

## Trump vs Hillary Clinton

The screenshot shows a Scala IDE interface with the following details:

- Title Bar:** workspace - LabSession1/src/main/scala/KMeansCluster.scala - Scala IDE
- File Menu:** File Edit Refactor Navigate Search Project Scala Run Window Help
- Toolbars:** Standard Java-like toolbars.
- Left Sidebar:** Package Explorer showing LabSession1 [De] and several Scala files: MLDataAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, and KMeansCluster.scala.
- Right Sidebar:** Outline view showing the structure of the KMeansCluster class.
- Bottom Panel:** Problems, Tasks, and Console tabs. The Console tab displays the output of the Scala application, which includes several WARN messages from BLAS and the final WSSSE value.

```

// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)
val kmeansModel = kmeans.fit(trainingData.cache())

val WSSSE = kmeansModel.computeCost(preparedData)
println(s"Within Set Sum of Squared Errors = $WSSSE")

kmeansModel.clusterCenters.foreach(println)

// Predict using the Test data
val kmeans_predictions = kmeansModel.transform(testData.cache())

```

```

<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (23 jan. 2019 23:41:13)
19\01\23 23:41:22  WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
19\01\23 23:41:24  WARN KMeans: The input data is not directly cached, which may hurt performance if its par
19\01\23 23:41:25  WARN KMeans: The input data was not directly cached, which may hurt performance if its par
Within Set Sum of Squared Errors = 0.021878868044038845
[0, 0.032986093324143435, -0.01236333530396223, -0.0031798667332623154, 0.0059727143961936235, 0.007790157214427986
[-0, 0.033925923846351635, -0.00668344071463627, 0.03682011705549324, 0.03600281044183408, -0.01744078671066506]
[-0, 0.00987763635136845, -0.008532787990671668, 0.05999176455514337, -0.00204857066578268]
[1, 0.48891895258037012E-4, -0.019211961401593633, 0.018017733238804786, 0.0013588984422639415, 0.83942320487635115]
[0, 0.030022306530736387, 0.0446722388467363395, 0.04131989484770508, 0.0012618876149645879, 0.026873297581914812]
[-0, 0.038386322059003365, 0.029889439864616306, -0.0036683802736644234, -0.009172540291079452, -0.017535421773988416
[0, 0.01851461119358243, -0.00713809529121435, 0.014970990750281029, -0.0175369172034916, 0.006173634139987576]
[-0, 0.04282763115285585, 0.0049481459893573865, 0.006508041643025726, -0.00636156608959523384, 2, 284593205743779E-4]
[-0, 0.020424492001025515, -0.013546348909254779, 0.014190261623717843, -0.0035708102566952057, -0.005052604315502
[-0, 0.005355714044089906, -0.03468180768704814, 0.0212638389513207, -0.006452379856879513, -0.0162225296869992286]
[0, 0.012190261623717843, -0.01597945136018974, 0.005378796078736375, 0.050281785608147, 0.052476147189736366]
[0, 0.01493617448348925, -0.03900246245149114, 0.05396284276503138, 0.029853448364883663, 0.018748549721203746]
[0, 0.052665059055600844, -0.03468156096226136, 0.01640065017023256, 0.04785842741174357, 0.016686638418052877]
[0, 0.03694196553745617, -0.0055232358009864885, -0.00833664460418125, 0.0251829245239223, -0.0020679884495808813] v

```

Figure 1 : KMeansCluster console results Hillary vs Trump

The screenshot shows a Scala IDE interface with the following details:

- Title Bar:** workspace - LabSession1/src/main/scala/KMeansCluster.scala - Scala IDE
- File Menu:** File Edit Refactor Navigate Search Project Scala Run Window Help
- Toolbars:** Standard Java-like toolbars.
- Left Sidebar:** Package Explorer showing LabSession1 [De] and several Scala files: MLDataAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, and KMeansCluster.scala.
- Right Sidebar:** Outline view showing the structure of the KMeansCluster class.
- Bottom Panel:** Problems, Tasks, and Console tabs. The Console tab displays the output of the Scala application, which includes the WSSSE value and a detailed prediction count table.

```

// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)
val kmeansModel = kmeans.fit(trainingData.cache())

val WSSSE = kmeansModel.computeCost(preparedData)
println(s"Within Set Sum of Squared Errors = $WSSSE")

kmeansModel.clusterCenters.foreach(println)

// Predict using the Test data
val kmeans_predictions = kmeansModel.transform(testData.cache())

