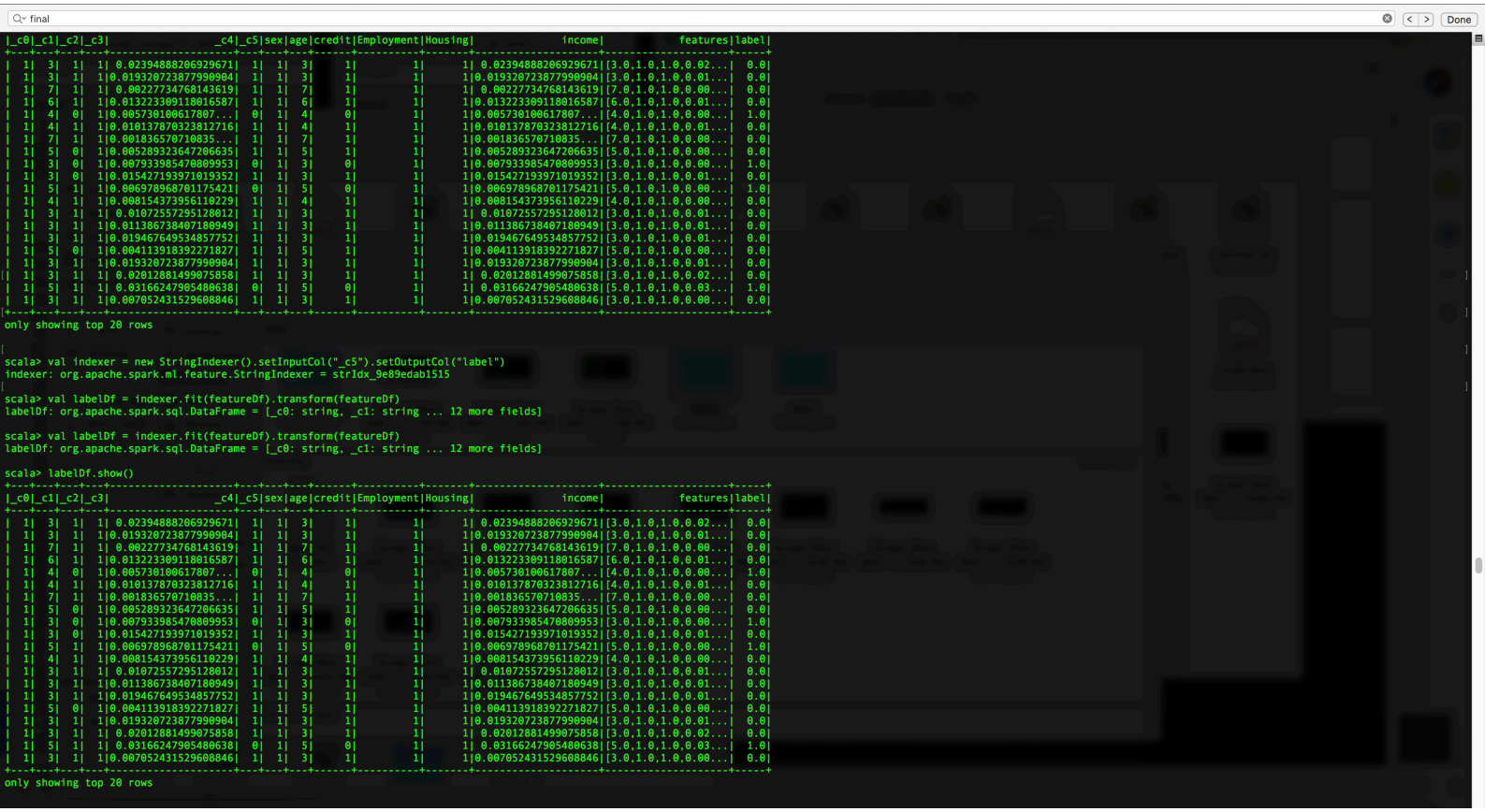
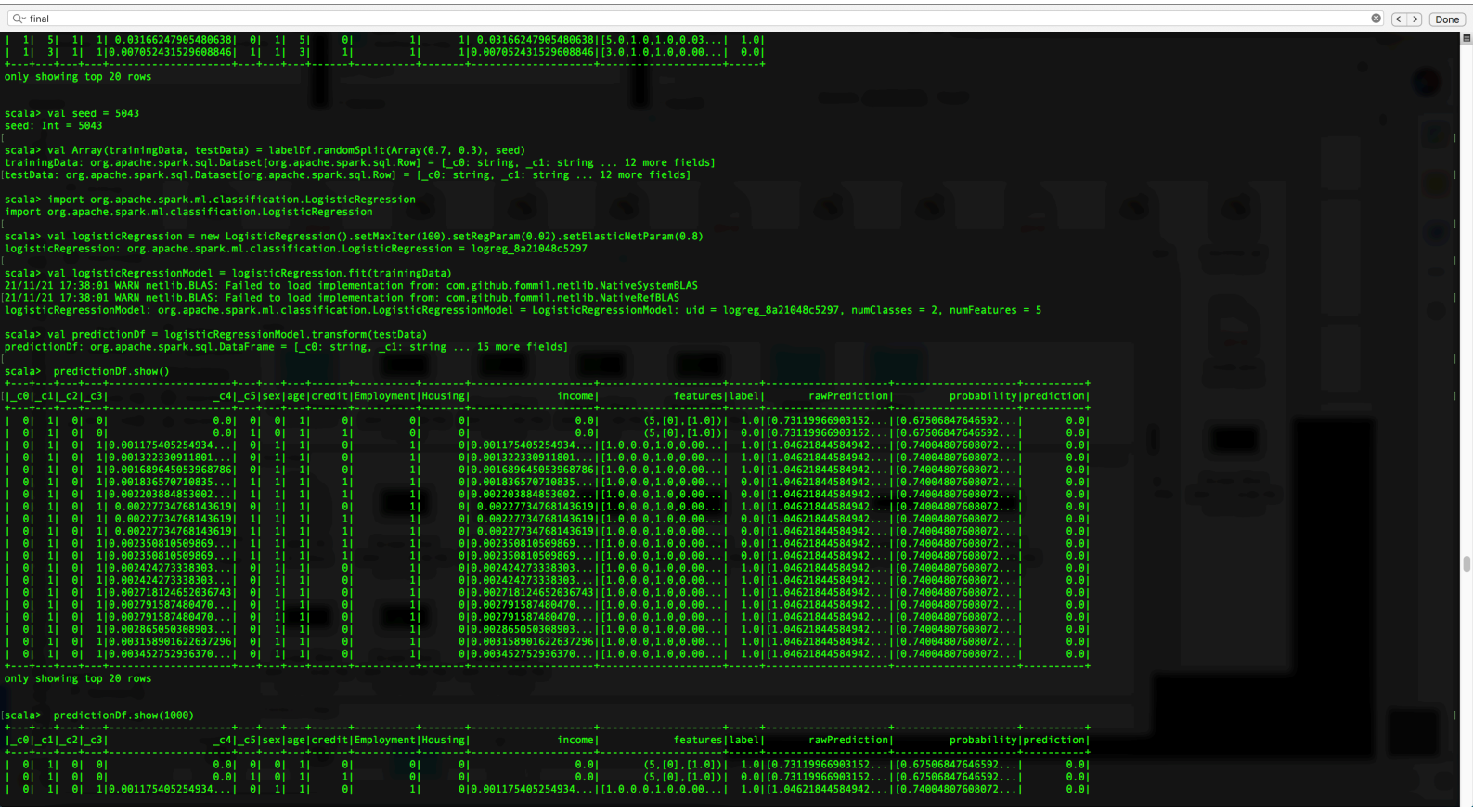
```
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.0-cdh6.3.4
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_181)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.ml.feature.VectorAssembler
import org.apache.spark.ml.feature.VectorAssembler

scala> import org.apache.spark.sql.types._
import org.apache.spark.sql.types._

scala>  var df = spark.read.option("header", "false").csv("Final_Project/all_input/finalcleandata.txt")
df: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 4 more fields]

scala> df.show()
+---+---+---+---+--------------------+---+
|_c0|_c1|_c2|_c3|                 _c4|_c5|
+---+---+---+---+--------------------+---+
|  1|  3|  1|  1| 0.02394888206929671|  1|
|  1|  3|  1|  1|0.019320723877990904|  1|
|  1|  7|  1|  1| 0.002277347681436l9|  1|
|  1|  6|  1|  1|0.013223309118016587|  1|
|  1|  4|  0|  1|0.005730100617807...|  0|
|  1|  4|  1|  1|0.010137070323812716|  1|
|  1|  7|  1|  1|0.001836570710835...|  1|
|  1|  5|  0|  1|0.005289323647206635|  1|
|  1|  3|  0|  1|0.007933985470809953|  0|
|  1|  3|  0|  1|0.015427193971019352|  1|
|  1|  5|  1|  1|0.006978968701175421|  0|
|  1|  4|  1|  1|0.008154373956110229|  1|
|  1|  3|  1|  1| 0.010725572951280l2|  1|
|  1|  3|  1|  1|0.011386738407180949|  1|
|  1|  3|  1|  1|0.019467649534857752|  1|
|  1|  5|  0|  1|0.004113918392271827|  1|
|  1|  3|  1|  1|0.019320723877990904|  1|
|  1|  3|  1|  1| 0.020128814990758l8|  1|
|  1|  5|  1|  1| 0.03166247905480638|  0|
|  1|  3|  1|  1|0.007052431529608846|  1|
+---+---+---+---+--------------------+---+
only showing top 20 rows

scala> var df_final_1 =  df.withColumn("sex",col("_c3").cast(IntegerType)).withColumn("age",col("_c1").cast(IntegerType)).withColumn("credit",col("_c5").cast(IntegerType)).withColumn("Employment",col("_c3").cast(IntegerType)).withColumn
("Housing",col("_c0").cast(IntegerType)).withColumn("income",col("_c4").cast(DoubleType))
df_final_1: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 10 more fields]

scala> df_final_1.show()
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|
```

