

4/23/2022

**Submitted to : Dr. Saber Amini**



**Assignment 2 – SparkML Machine Learning Algorithm**

BDAT1008- Data Collection and Curation

**Table of Content**

Introduction

* SparkML…………………………………………………………………………………………..2
* Decision Trees and Random Forest Classifier model….………....................................2
* Dataset Background……………………………………………………..................................2
* Assignment Objective……………………………………………………………………………3
* Limitations………………………………………………………………………………………...3

Assignment Approach and Methodology

* Business Questions……………………………………………………………………………...3
* Answers to Business Questions………………………………………………………………..3
* Solution Walkthrough.……………………………………………………………….................5
* Conclusion………………………………………………………………………………………10
* Recommendation……………………………………………………………………………….10

Appendix…………………………………………………………………………………………………10

References……………………………………………………………………………………………….18

**Introduction – SparkML**

*spark.ml* is a new package launched in Spark 1.2. Its goal is to provide a uniform set of high-level APIs that will help users to create and tune practical machine learning pipelines. Spark ML standardizes APIs for machine learning algorithms to make it more easier to merge multiple algorithms into a single workflow or pipeline.

**Decision Trees Vs Random Forest Classifier model**

Decision Tree is a popular method for the machine learning classification and regression. Decision trees are commonly used since they are simple, fast and performs well on large data and it is also capable of handling categorical features. The main drawback of decision trees are the risk of overfitting. Random Forest can be used to resolve this problem, as it combines many decision trees to reduce the risk of overfitting. Random forest is nothing but an ensemble learning algorithm of decision tree.

**Dataset Background**

The data set contains severity, demographic and geographic information about all probable and confirmed cases reported to and managed by Toronto Public Health ever since the first case of Covid 19 was reported in January 2020. The data recorded are extracted from the provincial Case & Contact Management System (CCM). The fields listed in the dataset are,



**Objective**

The Objective of the dataset is to develop and validate a model using Apache SparkML algorithms to predict the Fatal cases rates in Toronto city with a higher accuracy.

**Limitations**

The major concern is the dataset is dynamic (i.e.,) it is not constant as the updation to the dataset is happening every week.

**Assignment Approach and Methodology**

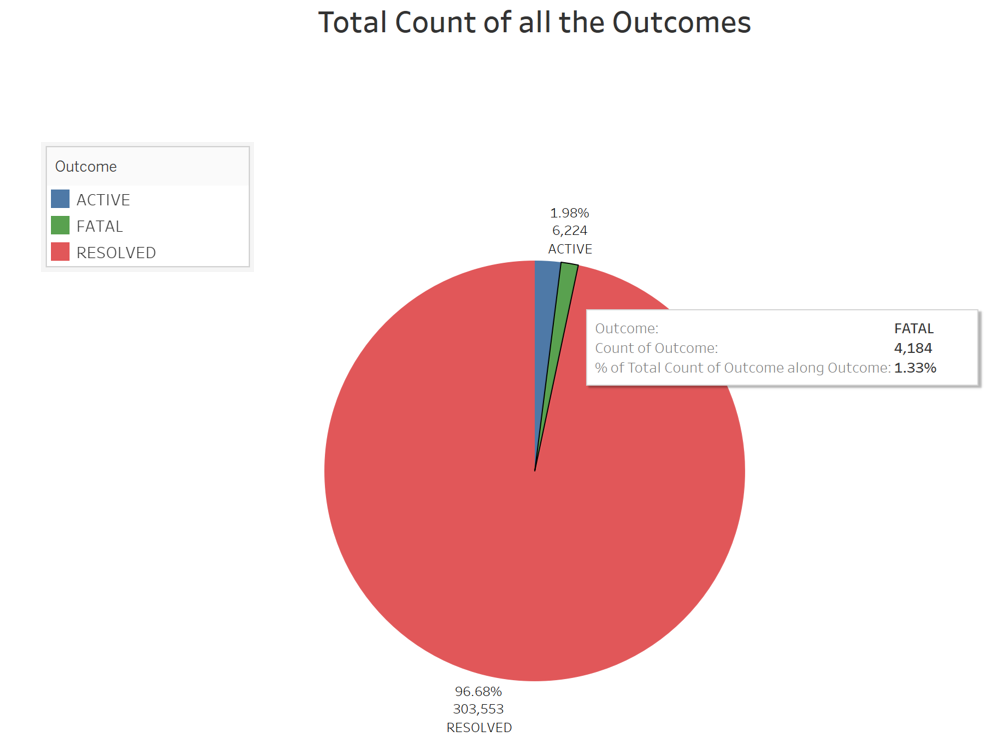
**Business Questions**

1. How much is the Fatal ratio in the dataset?
2. What is the relationship between fatal cases and source of Infection?
3. In which months of the years 2021 and 2022 the fatality rate is higher according to gender?
4. In which Age group the fatality ratio is higher?

**Answers to the Business questions**

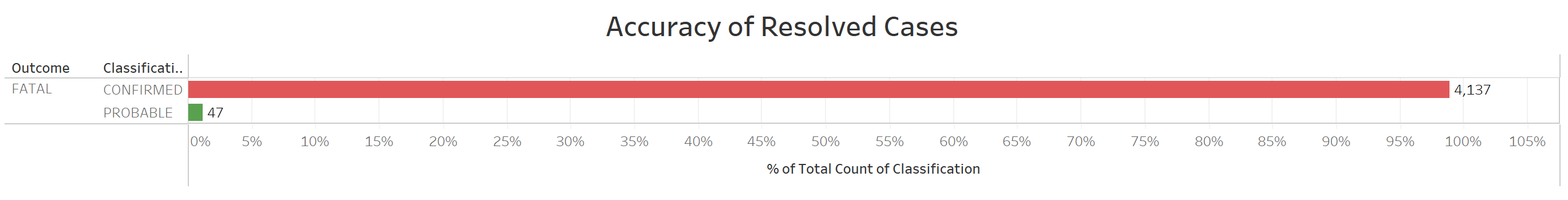
The following visualizations are created using Tableau for better understanding of the dataset.

1. **How much is the Fatal ratio in the dataset?**

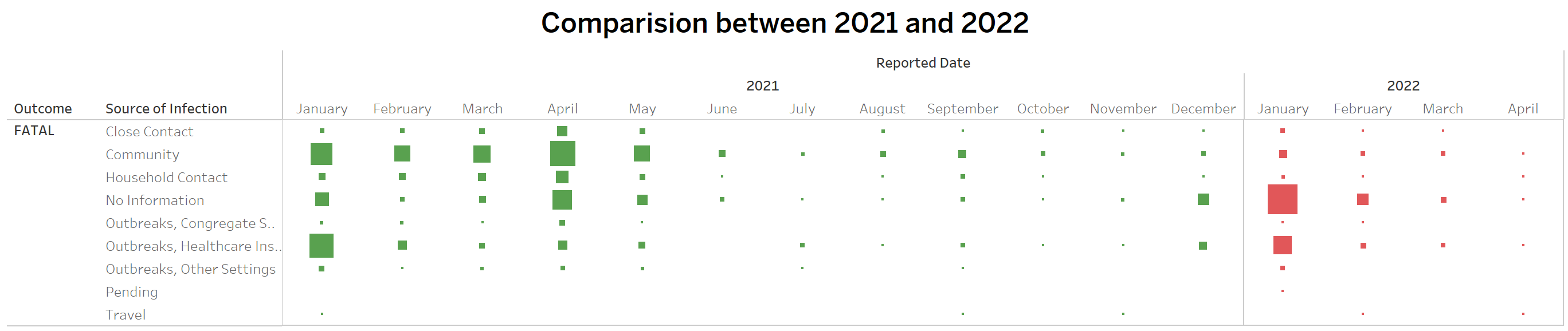


There are totally 6,224 (2%) Active cases, **4,184 (1.3%) Fatal cases** and 303,553 (96.6%) Resolved cases in the dataset. The following comparisons are made to improve the accuracy of the prediction model.

According to the dataset, 99% of the Fatal cases are ‘Confirmed’ according to standard criteria.