```

```

<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (23 jan. 2019 23:41:13)
[Stage 18:=====] (117 + 4) / 200
[Stage 18:=====] (186 + 4) / 200
+-----+-----+
|prediction|count|
+-----+-----+
|      8|    2|
|      3|    2|
|      4|    2|
|     13|    1|
|     16|    1|
|      6|    1|
|      0|    1|
|     17|    1|
|     11|    1|
|     12|    1|
|      9|    1|
+-----+-----+

```

Figure 2 : KMeansCluster console results Hillary vs Trump – predict count algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

File Edit Refactor Navigate Search Project Scala Run Window Help

Package... MLDataAnalysis.scala TwitterAnalysis.scala CollectingTweets.scala FeatureExtractor.scala KMeansCluster.scala

```
// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)
val kmeansModel = kmeans.fit(trainingData.cache())

val WSSSE = kmeansModel.computeCost(preparedData)
println(s"Within Set Sum of Squared Errors = $WSSSE")

kmeansModel.clusterCenters.foreach(println)

// Predict using the Test data
val kmeans_predictions = kmeansModel.transform(testData.cache())
```

Problems Tasks Console

KMeansCluster\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:41:13)

country	followers	friends	hashtags	lang	likes	location	name	retweets
Other	0	0	[]	en	0	Other	VoteTrump	0
Other	0	0	[]	es	0	Other	camila pacheco	0
Other	0	9	[]	en	0	Other	eric westendarp	0
Other	41	20	[]	tr	0	late 90's	weaboo trash (ipek)	0
Other	283	263	[]	en	0	Other	owen greene	0
Other	302	458	[]	en	0	Montréal, Canada	Anouk Charles	0
Other	479	691	[]	en	0	Other	Ladybird	0
Other	570	673	[Obama]	en	0	Kansas City, MO	Lore Meltzer	0
Other	683	687	[Trump2016]	en	0	Colorado	MAGS	0
Other	824	379	[]	en	0	Other	MONIQUE SOURDIF	0
Other	1010	898	[Democrat, Hillar...	en	0	For any warrior t...	Task Force Freedom	0
Other	1308	706	[]	en	0	Everywhere	Ann Foster	0
Other	2136	784	[]	en	0	New York, NY	Adam Gurri	0
Other	16866	14395	[]	en	0	Middle of Boswash...	Bill Nigh	0

790M of 990M

Figure 3 : KMeansCluster console results Hillary vs Trump – WordToVec algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

File Edit Refactor Navigate Search Project Scala Run Window Help

Package... MLDataAnalysis.scala TwitterAnalysis.scala CollectingTweets.scala FeatureExtractor.scala KMeansCluster.scala

```
// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)
val kmeansModel = kmeans.fit(trainingData.cache())

val WSSSE = kmeansModel.computeCost(preparedData)
println(s"Within Set Sum of Squared Errors = $WSSSE")

kmeansModel.clusterCenters.foreach(println)

// Predict using the Test data
val kmeans_predictions = kmeansModel.transform(testData.cache())
```

Problems Tasks Console

KMeansCluster\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:41:13)

name	retweets	source	text	time_zone
VoteTrump	0	<a href="http://t...">@realDonaldTrump	Other	[@realdonaldtrum
camila pacheco	0	<a href="http://w... No tiene sentido ...	Other	[no, tiene, sent
eric westendarp	0	<a href="http://t... @HillaryClinton s...	Other	[@hillaryclinton
weaboo trash (ipek)	0	<a href="http://t... bilemiyorum, tr...	Istanbul	[bilemiyorum, tr...
owen greene	0	<a href="http://t... @WORLDSTARHIPH...	Other	[@worldstarh...
Anouk Charles	0	<a href="http://t... RT @lyssaneel: Mo...	Quito	[rt, @lyssaneel:
Ladybird	0	<a href="http://t... RT @Women4Trump: ...	Atlantic Time (Ca...	[rt, @Women4Trum
Lore Meltzer	0	<a href="http://t... @SlimTim925 @Trum...	Central Time (US ...	[@SlimTim925, @T...
MAGS	0	<a href="http://t... RT @Rockprincess8...	Mountain Time (US ...	[rt, @Rockprinc...
MONIQUE SOURDIF	0	<a href="http://t... RT @Thyefan: If @...	Atlantic Time (Ca...	[rt, @Thyefan: If @...
Task Force Freedom	0	<a href="http://t... Not unlike an ere...	Other	[not, unlike, ar...
Ann Foster	0	<a href="http://t... RT @thejeffoneal...	Saskatchewan	[rt, @thejeffone...
Adam Gurri	0	<a href="http://t... RT @pxdelaney It def...	Eastern Time (US ...	[@pxdelaney, it, ...
Bill Nigh	0	<a href="http://t... RT @borzou: Would...	Eastern Time (US ...	[rt, @borzou:, w...