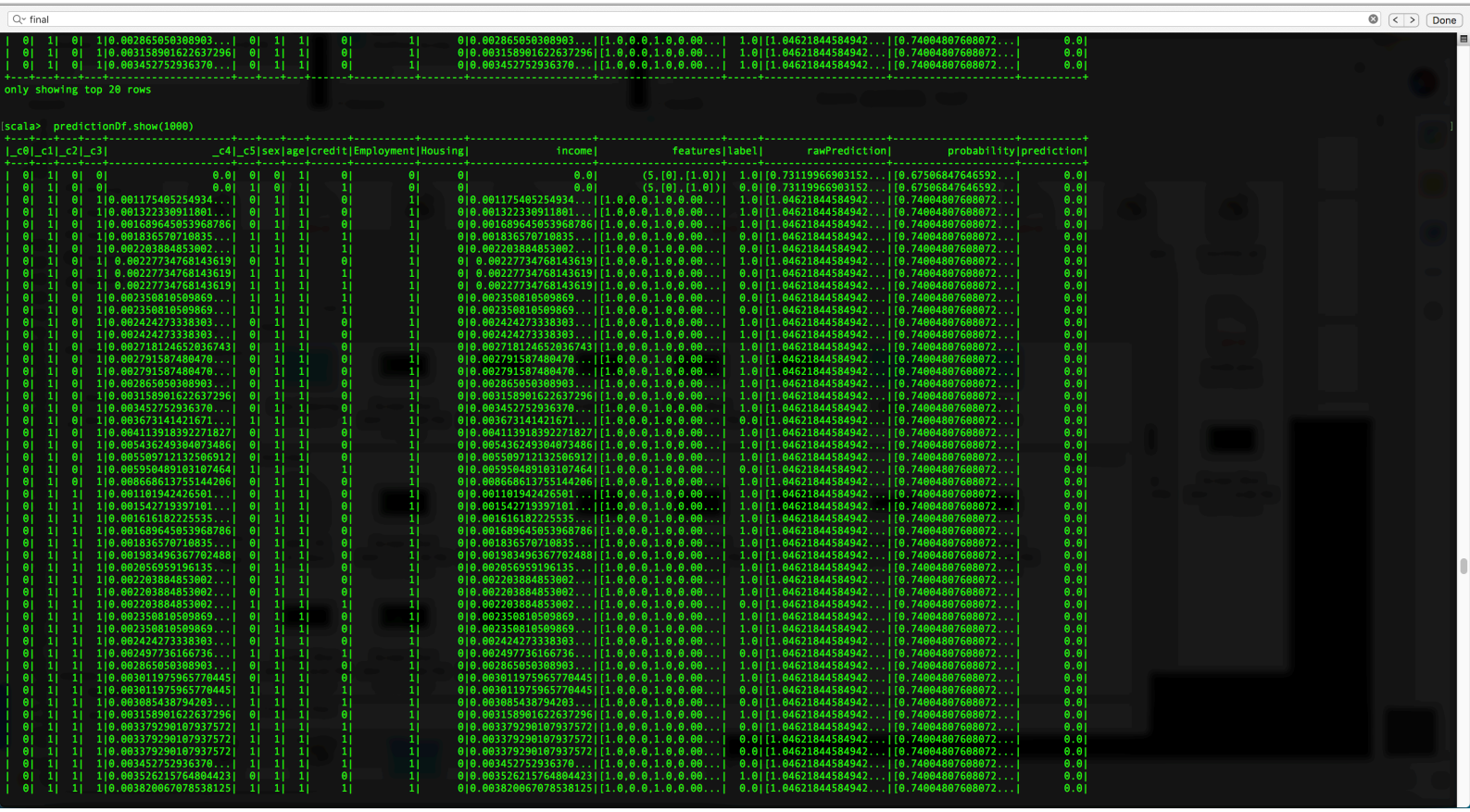
```
|  1|  5|  1|  1| 0.03166247905480638|  0|
|  1|  3|  1|  1|0.007052431529608846|  1|
+---+---+---+---+--------------------+---+
only showing top 20 rows

scala> var df_final_1 =  df.withColumn("sex",col("_c3").cast(IntegerType)).withColumn("age",col("_c1").cast(IntegerType)).withColumn("credit",col("_c5").cast(IntegerType)).withColumn("Employment",col("_c3").cast(IntegerType)).withColumn("Housing",col("_c0").cast(IntegerType)).withColumn("income",col("_c4").cast(DoubleType))
df_final_1: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 10 more fields]

scala> df_final_1.show()
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+
|  1|  3|  1|  1| 0.02394888206929671|  1|  1|  3|     1|         1|      1| 0.02394888206929671|
|  1|  3|  1|  1|0.019320723877990904|  1|  1|  3|     1|         1|      1|0.019320723877990904|
|  1|  7|  1|  1| 0.00227734768143619|  1|  1|  7|     1|         1|      1| 0.00227734768143619|
|  1|  6|  1|  1|0.013223309118016587|  1|  1|  6|     1|         1|      1|0.013223309118016587|
|  1|  4|  0|  1|0.005730100617807...|  0|  1|  4|     0|         1|      1|0.005730100617807...|
|  1|  4|  1|  1|0.010137870323812716|  1|  1|  4|     1|         1|      1|0.010137870323812716|
|  1|  7|  1|  1|0.001836570710835...|  1|  1|  7|     1|         1|      1|0.001836570710835...|
|  1|  5|  0|  1|0.005289323647206635|  1|  1|  5|     1|         1|      1|0.005289323647206635|
|  1|  3|  0|  1|0.007933985470809953|  0|  1|  3|     0|         1|      1|0.007933985470809953|
|  1|  3|  0|  1|0.015427193971019352|  1|  1|  3|     1|         1|      1|0.015427193971019352|
|  1|  5|  1|  1|0.006978968701175421|  0|  1|  5|     0|         1|      1|0.006978968701175421|
|  1|  4|  1|  1|0.008154373956110229|  1|  1|  4|     1|         1|      1|0.008154373956110229|
|  1|  3|  1|  1| 0.01072557295128012|  1|  1|  3|     1|         1|      1| 0.01072557295128012|
|  1|  3|  1|  1|0.011386738407180949|  1|  1|  3|     1|         1|      1|0.011386738407180949|
|  1|  3|  1|  1|0.019467649534857752|  1|  1|  3|     1|         1|      1|0.019467649534857752|
|  1|  5|  0|  1|0.004113918392271827|  1|  1|  5|     1|         1|      1|0.004113918392271827|
|  1|  3|  1|  1|0.019320723877990904|  1|  1|  3|     1|         1|      1|0.019320723877990904|
|  1|  3|  1|  1| 0.02012881499075858|  1|  1|  3|     1|         1|      1| 0.02012881499075858|
|  1|  5|  1|  1| 0.03166247905480638|  0|  1|  5|     0|         1|      1| 0.03166247905480638|
|  1|  3|  1|  1|0.007052431529608846|  1|  1|  3|     1|         1|      1|0.007052431529608846|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+
only showing top 20 rows

scala> val cols = Array("age","Housing","Employment","income","sex")
cols: Array[String] = Array(age, Housing, Employment, income, sex)

scala>  val assembler = new VectorAssembler().setInputCols(cols).setOutputCol("features")
assembler: org.apache.spark.ml.feature.VectorAssembler = vecAssembler_f1b23933f5d3

scala>  val featureDf = assembler.transform(df_final_1)
featureDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 11 more fields]

scala> featureDf.show()
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+
|  1|  3|  1|  1| 0.02394888206929671|  1|  1|  3|     1|         1|      1| 0.02394888206929671|[3.0,1.0,1.0,0.02...|
|  1|  3|  1|  1|0.019320723877990904|  1|  1|  3|     1|         1|      1|0.019320723877990904|[3.0,1.0,1.0,0.01...|
|  1|  7|  1|  1| 0.00227734768143619|  1|  1|  7|     1|         1|      1| 0.00227734768143619|[7.0,1.0,1.0,0.00...|
|  1|  6|  1|  1|0.013223309118016587|  1|  1|  6|     1|         1|      1|0.013223309118016587|[6.0,1.0,1.0,0.01...|
|  1|  4|  0|  1|0.005730100617807...|  0|  1|  4|     0|         1|      1|0.005730100617807...|[4.0,1.0,1.0,0.00...|
|  1|  4|  1|  1|0.010137870323812716|  1|  1|  4|     1|         1|      1|0.010137870323812716|[4.0,1.0,1.0,0.01...|
|  1|  7|  1|  1|0.001836570710835...|  1|  1|  7|     1|         1|      1|0.001836570710835...|[7.0,1.0,1.0,0.00...|
|  1|  5|  0|  1|0.005289323647206635|  1|  1|  5|     1|         1|      1|0.005289323647206635|[5.0,1.0,1.0,0.00...|
```