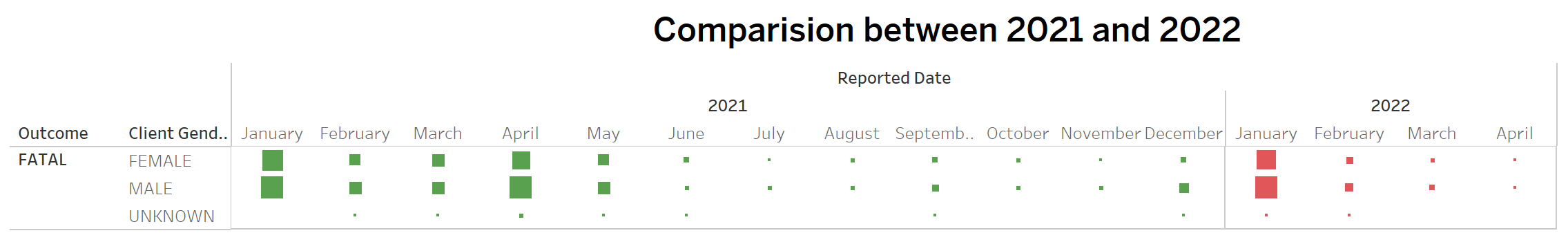


1. **What is the relationship between Fatal cases and source of Infection?**



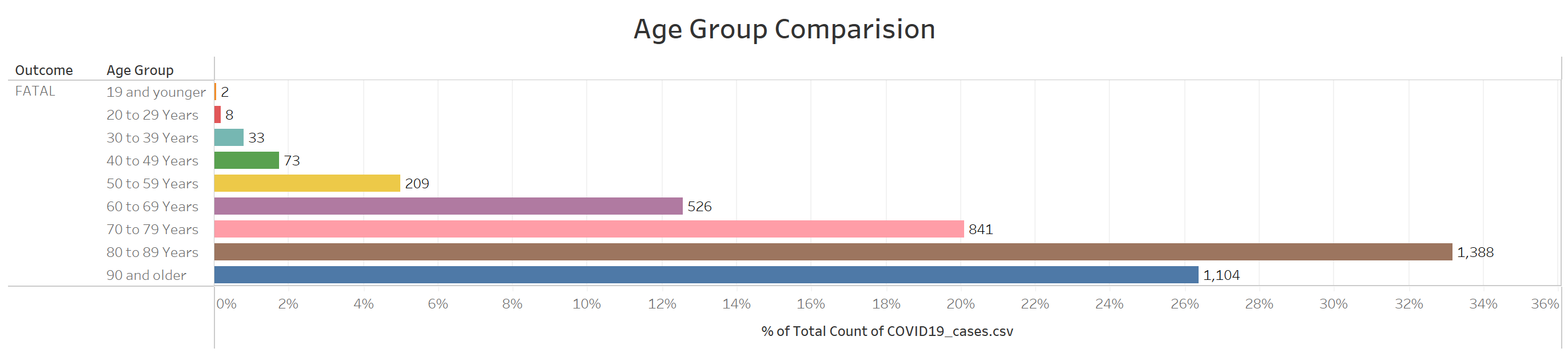
One of the factors that may determine the accuracy of prediction model is ‘Source of Infection’. The Fatal cases are stated based on the results of Hospitalized cases. As per the visual it is evident that community and Healthcare Institutions are the two significant sources of infection for the fatality outcome.

1. **In which months of the years 2021 and 2022 the fatality rate is higher according to gender?**



The above heat map depicts the fatal cases according to each month of the years 2021 and 2022 by gender. In the year 2021, the fatality rate is higher in both the gender in the months between January – May. And in 2022, fatality rate is higher in both the gender in the month of January.

1. **In which Age group the fatality ratio is higher?**



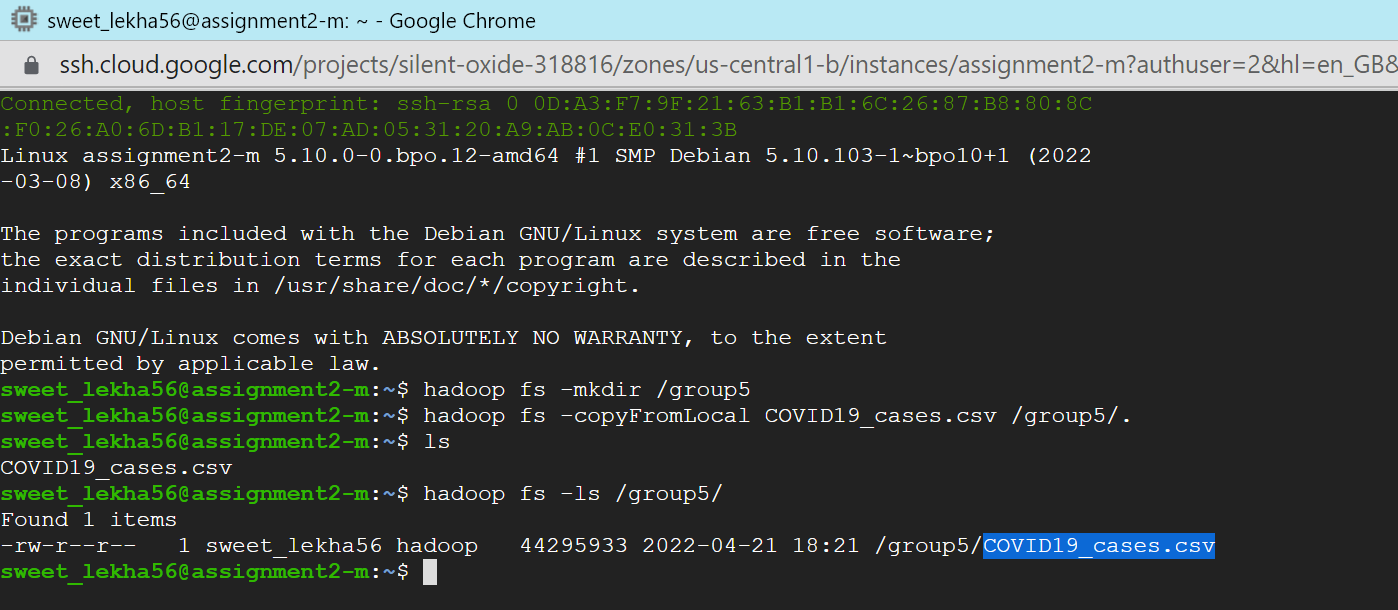
Another important factors that may determine the accuracy of prediction model is ‘Age Groups’. The above chart depicts the comparisons made between the columns ‘Fatal’ and ‘Age groups’. It is very clear from the above visual that fatality ratio is higher in senior citizens above 60 years of age.

Other factors that will contributed to determining the accuracy of prediction model are ‘Currently Hospitalized’, ‘Currently in ICU’, ‘Currently Intubated’, ‘Ever Hospitalized’, ‘Ever in ICU’ and ‘Ever Intubated’ with results ‘yes’.

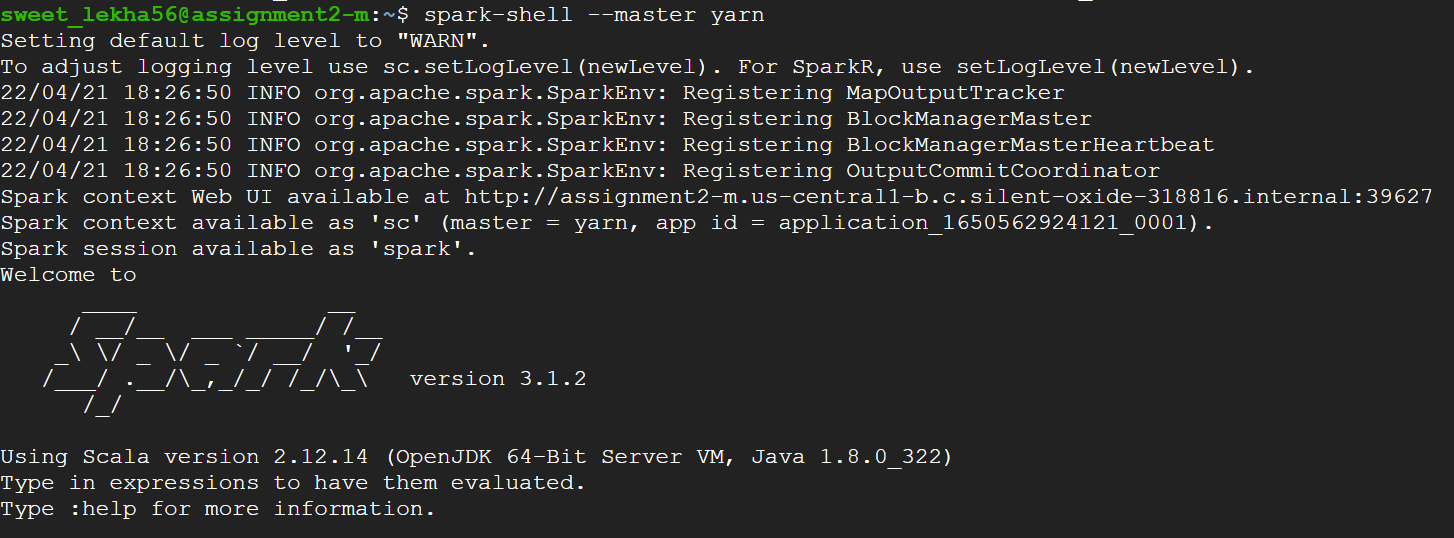
**Solution Walkthrough:**

The below mentioned steps are executed in Google Cloud Platform (GCP) to implement the machine learning algorithm using SparkML library in Scala to predict the accuracy of “FATAL” cases.

