

Create Riskactor from Spark Using Zeppelin

Code

```
%spark2
val geolocation_temp2 = hiveContext.sql("SELECT driverid, count(driverid) occurrence from geolocation_temp1 where
event!='normal' group by driverid")
```

Error Logs

```
org.apache.spark.sql.AnalysisException: cannot resolve ''event'' given input columns: [column9, column2, column4, column5,
column3, column10, column7, column1, column8, column6]; line 1 pos 72;
'Aggregate ['driverid], ['driverid, 'count('driverid) AS occurrence#85]
+- 'Filter NOT ('event = normal)
+- SubqueryAlias geolocation_temp1
+- Project [column1#12, column2#13, column3#14, column4#15, column5#16, column6#17, column7#18, column8#19,
column9#20, column10#21]
+- SubqueryAlias geolocation
+- HiveTableRelation `default`.geolocation, org.apache.hadoop.hive.ql.io.orc.OrcSerde, [column1#12, column2#13,
column3#14, column4#15, column5#16, column6#17, column7#18, column8#19, column9#20, column10#21]
at org.apache.spark.sql.catalyst.analysis.package$AnalysisErrorAt.failAnalysis(package.scala:42)
at
org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1$$anonfun$apply$2.applyOrElse(CheckAnalysis
.scala:88)
at
org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1$$anonfun$apply$2.applyOrElse(CheckAnalysis
.scala:85)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$transformUp$1.apply(TreeNode.scala:289)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$transformUp$1.apply(TreeNode.scala:289)
at org.apache.spark.sql.catalyst.trees.CurrentOrigin$.withOrigin(TreeNode.scala:70)
at org.apache.spark.sql.catalyst.trees.TreeNode.transformUp(TreeNode.scala:288)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$3.apply(TreeNode.scala:286)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$3.apply(TreeNode.scala:286)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$4.apply(TreeNode.scala:306)
at org.apache.spark.sql.catalyst.trees.TreeNode.mapProductIterator(TreeNode.scala:187)
at org.apache.spark.sql.catalyst.trees.TreeNode.mapChildren(TreeNode.scala:304)
at org.apache.spark.sql.catalyst.trees.TreeNode.transformUp(TreeNode.scala:286)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$3.apply(TreeNode.scala:286)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$3.apply(TreeNode.scala:286)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$4.apply(TreeNode.scala:306)
at org.apache.spark.sql.catalyst.trees.TreeNode.mapProductIterator(TreeNode.scala:187)
at org.apache.spark.sql.catalyst.trees.TreeNode.mapChildren(TreeNode.scala:304)
at org.apache.spark.sql.catalyst.trees.TreeNode.transformUp(TreeNode.scala:286)
at org.apache.spark.sql.catalyst.plans.QueryPlan$$anonfun$transformExpressionsUp$1.apply(QueryPlan.scala:268)
at org.apache.spark.sql.catalyst.plans.QueryPlan$$anonfun$transformExpressionsUp$1.apply(QueryPlan.scala:268)
at org.apache.spark.sql.catalyst.plans.QueryPlan.transformExpression$1(QueryPlan.scala:279)
at
org.apache.spark.sql.catalyst.plans.QueryPlan.org$apache$spark$sql$catalyst$plans$QueryPlan$$recursiveTransform$1(Query
Plan.scala:289)
at org.apache.spark.sql.catalyst.plans.QueryPlan$$anonfun$6.apply(QueryPlan.scala:298)
at org.apache.spark.sql.catalyst.trees.TreeNode.mapProductIterator(TreeNode.scala:187)
at org.apache.spark.sql.catalyst.plans.QueryPlan.mapExpressions(QueryPlan.scala:298)
at org.apache.spark.sql.catalyst.plans.QueryPlan.transformExpressionsUp(QueryPlan.scala:268)
at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1.apply(CheckAnalysis.scala:85)
at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$$anonfun$checkAnalysis$1.apply(CheckAnalysis.scala:78)
at org.apache.spark.sql.catalyst.trees.TreeNode.foreachUp(TreeNode.scala:127)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$foreachUp$1.apply(TreeNode.scala:126)
at org.apache.spark.sql.catalyst.trees.TreeNode$$anonfun$foreachUp$1.apply(TreeNode.scala:126)
at scala.collection.immutable.List.foreach(List.scala:381)
at org.apache.spark.sql.catalyst.trees.TreeNode.foreachUp(TreeNode.scala:126)
at org.apache.spark.sql.catalyst.analysis.CheckAnalysis$class.checkAnalysis(CheckAnalysis.scala:78)
```

```
at org.apache.spark.sql.catalyst.analysis.Analyzer.checkAnalysis(Analyzer.scala:91)
at org.apache.spark.sql.execution.QueryExecution.assertAnalyzed(QueryExecution.scala:52)
at org.apache.spark.sql.Dataset$.ofRows(Dataset.scala:67)
at org.apache.spark.sql.SparkSession.sql(SparkSession.scala:637)
... 47 elided
```