799M of 990M

Figure 4 : KMeansCluster console results Hillary vs Trump WordToVec algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

```
// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)
val kmeansModel = kmeans.fit(trainingData.cache())

val WSSSE = kmeansModel.computeCost(preparedData)
println(s"Within Set Sum of Squared Errors = $WSSSE")

kmeansModel.clusterCenters.foreach(println)

// Predict using the Test data
val kmeans predictions = kmeansModel.transform(testData.cache())

```

Problems Tasks Console <terminated> KMeansCluster\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:41:13)

text	time_zone	words	filtered_words	features	prediction
ldTrump ...	Other	[@realdonaldtrump...]	[@realdonaldtrump...]	[0.03497288050138...	6
sentido ...	Other	[no, tiene, sentido, ...]	[tiene, sentido, ...]	[0.00738883782178...	3
linton s...	Other	[@hillaryclinton...]	[@hillaryclinton...]	[0.8175398240809...	17
un trump...	Istanbul	[@bilemiyorum, tru...	[@bilemiyorum, tru...	[0.01460648151114...	3
STARSHIPH...	Other	[rt, @wondstarhi...	[rt, @wondstarhi...	[0.0143728731139...	8
neel: Mo...	Quito	[rt, @lyssaneel..., ...]	[rt, @lyssaneel..., ...]	[0.00939659801382...	8
4Trump: ...	Atlantic Time (Ca...	[rt, @womend4trump...	[rt, @womend4trump...	[0.03413327461918...	13
25 @Trum...	Central Time (US...[rt, @slimtim925, @tr...	[@slimtim925, @tr...	[@slimtim925, @tr...	[0.01621496283914...	4
princess8...	Mountain Time (US...[rt, @rockprinces...	[rt, @rockprinces...	[rt, @rockprinces...	[0.04485551462857...	12
an: If @...	Atlantic Time (Ca...	[rt, @theyfan: i...	[rt, @theyfan: i...	[0.03348714043386...	9
@ an ere...	Other	[not, unlike, an...]	[unlike, erection...]	[0.01828751298792...	0
ffoneal...	Saskatchewan	[rt, @thejeffonea...	[rt, @thejeffonea...	[0.02433476052246...	4
iy It def...	Eastern Time (US ...[rt, @pxdelaney, it, ...]	[@pxdelaney, defi...	[@pxdelaney, defi...	[0.00731938433918...	16
ou: Would...[Eastern Time (US ...[rt, @borzou:, wo...	[rt, @borzou:, tr...	[rt, @borzou:, tr...	[rt, @borzou:, tr...	[0.02185672625306...	11

838M of 990M

Figure 5 : KMeansCluster console results Hillary vs Trump – WordToVec algorithm

workspace - LabSession1/src/main-scala/FeatureExtractor.scala - Scala IDE

```
FeatureExtractor extracts features from twitter tweets
based on meta data provided as parameters. It creates top 20
values for each feature and a separate features file with
tweets matching the filter criteria provided for each column
JSON output is generated in features folder in the same path

It accepts the following parameters -
Param 1 - Path to the tweets.json file
Param 2 - Meta data is in the following format
    column:alias:null:filter where
    column is column from json - entities.hashtags.text or user.location
    alias is name given in generated features json file - hashtags or location
    null is column name to be used for null check - entities.hashtag(0).text
```

Problems Tasks Console <terminated> FeatureExtractor\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:51:10)

```
[Stage 0:> (0 + 4) / 4]
[Stage 0:===== (3 + 1) / 4]

19/01/23 23:51:19 WARN Utils: Truncated the string representation of a plan since it was too large. This behavior
processing place.country 3
processing lang 3
processing user.location 3
processing user.time_zone 3
processing text 4
processing Hillary|hillary|Trump|trump
processing user.name 3
processing source 3
processing entities.hashtags.text 3
processing favorite_count 3
processing retweet_count 3
processing user.friends_count 3
processing user.followers_count 3
processing text like "%Hillary%" OR text like "%hillary%" OR text like "%Trump%" OR text like "%trump%" \c
```