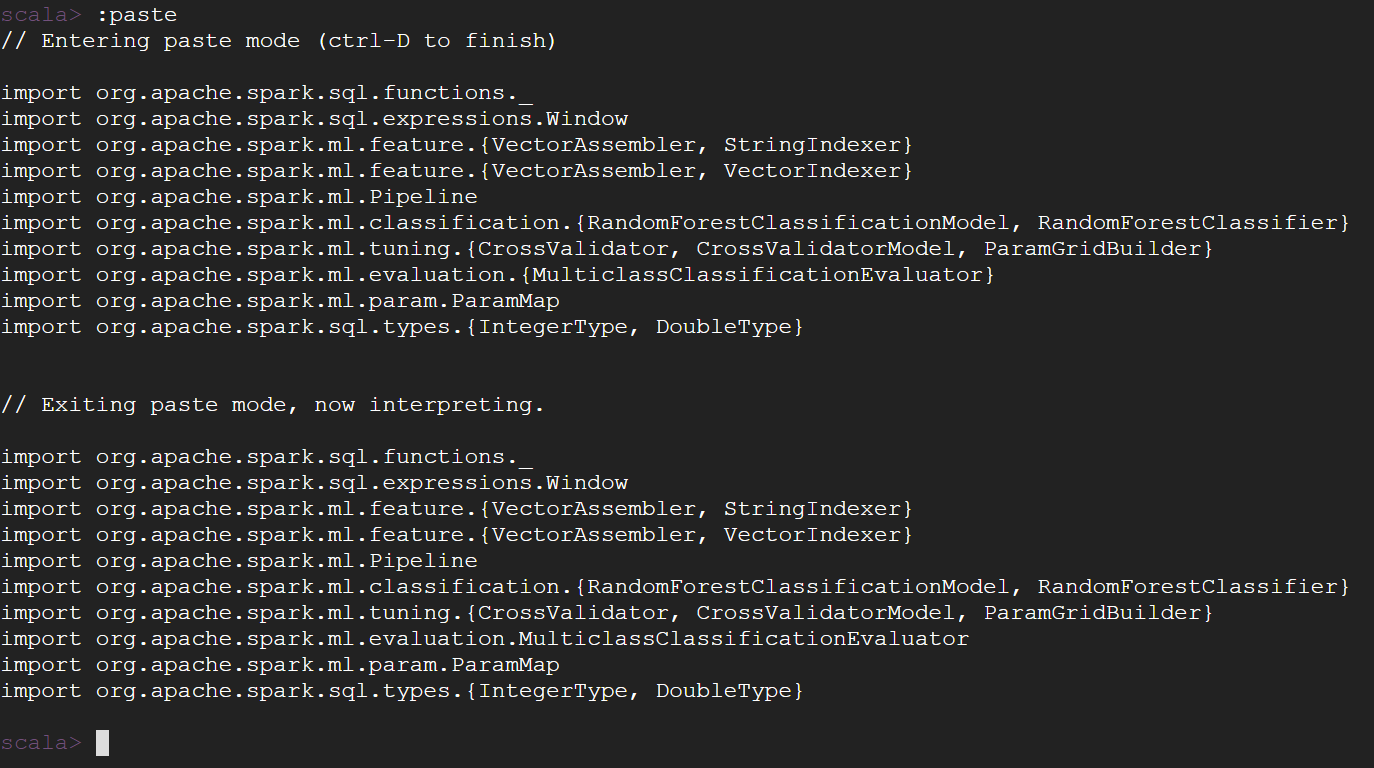
Copying the Dataset to Hadoop Cluster,



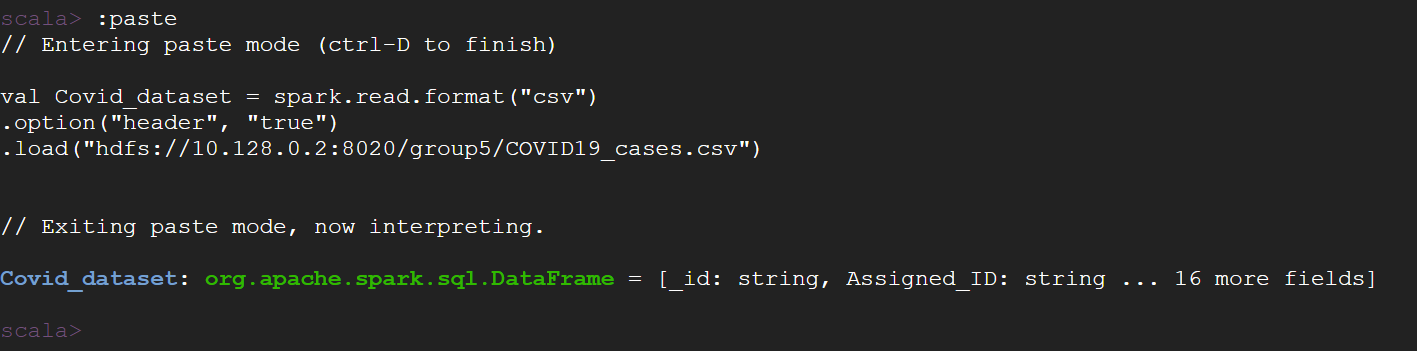
Initiating Spark version 3.1,



Importing necessary libraries,



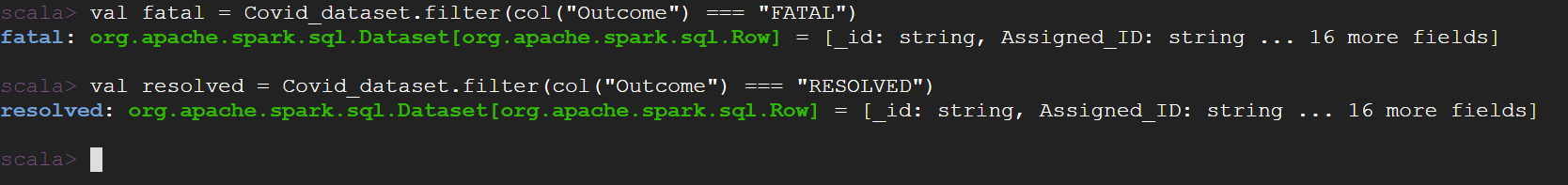
Loading the dataset,



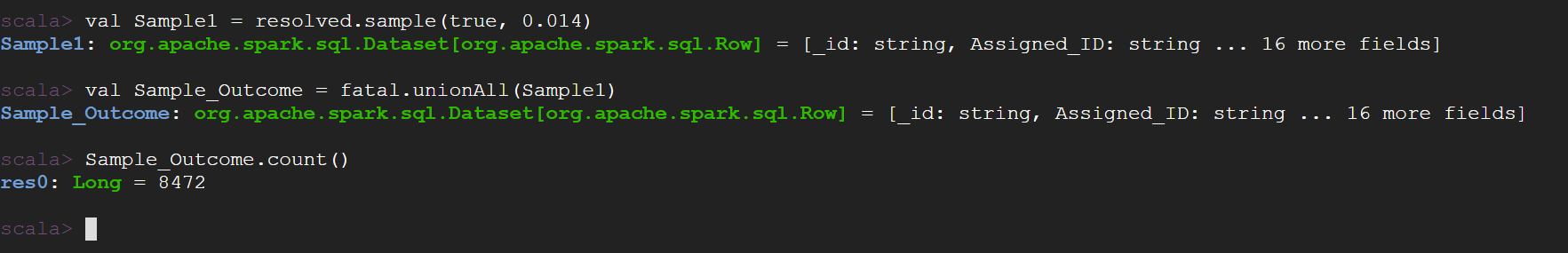
**Balancing the Dataset:**

As the Resolved cases are over 96% in the dataset, We are balancing the ‘Resolved cases’ according to ‘Fatal cases’ ratio to predict the accuracy.

