good(Python)
Import Notebook

```
import pyspark
```

Show result
Command took 0.05 seconds

```
df=sqlContext.sql('SELECT* FROM churn3_1_csv')
```

Command took 5.42 seconds

```
for churn in df.head(3):
  print(churn)
  print('\n')
```

Row(Age=42.0, Total_Purchase=11066.8, Years=7.22, Num_Sites=8.0, Churn=1) Row(Age=41.0, Total_Purchase=11916.22, Years=6.5, Num_Sites=11.0, Churn=1) Row(Age=38.0, Total_Purchase=12884.75, Years=6.67, Num_Sites=12.0, Churn=1)
Command took 5.09 seconds

```
df.printSchema()
```

root |-- Age: double (nullable = true) |-- Total_Purchase: double (nullable = true) |-- Years: double (nullable = true) |-- Num_Sites: double (nullable = true) |-- Churn: integer (nullable = true)
Command took 0.05 seconds

```
df.describe().show()
```

+-------+-----------------+-----------------+-----------------+------------------+-------------------+ |summary| Age| Total_Purchase| Years| Num_Sites| Churn| +-------+-----------------+-----------------+-----------------+------------------+-------------------+ | count| 900| 900| 900| 900| 900| | mean|41.81666666666667|10062.82403333334| 5.27315555555555| 8.587777777777777|0.16666666666666666| | stddev|6.127560416916251|2408.644531858096|1.2744490131946161.7648355920350969| 0.3728852122772358| | min| 22.0| 100.0| 1.0| 3.0| 0| | max| 65.0| 18026.01| 9.15| 14.0| 1| +-------+-----------------+-----------------+-----------------+------------------+-------------------+
Command took 3.06 seconds

```
df.columns
```

Out[7]: ['Age', 'Total_Purchase', 'Years', 'Num_Sites', 'Churn']
Command took 0.04 seconds

```
#Formating our data into features and label.
```

Command took 0.03 seconds

```
from pyspark.ml.feature import VectorAssembler
```

Command took 0.04 seconds

```
ass=VectorAssembler(inputCols=['Age', 'Total_Purchase', 'Years', 'Num_Sites'],outputCol='features')
```

Command took 0.09 seconds

```
output=ass.transform(df)
```

Command took 0.74 seconds

```
final_df=output.select('features','Churn')
```

Command took 0.07 seconds

```
#Test Train Split
train,test=final_df.randomSplit([0.7,0.3])
```

Command took 0.06 seconds

```
#Fit in model
from pyspark.ml.classification import LogisticRegression
```

Command took 0.04 seconds

```
Lg=LogisticRegression(featuresCol='features',labelCol='Churn')
```

Command took 0.14 seconds

```
fit_model=Lg.fit(train)
```

Command took 8.04 seconds

```
train_sum=fit_model.summary
```

Command took 0.06 seconds

```
train_sum.predictions.describe().show()
```

+-------+------------------+------------------+ |summary| Churn| prediction| +-------+------------------+------------------+ | count| 629| 629| | mean|0.17011128775834658|0.14308426073131955| | stddev|0.37602956799028053|0.35043743486543466| | min| 0.0| 0.0| | max| 1.0| 1.0| +-------+------------------+------------------+

Command took 1.44 seconds

```
#Evaluate results
from pyspark.ml.evaluation import BinaryClassificationEvaluator
```

Command took 0.03 seconds

```
pred_and_labels=fit_model.evaluate(test)
```

Command took 0.43 seconds

```
pred_and_labels.predictions.show()
```

+--------------------+-----+--------------------+--------------------+----------+ | features|Churn| rawPrediction| probability|prediction| +--------------------+-----+--------------------+--------------------+----------+ |[22.0,11254.38,4....| 0|[5.89717332643735...|[0.99726032713781...| 0.0| |[27.0,8628.8,5.3,...| 0|[6.93115631332765...|[0.99902408269843...| 0.0| |[29.0,8688.17,5.7...| 1|[3.61222968908919...|[0.97371780164937...| 0.0| |[29.0,12711.15,5....| 0|[6.08734858685923...|[0.99773372425208...| 0.0| |[29.0,13240.01,4....| 0|[8.22240364687139...|[0.99973150336577...| 0.0| |[30.0,8677.28,7.3...| 0|[5.37451360508052...|[0.99538819554529...| 0.0| |[30.0,11575.37,5....| 1|[5.00020585045489...|[0.99330851744224...| 0.0| |[31.0,9574.89,7.3...| 0|[3.74194059534869...|[0.97684100392550...| 0.0| |[31.0,10058.87,6....| 0|[5.65026421723741...|[0.99649573554349...| 0.0| |[31.0,10182.6,3.7...| 0|[5.97495930401166...|[0.99746483786426...| 0.0| |[31.0,11297.57,6....| 1|[1.40929039962492...|[0.80365399690131...| 0.0| |[31.0,11743.24,5....| 0|[7.72723276891047...|[0.99955953225271...| 0.0| |[32.0,8575.71,4.2...| 0|[4.28439571629112...|[0.98640541267322...| 0.0| |[32.0,9472.72,1.0...| 0|[4.85025964395829...|[0.99223443080323...| 0.0| |[32.0,12142.99,5....| 0|[6.34711886872998...|[0.99825127639046...| 0.0| |[33.0,7492.9,6.71...| 0|[5.60922412028183...|[0.99634946422876...| 0.0| |[33.0,7750.54,4.5...| 0|[5.51702222465506...|[0.99599828383694...| 0.0| |[33.0,10709.39,5....| 0|[7.70151925722718...|[0.99954806459371...| 0.0| |[33.0,13157.08,5....| 0|[1.82179545764817...|[0.86078142827480...| 0.0| |[33.0,14160.05,4....| 0|[6.32529931767959...|[0.99821276964841...| 0.0| +--------------------+-----+--------------------+--------------------+----------+ only showing top 20 rows

Command took 0.99 seconds

```
#Evaluating with AUC
evaluator=BinaryClassificationEvaluator(rawPredictionCol='prediction',labelCol='Churn')
```

Command took 0.04 seconds

```
AUC=evaluator.evaluate(pred_and_labels.predictions)
```

Command took 0.84 seconds

```
AUC
```

Out[48]: 0.6873725010199919
Command took 0.03 seconds