655M of 965M

Figure 6 : FeatureExtractor console results Hillary vs Trump

The screenshot shows the Eclipse IDE interface with the FeatureExtractor.scala file open. The code implements a FeatureExtractor class that processes tweets from a JSON file, extracting top 20 features based on meta data parameters. The execution console output shows the processing of various tweet fields like lang, user.location, user.time\_zone, text, and hashtags, along with entity counts. The results indicate two stages of processing, each involving four operations.

```

/*
FeatureExtractor extracts features from twitter tweets
based on meta data provided as parameters. It creates top 20
values for each feature and a separate features file with
tweets matching the filter criteria provided for each column
JSON output is generated in features folder in the same path

It accepts the following parameters -
Param 1 - Path to the tweets.json file
Param 2 - Meta data is in the following format
    column:alias:null:filter where
    column is column from json - entities.hashtags.text or user.location
    alias is name given in generated features json file - hashtags or location
    null is column name to be used for null check - entities.hashtag(0).text

```

```

<terminated> FeatureExtractor$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (23 jan. 2019 23:51:10)
processing lang 3
processing user.location 3
processing user.time_zone 3
processing text 4
processing Hillary|hillary|Trump|trump
processing user.name 3
processing source 3
processing entities.hashtags.text 3
processing favorite_count 3
processing retweet_count 3
processing user.friends_count 3
processing user.followers_count 3
(text like "%Hillary%" OR text like "%hillary%" OR text like "%Trump%" OR text like "%trump%" )

[Stage 1:> (0 + 4) / 4]
[Stage 1:=====

```

Figure 7 : FeatureExtractor console results Hillary vs Trump

## iPhone vs Android

The screenshot shows the Eclipse IDE interface with the FeatureExtractor.scala file open. The code is identical to Figure 7, but the execution results show processing for iPhone and Android devices instead of Hillary and Trump. The console output shows the processing of various tweet fields and entity counts, resulting in two stages of processing, each involving four operations.

```

/*
FeatureExtractor extracts features from twitter tweets
based on meta data provided as parameters. It creates top 20
values for each feature and a separate features file with
tweets matching the filter criteria provided for each column
JSON output is generated in features folder in the same path

It accepts the following parameters -
Param 1 - Path to the tweets.json file
Param 2 - Meta data is in the following format
    column:alias:null:filter where
    column is column from json - entities.hashtags.text or user.location

```

```

<terminated> FeatureExtractor$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (23 jan. 2019 23:53:09)
[Stage 0:> (0 + 4) / 4]

19/01/23 23:53:19 WARN Utils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting spark.debug.maxToStringSize in spark.conf.

processing place.country 3
processing lang 3
processing user.location 3
processing user.time_zone 3
processing text 4
processing iphone|iphone|Android|android
processing user.name 3
processing source 3
processing entities.hashtags.text 3
processing favorite_count 3
processing retweet_count 3
processing user.friends_count 3
processing user.followers_count 3
(text like "%iphone%" OR text like "%iPhone%" OR text like "%Android%" OR text like "%android%" )

[Stage 1:> (0 + 4) / 4]

```

Figure 8 : FeatureExtractor console results iPhone vs android

```

workspace - LabSession1/src/main/scala/KMeansCluster.scala - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package...  MLDatAnalysis.scala  TwitterAnalysis.scala  CollectingTweets.scala  FeatureExtractor.scala  KMeansCluster.scala
  // Remove frequently appearing words that have no meaning
  val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
  val stopWordsRemoved = remover.transform(tokenized)

  // Convert this to a vector form
  val word2Vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(0)
  val model = word2Vec.fit(stopWordsRemoved)
  val preparedData = model.transform(stopWordsRemoved)

Problems Tasks Console
<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (23 jan. 2019 23:54:08)
[19/01/23 23:54:18 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
[19/01/23 23:54:19 WARN KMeans: The input data is not directly cached, which may hurt performance if its parent RDDs are also J
[19/01/23 23:54:20 WARN KMeans: The input data was not directly cached, which may hurt performance if its parent RDDs are also J
Within Set Sum of Squared Errors = 0.08695761374068287
[0.04927530619833204, -0.0878137500823975, 0.020944449630772905, -0.049683113846307]
[-0.02558763110501221, 0.006073955883374979, -0.01517510462398257, -0.00936566949499376, 0.02049095127595104]
[0.019786404446513653, 0.0017831235486230734, -0.016315062688171074, 0.0011930618291864027, 0.006790101098326536]
[0.00243855358529696885, 0.03483075639664536, 0.05328904386617586, 0.0526679911793998, 0.01997281095156303]
[0.00400083293852897, -0.017340087219254, 0.010091515292495407, 0.04684450945581545, -8.618308571525478E-1]
[-0.004461569545213536, 0.001516765190452935, -0.01715427990769597, -0.01642266462382395, -0.015613047680526508]
[0.02446370338322595, -0.07117208502313588, 0.0943028290741697, 0.051101872862976364, -0.023082952136173847]
[0.016891390516388195, -0.04914133254186289, -0.0013861796734007917, -0.056648452150531936, -0.08249574814317905]
[0.017956994619453326, -0.032353946089599584, -0.002888637303848369, -0.02641282678814605, 0.013701688614673913]
[0.00521858316343412, -0.034846554092719435, 0.02874839599985122, -0.039022486108908446]
[0.010281541793231378, 0.05516788597921268, 0.04545735527777619, 0.025223750153376735, 0.04964611240613617]
[-0.03380667501139388, 0.03521113727959649, 0.026465342871545415, 0.037094513834537275, 0.04697388018913833]
[0.013562435538254002, 0.01672411887731869, 0.06112000617113981, 0.0299329779790358072, 0.05143102901903086]
[0.047475092516926956, -0.02440763582103141, 0.02159355108855434, 0.006630708878903658, 0.001520743802524147]
[-0.00823071522807533, 0.004186407858092676, 0.04358211506173988, 0.007548611550211906, 0.04828603210097009]
[1.9738493800768874E-4, 0.013700115723928149, 0.01518485574927755, -0.00267964842844855, -0.004788757791328675]
[-0.018076930959864289, 0.03467959758709185, 0.00587289002865461, 0.011247959861866192, 0.011967883214917189]
[0.029178646889825667, -0.09706957708265185, 0.03679156452029323, 0.034594338441578854, -0.017591980658471583]
[-9.798915679788305E-4, 0.0195495738785891, 0.007046817314057123, 0.0746028027053745, 0.04309091478630545]