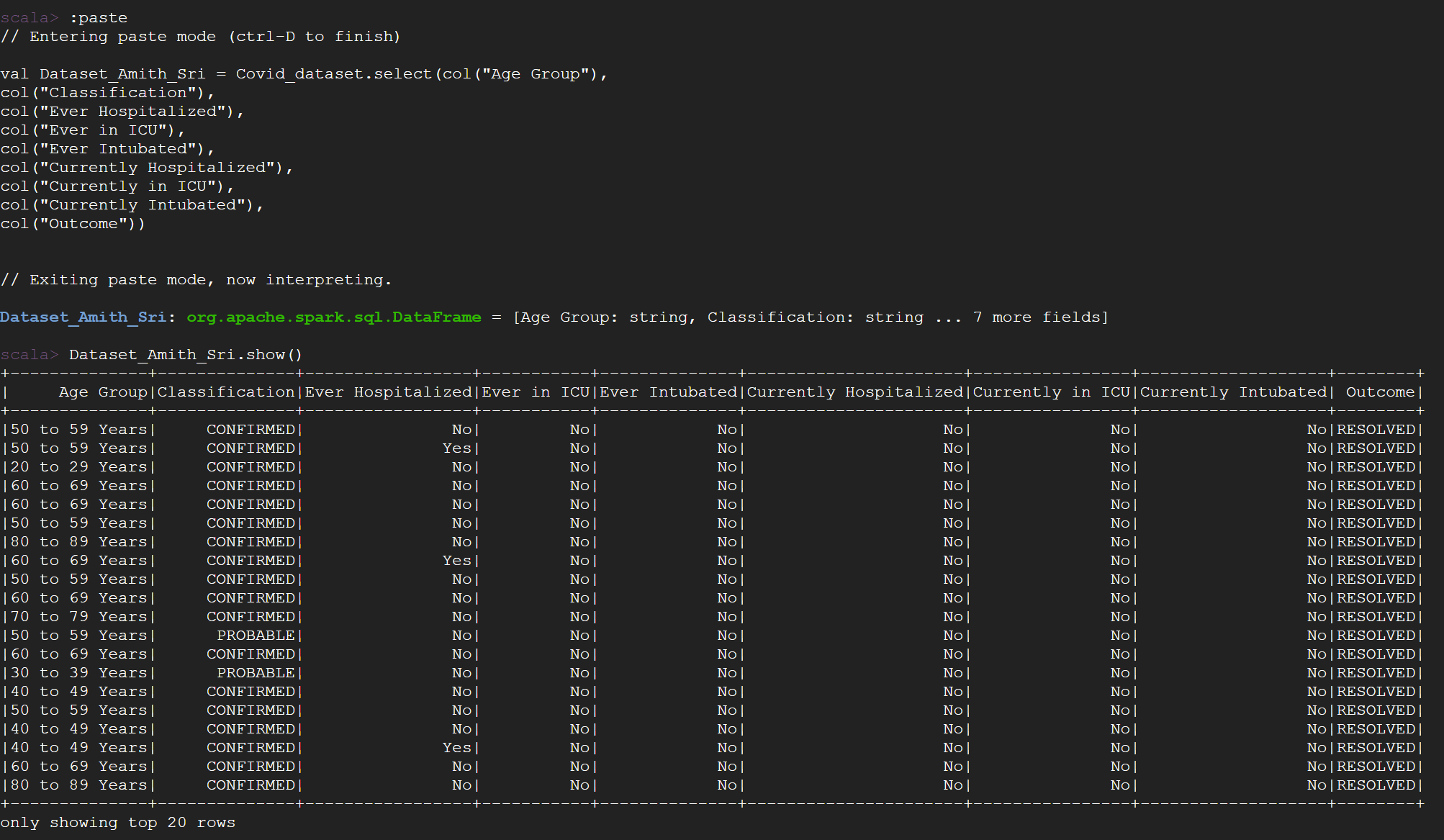
Creating two different data frames by filtering Fatal and Resolved values from the column ‘Outcome’



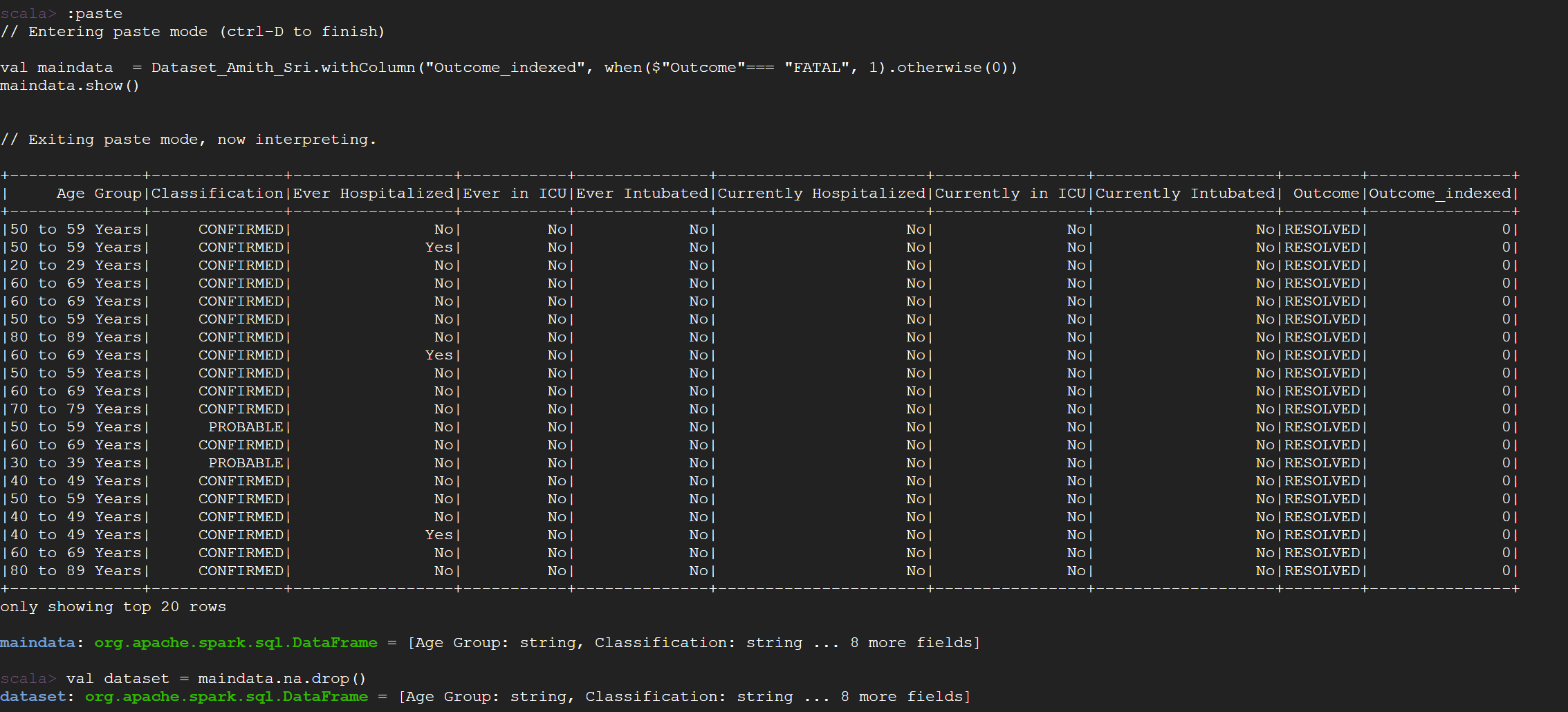
Taking a sample of Resolved cases matching the Fatality ratio. After code execution the total count of the data set is **8472.**



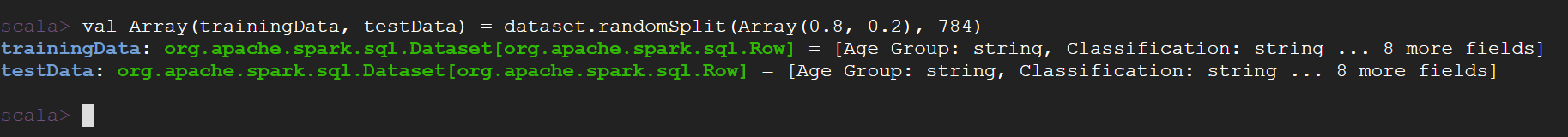
Creating a new data frame from the balanced dataset with only required columns



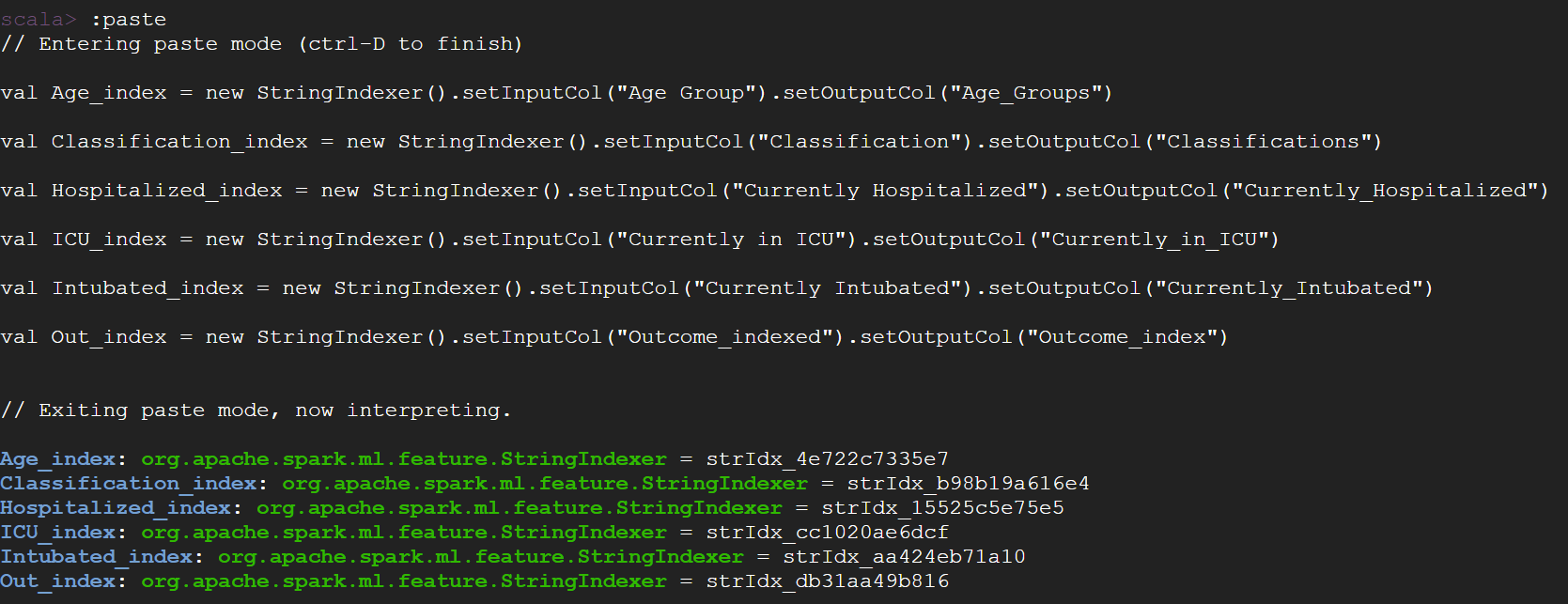
The outcome column has been filtered for only ‘FATAL’ values and the null values (if any present) from the dataset are dropped.



