

Class weight

```
giranabhi@bigdata-m: ~ -- Mozilla Firefox
https://ssh.cloud.google.com/projects/iron-zodiac-347119/zones/us-central1-b/instances/bigdata-m?authuser=0&hl=en_US&projectNumber=782503674315&useAdminProxy=true&trou

Connected, host fingerprint: ssh-rsa 0 77:91:Ad:1F:E9:AB:15:02:E3:1C:AA:6B:1B:25
:15:EB:A9:01:BD:F9:E8:B9:9E:AD:0D:40:E4:E8:FA:D0:5A:3D
Linux bigdata-m 5.10.0-0.bpo.12-amd64 #1 SMP Debian 5.10.103-1-bpo10+1 (2022-03-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
giranabhi@bigdata-m:~$ wget https://ckan0.cf.opendata.inter.prod-toronto.ca/dataset/64b54586-6180-4485-83eb-81e8fae3b8fe/resource/fff4ee65-3527-43be-9a8a-cb9401377dbc/download/COVID19%20cases.csv
--2022-04-13 21:26:55-- https://ckan0.cf.opendata.inter.prod-toronto.ca/dataset/64b54586-6180-4485-83eb-81e8fae3b8fe/resource/fff4ee65-3527-43be-9a8a-cb9401377dbc/download/COVID19%20cases.csv
Resolving ckan0.cf.opendata.inter.prod-toronto.ca (ckan0.cf.opendata.inter.prod-toronto.ca)... 108.156.120.94, 108.156.120.84, 108.156.120.125, ...
Connecting to ckan0.cf.opendata.inter.prod-toronto.ca (ckan0.cf.opendata.inter.prod-toronto.ca)|108.156.120.94|:443
... connected.
HTTP request sent, awaiting response... 200 OK
Length: 43646641 (42M) [application/octet-stream]
Saving to: 'COVID19 cases.csv'

COVID19 cases.csv          100%[=====] 41.62M  55.7MB/s   in 0.7s

2022-04-13 21:26:56 (55.7 MB/s) - 'COVID19 cases.csv' saved [43646641/43646641]

giranabhi@bigdata-m:~$ mv COVID19 cases.csv Covid19Cases.csv
mv: target 'Covid19Cases.csv' is not a directory
giranabhi@bigdata-m:~$ mv 'COVID19 cases.csv' Covid19Cases.csv
giranabhi@bigdata-m:~$ hadoop fs -mkdir /BigData
giranabhi@bigdata-m:~$ hadoop fs -copyFromLocal Covid19Cases.csv /BigData/.
giranabhi@bigdata-m:~$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/13 21:33:13 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
22/04/13 21:33:13 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
22/04/13 21:33:13 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
22/04/13 21:33:13 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Spark context Web UI available at http://bigdata-m.us-central1-b.c.iron-zodiac-347119.internal:46315
Spark context available as 'sc' (master = yarn, app id = application_1649885026980_0001).
Spark session available as 'spark'.
Welcome to

Type here to search
```

```
giranabhi@bigdata-m: ~ -- Mozilla Firefox
https://ssh.cloud.google.com/projects/iron-zodiac-347119/zones/us-central1-b/instances/bigdata-m?authuser=0&hl=en_US&projectNumber=782503674315&useAdminProxy=true&trou

Databricks version 3.1.2

Using Scala version 2.12.14 (OpenJDK 64-Bit Server VM, Java 1.8.0_322)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :paste
// Entering paste mode (ctrl-D to finish)

import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.ml.feature.{VectorAssembler, StringIndexer}
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.tuning.{CrossValidator, CrossValidatorModel, ParamGridBuilder}
import org.apache.spark.ml.evaluation.{MulticlassClassificationEvaluator}
import org.apache.spark.ml.param.ParamMap
import org.apache.spark.sql.types.{IntegerType, DoubleType}
import org.apache.spark.sql.DataFrame

val schema_covid19 = StructType( StructField("id", IntegerType, nullable = true) ::
  StructField("Assigned ID", IntegerType, nullable = true) ::
  StructField("Outbreak Associated", StringType, nullable = true) ::
  StructField("Age Group", StringType, nullable = true) ::
  StructField("Neighbourhood Name", StringType, nullable = true) ::
  StructField("FSA", StringType, nullable = true) ::
  StructField("Source of Infection", StringType, nullable = true) ::
  StructField("Classification", StringType, nullable = true) ::
