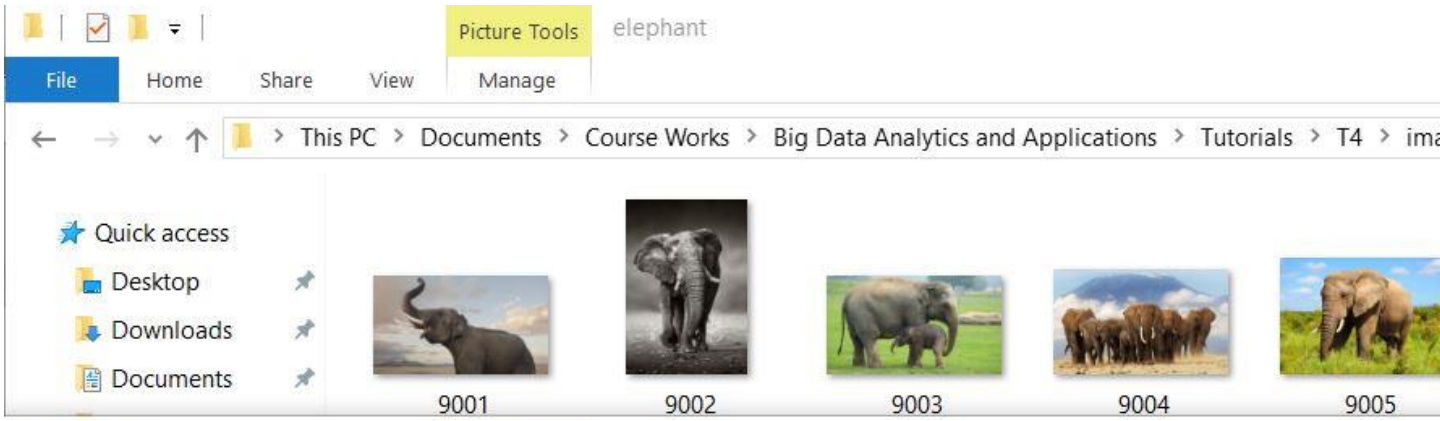
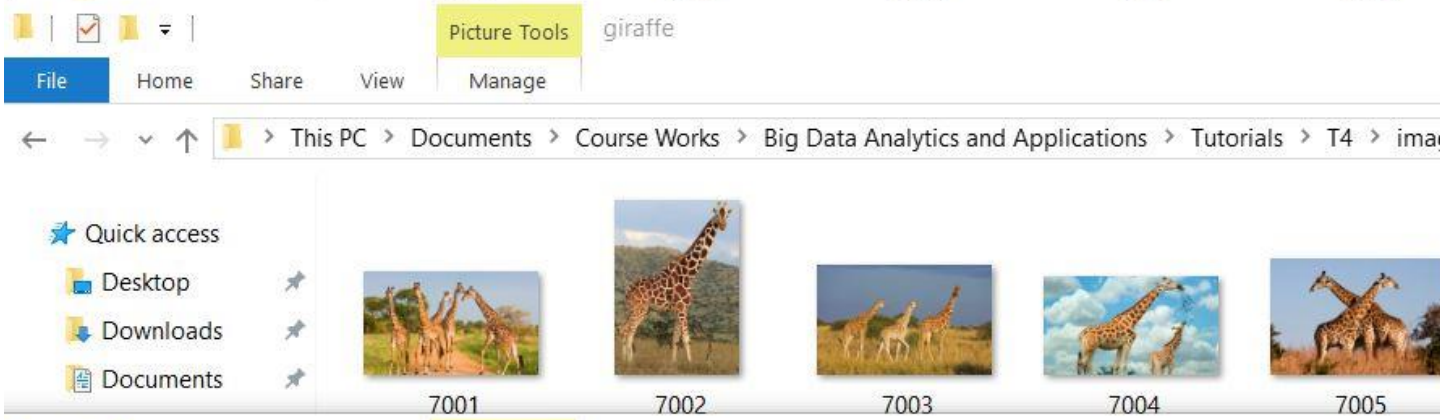
Task:

