Lab07

June 27, 2022

1 Lab 07: Spark for Machine Learning

A marketing agency has many customers that use their service to produce ads for the client/customer websites. They've noticed that they have quite a bit of churn in clients. They basically randomly assign account managers right now, but want you to create a machine learning model that will help predict which customers will churn (stop buying their service) so that they can correctly assign the customers most at risk to churn an account manager. Luckily they have some historical data. Can you help them out? Create a classification algorithm that will help classify whether or not a customer churned. Then the company can test this against incoming data for future customers to predict which customers will churn and assign them an account manager.

The data is saved as customer_churn.csv. Here are the fields and their definitions: Name : Name of the latest contact at Company Age: Customer Age Total_Purchase: Total Ads Purchased Account_Manager: Binary 0 = No manager, 1= Account manager assigned Years: Total Years as a customer Num_sites: Number of websites that use the service Onboard_date: Date that the name of the latest contact was onboarded Location: Client HQ Address Company: Name of Client Company

Once you've created the model and evaluated it, test out the model on some new data (you can think of this almost like a hold-out set) that your client has provided, saved under new_customers.csv. The client wants to know which customers are most likely to churn given this data (they don't have the label yet).

[1]:
```
from pyspark.sql import SparkSession
```

[5]:
```
spark = SparkSession.builder.appName('data').getOrCreate()
df = spark.read.csv('gs://lab07-spark-for-ml/customer_churn.csv',␣
 ↪inferSchema=True, header=True)
df.show()
```

```
+------------------+----+-------------+--------------+-----+---------+-------
-----------+------------------+--------------------+-----+
|             Names| Age|Total_Purchase|Account_Manager|Years|Num_Sites|
Onboard_date|           Location|             Company|Churn|
+------------------+----+-------------+--------------+-----+---------+-------
-----------+------------------+--------------------+-----+
|  Cameron Williams|42.0|      11066.8|             0| 7.22|
8.0|2013-08-30 07:00:40|10265 Elizabeth M…|          Harvey LLC|    1|
|      Kevin Mueller|41.0|     11916.22|             0|  6.5|
```

```
11.0|2013-08-13 00:38:46|6157 Frank Garden…|          Wilson PLC|    1|
|        Eric Lozano|38.0|      12884.75|              0| 6.67|
12.0|2016-06-29 06:20:07|1331 Keith Court …|Miller, Johnson a…|    1|
|      Phillip White|42.0|       8010.76|              0| 6.71|
10.0|2014-04-22 12:43:12|13120 Daniel Moun…|           Smith Inc|    1|
|     Cynthia Norton|37.0|       9191.58|              0| 5.56|
9.0|2016-01-19 15:31:15|765 Tricia Row Ka…|          Love-Jones|    1|
|   Jessica Williams|48.0|      10356.02|              0| 5.12|
8.0|2009-03-03 23:13:37|6187 Olson Mounta…|        Kelly-Warren|    1|
|        Eric Butler|44.0|      11331.58|              1| 5.23|
11.0|2016-12-05 03:35:43|4846 Savannah Roa…|   Reynolds-Sheppard|    1|
|      Zachary Walsh|32.0|       9885.12|              1| 6.92|
9.0|2006-03-09 14:50:20|25271 Roy Express…|          Singh-Cole|    1|
|        Ashlee Carr|43.0|       14062.6|              1| 5.46|
11.0|2011-09-29 05:47:23|3725 Caroline Str…|           Lopez PLC|    1|
|     Jennifer Lynch|40.0|       8066.94|              1| 7.11|
11.0|2006-03-28 15:42:45|363 Sandra Lodge …|       Reed-Martinez|    1|
|       Paula Harris|30.0|      11575.37|              1| 5.22|
8.0|2016-11-13 13:13:01|Unit 8120 Box 916…|Briggs, Lamb and …|    1|
|     Bruce Phillips|45.0|       8771.02|              1| 6.64|
11.0|2015-05-28 12:14:03|Unit 1895 Box 094…|    Figueroa-Maynard|    1|
|       Craig Garner|45.0|       8988.67|              1| 4.84|
11.0|2011-02-16 08:10:47|897 Kelley Overpa…|     Abbott-Thompson|    1|
|       Nicole Olson|40.0|       8283.32|              1|  5.1|
13.0|2012-11-22 05:35:03|11488 Weaver Cape…|Smith, Kim and Ma…|    1|
|     Harold Griffin|41.0|       6569.87|              1|  4.3|
11.0|2015-03-28 02:13:44|1774 Peter Row Ap…|Snyder, Lee and M…|    1|
|       James Wright|38.0|      10494.82|              1| 6.81|
12.0|2015-07-22 08:38:40|45408 David Path …|      Sanders-Pierce|    1|
|      Doris Wilkins|45.0|       8213.41|              1| 7.35|
11.0|2006-09-03 06:13:55|28216 Wright Moun…|Andrews, Adams an…|    1|
|Katherine Carpenter|43.0|      11226.88|              0| 8.08|
12.0|2006-10-22 04:42:38|Unit 4948 Box 481…|Morgan, Phillips …|    1|
|     Lindsay Martin|53.0|       5515.09|              0| 6.85|
8.0|2015-10-07 00:27:10|69203 Crosby Divi…|      Villanueva LLC|    1|
|        Kathy Curry|46.0|        8046.4|              1| 5.69|
8.0|2014-11-06 23:47:14|9569 Caldwell Cre…|Berry, Orr and Ca…|    1|
+-------------------+----+--------------+---------------+-----+---------+-------
------------+--------------------+--------------------+-----+
only showing top 20 rows
```