  StructField("Episode Date", DateType, nullable = true) ::
  StructField("Reported Date", DateType, nullable = true) ::
  StructField("Client Gender", StringType, nullable = true) ::
  StructField("Outcome", StringType, nullable = true) ::
  StructField("Currently Hospitalized", StringType, nullable = true) ::
  StructField("Currently in ICU", StringType, nullable = true) ::
  StructField("Currently Intubated", StringType, nullable = true) ::
  StructField("Ever Hospitalized", StringType, nullable = true) ::
  StructField("Ever in ICU", StringType, nullable = true) ::
)

Type here to search
```

```
girana@bigdata-m: ~ -- Mozilla Firefox
https://ssh.cloud.google.com/projects/iron-zodiac-347119/zones/us-central1-b/instances/bigdata-m?authuser=0&hl=en_US&projectNumber=782503674315&useAdminProxy=true&trou

// Entering paste mode (ctrl-D to finish)
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.ml.feature.{VectorAssembler, StringIndexer}
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.tuning.{CrossValidator, CrossValidatorModel, ParamGridBuilder}
import org.apache.spark.ml.evaluation.{MulticlassClassificationEvaluator}
import org.apache.spark.ml.param.ParamMap
import org.apache.spark.sql.types.{IntegerType, DoubleType}
import org.apache.spark.sql.DataFrame

val schema_covid19 = StructType( StructField("id", IntegerType, nullable = true) ::
    StructField("Assigned ID", IntegerType, nullable = true) ::
    StructField("Outbreak Associated", StringType, nullable = true) ::
    StructField("Age Group", StringType, nullable = true) ::
    StructField("Neighbourhood Name", StringType, nullable = true) ::
    StructField("FSA", StringType, nullable = true) ::
    StructField("Source of Infection", StringType, nullable = true) ::
    StructField("Classification", StringType, nullable = true) ::
    StructField("Episode Date", DateType, nullable = true) ::
    StructField("Reported Date", DateType, nullable = true) ::
    StructField("Client Gender", StringType, nullable = true) ::
    StructField("Outcome", StringType, nullable = true) ::
    StructField("Currently Hospitalized", StringType, nullable = true) ::
    StructField("Currently in ICU", StringType, nullable = true) ::
    StructField("Currently Intubated", StringType, nullable = true) ::
    StructField("Ever Hospitalized", StringType, nullable = true) ::
    StructField("Ever in ICU", StringType, nullable = true) ::
    StructField("Ever Intubated", StringType, nullable = true) :: Nil )

val raw_covid19_df = spark.read.format("csv").
    option("header", value = true).option("delimiter", ",").option("mode", "DROPMALFORMED").
    schema(schema_covid19).load("hdfs://BigData/Covid19Cases.csv").cache()

raw_covid19_df.printSchema()
raw_covid19_df.show(10)

val covid19_df = raw_covid19_df.filter(col("Outcome").isin("RESOLVED", "FATAL")).filter(col("Age Group").isNotNull)
```

```
girana@bigdata-m: ~ -- Mozilla Firefox
https://ssh.cloud.google.com/projects/iron-zodiac-347119/zones/us-central1-b/instances/bigdata-m?authuser=0&hl=en_US&projectNumber=782503674315&useAdminProxy=true&trou

    StructField("Ever Intubated", StringType, nullable = true) :: Nil )

val raw_covid19_df = spark.read.format("csv").
    option("header", value = true).option("delimiter", ",").option("mode", "DROPMALFORMED").
    schema(schema_covid19).load("hdfs://BigData/Covid19Cases.csv").cache()

raw_covid19_df.printSchema()
raw_covid19_df.show(10)

val covid19_df = raw_covid19_df.filter(col("Outcome").isin("RESOLVED", "FATAL")).filter(col("Age Group").isNotNull)
covid19_df.count()

val indexer = new StringIndexer()
    .setInputCol("Outcome")
    .setOutputCol("OutcomeIDX")

print(indexer)

