Apache Spark and Scala Certification Training Course

Apache Spark Streaming – Data Sources

Case Study: Spam Detection
Domain: Telecom
Module -11

A telecom software provider is building an application to monitor different telecom components in the production environment. For monitoring purpose, the applicationrelies on log files by parsing the log files and looking for potential warning or exceptions in the logs and reporting them. The POC we had been working on, for SPAM Detection on the data of telecom operator forum, has been accepted and the stakeholders has asked us to work on the real-time example for predicting SPAM messages.

Tasks:

This POC will focus on saved machine learning model for spam prediction with streaming data to do real-time prediction. Now with model and data pipeline ready,you are required to predict the spam message on the steaming data.

1. Modify the model application to train the model and persist it.
2. Create a new spark streaming application to predict the spam messages.
3. Application will connect to the flume to retrieve the data.
4. Load the model.
5. Predict the SPAM messages and print the SPAM in the logs.
6. Test the application by sending dummy data rows from the consumer.

STEP 1:
Creating the Scala File and upload in edgeNode
SpamMessage.scala

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.ml.feature.Tokenizer
import org.apache.spark.ml.feature.StopWordsRemover
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
import org.apache.spark.ml.{Pipeline, PipelineModel}
import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.feature.{OneHotEncoder, StringIndexer}
object SpamMessage {
 def main(args: Array[String]) {
 val spark = SparkSession.builder.appName("Spam Messages
Detection").getOrCreate()
```

```
 spark.sparkContext.setLogLevel("ERROR")
 // Load data in spark
 val raw
=spark.read.option("delimiter","\t").csv("/user/kaushikdey45edu/SpamMessage/SM
SSpamCollection").toDF("spam","message")
 // Extract words from the SMS message
 val tokenizer = new Tokenizer().setInputCol("message").setOutputCol("words")
 // Modify the stop words to include your custom words such as â-â
 val stopwords = new StopWordsRemover().getStopWords ++ Array("-")
 val remover = new
StopWordsRemover().setStopWords(stopwords).setInputCol("words").setOutputCol(
"filtered")
 // Create the features from SMS message using CountVectorizer
 val cvmodel = new
CountVectorizer().setInputCol("filtered").setOutputCol("features")
 val indexer = new StringIndexer().setInputCol("spam").setOutputCol("label")
 val lr = new
LogisticRegression().setMaxIter(10).setRegParam(0.3).setElasticNetParam(0.8)
 val pipeline = new Pipeline().setStages(Array(tokenizer, remover, cvmodel,indexer,
lr))
 val model = pipeline.fit(raw)

model.save("/user/kaushikdey45edu/SpamMessage/ModelSave/spam_message.mo
del")
 val output = model.transform(raw)
 println("--->", output.show())
 spark.stop()
 }
}
```

Folder Structure of Scala File
/home/kaushikdey45edu/SpamMessage/src/main/scala/SpamMessage.scala
Folder Structure of Jar File
/home/kaushikdey45edu/SpamMessage/target/scala-2.11/SpamMessage_2.11-1.0.jar

Buid.sbt file (/home/kaushikdey45edu/SpamMessage/build.sbt**)**

```
name :=" SpamMessage "
version :="1.0"
scalaVersion :="2.11.8"
val sparkVersion = "2.2.0"
libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % sparkVersion,
```