[6]:
```
df.columns
```

[6]:
```
['Names',
 'Age',
 'Total_Purchase',
```

```
 'Account_Manager',
 'Years',
 'Num_Sites',
 'Onboard_date',
 'Location',
 'Company',
 'Churn']
```

[7]:
```
from pyspark.ml.feature import VectorAssembler
```

[8]:
```
assembler = VectorAssembler(inputCols=['Age', 'Total_Purchase',␣
↪'Account_Manager', 'Years', 'Num_Sites'],
                            outputCol='features')
```

[9]:
```
output = assembler.transform(df)
```

[10]:
```
final_df = output.select('features', 'churn')
```

Test

[11]:
```
tr_churn, test_churn = final_df.randomSplit([0.7, 0.3])
```

Model

[13]:
```
from pyspark.ml.classification import LogisticRegression

lr_churn = LogisticRegression(labelCol='churn')
churn_model = lr_churn.fit(tr_churn)
tr_sum = churn_model.summary
tr_sum.predictions.describe().show()
```

```
+-------+-------------------+-------------------+
|summary|              churn|         prediction|
+-------+-------------------+-------------------+
|  count|                622|                622|
|   mean|0.16881028938906753|0.13504823151125403|
| stddev| 0.3748857466617767|0.34205015253427273|
|    min|                0.0|                0.0|
|    max|                1.0|                1.0|
+-------+-------------------+-------------------+
```

Evaluation of Results

[15]:
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator

pred_labels = churn_model.evaluate(test_churn)
pred_labels.predictions.show()
```

```
+------------------+-----+------------------+------------------+----------
+
|            features|churn|       rawPrediction|
probability|prediction|
+------------------+-----+------------------+------------------+----------
+
|[27.0,8628.8,1.0,…|    0|[6.10458935158166…|[0.99777237539554…|
0.0|
|[28.0,11128.95,1…|    0|[4.75126500374198…|[0.99143326538230…|
0.0|
|[28.0,11204.23,0…|    0|[2.18193615925427…|[0.89861560324320…|
0.0|
|[29.0,8688.17,1.0…|    1|[3.28388298808549…|[0.96387174476012…|
0.0|
|[29.0,9617.59,0.0…|    0|[4.86737628785937…|[0.99236521398876…|
0.0|
|[29.0,10203.18,1…|    0|[4.34685028637650…|[0.98721796632379…|
0.0|
|[30.0,10744.14,1…|    1|[2.31658930179719…|[0.91024167020813…|
0.0|
|[31.0,7073.61,0.0…|    0|[3.48808750612343…|[0.97034691561818…|
0.0|
|[31.0,11297.57,1…|    1|[1.46232605385164…|[0.81188818088283…|
0.0|
|[32.0,8617.98,1.0…|    1|[1.53567716399398…|[0.82283543313031…|
0.0|
|[32.0,9885.12,1.0…|    1|[2.30166323437797…|[0.90901469369116…|
0.0|
|[32.0,10716.75,0…|    0|[4.71490765052733…|[0.99111888738448…|
0.0|
|[32.0,11715.72,0…|    0|[3.64948314346816…|[0.97465453195916…|
0.0|
|[32.0,12547.91,0…|    0|[0.81030750267544…|[0.69217502738671…|
0.0|
|[33.0,7720.61,1.0…|    0|[2.13011130578569…|[0.89379557449471…|
0.0|
|[33.0,12115.91,1…|    0|[2.81737696150153…|[0.94360765094496…|
0.0|
|[33.0,13314.19,0…|    0|[3.09170214077243…|[0.95654916588488…|
0.0|
|[34.0,6131.92,0.0…|    0|[4.06809608425412…|[0.98317789176348…|
0.0|
|[34.0,6461.86,1.0…|    0|[4.71602944603067…|[0.99112875625785…|
0.0|
|[34.0,7818.13,0.0…|    0|[4.17792803854307…|[0.98490122940605…|
0.0|
+------------------+-----+------------------+------------------+----------
+
```

```
only showing top 20 rows
```

AUC

[16]:
```
churn_eval = BinaryClassificationEvaluator(rawPredictionCol='prediction',␣
 ↪labelCol='churn')
auc_churn = churn_eval.evaluate(pred_labels.predictions)
auc_churn
```

[16]:
```
0.74482594182165
```

Unlabeled Data Prediction

[18]:
```
fn_model = lr_churn.fit(final_df)
new_customers = spark.read.csv('gs://lab07-spark-for-ml/new_customers.csv',␣
 ↪inferSchema=True, header=True)
new_customers.printSchema()
```

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

[19]:
```
test_new_cust = assembler.transform(new_customers)
test_new_cust.printSchema()
```

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

[20]:
```
final_result = fn_model.transform(test_new_cust)
final_result.select('Company', 'prediction').show()
```

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

[ ]: