first session

```
import org.apache.spark.sql.types._
```

FINISHED

```
import org.apache.spark.sql.types._
```

Took 1 min 2 sec. Last updated by anonymous at December 28 2021, 5:48:56 PM.

```
val schemaChurn = new StructType().add("RowNumber", IntegerType).
add("CustomerId", IntegerType).
add("Surname", StringType).
add("CreditScore", IntegerType).
add("Geography", StringType).
add("Gender", StringType).
add("Age", IntegerType).
add("Tenure", IntegerType).
add("Balance", DoubleType).
add("NumOfProducts", IntegerType).
add("HasCrCard", IntegerType).
add("IsActiveMember", IntegerType).
add("EstimatedSalary", DoubleType).
add("Exited", IntegerType)
```

FINISHED

```
schemaChurn: org.apache.spark.sql.types.StructType = StructType(StructField(RowNumber,IntegerType,tru
e), StructField(CustomerId,IntegerType,true), StructField(Surname,StringType,true), StructField(Credit
Score,IntegerType,true), StructField(Geography,StringType,true), StructField(Gender,StringType,true),
 StructField(Age,IntegerType,true), StructField(Tenure,IntegerType,true), StructField(Balance,DoubleTy
pe,true), StructField(NumOfProducts,IntegerType,true), StructField(HasCrCard,IntegerType,true), Struct
Field(IsActiveMember,IntegerType,true), StructField(EstimatedSalary,DoubleType,true), StructField(Exit
ed,IntegerType,true))
```

Took 2 sec. Last updated by anonymous at December 28 2021, 5:49:58 PM.

```
val df = spark.read.format("csv").schema(schemaChurn).option("header","true").load("hdfs://2a3314f957
```

FINISHED

```
df: org.apache.spark.sql.DataFrame = [RowNumber: int, CustomerId: int ... 12 more fields]
```

Took 3 sec. Last updated by anonymous at December 28 2021, 5:50:56 PM.

```
val dfraw = df.drop("RowNumber","CustomerId","Surname").withColumn("label", $"Exited")
```

FINISHED

```
dfraw: org.apache.spark.sql.DataFrame = [CreditScore: int, Geography: string ... 10 more fields]
```

Took 2 sec. Last updated by anonymous at December 28 2021, 5:51:08 PM.

```
import org.apache.spark.ml.feature.{StringIndexer, OneHotEncoder, VectorAssembler}
val dfrawIndexer1 = new StringIndexer().setInputCol("Geography").setOutputCol("GeographyCat")
val dfrawIndexer2 = new StringIndexer().setInputCol("Gender").setOutputCol("GenderCat")
```

FINISHED

```
import org.apache.spark.ml.feature.{StringIndexer, OneHotEncoder, VectorAssembler}
dfrawIndexer1: org.apache.spark.ml.feature.StringIndexer = strIdx_05fb6538bd5c
dfrawIndexer2: org.apache.spark.ml.feature.StringIndexer = strIdx_b39f2f1b5a2b
```

Took 1 sec. Last updated by anonymous at December 28 2021, 5:51:16 PM.

first session

```
val dfrawIndexer11 = new OneHotEncoder().setInputCol("GeographyCat").setOutputCol("GeographyVect1")
val dfrawIndexer21 = new OneHotEncoder().setInputCol("GenderCat").setOutputCol("GenderVect")
```

FINISHED

```
warning: there were two deprecation warnings; re-run with -deprecation for details
dfrawIndexer11: org.apache.spark.ml.feature.OneHotEncoder = oneHot_285bbf36fb8a
dfrawIndexer21: org.apache.spark.ml.feature.OneHotEncoder = oneHot_56ea756af0ec
```

Took 0 sec. Last updated by anonymous at December 28 2021, 5:51:24 PM.

```
val va = new VectorAssembler().setOutputCol("features").setInputCols(Array("CreditScore","GeographyVect
    ,"IsActiveMember","EstimatedSalary"))
```

ABORT

```
va: org.apache.spark.ml.feature.VectorAssembler = vecAssembler_0b043472e325
```

Took 1 sec. Last updated by anonymous at December 28 2021, 5:51:32 PM.

```
import org.apache.spark.ml.feature.StandardScaler
val stdScaler = new StandardScaler().
setWithStd(true).
setWithMean(true).
setInputCol("features").
setOutputCol("scaledFeatures")
```

FINISHED

```
import org.apache.spark.ml.feature.StandardScaler
stdScaler: org.apache.spark.ml.feature.StandardScaler = stdScal_e3538e70e274
```

Took 1 sec. Last updated by anonymous at December 28 2021, 5:51:41 PM.

```
import org.apache.spark.ml.classification.LogisticRegression
val lr = new LogisticRegression
lr.setRegParam(0.01).setMaxIter(500).setFitIntercept(true).setFeaturesCol("scaledFeatures")
```

FINISHED

```
import org.apache.spark.ml.classification.LogisticRegression
lr: org.apache.spark.ml.classification.LogisticRegression = logreg_693dd3cb41f3
res1: org.apache.spark.ml.classification.LogisticRegression = logreg_693dd3cb41f3
```

Took 1 sec. Last updated by anonymous at December 28 2021, 5:51:49 PM.

```
import org.apache.spark.ml.Pipeline
val pipeline = new Pipeline().setStages(Array(dfrawIndexer1,dfrawIndexer2,dfrawIndexer11,dfrawIndexer2
```

FINISHED

```
import org.apache.spark.ml.Pipeline
pipeline: org.apache.spark.ml.Pipeline = pipeline_f5c3f16f9cae
```

Took 1 sec. Last updated by anonymous at December 28 2021, 5:51:56 PM.

```
val Array(trainingData, testData) = dfraw.randomSplit(Array(0.7,0.3),11L)

val model = pipeline.fit(trainingData)
```

SPARK JOB FINISHED

```
trainingData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [CreditScore: int, Geography: s
tring ... 10 more fields]
testData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [CreditScore: int, Geography: strin
g ... 10 more fields]
model: org.apache.spark.ml.PipelineModel = pipeline_f5c3f16f9cae
```

Took 27 sec. Last updated by anonymous at December 28 2021, 5:53:08 PM.

first session

```
val pred = model.transform(testData)
```

FINISHED

```
pred: org.apache.spark.sql.DataFrame = [CreditScore: int, Geography: string ... 19 more fields]
```

Took 0 sec. Last updated by anonymous at December 28 2021, 5:53:15 PM.

```
import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
val bceval = new BinaryClassificationEvaluator()

bceval.evaluate(pred)
```

SPARK JOB FINISHED

```
import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
bceval: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_a197fb806547
res2: Double = 0.7588551798048759
```

Took 5 sec. Last updated by anonymous at December 28 2021, 5:53:30 PM.

```
import org.apache.spark.ml.classification.LogisticRegressionModel

val lrmodel = model.stages(6).asInstanceOf[LogisticRegressionModel]

println(s"LR Model coefficients:\n${lrmodel.coefficients.toArray.mkString("\n")}")
```

FINISHED

```
LR Model coefficients:
-0.04687018669513266
-0.03235338277593241
0.30310830705423825
-0.24368790140026417
0.7160853652899726
-0.06519910614543586
0.15529435849765294
-0.034514927609777195
-0.015782223254386666
-0.4822332145289691
0.012215544968639276
import org.apache.spark.ml.classification.LogisticRegressionModel
lrmodel: org.apache.spark.ml.classification.LogisticRegressionModel = logreg_693dd3cb41f3
```

Took 1 sec. Last updated by anonymous at December 28 2021, 5:53:42 PM.

READY