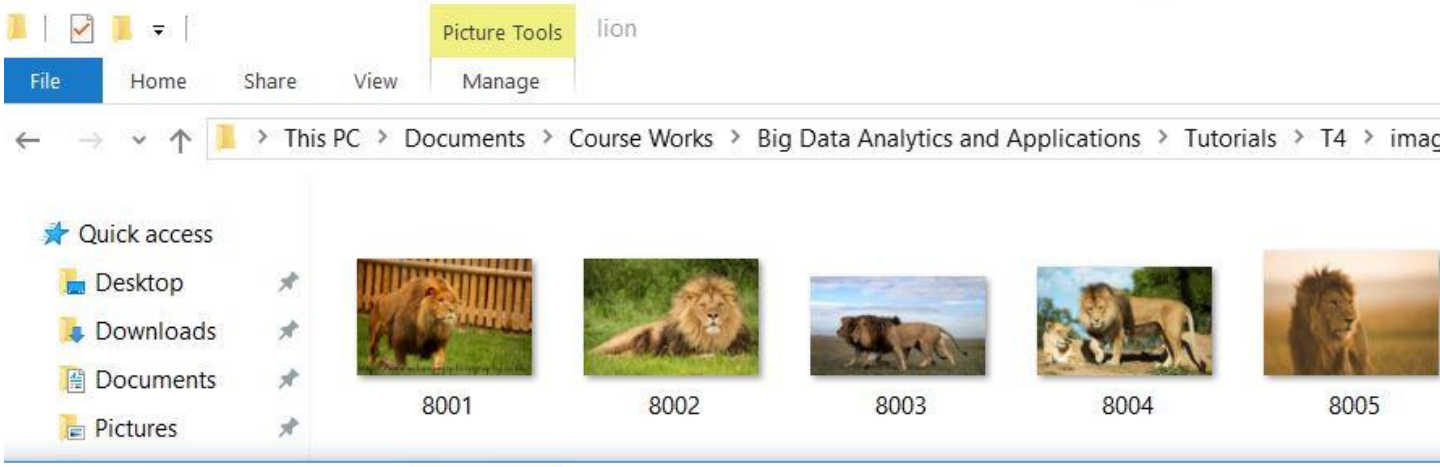
1. Create your own dataset for Image Classification Problem. Use the workflow as discussed in the Tutorial 4 Session using Decision Tree Algorithm. Report the accuracy and confusion matrix obtained. In the Wiki Page, include a brief description of your dataset and purpose behind image classification problem .

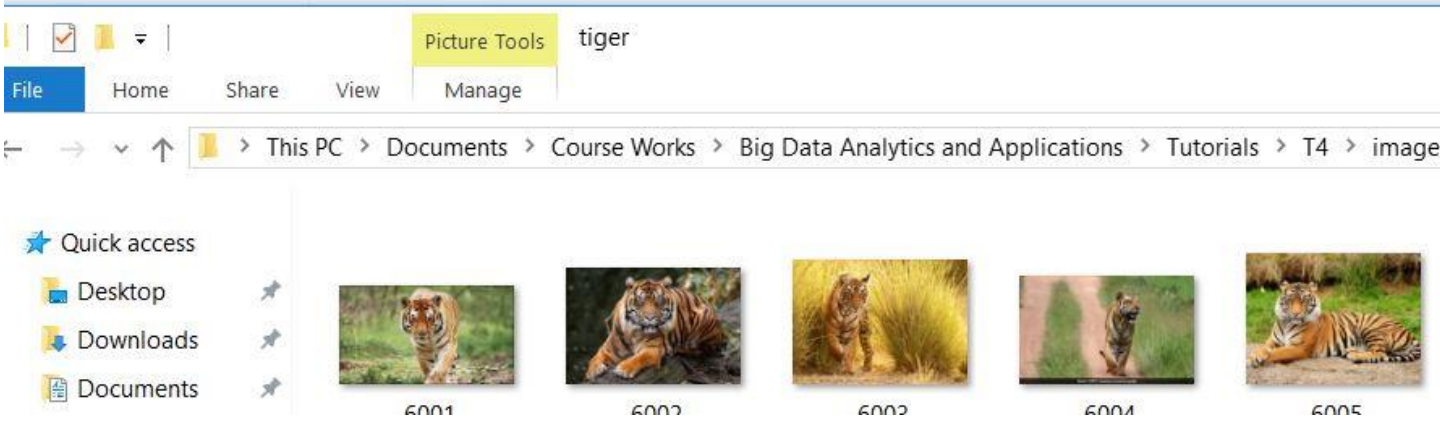
- I have taken animals images and trained the model.
- Training images contains four classes 'elephant','tiger','giraffe' and 'lion'. Each class contains 12 images.
- Following folder structures indicate the training data for the model.