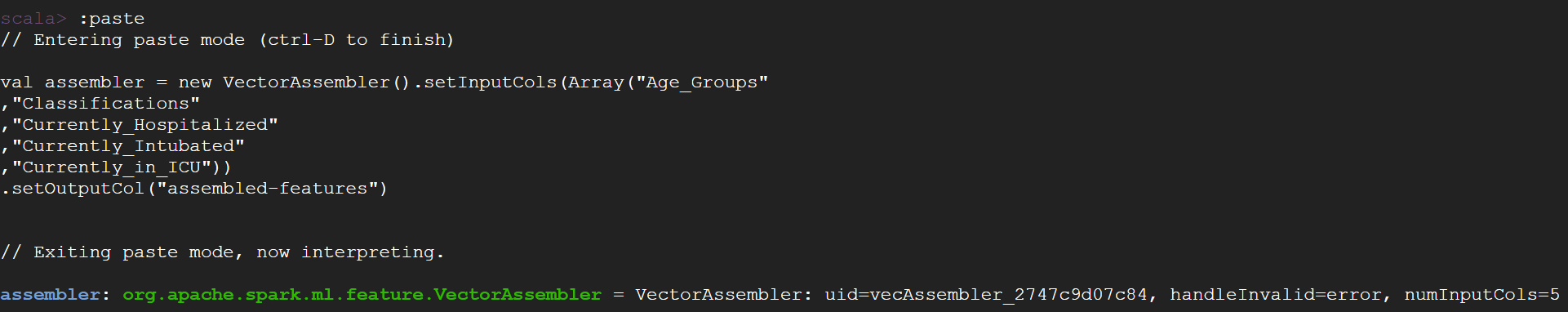
Splitting the dataset into training and test data typical 80 and 20 ratio and we give a seed so we have the same random data in each set.



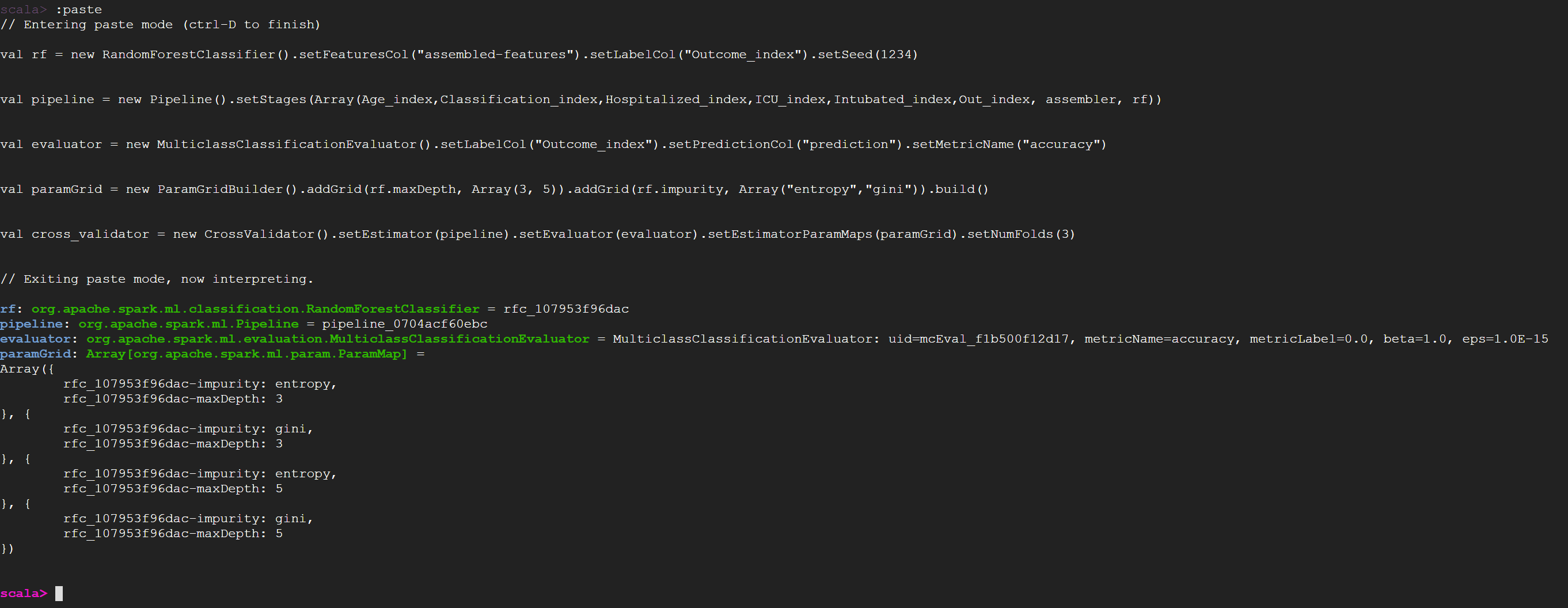
Indexing all the columns,



Using VectorAssembler function, Input columns and the Output column has been specified



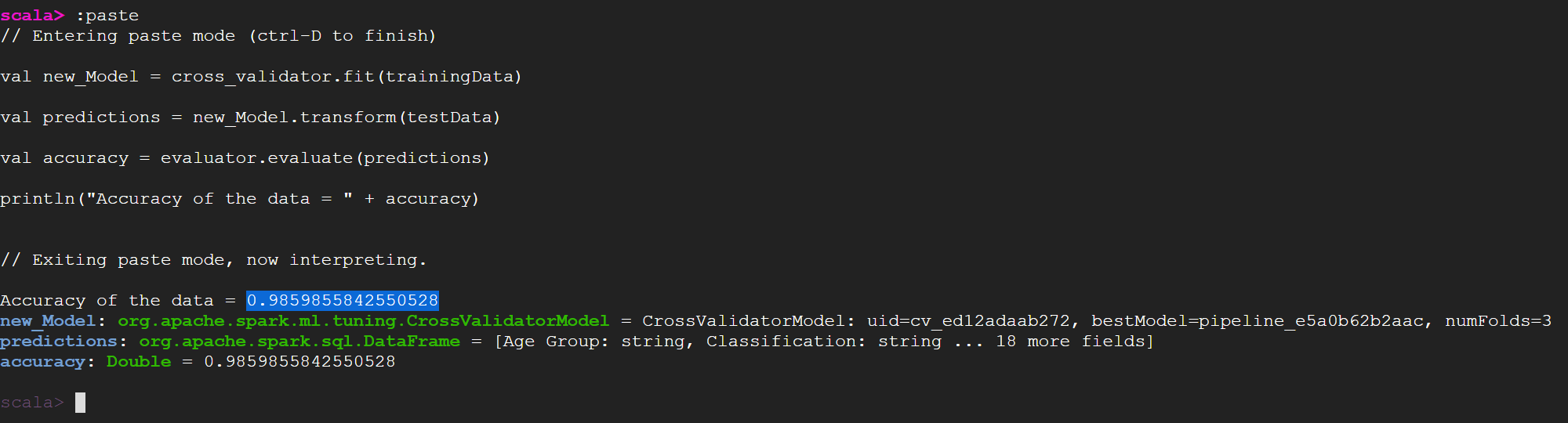
Applying Random Forest Classifier,



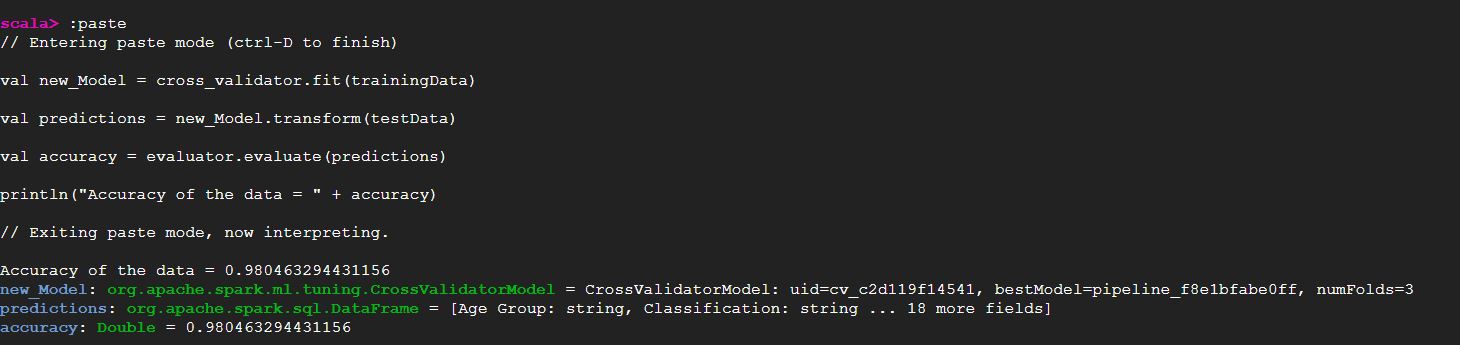
Accuracy of the model is calculated using Random Forest Classifier.