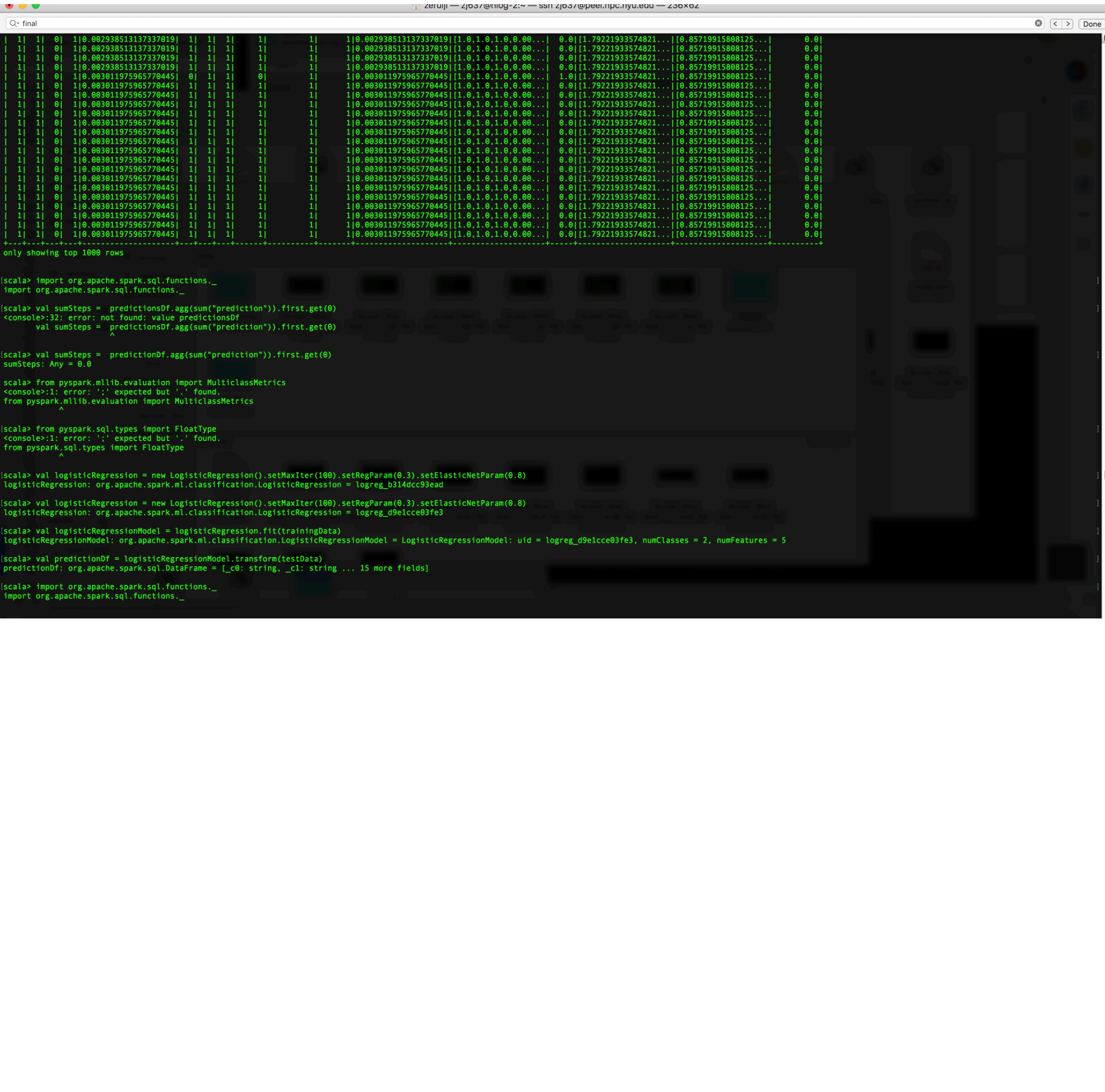
```
assembler: org.apache.spark.ml.feature.VectorAssembler = vecAssembler_f1b23933f5d3

scala>  val featureDf = assembler.transform(df_final_1)
featureDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 11 more fields]

scala> featureDf.show()
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+
|  1|  3|  1|  1| 0.02394888206929671|  1|  1|  3|     1|         1|      1| 0.02394888206929671|[3.0,1.0,1.0,0.02...|
|  1|  3|  1|  1|0.019320723877990904|  1|  1|  3|     1|         1|      1|0.019320723877990904|[3.0,1.0,1.0,0.01...|
|  1|  7|  1|  1| 0.00227734768143619|  1|  1|  7|     1|         1|      1| 0.00227734768143619|[7.0,1.0,1.0,0.00...|
|  1|  6|  1|  1|0.013223309118016587|  1|  1|  6|     1|         1|      1|0.013223309118016587|[6.0,1.0,1.0,0.01...|
|  1|  4|  0|  1|0.005730100617807...|  0|  1|  4|     0|         1|      1|0.005730100617807...|[4.0,1.0,1.0,0.00...|
|  1|  4|  1|  1|0.010137870323812716|  1|  1|  4|     1|         1|      1|0.010137870323812716|[4.0,1.0,1.0,0.01...|
|  1|  7|  1|  1|0.001836570710835...|  1|  1|  7|     1|         1|      1|0.001836570710835...|[7.0,1.0,1.0,0.00...|
|  1|  5|  0|  1|0.005289323647206635|  1|  1|  5|     1|         1|      1|0.005289323647206635|[5.0,1.0,1.0,0.00...|
|  1|  3|  0|  1|0.007933985470809953|  0|  1|  3|     0|         1|      1|0.007933985470809953|[3.0,1.0,1.0,0.00...|
|  1|  3|  0|  1|0.015427193971019352|  1|  1|  3|     1|         1|      1|0.015427193971019352|[3.0,1.0,1.0,0.01...|
|  1|  5|  1|  1|0.006978968701175421|  0|  1|  5|     0|         1|      1|0.006978968701175421|[5.0,1.0,1.0,0.00...|
|  1|  4|  1|  1|0.008154373956110229|  1|  1|  4|     1|         1|      1|0.008154373956110229|[4.0,1.0,1.0,0.00...|
|  1|  3|  1|  1| 0.01072557295128012|  1|  1|  3|     1|         1|      1| 0.01072557295128012|[3.0,1.0,1.0,0.01...|
|  1|  3|  1|  1|0.011386738407180949|  1|  1|  3|     1|         1|      1|0.011386738407180949|[3.0,1.0,1.0,0.01...|
|  1|  3|  1|  1|0.019467649534857752|  1|  1|  3|     1|         1|      1|0.019467649534857752|[3.0,1.0,1.0,0.01...|
|  1|  5|  0|  1|0.004113918392271827|  1|  1|  5|     1|         1|      1|0.004113918392271827|[5.0,1.0,1.0,0.00...|
|  1|  3|  1|  1|0.019320723877990904|  1|  1|  3|     1|         1|      1|0.019320723877990904|[3.0,1.0,1.0,0.01...|
|  1|  3|  1|  1| 0.02012881499075858|  1|  1|  3|     1|         1|      1| 0.02012881499075858|[3.0,1.0,1.0,0.02...|
|  1|  5|  1|  1| 0.03166247905480638|  0|  1|  5|     0|         1|      1| 0.03166247905480638|[5.0,1.0,1.0,0.03...|
|  1|  3|  1|  1|0.007052431529608846|  1|  1|  3|     1|         1|      1|0.007052431529608846|[3.0,1.0,1.0,0.00...|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+
only showing top 20 rows

scala> import org.apache.spark.ml.feature.StringIndexer
import org.apache.spark.ml.feature.StringIndexer

scala> val indexer = new StringIndexer().setInputCol("credit").setOutputCol("label")
indexer: org.apache.spark.ml.feature.StringIndexer = strIdx_3a9604d907a0

scala> val labelDf = indexer.fit(featureDf).transform(featureDf)
labelDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 12 more fields]

scala> labelDf.show()
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|label|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+
|  1|  3|  1|  1| 0.02394888206929671|  1|  1|  3|     1|         1|      1| 0.02394888206929671|[3.0,1.0,1.0,0.02...|  0.0|
|  1|  3|  1|  1|0.019320723877990904|  1|  1|  3|     1|         1|      1|0.019320723877990904|[3.0,1.0,1.0,0.01...|  0.0|
|  1|  7|  1|  1| 0.00227734768143619|  1|  1|  7|     1|         1|      1| 0.00227734768143619|[7.0,1.0,1.0,0.00...|  0.0|
|  1|  6|  1|  1|0.013223309118016587|  1|  1|  6|     1|         1|      1|0.013223309118016587|[6.0,1.0,1.0,0.01...|  0.0|
|  1|  4|  0|  1|0.005730100617807...|  0|  1|  4|     0|         1|      1|0.005730100617807...|[4.0,1.0,1.0,0.00...|  1.0|
|  1|  4|  1|  1|0.010137870323812716|  1|  1|  4|     1|         1|      1|0.010137870323812716|[4.0,1.0,1.0,0.01...|  0.0|
|  1|  7|  1|  1|0.001836570710835...|  1|  1|  7|     1|         1|      1|0.001836570710835...|[7.0,1.0,1.0,0.00...|  0.0|
|  1|  5|  0|  1|0.005289323647206635|  1|  1|  5|     1|         1|      1|0.005289323647206635|[5.0,1.0,1.0,0.00...|  0.0|
|  1|  3|  0|  1|0.007933985470809953|  0|  1|  3|     0|         1|      1|0.007933985470809953|[3.0,1.0,1.0,0.00...|  1.0|
|  1|  3|  0|  1|0.015427193971019352|  1|  1|  3|     1|         1|      1|0.015427193971019352|[3.0,1.0,1.0,0.01...|  0.0|
|  1|  5|  1|  1|0.006978968701175421|  0|  1|  5|     0|         1|      1|0.006978968701175421|[5.0,1.0,1.0,0.00...|  1.0|
|  1|  4|  1|  1|0.008154373956110229|  1|  1|  4|     1|         1|      1|0.008154373956110229|[4.0,1.0,1.0,0.00...|  0.0|
|  1|  3|  1|  1| 0.01072557295128012|  1|  1|  3|     1|         1|      1| 0.01072557295128012|[3.0,1.0,1.0,0.01...|  0.0|
|  1|  3|  1|  1|0.011386738407180949|  1|  1|  3|     1|         1|      1|0.011386738407180949|[3.0,1.0,1.0,0.01...|  0.0|
|  1|  3|  1|  1|0.019467649534857752|  1|  1|  3|     1|         1|      1|0.019467649534857752|[3.0,1.0,1.0,0.01...|  0.0|
|  1|  5|  0|  1|0.004113918392271827|  1|  1|  5|     1|         1|      1|0.004113918392271827|[5.0,1.0,1.0,0.00...|  0.0|
```

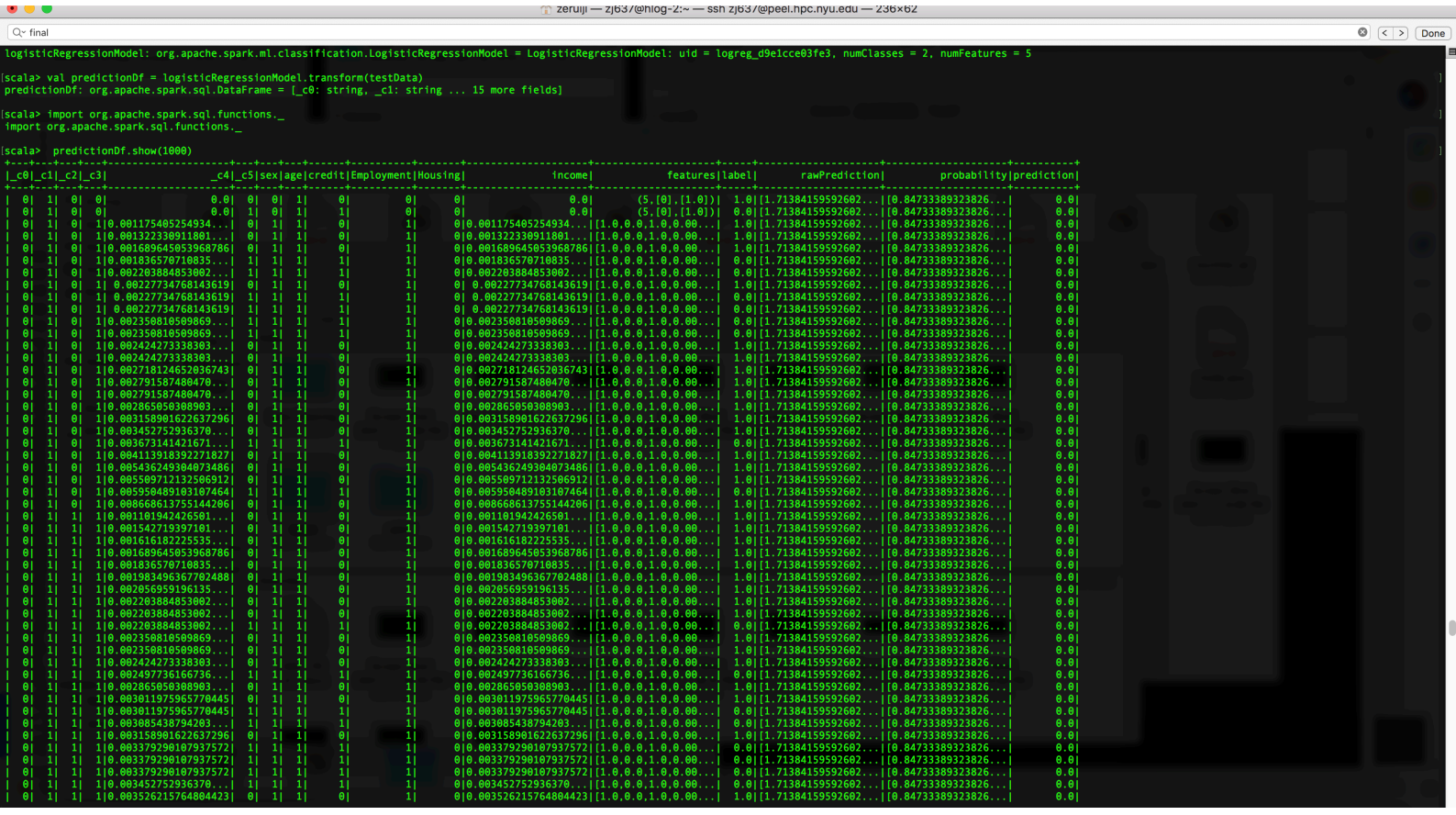

final
Done
|_c0|_c1|_c2|_c3|_c4|_c5|sex|age|credit|Employment|Housing|income|features|label|
only showing top 20 rows
scala> val indexer = new StringIndexer().setInputCol("_c5").setOutputCol("label")
indexer: org.apache.spark.ml.feature.StringIndexer = strIdx_9e89edab1515
scala> val labelDf = indexer.fit(featureDf).transform(featureDf)
labelDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 12 more fields]
scala> val labelDf = indexer.fit(featureDf).transform(featureDf)
labelDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 12 more fields]
scala> labelDf.show()
|_c0|_c1|_c2|_c3|_c4|_c5|sex|age|credit|Employment|Housing|income|features|label|
only showing top 20 rows