Training data


Picture Tools
elephant
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Tutorials
T4
Quick access
Desktop
Downloads
Documents
9001
9002
9003
9004
9005


Picture Tools
giraffe
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Tutorials
T4
Quick access
Desktop
Downloads
Documents
7001
7002
7003
7004
7005


Picture Tools
lion
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Tutorials
T4
Quick access
Desktop
Downloads
Documents
Pictures
8001
8002
8003
8004
8005


Picture Tools
tiger
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Tutorials
T4
Quick access
Desktop
Downloads
Documents

- Four image categories are taken in the source code. They are "elephant","giraffe","lion","tiger".

Image Categories

```
object IPApp {
  val featureVectorsCluster = new mutable.MutableList[String]

  val IMAGE_CATEGORIES = List("elephant", "giraffe", "lion", "tiger")
  /**
    *
    * @param sc     : SparkContext
    * @param images : Images list from the training set
    */
```

- Extracted the image descriptors and applied K-Means clustering algorithms for the extracted descriptors.

K-Means Implementation

```
val numClusters = 400
val numIterations = 20
val clusters = KMeans.train(parsedData, numClusters, numIterations)
```

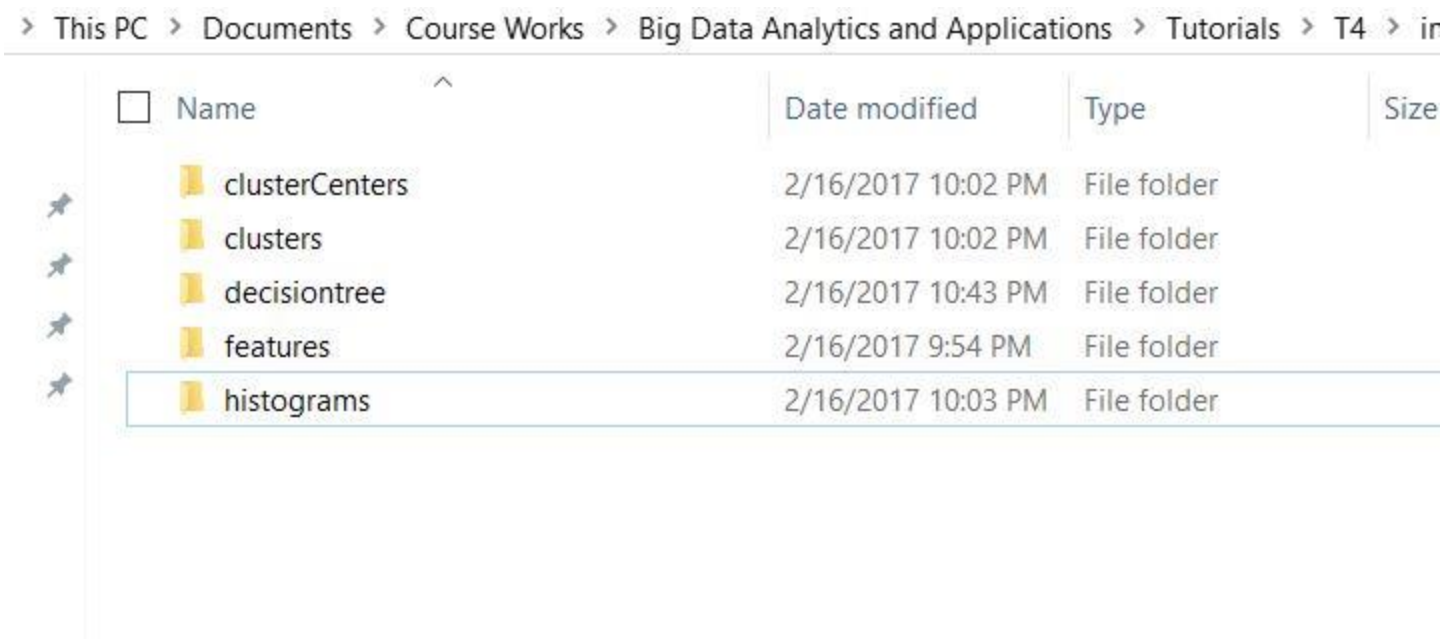
- Based on the generated clusters, Histograms were generated. After the generation of histograms, Decision tree algorithm was applied on the data to save the decision tree model.

