, YearsAtCompany=10, AttritionMod=0),
 Row(Age=37, Attrition='Yes', DistanceFromHome=2, EnvironmentSatisfaction=4,
Gender='Male', JobInvolvement=2, JobLevel=1, JobSatisfaction=3,
MaritalStatus='Single', MonthlyIncome=2090, NumCompaniesWorked=6,
PercentSalaryHike=15, PerformanceRating=3, TotalWorkingYears=7,
TrainingTimesLastYear=3, WorkLifeBalance=3, YearsAtCompany=0, AttritionMod=1),
 Row(Age=33, Attrition='No', DistanceFromHome=3, EnvironmentSatisfaction=4,
Gender='Female', JobInvolvement=3, JobLevel=1, JobSatisfaction=3,
MaritalStatus='Married', MonthlyIncome=2909, NumCompaniesWorked=1,
PercentSalaryHike=11, PerformanceRating=3, TotalWorkingYears=8,
TrainingTimesLastYear=3, WorkLifeBalance=3, YearsAtCompany=8, AttritionMod=0),
 Row(Age=27, Attrition='No', DistanceFromHome=2, EnvironmentSatisfaction=1,
Gender='Male', JobInvolvement=3, JobLevel=1, JobSatisfaction=2,
MaritalStatus='Married', MonthlyIncome=3468, NumCompaniesWorked=9,
PercentSalaryHike=12, PerformanceRating=3, TotalWorkingYears=6,
TrainingTimesLastYear=3, WorkLifeBalance=3, YearsAtCompany=2, AttritionMod=0)]
```

[27]:
```
df_drop1=df_drop1.withColumn("MaritalStatusMod",when((df_drop1.
↪MaritalStatus=='Yes'), lit(1)) .otherwise(lit(0)))
```

[28]:
```
df_drop1=df_drop1.withColumn("GenderMod",when((df_drop1.Gender=='Male'),␣
↪lit(1)) .otherwise(lit(0)))
```

[29]:
```
df_final=df_drop1.drop('Gender','MaritalStatus','Attrition')
```

[30]:
```
df_final.columns
```

[30]:
```
['Age',
 'DistanceFromHome',
 'EnvironmentSatisfaction',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MonthlyIncome',
 'NumCompaniesWorked',
 'PercentSalaryHike',
 'PerformanceRating',
 'TotalWorkingYears',
 'TrainingTimesLastYear',
 'WorkLifeBalance',
 'YearsAtCompany',
 'AttritionMod',
 'MaritalStatusMod',
 'GenderMod']
```

[31]:
```
df_final = df_final.withColumn("GenderMod",df_final["GenderMod"].
 ↪cast(IntegerType()))
df_final = df_final.withColumn("MaritalStatusMod",df_final["MaritalStatusMod"].
 ↪cast(IntegerType()))
df_final = df_final.withColumn("AttritionMod",df_final["AttritionMod"].
 ↪cast(IntegerType()))
```

[32]:
```
df_final.dtypes
```

[32]:
```
[('Age', 'int'),
 ('DistanceFromHome', 'int'),
 ('EnvironmentSatisfaction', 'int'),
 ('JobInvolvement', 'int'),
 ('JobLevel', 'int'),
 ('JobSatisfaction', 'int'),
 ('MonthlyIncome', 'int'),
 ('NumCompaniesWorked', 'int'),
 ('PercentSalaryHike', 'int'),
 ('PerformanceRating', 'int'),
 ('TotalWorkingYears', 'int'),
 ('TrainingTimesLastYear', 'int'),
 ('WorkLifeBalance', 'int'),
 ('YearsAtCompany', 'int'),
 ('AttritionMod', 'int'),
 ('MaritalStatusMod', 'int'),
 ('GenderMod', 'int')]
```

[33]:
```
dffinal1=df_final.toPandas()
```

[36]:
```
dffinal1
```

[36]:
```
      Age  DistanceFromHome  EnvironmentSatisfaction  JobInvolvement  \
0      41                 1                        2               3
1      49                 8                        3               2
2      37                 2                        4               2
3      33                 3                        4               3
4      27                 2                        1               3
…    …                …                        …               …
```

```
1465   36                23                    3            4
1466   39                 6                    4            2
1467   27                 4                    2            4
1468   49                 2                    4            2
1469   34                 8                    2            4

      JobLevel  JobSatisfaction  MonthlyIncome  NumCompaniesWorked  \
0            2                4           5993                   8
1            2                2           5130                   1
2            1                3           2090                   6
3            1                3           2909                   1
4            1                2           3468                   9
…          …              …            …                  …
1465         2                4           2571                   4
1466         3                1           9991                   4
1467         2                2           6142                   1
1468         2                2           5390                   2
1469         2                3           4404                   2