```

Figure 9 : KMeansCluster console results iPhone vs android

```

workspace - LabSession1/src/main/scala/KMeansCluster.scala - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package...  MLDatAnalysis.scala  TwitterAnalysis.scala  CollectingTweets.scala  FeatureExtractor.scala  KMeansCluster.scala
  // Remove frequently appearing words that have no meaning
  val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
  val stopWordsRemoved = remover.transform(tokenized)

  // Convert this to a vector form
  val word2Vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(0)
  val model = word2Vec.fit(stopWordsRemoved)
  val preparedData = model.transform(stopWordsRemoved)

Problems Tasks Console
<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (23 jan. 2019 23:54:08)
[Stage 20:=====
[Stage 20:=====
|prediction|count|
+-----+-----+
| 15 | 4 |
| 1 | 4 |
| 7 | 3 |
| 5 | 3 |
| 2 | 2 |
| 16 | 2 |
| 10 | 2 |
| 13 | 1 |
| 17 | 1 |
| 8 | 1 |
| 12 | 1 |
| 11 | 1 |
| 18 | 1 |
| 0 | 1 |
| 14 | 1 |
+-----+-----+
|prediction|count|
+-----+-----+
| 15 | 4 |
| 1 | 4 |
| 7 | 3 |
| 5 | 3 |
| 2 | 2 |
| 16 | 2 |
| 10 | 2 |
| 13 | 1 |
| 17 | 1 |
| 8 | 1 |
| 12 | 1 |
| 11 | 1 |
| 18 | 1 |
| 0 | 1 |
| 14 | 1 |
+-----+-----+

```

Figure 10 : KMeansCluster console results iPhone vs android – Predict count algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

```
// Remove frequently appearing words that have no meaning
val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
val stopWordsRemoved = remover.transform(tokenized)

// Convert this to a vector form
val word2vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(0)
val model = word2vec.fit(stopWordsRemoved)
val preparedData = model.transform(stopWordsRemoved)
```

terminated> KMeansCluster\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:54:08)