Decision Tree Implementation

```
val model = DecisionTree.trainClassifier(training, numClasses, categor
   impurity, maxDepth, maxBins)
```

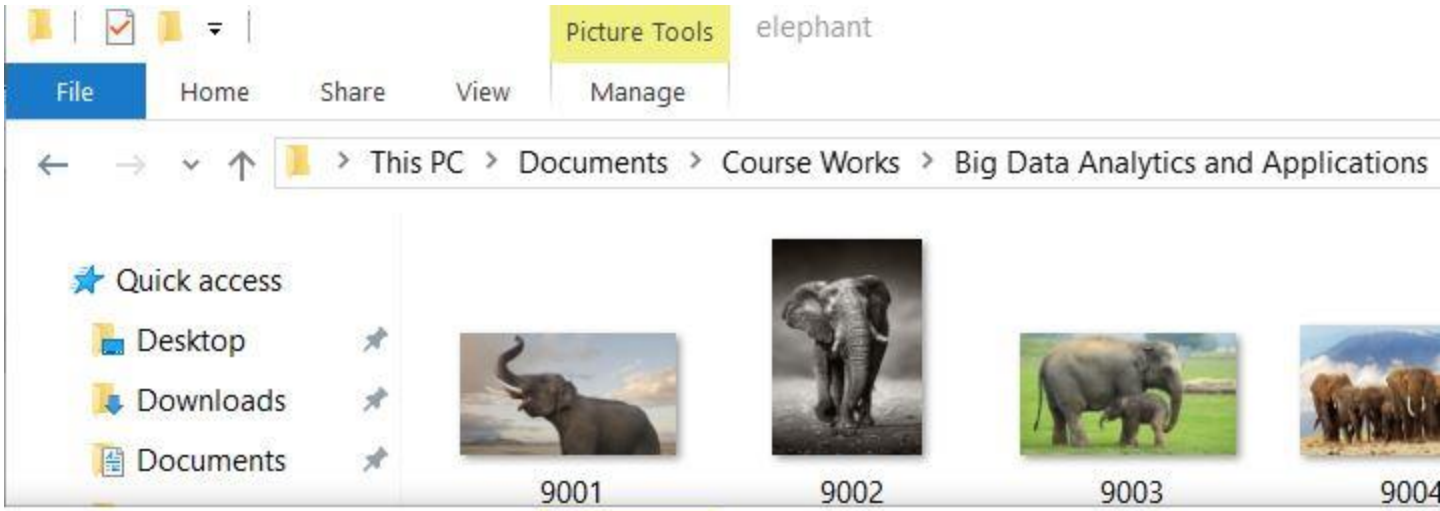
- Generated data was saved under 'features','clusters','clusterCenters','histograms' and 'decisiontree' folders.