      PercentSalaryHike  PerformanceRating  TotalWorkingYears  \
0                    11                  3                  8
1                    23                  4                 10
2                    15                  3                  7
3                    11                  3                  8
4                    12                  3                  6
…                  …                 …                 …
1465                 17                  3                 17
1466                 15                  3                  9
1467                 20                  4                  6
1468                 14                  3                 17
1469                 12                  3                  6

      TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  AttritionMod  \
0                         0                1               6             1
1                         3                3              10             0
2                         3                3               0             1
3                         3                3               8             0
4                         3                3               2             0
…                       …              …              …            …
1465                      3                3               5             0
1466                      5                3               7             0
1467                      0                3               6             0
1468                      3                2               9             0
1469                      3                4               4             0

      MaritalStatusMod  GenderMod
0                    0          0
```

```
1                    0          1
2                    0          1
3                    0          0
4                    0          1
…                  …        …
1465                 0          1
1466                 0          1
1467                 0          1
1468                 0          1
1469                 0          1

[1470 rows x 17 columns]
```

[37]:
```
featurescols =␣
 ↪['Age','DistanceFromHome','EnvironmentSatisfaction','JobInvolvement','JobLevel',
             ␣
 ↪'JobSatisfaction','MonthlyIncome','NumCompaniesWorked','PercentSalaryHike','PerformanceRati
             ␣
 ↪'TrainingTimesLastYear','WorkLifeBalance','YearsAtCompany','MaritalStatusMod','GenderMod']
from pyspark.ml.feature import VectorAssembler, StringIndexer, VectorIndexer,␣
 ↪MinMaxScaler,Normalizer

from pyspark.ml.classification import LogisticRegression

Vect =␣
 ↪VectorAssembler(inputCols=['Age','DistanceFromHome','EnvironmentSatisfaction','JobInvolveme
             ␣
 ↪'JobSatisfaction','MonthlyIncome','NumCompaniesWorked','PercentSalaryHike','PerformanceRati
             ␣
 ↪'TrainingTimesLastYear','WorkLifeBalance','YearsAtCompany','MaritalStatusMod','GenderMod'],
 ↪outputCol="features")
```

[38]:
```
catIdx = VectorIndexer(inputCol = Vect.getOutputCol(), outputCol =␣
 ↪"idxCatFeatures")
```

[39]:
```
train, test = df_final.randomSplit([0.8,0.2])
```

[40]:
```
train.count()
```

[40]:
```
1177
```

[41]:
```
test.count()
```

[41]:
```
293
```

[42]:
```
lr =␣
↪LogisticRegression(labelCol="AttritionMod",featuresCol="features",maxIter=10,regParam=0.
↪3)
```

[43]:
```
pipeline1 = Pipeline(stages=[Vect, catIdx, lr])
```

[44]:
```
pipelineModel = pipeline1.fit(train)
```

```
21/11/13 01:44:32 WARN com.github.fommil.netlib.BLAS: Failed to load
implementation from: com.github.fommil.netlib.NativeSystemBLAS
21/11/13 01:44:33 WARN com.github.fommil.netlib.BLAS: Failed to load
implementation from: com.github.fommil.netlib.NativeRefBLAS
```

[45]:
```
prediction1 = pipelineModel.transform(test)
```

[46]:
```
predicted=prediction1.select("AttritionMod","prediction")
predicted.show(100, truncate=False)
```

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

[114]:
```
tp = float(predicted.filter("prediction == 1.0 AND AttritionMod== 1").count())
fp = float(predicted.filter("prediction == 1.0 AND AttritionMod == 0").count())
tn = float(predicted.filter("prediction == 0.0 AND AttritionMod== 0").count())
fn = float(predicted.filter("prediction == 0.0 AND AttritionMod == 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()
```

```
---------------------------------------------------------------------------
```

```
ZeroDivisionError                         Traceback (most recent call last)
/tmp/ipykernel_8107/768638879.py in <module>
      3 tn = float(predicted.filter("prediction == 0.0 AND AttritionMod== 0").
 ↪count())
      4 fn = float(predicted.filter("prediction == 0.0 AND AttritionMod == 1").
 ↪count())
----> 5 pr = tp / (tp + fp)
      6 re = tp / (tp + fn)
      7 metrics = spark.createDataFrame([

ZeroDivisionError: float division by zero
```

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

[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|
|           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|
+------------+----------+
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
135374497,3.6269792461350723,0.6516627363731139,0.0,3.551973013434531 5,8.3143137
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
993,2.7202344346013043,1.1123721481049536,2.401912877583144,3.551973013434531 5,8
.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|
|           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|
|            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|
|            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|
|            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
```

[ ]:

```

```