country	followers	friends	hashtags	lang	likes	location	name	retweets	sou
Other	0	0	[MTVSTARS]	fr	0	Other	sidener loe	0	<a href="http://t
Other	0	0	[android, gameins...	en	0	Other	deserie	0	<a href="https://
Other	0	5	[android, android...]	it	0	Other	patria	0	<a href="http://k
Other	2	4	[android, android...]	fr	0	Other	Karoline Günter	0	
Other	2	23	[iPhone, iPhoneGa...]	en	0	Other	decretion	0	
Other	4	17	[]	ja	0	Other	Nada	0	<a href="https://
Other	6	67	[]	in	0	神奈川県東部]	え	0	<a href="http:
Other	12	26	[]	en	0	Other	makgad	0	<a href="http://t
Other	14	78	[]	ja	0	Other	J&J Farm	0	<a href="http://t
Other	20	99	[]	en	0	JKs	MICO	0	<a href="http://t
Other	20	1264	[]	ru	0	Other	Septex	0	<a href="http://t
Other	23	63	[iphone6senametro...]	fr	0	Иркутск, Иркутска...	アリナ Романова	0	<a href="https://
Other	46	55	[androidgames, ga...]	en	0	Other	sam	0	<a href="http://t
Other	126	257	[]	en	0	tech game reviews	Shirleybryantideas	0	<a href="http://i
Other	236	85	[]	en	0	Other	Fox	0	<a href="http://t
Other	592	603	[]	ja	0	Toronto, Ontario	MarketingForJustice	0	<a href="http://i
Other	775	1	[iphone, retweet, ...]	ja	0	血飛沫がしゃくなげな天井『直射日光厳禁』もさでブッショキアホ場囃子】	0	<a href="http://h	
Other	798	404	[latergram, iphon...]	en	0	Other	Vinny Scans	0	<a href="http://m
Other	839	587	[]	ja	0	Zürich, Switzerland	Philippe Wiget	0	<a href="http://i
Other	839	587	[]	ja	0	有楽町ヒル舞浜ホルダ正時代古】	裕子ウラ大戦19周年!!】	0	<a href="

633M of 977M

Figure 11 : KMeansCluster console results iPhone vs Android – WordToVec algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

```
// Remove frequently appearing words that have no meaning
val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
val stopWordsRemoved = remover.transform(tokenized)

// Convert this to a vector form
val word2vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(0)
val model = word2vec.fit(stopWordsRemoved)
val preparedData = model.transform(stopWordsRemoved)
```

terminated> KMeansCluster\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:54:08)

source	text	time_zone	words	filtered_words	features
[<a href="http://t...]	[RT @DokPhone: En ...]	[Other]	[rt, @dokphone, ...]	[@.0273901718028...	
[<a href="https://...]	i completely fell...	[Other]	[i, completely, f...]	[@.01356243553825...	
[<a href="http://w...]	[I completamente "Pa...	[Other]	[ho, completato, ...]	[@.0185646452527...	
[<a href="http://...]	Ich habe 1.050 Na...	[Other]	[ich, habe, 1,050...]	[@.02296363301575...	
[<a href="https://...]	I'ai récupéré 93,...	[Other]	[i'ai, récupér...]	[@.04982904931530...	
[<a href="https://...]	I have bought 'B...	[Other]	[i, have, bought, ...]	[@.01591209948269...	
[<a href="http://t...]	[@anco_01887 仕事が...]	[Tokyo]	[@anco_01887, 仕事が...]	[@.0097874...	
[<a href="http://b...]	[RT @isayahbusya:...	[Other]	[rt, @isayahbusya...	[@.0046056468829...	
[<a href="http://t...]	... No more passport...	[Other]	[no, more, passpo...]	[@.01782512357458...	
[<a href="http://t...]	[RT @luzabs: カラカワ...]	[Other]	[rt, @luzabs:, カ...]	[@.02696783669913...	
[<a href="http://t...]	[#aaronlawd would ... Pacific Time (US ...]	[...@aaronlawd, woul...]	[@.0028715332570...		
[<a href="https://...]	[RT @games4mob: Cx...]	[Other]	[rt, @games4mob, ...]	[@.0015371879562...	
[<a href="http://t...]	[#iphone6senametro...]	[Other]	[#iphone6senametr...]	[@.0063178502023...	
[<a href="http://t...]	Finally got the g...	[Other]	[finally, got, th...]	[@.02038149342227...	
[<a href="http://t...]	[@SamsungMobileUS ...]	[Other]	[@SamsungMobileUS...	[@.00229707924...	
[<a href="http://t...]	[There are over 1...]	[Other]	[there, are, ove...]	[@.0061667292616...	
[&plus;]	[@a href="http://t...]	[RT @Nadjastaff: i...]	[@.021]		
[<a href="http://m...]	[A 2010 Toyota Cam...]	[Eastern Time (US ...]	[a, 2010, toyota, ...]	[@.0025678009260...	
[<a href="http://t...]	[Fall is gone by n...]	[Greenland]	[fall, is, gone, -]	[@.010190998894...	
[!]	[@ a href="http://t...]	Hawaii	[@! androidからキャス配...]	[@! androidからキャス配...]	