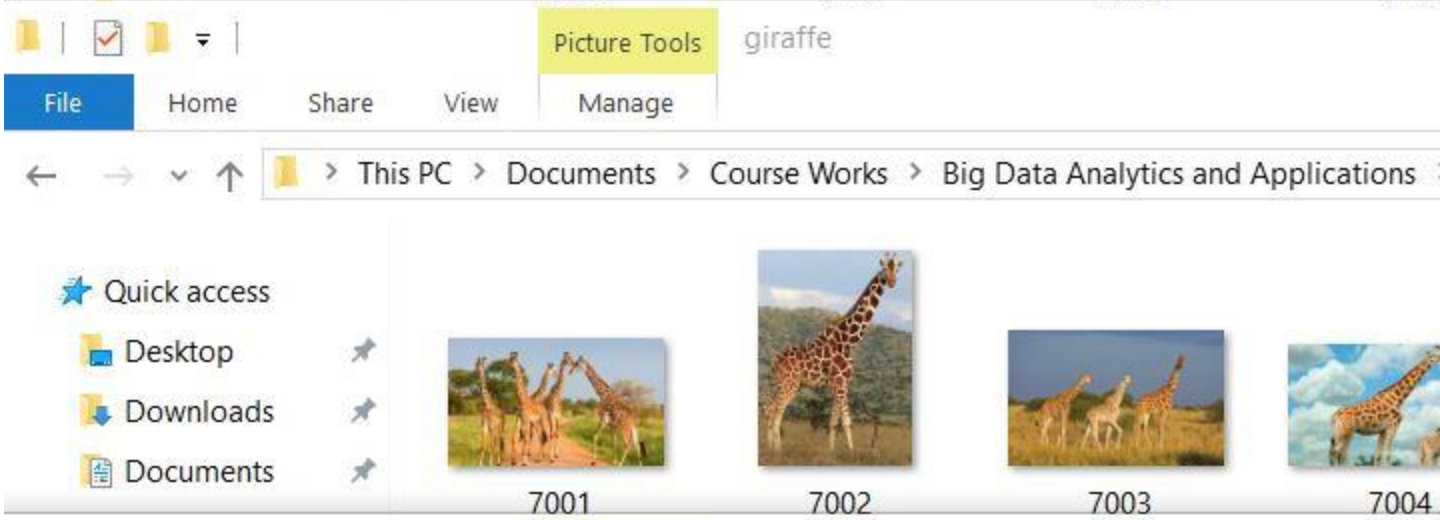
Saved models

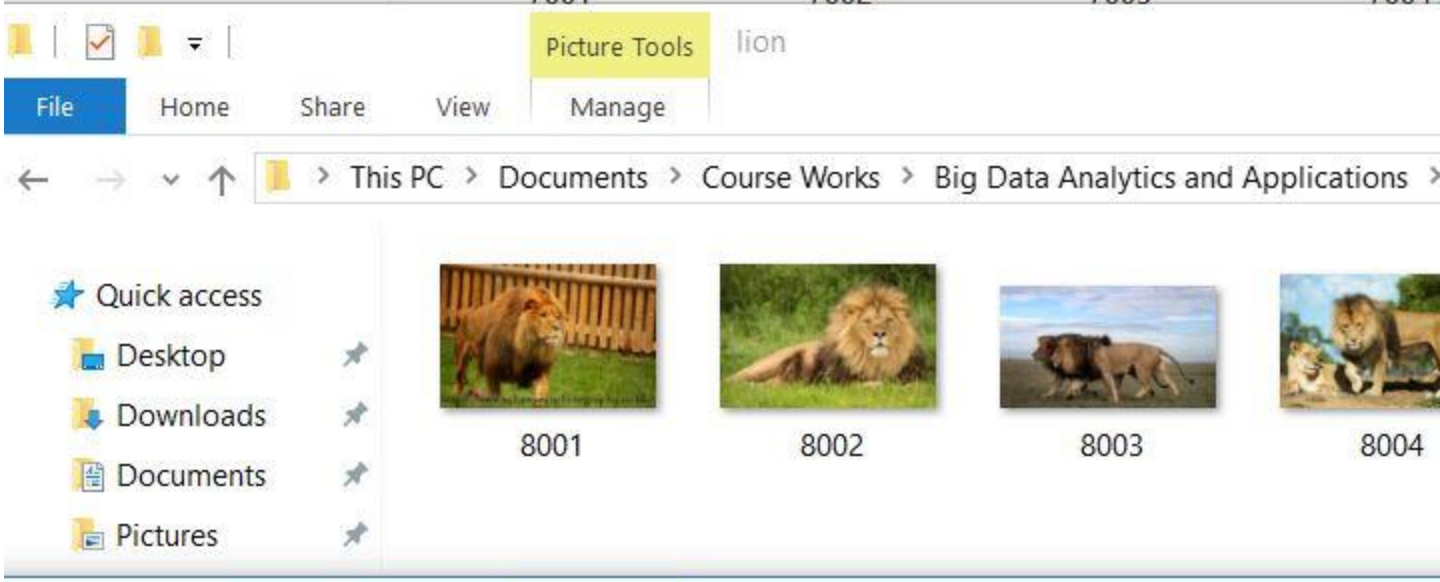


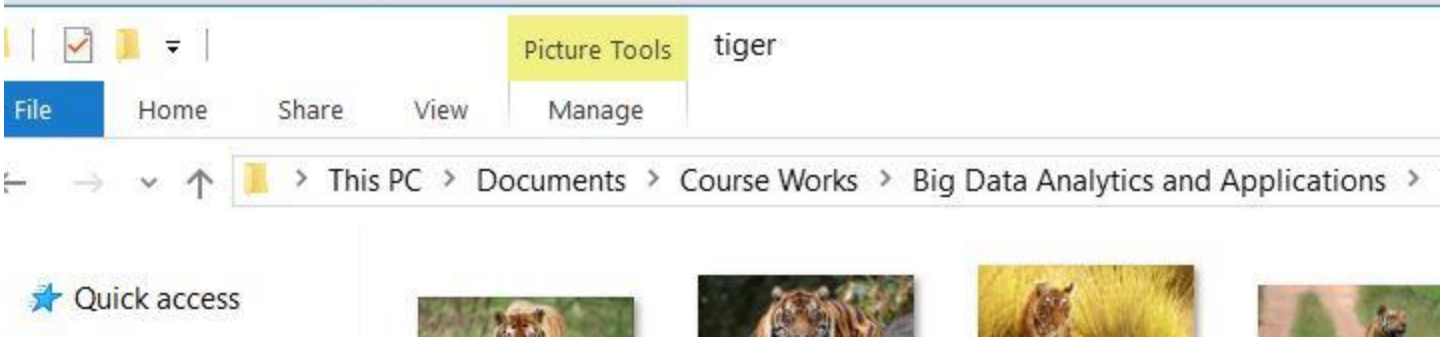
- Test data contains the same four classes and each class contains 5 images.

Test Data


Picture Tools
elephant
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Quick access
Desktop
Downloads
Documents