```
"org.apache.spark" %% "spark-sql" % sparkVersion,
"org.apache.spark" %% "spark-mllib" % sparkVersion,
"org.apache.spark" %% "spark-streaming-flume" % "2.1.0"
)
```

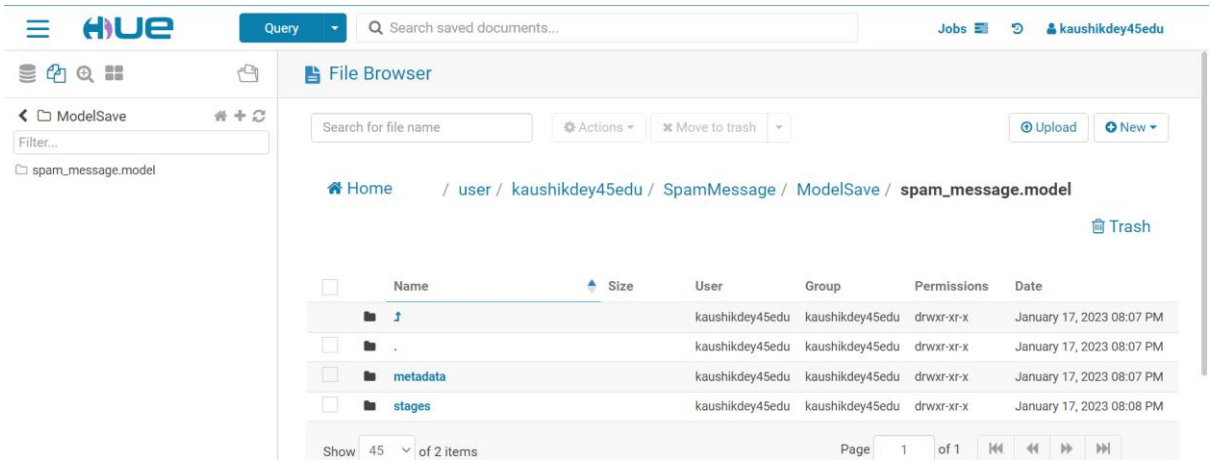
Create Folder in HDFS and upload the SMSSpamCollection file.

1) hadoop fs -mkdir SpamMessage
2) Upload the csv file inside SpamMessage (SMSSpamCollection)
3) hadoop fs -mkdir /user/kaushikdey45edu/SpamMessage/ModelSave
4) upload the Model inside Model Save(spam_message.model)

Run the Jar File

[kaushikdey45edu@ip-10-1-1-204 ~]$ spark-submit --class "SpamMessage" --master yarn /home/kaushikdey45edu/SpamMessage/target/scala-2.11/SpamMessage_2.11-1.0.jar

```
minRegisteredResourcesRatio: 0.8
```



STEP 2:

Create a Table in MYSQL

1. First login into mySQL instance mysql -u kaushikdey45edu -p
2. Provide the password as per Big Data lab
3. create table spam_message(message varchar(140),prediction double);
4. select * from spam_message;

STEP 3:

Create New Spark Streaming Application to predict the spam Message

SpamStreaming.Scala

```
import org.apache.spark.sql.SparkSession
//import org.apache.spark.ml.feature.QuantileDiscretizer
//import org.apache.spark.sql.types._
//import org.apache.spark.ml.feature.VectorAssembler
//import org.apache.spark.ml.linalg.Vectors
//import org.apache.spark.ml.Pipeline
import org.apache.spark.sql.functions._
//import org.apache.spark.ml.evaluation.RegressionEvaluator
//import
org.apache.spark.ml.regression.{RandomForestRegressionModel,RandomForestRegressor}
import org.apache.spark._
import org.apache.spark.streaming._
import org.apache.spark.sql.Encoders
import org.apache.spark.ml._
case class Message(message:String)
object SpamStreaming {
 def main(args: Array[String]) {
 val conf = new SparkConf().setAppName("SpamStreaming")
 val ssc = new StreamingContext(conf, Seconds(10))
 val lines = ssc.textFileStream("/user/kaushikdey45edu/spam_message")
 lines.foreachRDD {
    rdd =>val spark=SparkSession.builder().getOrCreate()
              import spark.implicits._
    val rawRdd = rdd.map(Message(_))
    val raw = spark.createDataFrame(rawRdd)
    val pipeline =
PipelineModel.read.load("/user/kaushikdey45edu/SpamMessage/ModelSave/spam_messag
e.model")
    val predictions = pipeline.transform(raw)
    val prop = new java.util.Properties
    prop.put("driver", "com.mysql.jdbc.Driver");
    prop.put("url", "jdbc:jdbc:mysql://ip-10-1-1-204.ap-south-
1.compute.internal/kaushikdey45edu");
    prop.put("user", "kaushikdey45edu");
    prop.put("password", "PurpleCrow52@");
    predictions.select("message","prediction").write.mode("append").jdbc(
                                    prop.getProperty("url"), "spam_message", prop)
            }
 ssc.start()
 ssc.awaitTermination()
 }
 }
```

Run the Jar File

```
[kaushikdey45edu@ip-10-1-1-204 ~]$ spark-submit --class "
SpamStreaming" --master yarn
/home/kaushikdey45edu/SpamStreming/target/scala-2.11/
spamstreaming_2.11-1.0.jar
```