653M of 977M

Figure 12 : KMeansCluster console results iPhone vs android – WordToVec algorithm

The screenshot shows the Eclipse IDE interface with the following details:

- Project Bar:** workspace - LabSession1/src/main/scala/KMeansCluster.scala - Scala IDE
- File Menu:** File Edit Refactor Navigate Search Project Scala Run Window Help
- Quick Access:** Package... MLDatAnalysis.scala TwitterAnalysis.scala CollectingTweets.scala FeatureExtractor.scala KMeansCluster.scala
- Outline View:** Shows the structure of the KMeansCluster class, including imports, main method, and various fields and methods.
- Code Editor:** Displays the Scala code for K-Means clustering, including removing stop words, tokenizing, and fitting a Word2Vec model to the filtered words.
- Problems View:** Shows no errors or warnings.
- Tasks View:** Shows no tasks.
- Console View:** Displays the command-line output of the Scala application running in Java, showing the execution of `KMeansCluster$` and the resulting data frame.
- Data View:** Shows the content of the data frame, including columns: text, time\_zone, words, filtered\_words, features, and prediction.

The data frame content is as follows:

text	time_zone	words	filtered_words	features	prediction
ie: En ...	Other	[rt, @dokphone; ...]	[rt, @dokphone; ...]	[-0.0273901718028...]	11
ly fell...	Other	[i, completely, f...	[completely, fell...	[0.01356243553825...	12
ito 'Pa...	Other	[ho, completato, ...	[ho, completato, ...]	[-0.18185646452525...	5
,050 Na...	Other	[ich, habe, 1, 050...	[ich, habe, 1, 050...	[0.02296363301575...	17
iré 93,...	Other	[j'ai, récupéré, ...]	[j'ai, récupéré, ...]	[0.04982904931530...	0
ght 'B...	Other	[i, have, bought, ...]	[bought, 'brigian...	[0.01591209948269...	13
1887 仕事がい...]	Tokyo	[@anco_o1887, 仕事が...]	[@anco_o1887, 仕事が...]	[0.00978743315984...]	8
usyabusa...	Other	[rt, @isyahbusaya; ...]	[rt, @isyahbusaya; ...]	[-0.00460564648829...	18
isport...	Other	[no, more, passspo...	[passport, page, ...]	[0.0782512357458...	2
: カラカワ...	Other	[rt, @luzabs; ...]	[rt, @luzabs; ...]	[0.02696733669913...]	10
would ... Pacific Time (US ...	@aronlawd, woul...	[@aronlawd, mind, ...]	[@aronlawd, mind, ...]	[-0.0028715332576...	15
mob: Cx...	Other	[rt, @games4mob; ...]	[rt, @games4mob; ...]	[-0.0015371879562...	15
iametro...	Other	[#iphonēsemantri...	[#iphonēsemantri...	[-0.0063178562023...	14
: the g...	Other	[finally, got, ga...	[finally, got, ga...	[0.02038149342227...	15
fileUS ...	Other	[@samsungmobileus...	[@samsungmobileus...	[-0.0072297767924...	16
over 1...	Other	["there, are, ove...	["there, are, ove...	[0.0061667292611...	1
. [RT @Nadjasta...	i...]	Other	[rt, @nadjastaff; ...]	[0.0218229934187...]	1
sta Cam ... Eastern Time (US ...	[a, 2010, toyota; ...]	[@2010, toyota, ca...	[@2010, toyota, ca...	[0.0025678009260...	5
be n by ...	Greenland	[fall, is, gone, _...	[fall, gone, _...	[-0.0121909988894...	1
モ ! Androidからキャス配信 ...	Hawaii	[モ ! androidからキャス配...]	[モ ! androidからキャス配...]	[0.05511468773086...	15

Figure 13 : KMMeansCluster console results iphone vs android – WordToVec algorithm

## Trump/Hillary vs iPhone/Android

The screenshot shows the Scala IDE interface with the following details:

- File Menu:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Toolbar:** Standard Java-like icons for file operations, search, and run.
- Quick Access:** Shows recent files like MLDataAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, and KMeansCluster.scala.
- Outline View:** Shows the class structure of KMeansCluster, including imports, methods, and fields.
- Code Editor:** The KMeansCluster.scala file is open, showing Scala code for data processing and machine learning. It includes imports for sparkConf, sc, sqlContext, and various utility classes. The code performs word tokenization, removes stop words, converts words to vectors, splits the data, and trains a k-means model.
- Console:** Shows the Scala application's output. It prints the results of a division operation:  $(\theta + 4) / 4$  and  $(2 + 2) / 4$ . Below this, there is a warning message about truncating a string representation of a plan due to size.