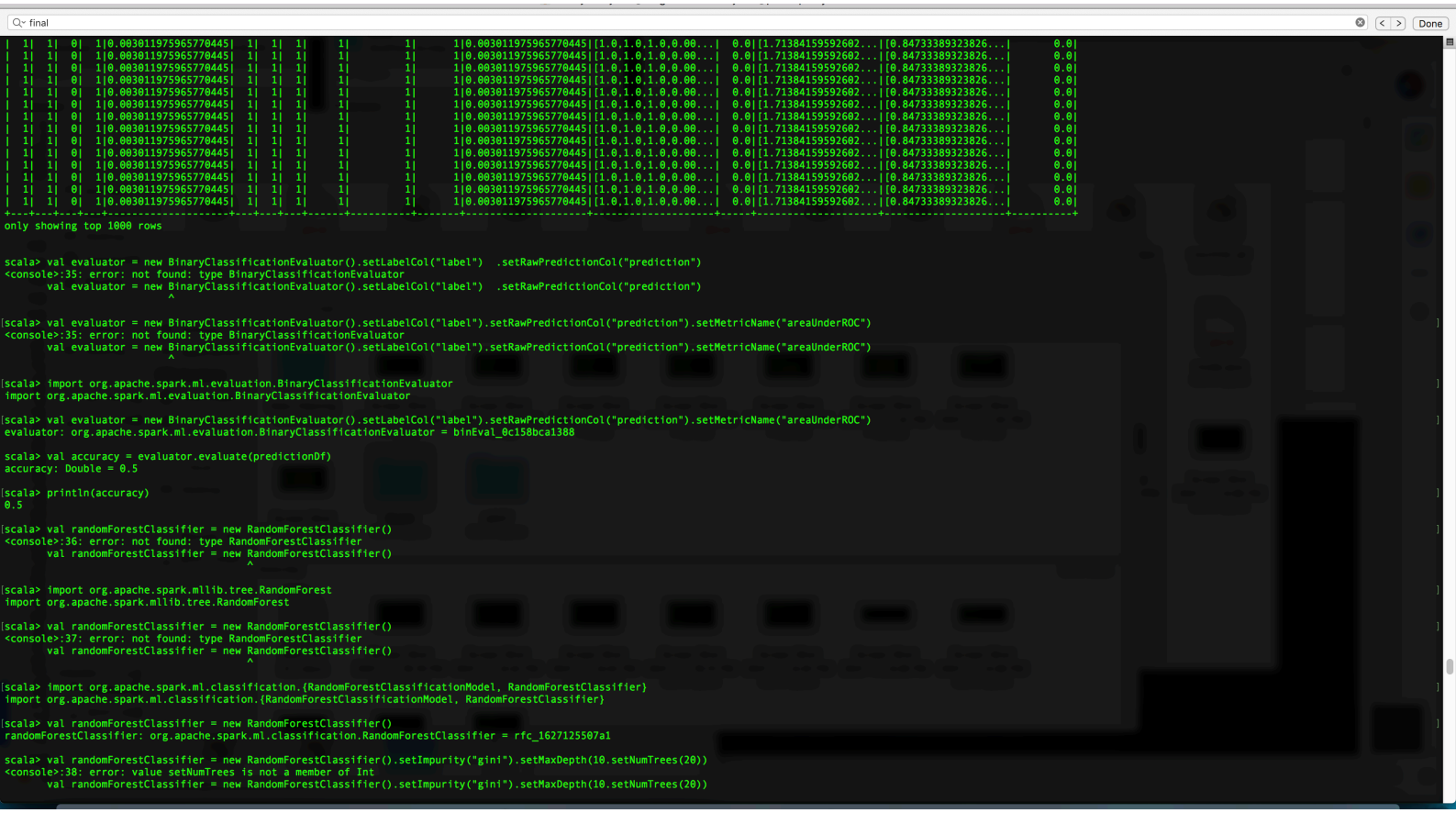


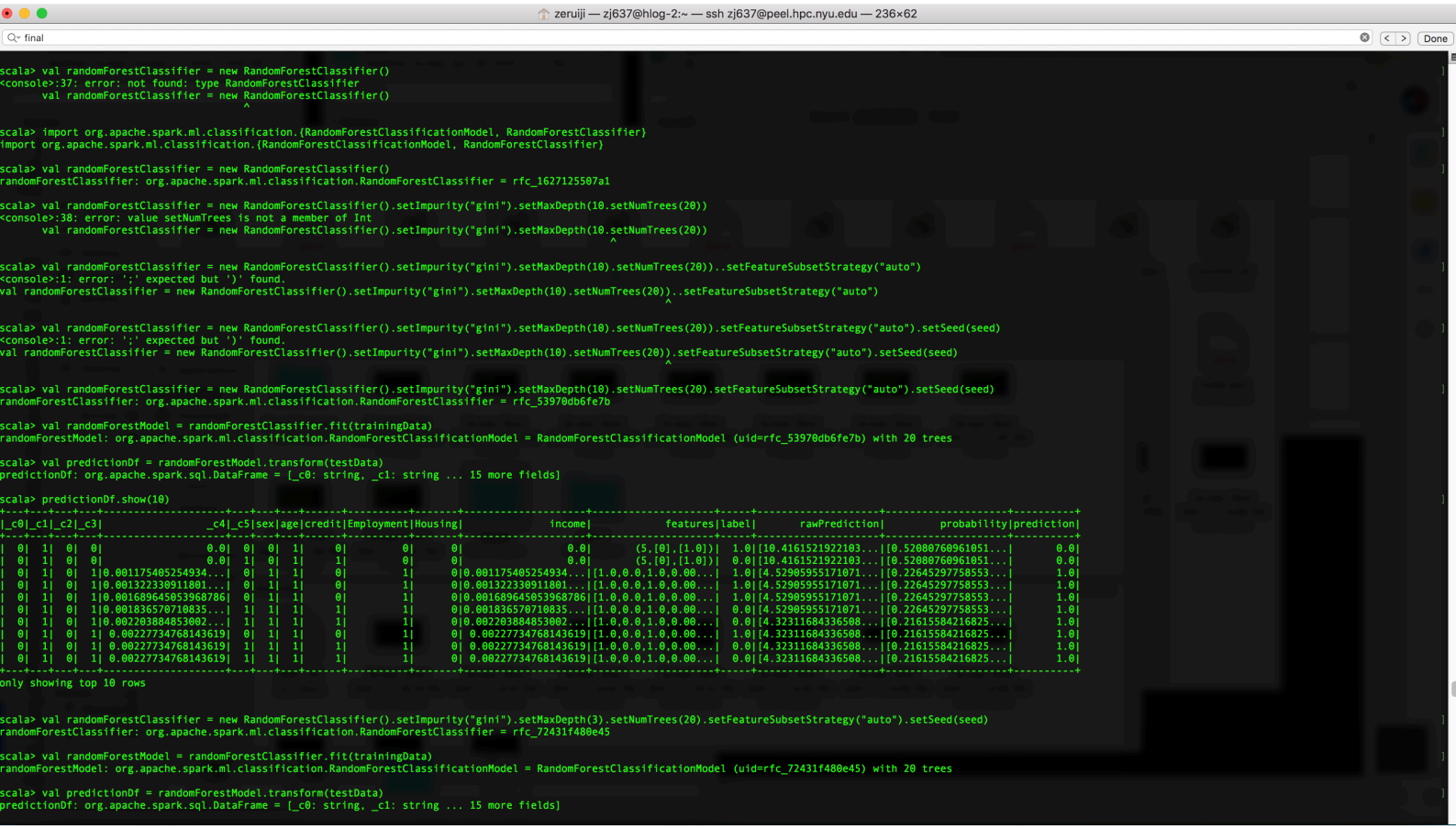
```
|  1|  5|  1|  1| 0.031662479054806381  0|  1|  5|     0|         1|      1| 0.031662479054806381[5.0,1.0,1.0,0.03...|  1.0|
|  1|  3|  1|  1|0.0070524315296088461  1|  1|  3|     1|         1|      1|0.0070524315296088461[3.0,1.0,1.0,0.00...|  0.0|
+---+---+---+---+----------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+
only showing top 20 rows


scala> val seed = 5043
seed: Int = 5043

scala> val Array(trainingData, testData) = labelDf.randomSplit(Array(0.7, 0.3), seed)
trainingData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [_c0: string, _c1: string ... 12 more fields]
testData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [_c0: string, _c1: string ... 12 more fields]

scala> import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.classification.LogisticRegression

scala> val logisticRegression = new LogisticRegression().setMaxIter(100).setRegParam(0.02).setElasticNetParam(0.8)
logisticRegression: org.apache.spark.ml.classification.LogisticRegression = logreg_8a21048c5297

scala> val logisticRegressionModel = logisticRegression.fit(trainingData)
21/11/21 17:38:01 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
21/11/21 17:38:01 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
logisticRegressionModel: org.apache.spark.ml.classification.LogisticRegressionModel = LogisticRegressionModel: uid = logreg_8a21048c5297, numClasses = 2, numFeatures = 5

scala> val predictionDf = logisticRegressionModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]

scala>  predictionDf.show()
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|label|       rawPrediction|         probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|  0|  1|  0|  0|                 0.0|  0|  0|  1|     0|         0|      0|                 0.0|       (5,[0],[1.0])|  1.0|[0.73119966903152...|[0.67506847646592...|       0.0|
|  0|  1|  0|  0|                 0.0|  1|  0|  1|     1|         0|      0|                 0.0|       (5,[0],[1.0])|  0.0|[0.73119966903152...|[0.67506847646592...|       0.0|
|  0|  1|  0|  1|0.001175405254934...|  0|  1|  1|     0|         1|      0|0.001175405254934...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.001322330911801...|  0|  1|  1|     0|         1|      0|0.001322330911801...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.001689645053968786|  0|  1|  1|     0|         1|      0|0.001689645053968786|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.001836570710835...|  1|  1|  1|     1|         1|      0|0.001836570710835...|[1.0,0,0,1,0,0.00...|  0.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002203884853002...|  1|  1|  1|     1|         1|      0|0.002203884853002...|[1.0,0,0,1,0,0.00...|  0.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1| 0.002277347681436191  0|  1|  1|     0|         1|      0| 0.002277347681436191[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1| 0.002277347681436191  1|  1|  1|     1|         1|      0| 0.002277347681436191[1.0,0,0,1,0,0.00...|  0.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1| 0.002277347681436191  1|  1|  1|     1|         1|      0| 0.002277347681436191[1.0,0,0,1,0,0.00...|  0.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002350810509869...|  1|  1|  1|     1|         1|      0|0.002350810509869...|[1.0,0,0,1,0,0.00...|  0.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002350810509869...|  1|  1|  1|     1|         1|      0|0.002350810509869...|[1.0,0,0,1,0,0.00...|  0.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002424273338303...|  0|  1|  1|     0|         1|      0|0.002424273338303...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002424273338303...|  0|  1|  1|     0|         1|      0|0.002424273338303...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002718124652036743|  0|  1|  1|     0|         1|      0|0.002718124652036743|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002791587480470...|  0|  1|  1|     0|         1|      0|0.002791587480470...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002791587480470...|  0|  1|  1|     0|         1|      0|0.002791587480470...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.002865050308903...|  0|  1|  1|     0|         1|      0|0.002865050308903...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.003158901622637296|  0|  1|  1|     0|         1|      0|0.003158901622637296|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
|  0|  1|  0|  1|0.003452752936370...|  0|  1|  1|     0|         1|      0|0.003452752936370...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
only showing top 20 rows


scala>  predictionDf.show(1000)
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|label|       rawPrediction|         probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|  0|  1|  0|  0|                 0.0|  0|  0|  1|     0|         0|      0|                 0.0|       (5,[0],[1.0])|  1.0|[0.73119966903152...|[0.67506847646592...|       0.0|
|  0|  1|  0|  0|                 0.0|  1|  0|  1|     1|         0|      0|                 0.0|       (5,[0],[1.0])|  0.0|[0.73119966903152...|[0.67506847646592...|       0.0|
|  0|  1|  0|  1|0.001175405254934...|  0|  1|  1|     0|         1|      0|0.001175405254934...|[1.0,0,0,1,0,0.00...|  1.0|[1.04621844584942...|[0.74004807608072...|       0.0|
```