9001
9002
9003
9004


Picture Tools
giraffe
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Quick access
Desktop
Downloads
Documents
7001
7002
7003
7004

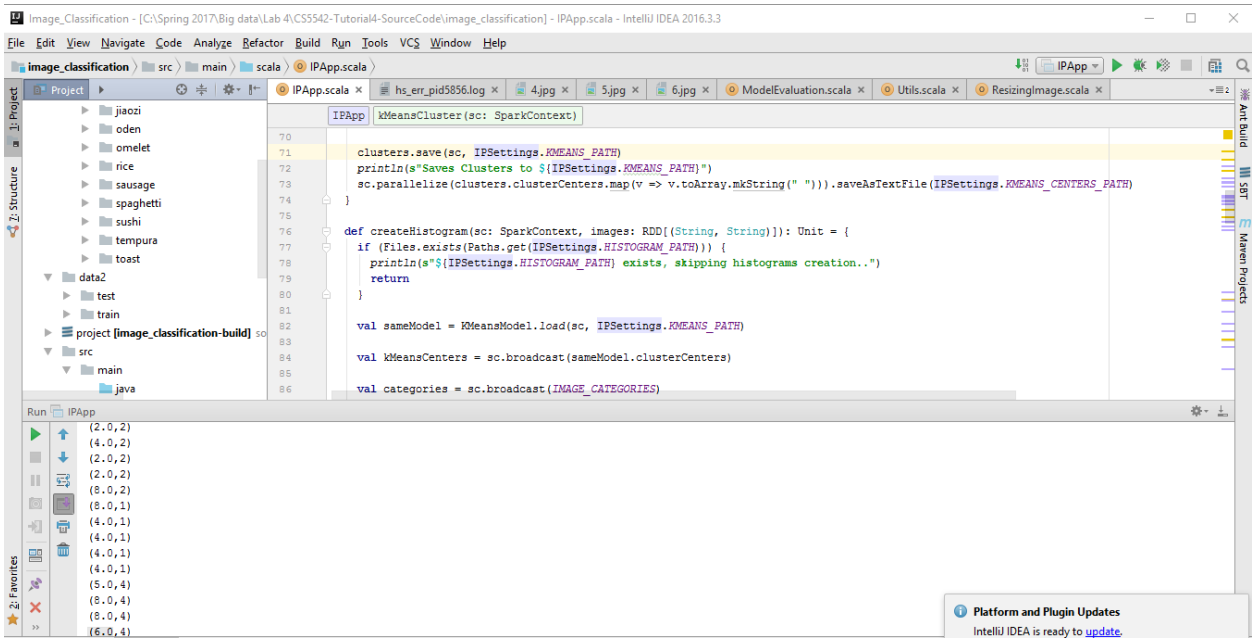

Picture Tools
lion
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Quick access
Desktop
Downloads
Documents
Pictures
8001
8002
8003
8004