Algorithm Contraption and the Accuracy of the model is shown below,

The **Fatal cases** of the outcome column in the dataset is **98% Accurate**.



The above procedure has been followed to predict the accuracy of ‘ACTIVE’ cases also.



The **Active cases** of the outcome column in the dataset isalso **98%** Accurate.

**Conclusion**

This approach of predicting model using Machine learning helped us to determine the accuracy of Fatal cases of Covid 19 infection of the years 2020, 2021 and 2022 in Toronto. Using Random Forest classifier model, accuracy rate is higher with 98%.

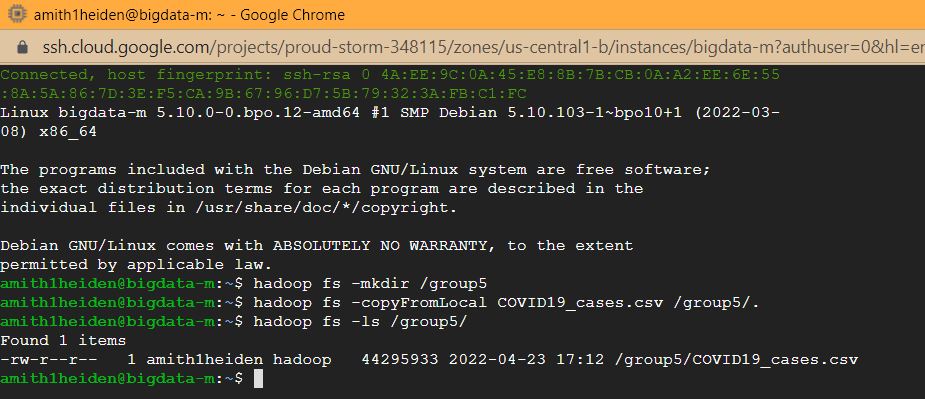
**Recommendations**

* Firstly, after analyzing the visuals, the fatality ratio in Senior citizens above age 60 are higher compared to the other age groups. So, the government should impose some restriction (like public gathering) on those age group people so that the fatality rate can be controlled.
* Secondly, community and Healthcare Institutions are the two significant sources of infection, therefore the government should concentrate on these areas and take proper measure to protect the people.

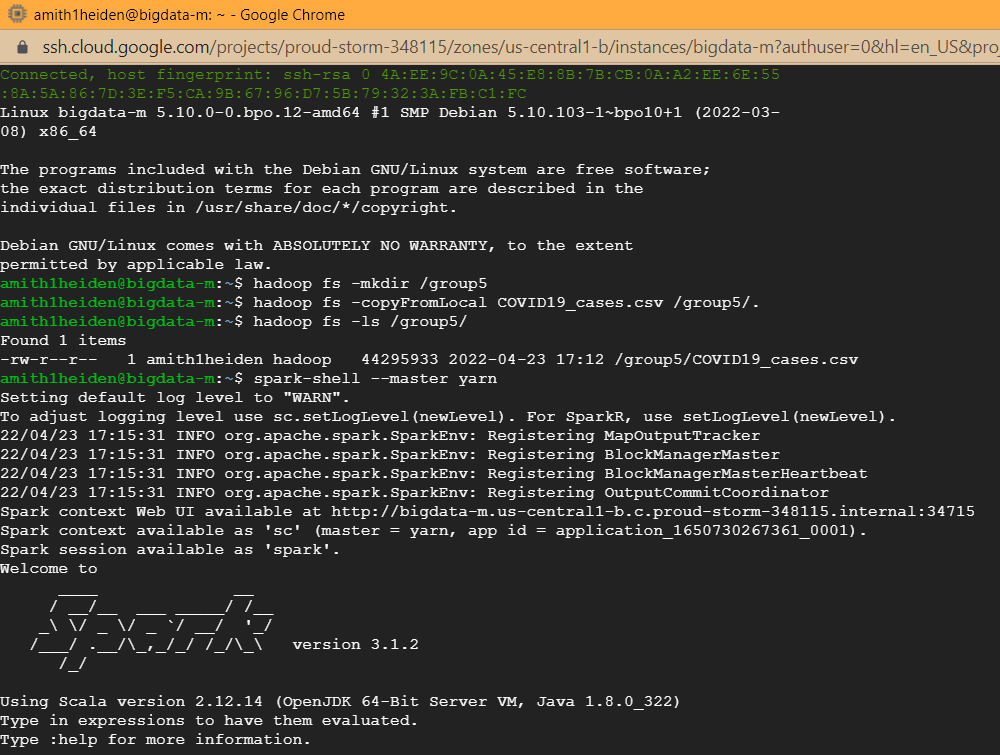
**Appendix**

Solution walkthrough for predicting the Accuracy of ‘ACTIVE’ Cases.

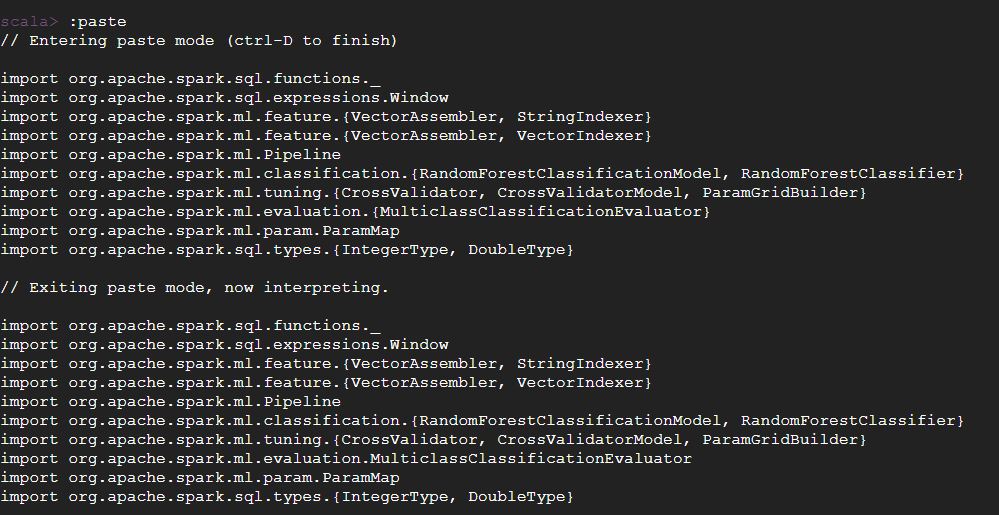
Copying the Dataset to Hadoop Cluster,



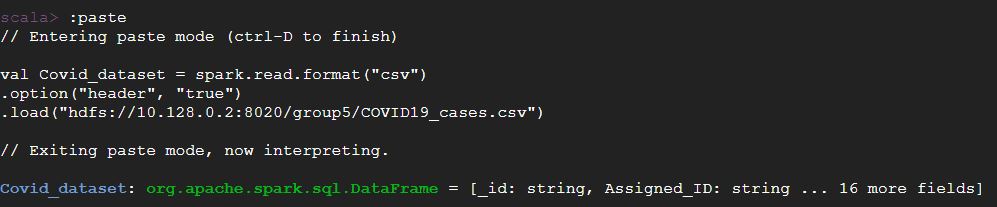
Initiating Spark version 3.1,



Importing necessary libraries,



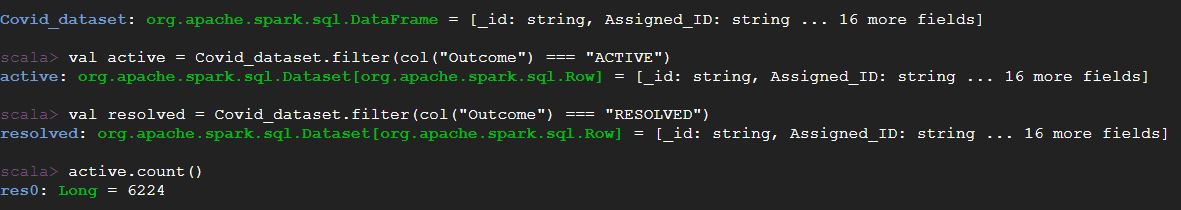
Loading the dataset,



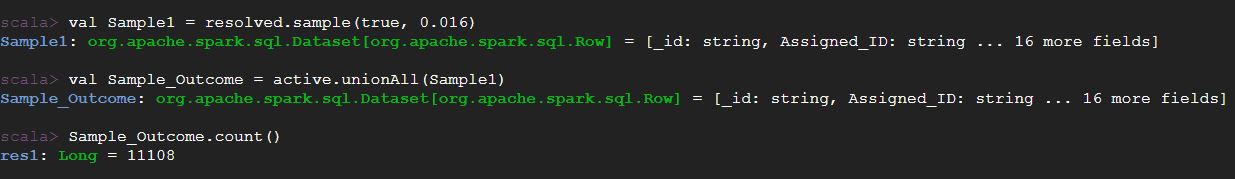
**Balancing the Dataset:**

As the Resolved cases are over 96% in the dataset, We are balancing the ‘Resolved cases’ according to ‘Active cases’ ratio to predict the accuracy.