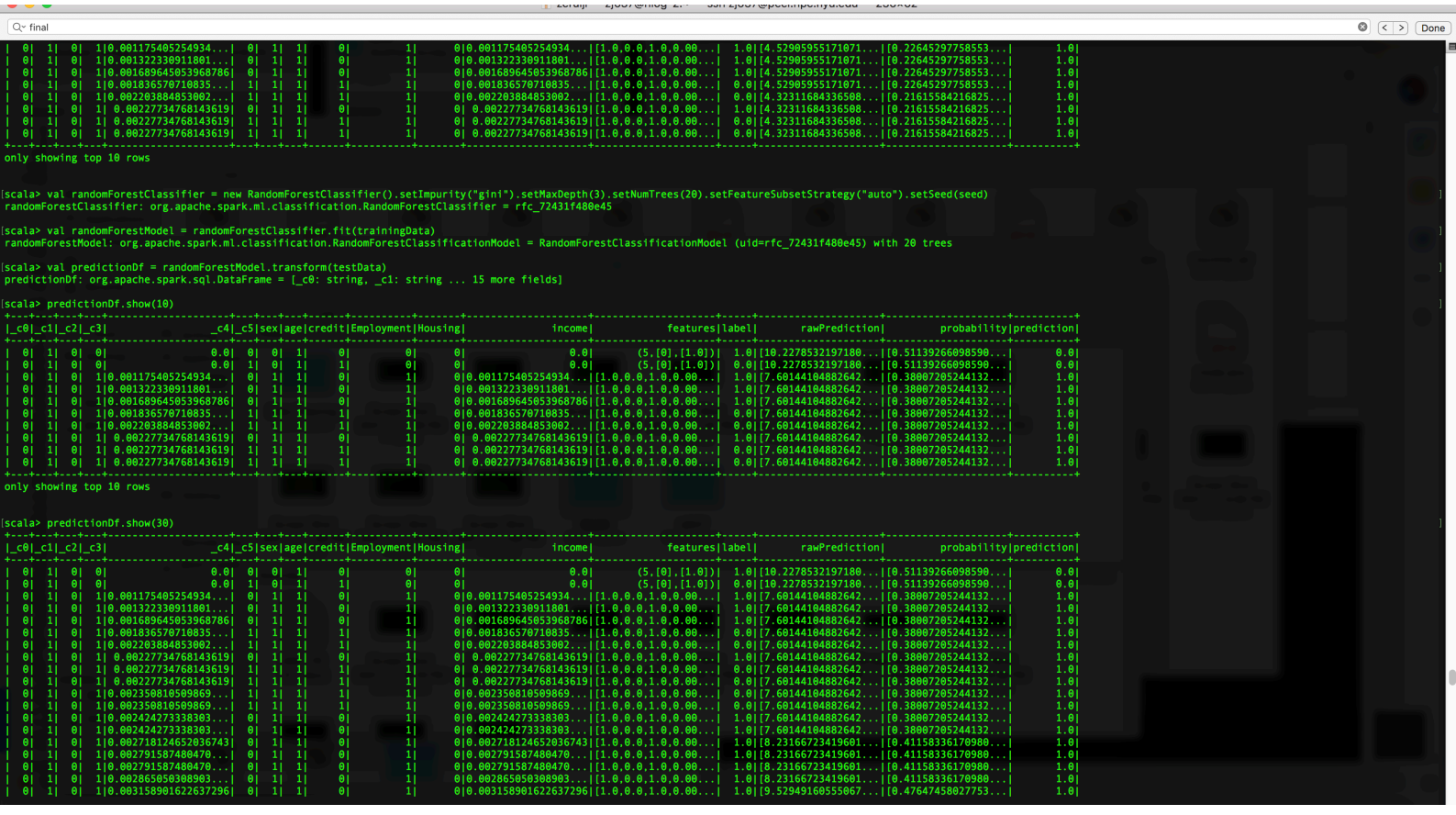

final
Done
only showing top 20 rows
scala> predictionDf.show(1000)

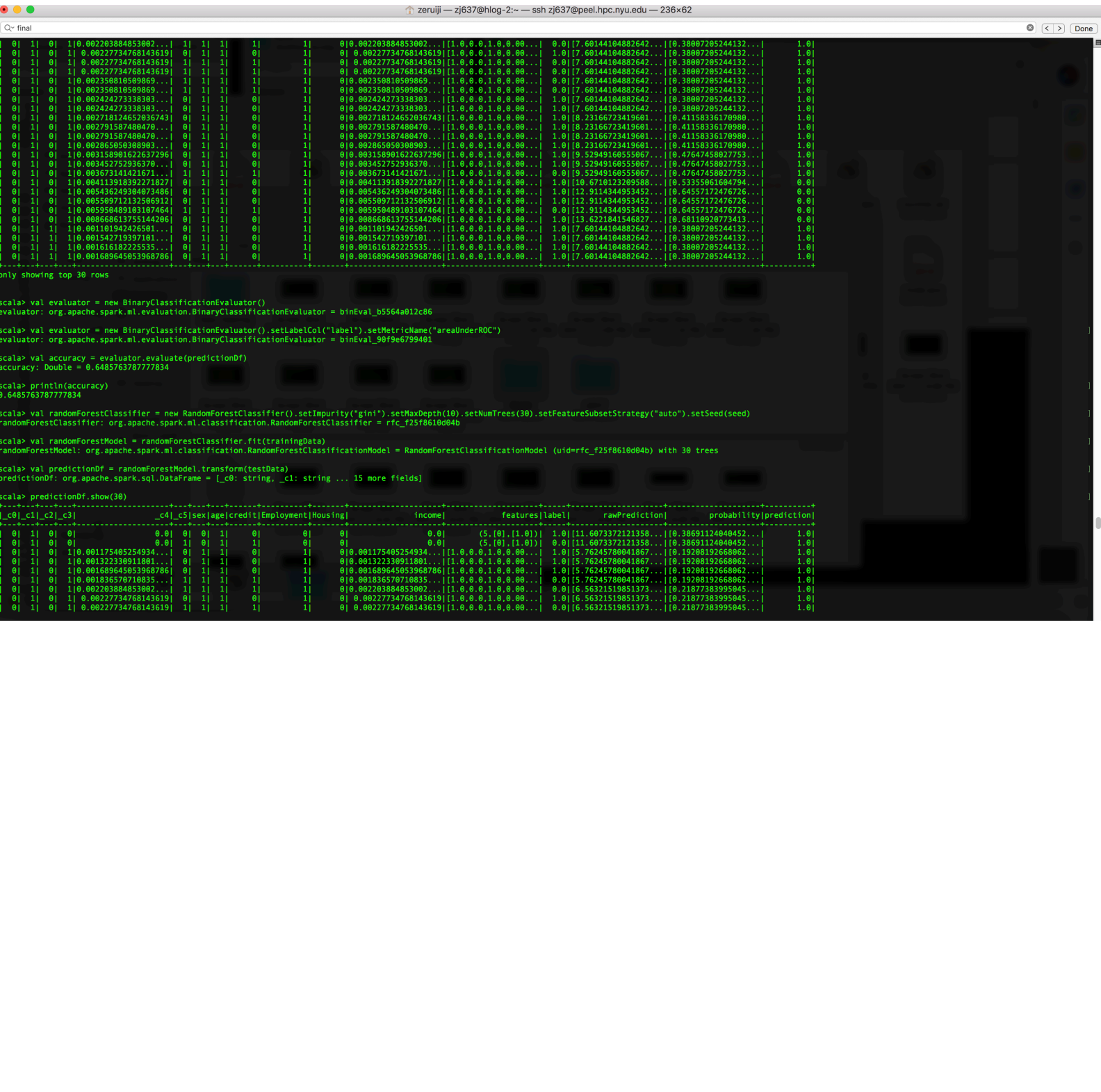

zeroji — zj637@nlog-2:~ — ssh zj637@peel.hpc.nyu.edu — 238×62
final
Done
| 1| 1| 0| 1|0.0029385131373337019| 1| 1| 1| 1| 1| 1|0.0029385131373337019|[1.0,1.0,1.0,0.00...| 0.0|[1.792219333574821...|[0.85719915808125...| 0.0|
only showing top 1000 rows
[scala> import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._
[scala> val sumSteps = predictionsDf.agg(sum("prediction")).first.get(0)
<console>:32: error: not found: value predictionsDf
val sumSteps = predictionsDf.agg(sum("prediction")).first.get(0)
^
[scala> val sumSteps = predictionDf.agg(sum("prediction")).first.get(0)
sumSteps: Any = 0.0
scala> from pyspark.mllib.evaluation import MulticlassMetrics
<console>:1: error: ';' expected but '.' found.
from pyspark.mllib.evaluation import MulticlassMetrics
^
[scala> from pyspark.sql.types import FloatType
<console>:1: error: ';' expected but '.' found.
from pyspark.sql.types import FloatType
^
[scala> val logisticRegression = new LogisticRegression().setMaxIter(100).setRegParam(0.3).setElasticNetParam(0.8)
logisticRegression: org.apache.spark.ml.classification.LogisticRegression = logreg_b314dcc93ead
[scala> val logisticRegression = new LogisticRegression().setMaxIter(100).setRegParam(0.3).setElasticNetParam(0.8)
logisticRegression: org.apache.spark.ml.classification.LogisticRegression = logreg_d9e1cce03fe3
[scala> val logisticRegressionModel = logisticRegression.fit(trainingData)
logisticRegressionModel: org.apache.spark.ml.classification.LogisticRegressionModel = LogisticRegressionModel: uid = logreg_d9e1cce03fe3, numClasses = 2, numFeatures = 5
[scala> val predictionDf = logisticRegressionModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]
[scala> import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._