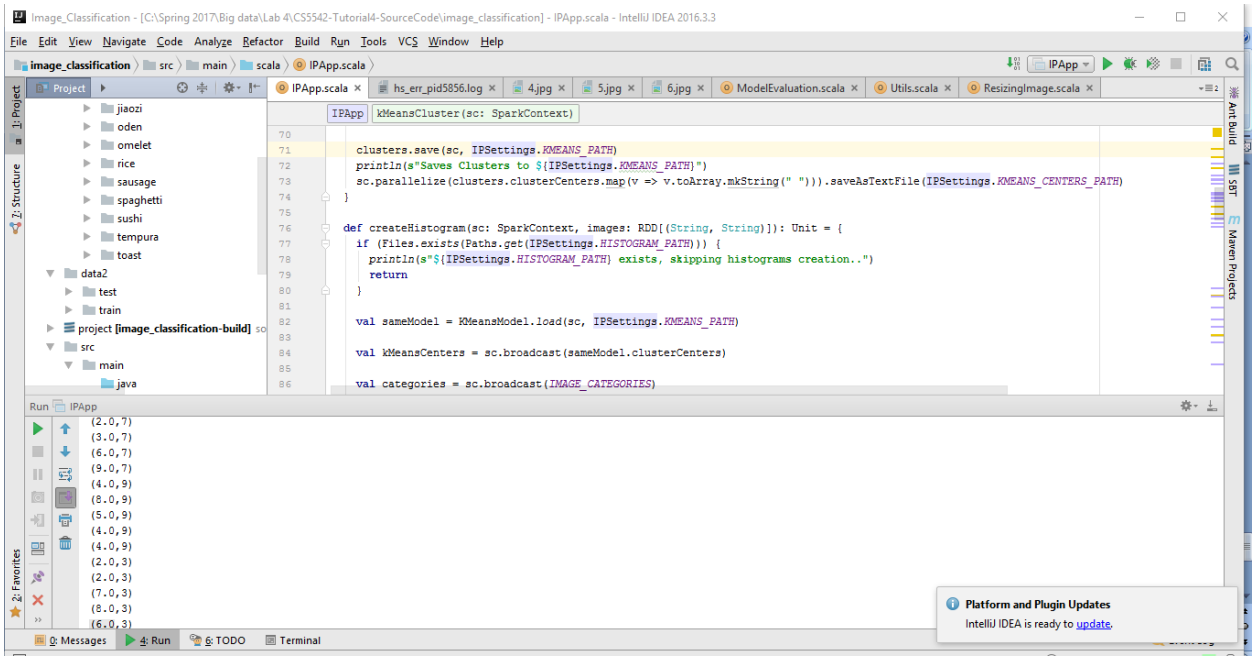

Picture Tools
tiger
File
Home
Share
View
Manage
This PC
Documents
Course Works
Big Data Analytics and Applications
Quick access