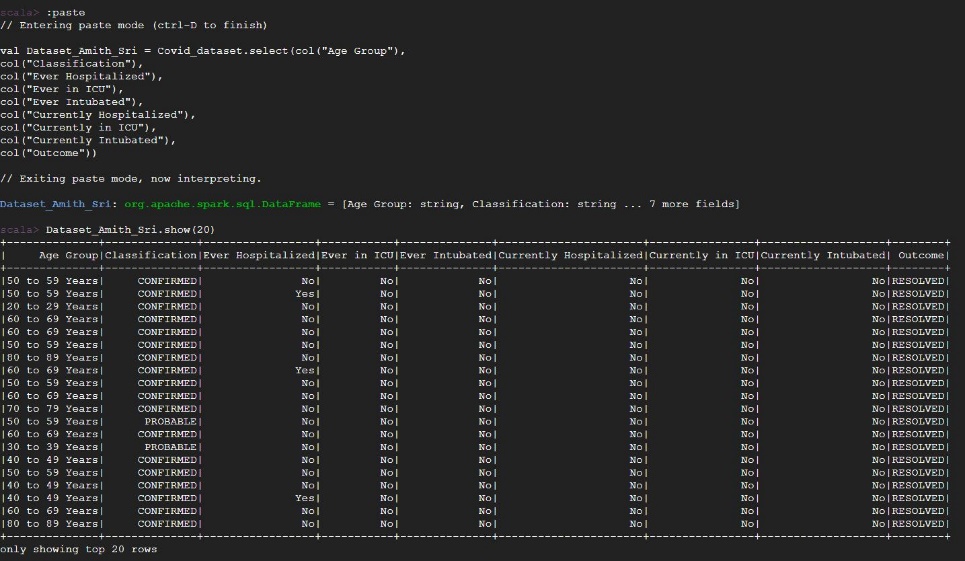
Creating two different data frames by filtering Active and Resolved values from the column ‘Outcome’



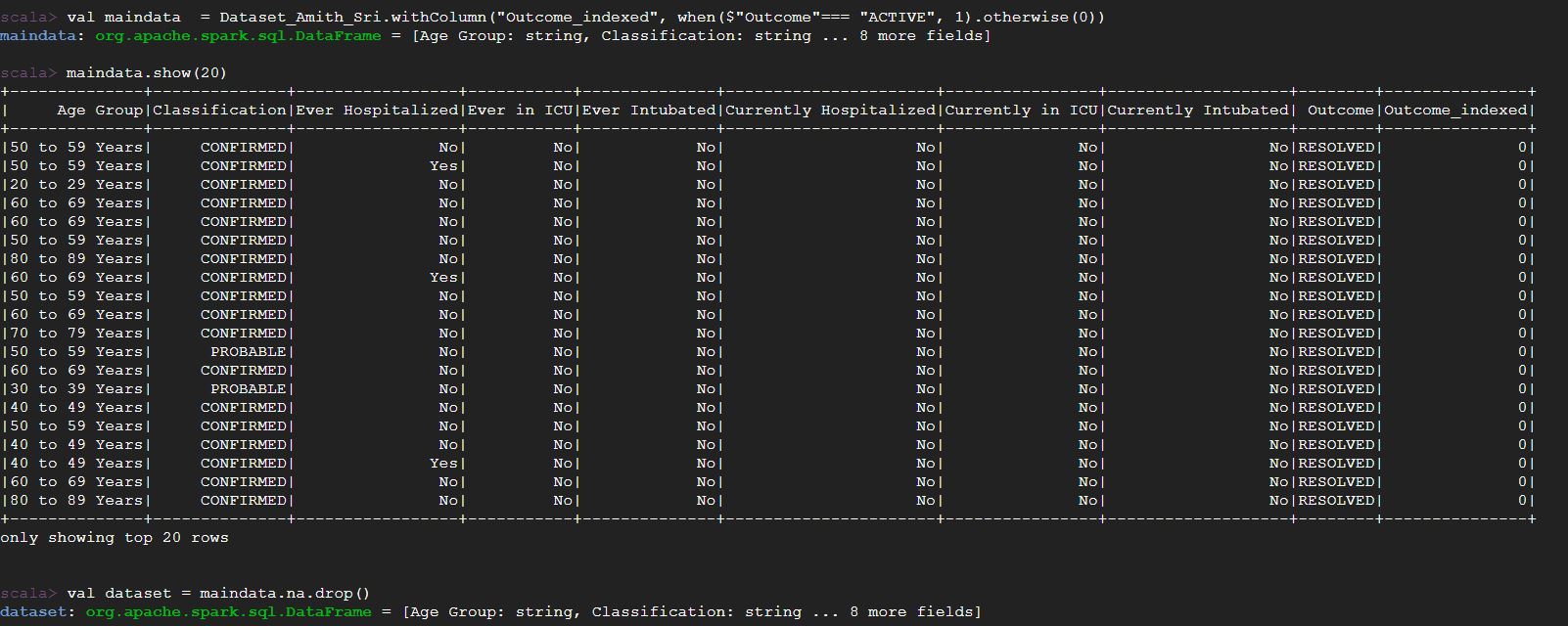
Taking a sample of Resolved cases matching the Active ratio. After code execution the total count of the data set is **11108.**



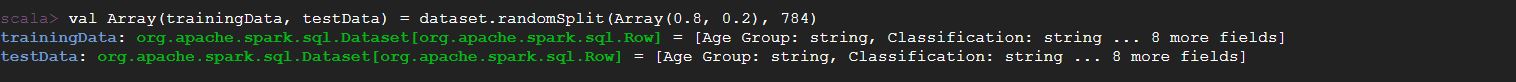
Creating a new data frame from the balanced dataset with only required columns



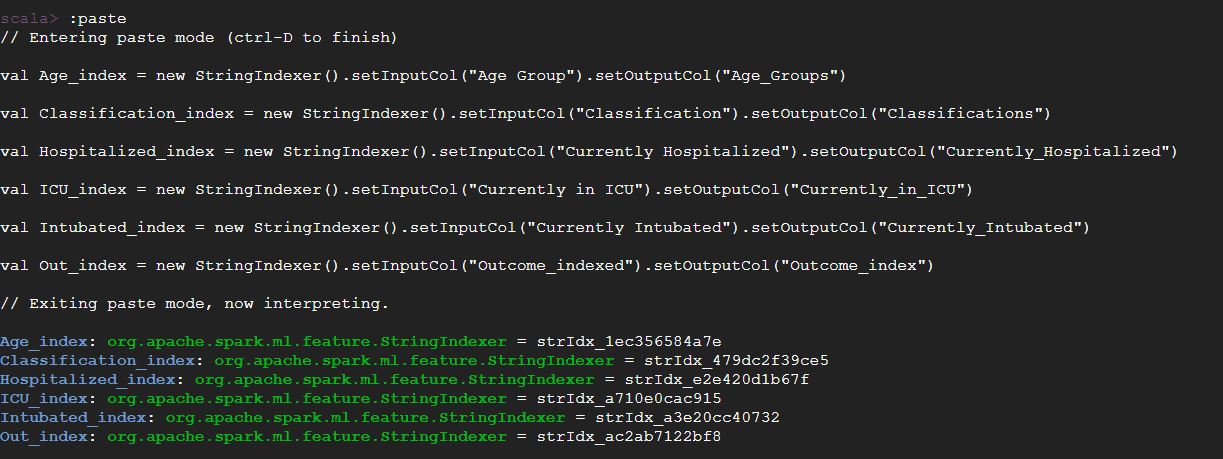
The outcome column has been filtered for only ‘Active’ values and the null values (if any present) from the dataset are dropped.



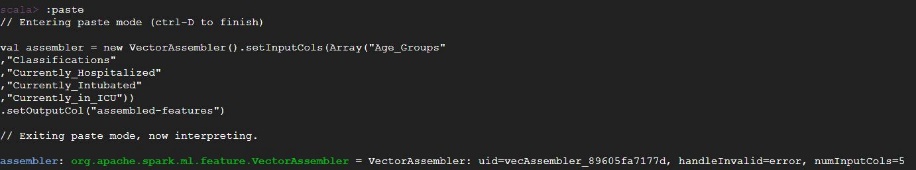
Splitting the dataset into training and test data typical 80 and 20 ratio and we give a seed so we have the same random data in each set.