zeruiji — zj637@hlog-2:~ — ssh zj637@peel.hpc.nyu.edu — 236×62
final
Done
logisticRegressionModel: org.apache.spark.ml.classification.LogisticRegressionModel = LogisticRegressionModel: uid = logreg_d9e1cce03fe3, numClasses = 2, numFeatures = 5
[scala> val predictionDf = logisticRegressionModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]
[scala> import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._
[scala> predictionDf.show(1000)
+---+---+---+---+--------------------+---+---+---+------+---------+-------+--------------------+--------------+----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3| _c4|_c5|sex|age|credit|Employment|Housing| income| features|label| rawPrediction| probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+---------+-------+--------------------+--------------+----+--------------------+--------------------+----------+
0	1	0	0	0.0	0	0	1	0	0	0	0.0	(5,[0],[1.0])	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	0	0.0	1	0	1	1	0	0	0.0	(5,[0],[1.0])	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0011754052549344...	0	1	1	0	1	0	0.0011754052549344...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0013223309118013...	0	1	1	0	1	0	0.0013223309118013...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0016896450539687...	0	1	1	0	1	0	0.0016896450539687...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0018365707108353...	1	1	1	1	1	0	0.0018365707108353...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0022038848530027...	1	1	1	1	1	0	0.0022038848530027...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.002277347681436...	0	1	1	0	1	0	0.002277347681436...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.002277347681436...	1	1	1	1	1	0	0.002277347681436...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.002277347681436...	1	1	1	1	1	0	0.002277347681436...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0023508105098693...	1	1	1	1	1	0	0.0023508105098693...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0023508105098693...	1	1	1	1	1	0	0.0023508105098693...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0024242473338303...	0	1	1	0	1	0	0.0024242473338303...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0024242473338303...	0	1	1	0	1	0	0.0024242473338303...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0027181246520367...	0	1	1	0	1	1	0.0027181246520367...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0027915874804705...	0	1	1	0	1	0	0.0027915874804705...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0027915874804705...	0	1	1	0	1	0	0.0027915874804705...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0028650503089033...	0	1	1	0	1	0	0.0028650503089033...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0031589016226372...	0	1	1	0	1	0	0.0031589016226372...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0034527529360370...	0	1	1	0	1	0	0.0034527529360370...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0036731414216715...	1	1	1	1	1	0	0.0036731414216715...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0041139183922718...	0	1	1	0	1	0	0.0041139183922718...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0054362493040734...	0	1	1	0	1	0	0.0054362493040734...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0055097121325069...	0	1	1	0	1	0	0.0055097121325069...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0059504891031074...	1	1	1	1	1	0	0.0059504891031074...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	0	1	0.0086686137551442...	0	1	1	0	1	0	0.0086686137551442...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0011019424265010...	0	1	1	0	1	0	0.0011019424265010...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0015427193970101...	0	1	1	0	1	0	0.0015427193970101...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0016161822255535...	0	1	1	0	1	0	0.0016161822255535...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0016896450539687...	0	1	1	0	1	0	0.0016896450539687...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0018365707108353...	0	1	1	0	1	0	0.0018365707108353...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0019834963677024...	0	1	1	0	1	0	0.0019834963677024...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0020569591961353...	0	1	1	0	1	0	0.0020569591961353...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0022038848530027...	0	1	1	0	1	0	0.0022038848530027...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0022038848530027...	0	1	1	0	1	1	0.0022038848530027...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0022038848530027...	1	1	1	1	1	1	0.0022038848530027...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0023508105098693...	0	1	1	0	1	0	0.0023508105098693...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0023508105098693...	0	1	1	0	1	0	0.0023508105098693...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0024242473338303...	0	1	1	0	1	0	0.0024242473338303...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0024977361667363...	1	1	1	1	1	0	0.0024977361667363...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0028650503089033...	0	1	1	0	1	0	0.0028650503089033...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0030119759657704...	0	1	1	0	1	0	0.0030119759657704...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0030119759657704...	1	1	1	1	1	0	0.0030119759657704...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0030854387942035...	1	1	1	1	1	0	0.0030854387942035...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0031589016226372...	0	1	1	0	1	0	0.0031589016226372...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0033792901079375...	1	1	1	1	1	0	0.0033792901079375...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0033792901079375...	1	1	1	1	1	0	0.0033792901079375...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0033792901079375...	1	1	1	1	1	0	0.0033792901079375...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0034527529360370...	1	1	1	1	1	0	0.0034527529360370...	(5,[1,3],[1.0,...	0.0	[1.71384159592602...	[0.84733389323826...	0.0
0	1	1	1	0.0035262157648044...	0	1	1	0	1	0	0.0035262157648044...	(5,[1,3],[1.0,...	1.0	[1.71384159592602...	[0.84733389323826...	0.0


final
Done
| 1| 1| 0| 1|0.003011975965770445| 1| 1| 1| 1| 1| 1|0.003011975965770445|[1.0,1.0,1.0,0.00...| 0.0|[1.713841595926 02...|[0.847333893238 26...| 0.0|
only showing top 1000 rows
scala> val evaluator = new BinaryClassificationEvaluator().setLabelCol("label") .setRawPredictionCol("prediction")
<console>:35: error: not found: type BinaryClassificationEvaluator
val evaluator = new BinaryClassificationEvaluator().setLabelCol("label") .setRawPredictionCol("prediction")
^
scala> val evaluator = new BinaryClassificationEvaluator().setLabelCol("label").setRawPredictionCol("prediction").setMetricName("areaUnderROC")
<console>:35: error: not found: type BinaryClassificationEvaluator
val evaluator = new BinaryClassificationEvaluator().setLabelCol("label").setRawPredictionCol("prediction").setMetricName("areaUnderROC")
^
scala> import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
scala> val evaluator = new BinaryClassificationEvaluator().setLabelCol("label").setRawPredictionCol("prediction").setMetricName("areaUnderROC")
evaluator: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_0c158bca1388
scala> val accuracy = evaluator.evaluate(predictionDf)
accuracy: Double = 0.5
scala> println(accuracy)
0.5
scala> val randomForestClassifier = new RandomForestClassifier()
<console>:36: error: not found: type RandomForestClassifier
val randomForestClassifier = new RandomForestClassifier()
^
scala> import org.apache.spark.mllib.tree.RandomForest
import org.apache.spark.mllib.tree.RandomForest
scala> val randomForestClassifier = new RandomForestClassifier()
<console>:37: error: not found: type RandomForestClassifier
val randomForestClassifier = new RandomForestClassifier()
^
scala> import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
scala> val randomForestClassifier = new RandomForestClassifier()
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_1627125507a1
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10.setNumTrees(20))
<console>:38: error: value setNumTrees is not a member of Int
val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10.setNumTrees(20))


zeruiji — zj637@hlog-2:~ — ssh zj637@peel.hpc.nyu.edu — 236×62
final
Done
scala> val randomForestClassifier = new RandomForestClassifier()
<console>:37: error: not found: type RandomForestClassifier
val randomForestClassifier = new RandomForestClassifier()
^
scala> import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
scala> val randomForestClassifier = new RandomForestClassifier()
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_1627125507a1
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10.setNumTrees(20))
<console>:38: error: value setNumTrees is not a member of Int
val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10.setNumTrees(20))
^
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10).setNumTrees(20))..setFeatureSubsetStrategy("auto")
<console>:1: error: ';' expected but ')' found.
val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10).setNumTrees(20))..setFeatureSubsetStrategy("auto")
^
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10).setNumTrees(20)).setFeatureSubsetStrategy("auto").setSeed(seed)
<console>:1: error: ';' expected but ')' found.
val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10).setNumTrees(20)).setFeatureSubsetStrategy("auto").setSeed(seed)
^
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10).setNumTrees(20).setFeatureSubsetStrategy("auto").setSeed(seed)
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_53970db6fe7b
scala> val randomForestModel = randomForestClassifier.fit(trainingData)
randomForestModel: org.apache.spark.ml.classification.RandomForestClassificationModel = RandomForestClassificationModel (uid=rfc_53970db6fe7b) with 20 trees
scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]
scala> predictionDf.show(10)
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3| _c4|_c5|sex|age|credit|Employment|Housing| income| features|label| rawPrediction| probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
0	1	0	0	0.0	0	0	1	0	0	0	0.0	(5,[0],[1.0])	1.0	[10.4161521922103...	[0.52080760961051...	0.0
0	1	0	0	0.0	1	0	1	1	0	0	0.0	(5,[0],[1.0])	0.0	[10.4161521922103...	[0.52080760961051...	0.0
0	1	0	1	0.001175405254934...	0	1	1	0	1	0	0.001175405254934...	[1.0,0,0.1,0,0.00...	1.0	[4.52905955171071...	[0.22645297758553...	1.0
0	1	0	1	0.001322330911801...	0	1	1	0	1	0	0.001322330911801...	[1.0,0,0.1,0,0.00...	1.0	[4.52905955171071...	[0.22645297758553...	1.0
0	1	0	1	0.001689645053968786	0	1	1	0	1	0	0.001689645053968786	[1.0,0,0.1,0,0.00...	1.0	[4.52905955171071...	[0.22645297758553...	1.0
0	1	0	1	0.001836570710835...	1	1	1	1	1	0	0.001836570710835...	[1.0,0,0.1,0,0.00...	0.0	[4.52905955171071...	[0.22645297758553...	1.0
0	1	0	1	0.002203884853002...	1	1	1	1	1	0	0.002203884853002...	[1.0,0,0.1,0,0.00...	0.0	[4.32311684336508...	[0.21615584216825...	1.0
0	1	0	1	0.00227734768143619	0	1	1	0	1	0	0.00227734768143619	[1.0,0,0.1,0,0.00...	1.0	[4.32311684336508...	[0.21615584216825...	1.0
0	1	0	1	0.00227734768143619	1	1	1	1	1	0	0.00227734768143619	[1.0,0,0.1,0,0.00...	0.0	[4.32311684336508...	[0.21615584216825...	1.0
0	1	0	1	0.00227734768143619	1	1	1	1	1	0	0.00227734768143619	[1.0,0,0.1,0,0.00...	0.0	[4.32311684336508...	[0.21615584216825...	1.0
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
only showing top 10 rows
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(3).setNumTrees(20).setFeatureSubsetStrategy("auto").setSeed(seed)
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_72431f480e45
scala> val randomForestModel = randomForestClassifier.fit(trainingData)
randomForestModel: org.apache.spark.ml.classification.RandomForestClassificationModel = RandomForestClassificationModel (uid=rfc_72431f480e45) with 20 trees
scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]