val covid19_df1 = indexer.fit(covid19_df).transform(covid19_df)

/*import scala.collection.mutable.ListBuffer*/
var f_indexers = new Array[org.apache.spark.ml.PipelineStage](0)
val featuresList = List("Outbreak Associated", "Age Group", "Source of Infection", "Client Gender", "Ever Hospitalized",
    "Ever in ICU", "Ever Intubated")

for (feature <- featuresList) {
    print(feature)
    val f_indexer = new StringIndexer().setInputCol(feature).setOutputCol(feature+ " IDX")
    print(f_indexer)
    f_indexers = f_indexers ++ f_indexer
}
f_indexers

val fpipeline = new Pipeline()
    .setStages(f_indexers)

val covid19_df2= fpipeline.fit(covid19_df1).transform(covid19_df1)

covid19_df2.filter(col("Outcome")==="FATAL").show(2)
```

```
gir nabhi@bigdata-m: ~ -- Mozilla Firefox
https://ssh.cloud.google.com/projects/iron-zodiac-347119/zones/us-central1-b/instances/bigdata-m?authuser=0&hl=en_US&projectNumber=782503674315&useAdminProxy=true&trou

val covid19_df2= fpipeline.fit(covid19_df1).transform(covid19_df1)
covid19_df2.filter(col("Outcome")==="FATAL").show(2)

// Adding Weights Column to balance the 'Outcome' Weightage equally for both the values
def balance(data: DataFrame): DataFrame = {
  val resolved = data.filter(data("OutcomeIDX") === 0).count
  val dataSize = data.count
  val balancingFactor = (dataSize - resolved).toDouble / dataSize

  val weights = udf { d: Double =>
    if (d == 0.0) {
      1 * balancingFactor
    }
    else {
      (1 * (1.0 - balancingFactor))
    }
  }

  val covid19_weighted = covid19_df2.withColumn("OutcomeWeights", weights(covid19_df1("OutcomeIDX")))
  covid19_weighted
}

val covid19_weighted = balance(covid19_df2)
covid19_weighted.filter(col("Outcome")==="FATAL").show(2)

val assembler = new VectorAssembler()
.setInputCols(Array("Outbreak Associated IDX","Age Group IDX","Source of Infection IDX","Client Gender IDX",
                    "Ever Hospitalized IDX",
                    "Ever in ICU IDX","Ever Intubated IDX"))
.setOutputCol("assembled-features")

val rf = new RandomForestClassifier()
.setFeaturesCol("assembled-features")
.setLabelCol("OutcomeIDX")
.setWeightCol("OutcomeWeights")
.setSeed(42)

Type here to search 9°C 17:36 13-04-2022
```

```
gir nabhi@bigdata-m: ~ -- Mozilla Firefox
https://ssh.cloud.google.com/projects/iron-zodiac-347119/zones/us-central1-b/instances/bigdata-m?authuser=0&hl=en_US&projectNumber=782503674315&useAdminProxy=true&trou

.setWeightCol("OutcomeWeights")
.setSeed(42)

val pipeline = new Pipeline()
.setStages(Array(assembler, rf))

val evaluator = new MulticlassClassificationEvaluator()
.setLabelCol("OutcomeIDX")
.setPredictionCol("prediction")
.setMetricName("accuracy")

val paramGrid = new ParamGridBuilder()
.addGrid(rf.maxDepth, Array(3, 5))
.addGrid(rf.impurity, Array("entropy", "gini")).build()

val cross_validator = new CrossValidator()
.setEstimator(pipeline)
.setEvaluator(evaluator)
.setEstimatorParamMaps(paramGrid)
.setNumFolds(3)

/* val trainData = covid19_weighted.sample("OutcomeIDX", fractions=(0.0: 0.09, 1.0: 0.7), seed=42)*/
val Array(trainingData, testData) = covid19_weighted.randomSplit(Array(0.8, 0.2), 42)

val cvModel = cross_validator.fit(trainingData)

val predictions = cvModel.transform(testData)

val accuracy = evaluator.evaluate(predictions)

// Exiting paste mode, now interpreting.

<console>:84: warning: a pure expression does nothing in statement position; multiline expressions may require enclosing parentheses
    f_indexers
    ^
root
|-- _id: integer (nullable = true)
|-- Assigned_ID: integer (nullable = true)
|-- Outbreak Associated: string (nullable = true)
|-- Age Group: string (nullable = true)

Type here to search 9°C 17:36 13-04-2022
```