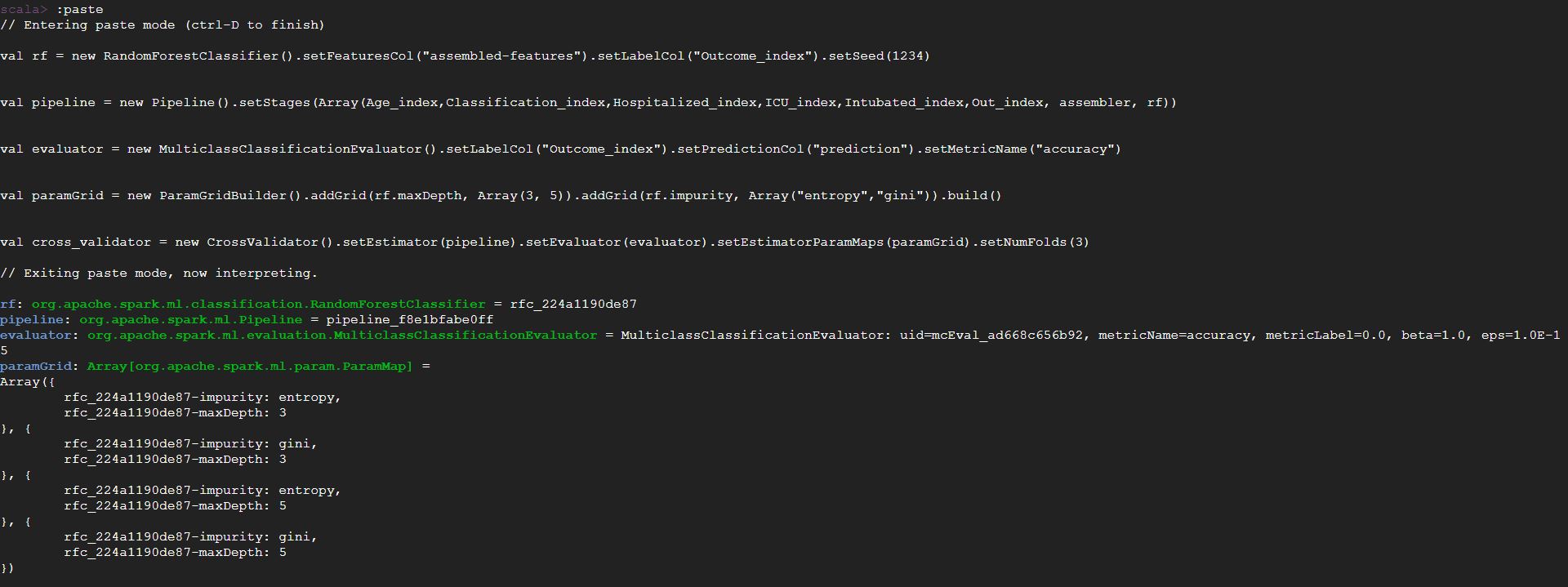
Indexing all the columns,



Using VectorAssembler function, Input columns and the Output column has been specified



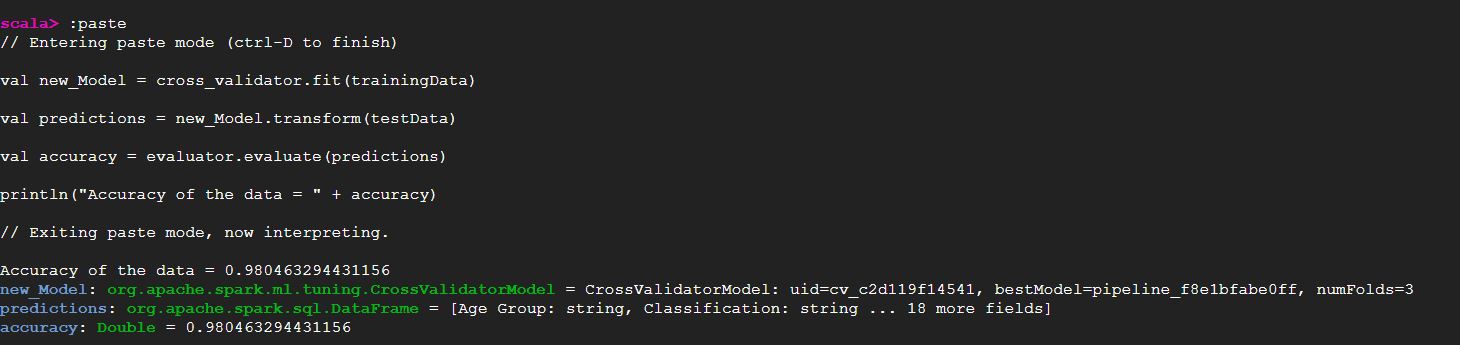
Applying Random Forest Classifier,



Accuracy of the model is calculated using Random Forest Classifier.

Algorithm Contraption and the Accuracy of the model is shown below,

The **Active cases** of the outcome column in the dataset is **98% Accurate**.



**Scala Codes**

Copying the Dataset to Hadoop Cluster,

Hadoopfs -mkdir /group5

hadoop fs -copyFromLocal COVID19\_cases.csv /group5/.

hadoop fs -ls /group5/

Initiating Spark version 3.1,

spark-shell --master yarn

Importing necessary libraries,

import org.apache.spark.sql.functions.\_

import org.apache.spark.sql.expressions.Window

import org.apache.spark.ml.feature.{VectorAssembler, StringIndexer}

import org.apache.spark.ml.feature.{VectorAssembler, VectorIndexer}

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}

Loading the dataset,

val Covid\_dataset = spark.read.format("csv")

.option("header", "true")

.load("hdfs://10.128.0.2:8020/group5/COVID19\_cases.csv")

Balancing the Dataset:

val active = Covid\_dataset.filter(col("Outcome") === "ACTIVE")

val resolved = Covid\_dataset.filter(col("Outcome") === "RESOLVED")

fatal.count()

val Sample1 = resolved.sample(true, 0.016)

val Sample\_Outcome = active.unionAll(Sample1)

Sample\_Outcome.count()

Creating a new data frame from the balanced dataset with only required columns

val Dataset\_Amith\_Sri = Covid\_dataset.select(col("Age Group"),

col("Classification"),

col("Ever Hospitalized"),

col("Ever in ICU"),

col("Ever Intubated"),

col("Currently Hospitalized"),

col("Currently in ICU"),

col("Currently Intubated"),

col("Outcome"))

The outcome column has been filtered for only ‘FATAL’ values and the null values (if any present) from the dataset are dropped.

val maindata = Dataset\_Amith\_Sri.withColumn("Outcome\_indexed", when($"Outcome"=== "ACTIVE", 1).otherwise(0))

maindata.show()

val dataset = maindata.na.drop()

Splitting the dataset into training and test data typical 80 and 20 ratio and we give a seed so we have the same random data in each set.

val Array(trainingData, testData) = dataset.randomSplit(Array(0.8, 0.2), 784)

Indexing all the columns,

val Age\_index = new StringIndexer().setInputCol("Age Group").setOutputCol("Age\_Groups")

val Classification\_index = new StringIndexer().setInputCol("Classification").setOutputCol("Classifications")

val Hospitalized\_index = new StringIndexer().setInputCol("Currently Hospitalized").setOutputCol("Currently\_Hospitalized")

val ICU\_index = new StringIndexer().setInputCol("Currently in ICU").setOutputCol("Currently\_in\_ICU")

val Intubated\_index = new StringIndexer().setInputCol("Currently Intubated").setOutputCol("Currently\_Intubated")

val Out\_index = new StringIndexer().setInputCol("Outcome\_indexed").setOutputCol("Outcome\_index")

Using VectorAssembler function, Input columns and the Output column has been specified

val assembler = new VectorAssembler().setInputCols(Array("Age\_Groups"

,"Classifications"

,"Currently\_Hospitalized"

,"Currently\_Intubated"

,"Currently\_in\_ICU"))

.setOutputCol("assembled-features")

Applying Random Forest Classifier,

val rf = new RandomForestClassifier().setFeaturesCol("assembled-features").setLabelCol("Outcome\_index").setSeed(1234)

val pipeline = new Pipeline().setStages(Array(Age\_index,Classification\_index,Hospitalized\_index,ICU\_index,Intubated\_index,Out\_index, assembler, rf))

val evaluator = new MulticlassClassificationEvaluator().setLabelCol("Outcome\_index").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)

Calculating the accuracy of the model

val new\_Model = cross\_validator.fit(trainingData)

val predictions = new\_Model.transform(testData)

val accuracy = evaluator.evaluate(predictions)

println("Accuracy of the data = " + accuracy)

**References**

College, C. (2022, 04 20). blackboard . Retrieved from blackboard:

<https://gc.blackboard.com/ultra/courses/_328785_1/cl/outline>

Open Data Dataset. (n.d.). City of Toronto Open Data Portal, from <https://open.toronto.ca/dataset/covid-19-cases-in-toronto/> , Accessed 16 Apr. 2022