final
Done
0	1	0	1	0.0011754052549 34...	0	1	1	0	1	0	0.0011754052549 34...	[1.0,0.0,1.0,0.00...	1.0	[4.529059551710 71...	[0.2264529775855 3...	1.0
0	1	0	1	0.0013223309118 01...	0	1	1	0	1	0	0.0013223309118 01...	[1.0,0.0,1.0,0.00...	1.0	[4.529059551710 71...	[0.2264529775855 3...	1.0
0	1	0	1	0.0016896450539 68786	0	1	1	0	1	0	0.0016896450539 68786	[1.0,0.0,1.0,0.00...	1.0	[4.529059551710 71...	[0.2264529775855 3...	1.0
0	1	0	1	0.0018365707108 35...	1	1	1	1	1	0	0.0018365707108 35...	[1.0,0.0,1.0,0.00...	0.0	[4.529059551710 71...	[0.2264529775855 3...	1.0
0	1	0	1	0.0022038848530 02...	1	1	1	1	1	0	0.0022038848530 02...	[1.0,0.0,1.0,0.00...	0.0	[4.323116843365 08...	[0.2161558421682 5...	1.0
0	1	0	1	0.002277347681436 19	0	1	1	0	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	1.0	[4.323116843365 08...	[0.2161558421682 5...	1.0
0	1	0	1	0.002277347681436 19	1	1	1	1	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	0.0	[4.323116843365 08...	[0.2161558421682 5...	1.0
0	1	0	1	0.002277347681436 19	1	1	1	1	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	0.0	[4.323116843365 08...	[0.2161558421682 5...	1.0
only showing top 10 rows
[scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(3).setNumTrees(20).setFeatureSubsetStrategy("auto").setSeed(seed)
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_72431f480e45
[scala> val randomForestModel = randomForestClassifier.fit(trainingData)
randomForestModel: org.apache.spark.ml.classification.RandomForestClassificationModel = RandomForestClassificationModel (uid=rfc_72431f480e45) with 20 trees
[scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]
[scala> predictionDf.show(10)
_c0	_c1	_c2	_c3	_c4	_c5	sex	age	credit	Employment	Housing	income	features	label	rawPrediction	probability	prediction
0	1	0	0	0.0	0	0	1	0	0	0	0.0	(5,[0],[1.0])	1.0	[10.227853219718 0...	[0.5113926609859 0...	0.0
0	1	0	0	0.0	1	0	1	1	0	0	0.0	(5,[0],[1.0])	0.0	[10.227853219718 0...	[0.5113926609859 0...	0.0
0	1	0	1	0.0011754052549 34...	0	1	1	0	1	0	0.0011754052549 34...	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0013223309118 01...	0	1	1	0	1	0	0.0013223309118 01...	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0016896450539 68786	0	1	1	0	1	0	0.0016896450539 68786	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0018365707108 35...	1	1	1	1	1	0	0.0018365707108 35...	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0022038848530 02...	1	1	1	1	1	0	0.0022038848530 02...	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.002277347681436 19	0	1	1	0	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.002277347681436 19	1	1	1	1	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
only showing top 10 rows
[scala> predictionDf.show(30)
_c0	_c1	_c2	_c3	_c4	_c5	sex	age	credit	Employment	Housing	income	features	label	rawPrediction	probability	prediction
0	1	0	0	0.0	0	0	1	0	0	0	0.0	(5,[0],[1.0])	1.0	[10.227853219718 0...	[0.5113926609859 0...	0.0
0	1	0	0	0.0	1	0	1	1	0	0	0.0	(5,[0],[1.0])	0.0	[10.227853219718 0...	[0.5113926609859 0...	0.0
0	1	0	1	0.0011754052549 34...	0	1	1	0	1	0	0.0011754052549 34...	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0013223309118 01...	0	1	1	0	1	0	0.0013223309118 01...	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0016896450539 68786	0	1	1	0	1	0	0.0016896450539 68786	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0018365707108 35...	1	1	1	1	1	0	0.0018365707108 35...	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0022038848530 02...	1	1	1	1	1	0	0.0022038848530 02...	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.002277347681436 19	0	1	1	0	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.002277347681436 19	1	1	1	1	1	0	0.002277347681436 19	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0023508105096 89...	1	1	1	1	1	0	0.0023508105096 89...	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0023508105096 89...	1	1	1	1	1	0	0.0023508105096 89...	[1.0,0.0,1.0,0.00...	0.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0024242733383 03...	0	1	1	0	1	0	0.0024242733383 03...	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0024242733383 03...	0	1	1	0	1	0	0.0024242733383 03...	[1.0,0.0,1.0,0.00...	1.0	[7.601441048826 42...	[0.3800720524413 2...	1.0
0	1	0	1	0.0027181246523 06743	0	1	1	0	1	0	0.0027181246523 06743	[1.0,0.0,1.0,0.00...	1.0	[8.231667234196 01...	[0.4115833617098 0...	1.0
0	1	0	1	0.0027915874804 70...	0	1	1	0	1	0	0.0027915874804 70...	[1.0,0.0,1.0,0.00...	1.0	[8.231667234196 01...	[0.4115833617098 0...	1.0
0	1	0	1	0.0027915874804 70...	0	1	1	0	1	0	0.0027915874804 70...	[1.0,0.0,1.0,0.00...	1.0	[8.231667234196 01...	[0.4115833617098 0...	1.0
0	1	0	1	0.0028650503089 03...	0	1	1	0	1	0	0.0028650503089 03...	[1.0,0.0,1.0,0.00...	1.0	[8.231667234196 01...	[0.4115833617098 0...	1.0
0	1	0	1	0.0031589016226 37296	0	1	1	0	1	0	0.0031589016226 37296	[1.0,0.0,1.0,0.00...	1.0	[9.529491605550 67...	[0.4764745802775 3...	1.0

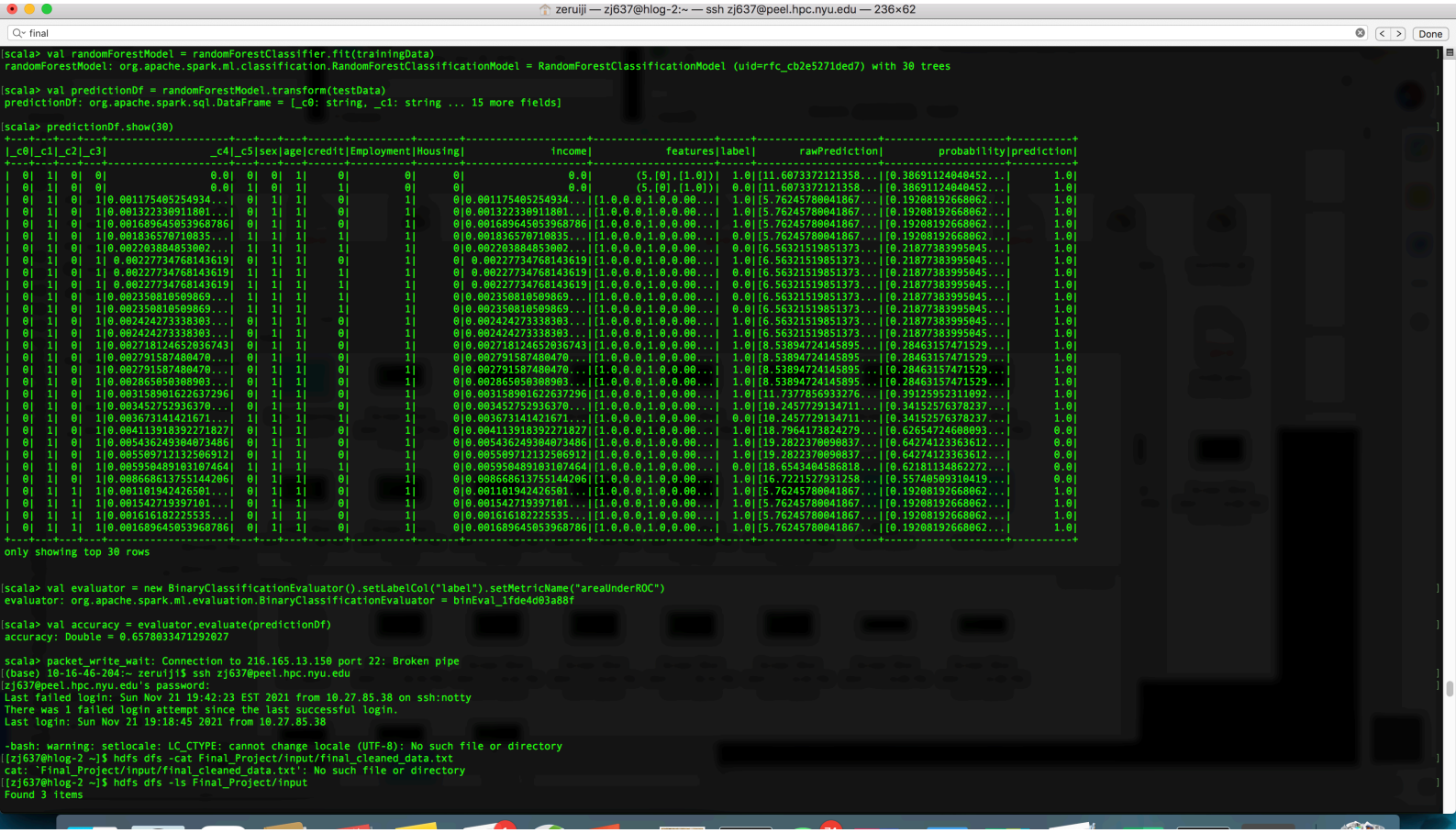

zeruiji — zj637@hlog-2:~ — ssh zj637@peel.hpc.nyu.edu — 236×62
final
Done
only showing top 30 rows
scala> val evaluator = new BinaryClassificationEvaluator()
evaluator: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_b5564a012c86
scala> val evaluator = new BinaryClassificationEvaluator().setLabelCol("label").setMetricName("areaUnderROC")
evaluator: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_90f9e6799401
scala> val accuracy = evaluator.evaluate(predictionDf)
accuracy: Double = 0.6485763787777834
scala> println(accuracy)
0.6485763787777834
scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(10).setNumTrees(30).setFeatureSubsetStrategy("auto").setSeed(seed)
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_f25f8610d04b
scala> val randomForestModel = randomForestClassifier.fit(trainingData)
randomForestModel: org.apache.spark.ml.classification.RandomForestClassificationModel = RandomForestClassificationModel (uid=rfc_f25f8610d04b) with 30 trees
scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]
scala> predictionDf.show(30)