f_indexers
^

Windows taskbar showing search bar, taskbar icons (including Edge, Mail, File Explorer, Chrome, OneDrive, Word, and a folder), system tray icons (including network, volume, and battery), and the time 17:36.

No | No | No | No | No | No |

Outbreak AssociatedstrIdx ad56e191639dAge GroupstrIdx 95b64660e88fSource of InfectionstrIdx 4ba2576cce0fClient GenderstrIdx 5902bcd94856Ever HospitalizedstrIdx 3f4c

```
action IDX|Client Gender IDX|Ever Hospitalized IDX|Ever in ICU IDX|Ever Intubated IDX|
```

ClientID	ClientName	Gender	Age	Source of Infection	Outbreak Associated	Outcome
1	John Doe	Male	45	Healthcare Facility	Yes	Recovered
2	Jane Smith	Female	32	Community	No	Recovered
3	Michael Johnson	Male	58	Healthcare Facility	Yes	Deceased
4	Sarah Williams	Female	28	Community	No	Recovered
5	David Brown	Male	65	Healthcare Facility	Yes	Recovered
6	Emily Davis	Female	41	Community	No	Recovered
7	Robert Miller	Male	72	Healthcare Facility	Yes	Deceased
8	Lisa Anderson	Female	35	Community	No	Recovered
9	Christopher Lee	Male	50	Healthcare Facility	Yes	Recovered
10	Amanda White	Female	25	Community	No	Recovered

77	80	Sporadic	70 to 79 Years	Victoria Village	M4A	Travel	CONFIRMED	2020-03-11	2020-03-13	MALE	FATAL
----	----	----------	----------------	------------------	-----	--------	-----------	------------	------------	------	-------

Windows taskbar showing search bar, taskbar icons (Edge, Mail, File Explorer, Chrome, VS Code, WhatsApp, Word, Firefox), system tray (9°C, 17:36, 13-04-2022, ENG, 122).

263	278	No	Sporadic	60 to 69 Years	No	Niagara	M5V	Yes	Yes	Community	Yes	CONFIRMED	2020-03-16	2020-03-22	0.0	5.0	MALE	FATAL
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
only showing top 2 rows																		
_id	Assigned_ID	Outbreak Associated	Age Group	Neighbourhood Name	FSA	Source of Infection	Classification	Episode Date	Reported Date	Client Gender	Outcome	Currently Hospitalized	Currently in ICU	Currently Intubated	Ever Hospitalized	Ever in ICU	Ever Intubated	OutcomeWeights
77	80	No	Sporadic	70 to 79 Years	Victoria Village	M4A	Travel	CONFIRMED	2020-03-11	2020-03-13	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
263	278	No	Sporadic	60 to 69 Years	Niagara	M5V	Community	CONFIRMED	2020-03-16	2020-03-22	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
only showing top 2 rows																		

```

import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import org.apache.spark.sql.expressions.Window
import org.apache.spark.ml.feature.{VectorAssembler, StringIndexer}
import org.apache.spark.ml.Pipeline
import org.apache.spark.ml.classification.{RandomForestClassificationModel, RandomForestClassifier}
import org.apache.spark.ml.tuning.{CrossValidator, CrossValidatorModel, ParamGridBuilder}
import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
import org.apache.spark.ml.param.ParamMap
import org.apache.spark.sql.types.{IntegerType, DoubleType}
import org.apache.spark.sql.DataFrame
schema_covid19: org.apache.spark.sql.types.StructType = StructType(StructField(_id,IntegerType,true), StructField(Assigned_ID,IntegerType,true...

```