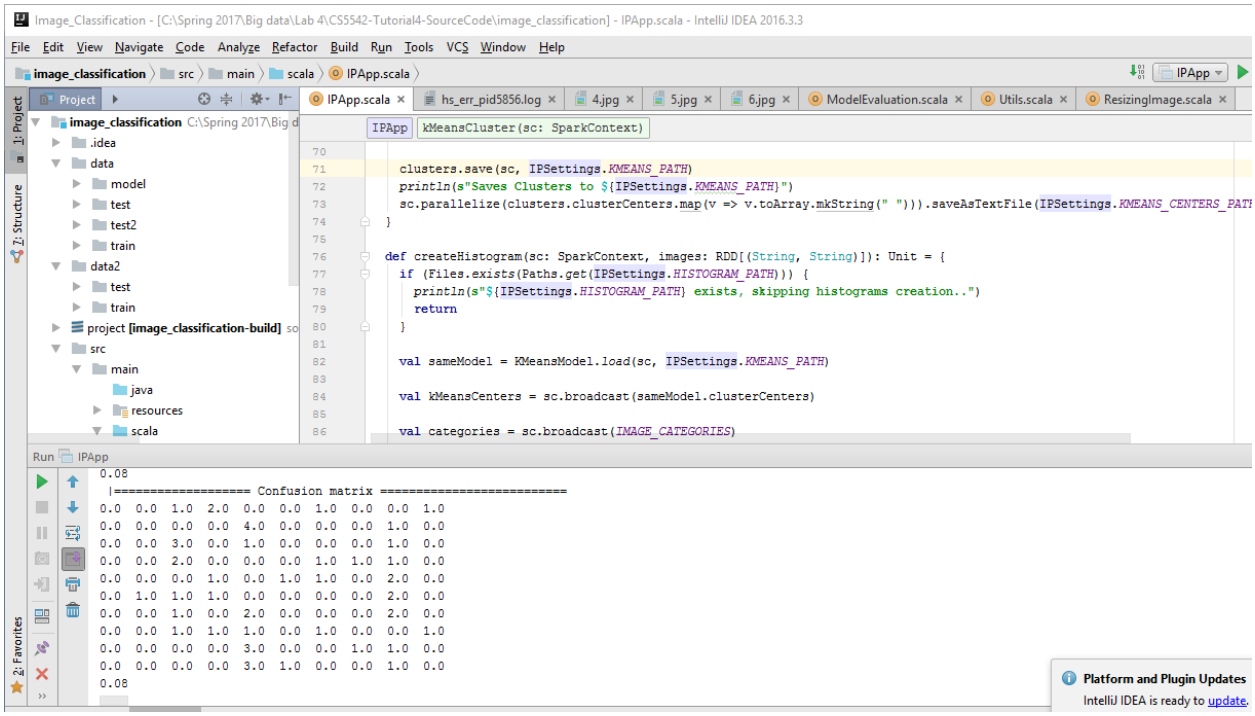
- For each and every image in test data, we will predict the image category based on the trained model.After running the test data on the trained model, following results were recorded.

Results

- Following confusion matrix and an accuracy of 70% was recorded from this task.






Image_Classification - [C:\Spring 2017\Big data\Lab 4\CS5542-Tutorial4-SourceCode\image_classification] - IPApp.scala - IntelliJ IDEA 2016.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
image_classification src main scala IPApp.scala
IPApp
Project
1: Project
7: Structure
2: Favorites
image_classification C:\Spring 2017\Big d
.idea
data
model
test
test2
train
data2
test
train
project [image_classification-build]
src
main
java
resources
scala
IPApp.scala
hs_err_pid5856.log
4.jpg
5.jpg
6.jpg
ModelEvaluation.scala
Utils.scala
ResizingImage.scala
IPApp
kMeansCluster(sc: SparkContext)
clusters.save(sc, IPSettings.KMEANS_PATH)
println(s"Saves Clusters to ${IPSettings.KMEANS_PATH}")
sc.parallelize(clusters.clusterCenters.map(v => v.toArray.mkString(" "))).saveAsTextFile(IPSettings.KMEANS_CENTERS_PATH
}
def createHistogram(sc: SparkContext, images: RDD[(String, String)]): Unit = {
if (Files.exists(Paths.get(IPSettings.HISTOGRAM_PATH))) {
println(s"${IPSettings.HISTOGRAM_PATH} exists, skipping histograms creation..")
return
}
val sameModel = KMeansModel.load(sc, IPSettings.KMEANS_PATH)
val kMeansCenters = sc.broadcast(sameModel.clusterCenters)
val categories = sc.broadcast(IMAGE_CATEGORIES)
Run IPApp
0.08
|=================== Confusion matrix ==========================
0.0 0.0 1.0 2.0 0.0 0.0 1.0 0.0 0.0 1.0
0.0 0.0 0.0 0.0 4.0 0.0 0.0 0.0 1.0 0.0
0.0 0.0 3.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0
0.0 0.0 2.0 0.0 0.0 0.0 1.0 1.0 1.0 0.0
0.0 0.0 0.0 1.0 0.0 1.0 1.0 0.0 2.0 0.0
0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 2.0 0.0
0.0 0.0 1.0 0.0 2.0 0.0 0.0 0.0 2.0 0.0
0.0 0.0 1.0 1.0 1.0 0.0 1.0 0.0 0.0 1.0
0.0 0.0 0.0 0.0 3.0 0.0 0.0 1.0 1.0 0.0
0.0 0.0 0.0 0.0 3.0 1.0 0.0 0.0 1.0 0.0
0.08
Platform and Plugin Updates
IntelliJ IDEA is ready to update.