zeruiji — zj637@hlog-2:~ — ssh zj637@peel.hpc.nyu.edu — 236×62

final

Done

```
scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]

scala> predictionDf.show(30)
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|label|       rawPrediction|         probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|  0|  1|  0|  0|                 0.0|  0|  0|  1|     0|         0|      0|                 0.0|       (5,[0],[1.0])|  1.0|[11.6073372121358...|[0.38691124040452...|       1.0|
|  0|  1|  0|  0|                 0.0|  1|  0|  1|     1|         0|      0|                 0.0|       (5,[0],[1.0])|  0.0|[11.6073372121358...|[0.38691124040452...|       1.0|
|  0|  1|  0|  1|0.001175405254934...|  0|  1|  1|     0|         1|      0|0.001175405254934...|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  0|  1|0.001322330911801...|  0|  1|  1|     0|         1|      0|0.001322330911801...|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  0|  1|0.001689645053968786|  0|  1|  1|     0|         1|      0|0.001689645053968786|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  0|  1|0.001836570710835...|  1|  1|  1|     1|         1|      0|0.001836570710835...|[1.0,0.0,1.0,0.00...|  0.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  0|  1|0.002203884853002...|  1|  1|  1|     1|         1|      0|0.002203884853002...|[1.0,0.0,1.0,0.00...|  0.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1| 0.002277347681436...|  0|  1|  1|     0|         1|      0| 0.002277347681436...|[1.0,0.0,1.0,0.00...|  1.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1| 0.002277347681436...|  1|  1|  1|     1|         1|      0| 0.002277347681436...|[1.0,0.0,1.0,0.00...|  0.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1| 0.002277347681436...|  1|  1|  1|     1|         1|      0| 0.002277347681436...|[1.0,0.0,1.0,0.00...|  0.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1|0.002350810509869...|  1|  1|  1|     1|         1|      0|0.002350810509869...|[1.0,0.0,1.0,0.00...|  0.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1|0.002350810509869...|  1|  1|  1|     1|         1|      0|0.002350810509869...|[1.0,0.0,1.0,0.00...|  0.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1|0.002424273338303...|  0|  1|  1|     0|         1|      0|0.002424273338303...|[1.0,0.0,1.0,0.00...|  1.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1|0.002424273338303...|  0|  1|  1|     0|         1|      0|0.002424273338303...|[1.0,0.0,1.0,0.00...|  1.0|[6.56321519851373...|[0.21877383995045...|       1.0|
|  0|  1|  0|  1|0.002718124652036743|  0|  1|  1|     0|         1|      0|0.002718124652036743|[1.0,0.0,1.0,0.00...|  1.0|[8.53894724145895...|[0.28463157471529...|       1.0|
|  0|  1|  0|  1|0.002791587480470...|  0|  1|  1|     0|         1|      0|0.002791587480470...|[1.0,0.0,1.0,0.00...|  1.0|[8.53894724145895...|[0.28463157471529...|       1.0|
|  0|  1|  0|  1|0.002791587480470...|  0|  1|  1|     0|         1|      0|0.002791587480470...|[1.0,0.0,1.0,0.00...|  1.0|[8.53894724145895...|[0.28463157471529...|       1.0|
|  0|  1|  0|  1|0.002865050308903...|  0|  1|  1|     0|         1|      0|0.002865050308903...|[1.0,0.0,1.0,0.00...|  1.0|[8.53894724145895...|[0.28463157471529...|       1.0|
|  0|  1|  0|  1|0.003158901622637296|  0|  1|  1|     0|         1|      0|0.003158901622637296|[1.0,0.0,1.0,0.00...|  1.0|[11.7280998294532...|[0.39093666098177...|       1.0|
|  0|  1|  0|  1|0.003452752936370...|  0|  1|  1|     0|         1|      0|0.003452752936370...|[1.0,0.0,1.0,0.00...|  1.0|[10.3280225334677...|[0.34426741778225...|       1.0|
|  0|  1|  0|  1|0.003673141421671...|  1|  1|  1|     1|         1|      0|0.003673141421671...|[1.0,0.0,1.0,0.00...|  0.0|[10.3280225334677...|[0.34426741778225...|       1.0|
|  0|  1|  0|  1|0.004113918392271827|  0|  1|  1|     0|         1|      0|0.004113918392271827|[1.0,0.0,1.0,0.00...|  1.0|[18.7964173824279...|[0.62654724608093...|       0.0|
|  0|  1|  0|  1|0.005436249304073486|  0|  1|  1|     0|         1|      0|0.005436249304073486|[1.0,0.0,1.0,0.00...|  1.0|[19.2460555554064...|[0.64153518518021...|       0.0|
|  0|  1|  0|  1|0.005509712132506912|  0|  1|  1|     0|         1|      0|0.005509712132506912|[1.0,0.0,1.0,0.00...|  1.0|[19.2460555554064...|[0.64153518518021...|       0.0|
|  0|  1|  0|  1|0.005950489103107464|  1|  1|  1|     1|         1|      0|0.005950489103107464|[1.0,0.0,1.0,0.00...|  0.0|[18.6181590050046...|[0.62060530016682...|       0.0|
|  0|  1|  0|  1|0.008668613755144206|  0|  1|  1|     0|         1|      0|0.008668613755144206|[1.0,0.0,1.0,0.00...|  1.0|[16.6345742577776...|[0.55448580859258...|       0.0|
|  0|  1|  1|  1|0.001101942426501...|  0|  1|  1|     0|         1|      0|0.001101942426501...|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  1|  1|0.001542719397101...|  0|  1|  1|     0|         1|      0|0.001542719397101...|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  1|  1|0.001616182225535...|  0|  1|  1|     0|         1|      0|0.001616182225535...|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
|  0|  1|  1|  1|0.001689645053968786|  0|  1|  1|     0|         1|      0|0.001689645053968786|[1.0,0.0,1.0,0.00...|  1.0|[5.76245780041867...|[0.19208192668062...|       1.0|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
only showing top 30 rows


scala> val evaluator = new BinaryClassificationEvaluator().setLabelCol("label").setMetricName("areaUnderROC")
evaluator: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_3caf0c7e684a

scala> val accuracy = evaluator.evaluate(predictionDf)
accuracy: Double = 0.6577528877401801

scala> val randomForestClassifier = new RandomForestClassifier().setImpurity("gini").setMaxDepth(20).setNumTrees(30).setFeatureSubsetStrategy("auto").setSeed(seed)
randomForestClassifier: org.apache.spark.ml.classification.RandomForestClassifier = rfc_cb2e5271ded7

scala> val randomForestModel = randomForestClassifier.fit(trainingData)
randomForestModel: org.apache.spark.ml.classification.RandomForestClassificationModel = RandomForestClassificationModel (uid=rfc_cb2e5271ded7) with 30 trees

scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]

scala> predictionDf.show(30)
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3|                 _c4|_c5|sex|age|credit|Employment|Housing|              income|            features|label|       rawPrediction|         probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|  0|  1|  0|  0|                 0.0|  0|  0|  1|     0|         0|      0|                 0.0|       (5,[0],[1.0])|  1.0|[11.6073372121358...|[0.38691124040452...|       1.0|
|  0|  1|  0|  0|                 0.0|  1|  0|  1|     1|         0|      0|                 0.0|       (5,[0],[1.0])|  0.0|[11.6073372121358...|[0.38691124040452...|       1.0|
```


zeruiji — zj637@hlog-2:~ — ssh zj637@peel.hpc.nyu.edu — 236×62
final
Done
[scala> val randomForestModel = randomForestClassifier.fit(trainingData)
randomForestModel: org.apache.spark.ml.classification.RandomForestClassificationModel = RandomForestClassificationModel (uid=rfc_cb2e5271ded7) with 30 trees
[scala> val predictionDf = randomForestModel.transform(testData)
predictionDf: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 15 more fields]
[scala> predictionDf.show(30)
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
|_c0|_c1|_c2|_c3| _c4|_c5|sex|age|credit|Employment|Housing| income| features|label| rawPrediction| probability|prediction|
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
0	1	0	0	0.0	0	0	1	0	0	0	0.0	(5,[0],[1.0])	1.0	[11.6073372121358...	[0.38691124040452...	1.0
0	1	0	0	0.0	1	0	1	1	0	0	0.0	(5,[0],[1.0])	0.0	[11.6073372121358...	[0.38691124040452...	1.0
0	1	0	1	0.001175405254934...	0	1	1	0	1	0	0.001175405254934...	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	0	1	0.001322330911801...	0	1	1	0	1	0	0.001322330911801...	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	0	1	0.001689645053968786	0	1	1	0	1	0	0.001689645053968786	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	0	1	0.001836570710835...	1	1	1	1	1	0	0.001836570710835...	[1.0,0,0,1,0,0.00...	0.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	0	1	0.002203884853002...	1	1	1	1	1	0	0.002203884853002...	[1.0,0,0,1,0,0.00...	0.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.00227734768143619	0	1	1	0	1	0	0.00227734768143619	[1.0,0,0,1,0,0.00...	1.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.00227734768143619	1	1	1	1	1	0	0.00227734768143619	[1.0,0,0,1,0,0.00...	0.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.00227734768143619	1	1	1	1	1	0	0.00227734768143619	[1.0,0,0,1,0,0.00...	0.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.002350810509869...	1	1	1	1	1	0	0.002350810509869...	[1.0,0,0,1,0,0.00...	0.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.002350810509869...	1	1	1	1	1	0	0.002350810509869...	[1.0,0,0,1,0,0.00...	0.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.002424273338303...	0	1	1	0	1	0	0.002424273338303...	[1.0,0,0,1,0,0.00...	1.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.002424273338303...	0	1	1	0	1	0	0.002424273338303...	[1.0,0,0,1,0,0.00...	1.0	[6.56321519851373...	[0.21877383995045...	1.0
0	1	0	1	0.002718124652036743	0	1	1	0	1	0	0.002718124652036743	[1.0,0,0,1,0,0.00...	1.0	[8.53894724145895...	[0.28463157471529...	1.0
0	1	0	1	0.002791587480470...	0	1	1	0	1	0	0.002791587480470...	[1.0,0,0,1,0,0.00...	1.0	[8.53894724145895...	[0.28463157471529...	1.0
0	1	0	1	0.002791587480470...	0	1	1	0	1	0	0.002791587480470...	[1.0,0,0,1,0,0.00...	1.0	[8.53894724145895...	[0.28463157471529...	1.0
0	1	0	1	0.002865050308903...	0	1	1	0	1	0	0.002865050308903...	[1.0,0,0,1,0,0.00...	1.0	[8.53894724145895...	[0.28463157471529...	1.0
0	1	0	1	0.003158901622637296	0	1	1	0	1	0	0.003158901622637296	[1.0,0,0,1,0,0.00...	1.0	[11.7377856933276...	[0.39125952311092...	1.0
0	1	0	1	0.003452752936370...	0	1	1	0	1	0	0.003452752936370...	[1.0,0,0,1,0,0.00...	1.0	[10.2457729134711...	[0.34152576378237...	1.0
0	1	0	1	0.003673141421671...	1	1	1	1	1	0	0.003673141421671...	[1.0,0,0,1,0,0.00...	0.0	[10.2457729134711...	[0.34152576378237...	1.0
0	1	0	1	0.004113918392271827	0	1	1	0	1	0	0.004113918392271827	[1.0,0,0,1,0,0.00...	1.0	[18.7964173824279...	[0.62654724608093...	0.0
0	1	0	1	0.005436249304073486	0	1	1	0	1	0	0.005436249304073486	[1.0,0,0,1,0,0.00...	1.0	[19.2822370090837...	[0.64274123363612...	0.0
0	1	0	1	0.005509712132506912	0	1	1	0	1	0	0.005509712132506912	[1.0,0,0,1,0,0.00...	1.0	[19.2822370090837...	[0.64274123363612...	0.0
0	1	0	1	0.005950489103107464	1	1	1	1	1	0	0.005950489103107464	[1.0,0,0,1,0,0.00...	0.0	[18.6543404586818...	[0.62181134862272...	0.0
0	1	0	1	0.008668613755144206	0	1	1	0	1	0	0.008668613755144206	[1.0,0,0,1,0,0.00...	1.0	[16.7221527931258...	[0.55740509310419...	0.0
0	1	1	1	0.001101942426501...	0	1	1	0	1	0	0.001101942426501...	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	1	1	0.001542719397101...	0	1	1	0	1	0	0.001542719397101...	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	1	1	0.001616182225535...	0	1	1	0	1	0	0.001616182225535...	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
0	1	1	1	0.001689645053968786	0	1	1	0	1	0	0.001689645053968786	[1.0,0,0,1,0,0.00...	1.0	[5.76245780041867...	[0.19208192668062...	1.0
+---+---+---+---+--------------------+---+---+---+------+----------+-------+--------------------+--------------------+-----+--------------------+--------------------+----------+
only showing top 30 rows
[scala> val evaluator = new BinaryClassificationEvaluator().setLabelCol("label").setMetricName("areaUnderROC")
evaluator: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_1fde4d03a88f
[scala> val accuracy = evaluator.evaluate(predictionDf)
accuracy: Double = 0.6578033471292027
scala> packet_write_wait: Connection to 216.165.13.150 port 22: Broken pipe
[(base) 10-16-46-204:~ zeruiji$ ssh zj637@peel.hpc.nyu.edu
[zj637@peel.hpc.nyu.edu's password:
Last failed login: Sun Nov 21 19:42:23 EST 2021 from 10.27.85.38 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Sun Nov 21 19:18:45 2021 from 10.27.85.38
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[zj637@hlog-2 ~]$ hdfs dfs -cat Final_Project/input/final_cleaned_data.txt
cat: `Final_Project/input/final_cleaned_data.txt': No such file or directory
[zj637@hlog-2 ~]$ hdfs dfs -ls Final_Project/input
Found 3 items