Figure 14 : FeatureExtractor console results Hillary vs Trump & iphone vs android

```

// Remove frequently appearing words that have no meaning
val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
val stopWordsRemoved = remover.transform(tokenized)

// Convert this to a vector form
val word2Vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(0)
val model = word2Vec.fit(stopWordsRemoved)
val preparedData = model.transform(stopWordsRemoved)

// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)

```

terminated> FeatureExtractor\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_191\bin\javaw.exe (23 jan. 2019 23:58:36)

processing user.location 3  
processing user.time\_zone 3  
processing text 4  
processing iphone|iPhone|Android|android  
processing user.name 3  
processing source 3  
processing entities.hashtags.text 4  
processing Hillary|hillary|Trump|trump  
processing favorite\_count 3  
processing retweet\_count 3  
processing user.friends\_count 3  
processing user.followers\_count 3  
(text like "%iPhone%" OR text like "%Android%" OR text like "%android%") AND (array\_contains(entities.hashtags.text, "hillary") OR array\_contains(entities.hashtags.text, "Trump") OR array\_contains(entities.hashtags.text, "trump"))  
[Stage 1:> (0 + 0) / 4]  
[Stage 1:> (0 + 4) / 4]

Figure 15 : FeatureExtractor console results Hillary vs Trump & iphone vs android

```

// Remove frequently appearing words that have no meaning
val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
val stopWordsRemoved = remover.transform(tokenized)

// Convert this to a vector form
val word2Vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(0)
val model = word2Vec.fit(stopWordsRemoved)
val preparedData = model.transform(stopWordsRemoved)

// Split Training and Testing data in the ratio 70:30
val Array(trainingData, testData) = preparedData.randomSplit(Array(0.7, 0.3), seed = 1234L)
// Train k-means model.
val kmeans = new KMeans().setK(20).setSeed(1L)

entities.hashtags.text, "hillary") OR array_contains(entities.hashtags.text, "Trump") OR array_contains(entities.hashtags.text, "trump"))

```

Figure 16 : FeatureExtractor console results Hillary vs Trump & iphone vs android

The screenshot shows a Scala IDE interface with the following details:

- Title Bar:** workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE
- Left Sidebar:** Package... (LabSession1), MLDataAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, KMeansCluster.scala.
- Code Editor:** The KMeansCluster.scala file contains Scala code for performing k-means clustering on word vectors. It includes imports for sparkConf, sc, sqlContext, Word2Vec, StopWordsRemover, and Tokenizer. The code removes stop words, converts tokens to vectors, splits the data, trains a k-means model, and calculates the WSSSE.
- Console Output:**

```

<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (24 Jan. 2019 00:03:04)
19/01/24 00:03:14 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
19/01/24 00:03:14 WARN KMeans: The input data is not directly cached, which may hurt performance if its parent RDDs are also uncached.
19/01/24 00:03:16 WARN KMeans: The input data was not directly cached, which may hurt performance if its parent RDDs are also uncached.

Within Set Sum of Squared Errors = 0.15847182882952934
[-0.017975139069474403, -0.061041501048259995, 0.0324699439936214, -0.03717928857632555, -0.01837244184894694]
[-0.03370059979484276, -0.05860554572829963, 0.0324698424889976, 0.1131775962813711]
[0.002499055612743652, 0.02754486549382886, 0.05088537822393653, -4.8473137232276286E-4, -0.05790054673149479]
[0.022511393088613593, 0.008425731118899429, -0.009456355413743029, 0.013635176744542074, -0.0278978893968982]
[-0.04238497517847767, -0.118398898372837342, 0.0278213802018128338, 0.00455847024938203]
[0.009251300956982336, 0.00858445426468078, 0.012305666606676393, -0.004808063126018964, -0.0037807993794566983]
[0.007693494631799668, -0.028123615116237306, 0.010256712836342267, 0.010236754588214627, 0.02829547782897135]
[-0.021655241293289388, -0.03858149817067602, 0.023696969195787624, -0.05363578511721859, 0.008007157113252564]
[-0.01270068519247266, 0.02134136902168393, 3.3622485800431325E-4, -0.054306764513827294, -0.01754545348768051]
[0.0065270335710790415, -0.0208953154250021304, -4.7376441712942096E-4, -0.0010259677624927768, -0.00946715025824637]
[0.05592789091247504, 0.050696158132451034, 0.04382613729139715, -0.0025694784186371077, -0.024747545446636932]

```
- Right Sidebar:** Quick Access, Outline (showing imports like import declarat, KMeansCluster, main(args: Ar, sparkConf, sc, sqlContext, import dec, output, data, tokenizer, remover, stopWords, word2Vec, model, preparedD, trainingDa, testData, kmeans, kmeansMc, WSSSE, kmeans\_pr).

Figure 17 : KMeansCluster console results Hillary vs Trump & iphone vs android

The screenshot shows a Scala IDE interface with the following details:

- Title Bar:** workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE
- Left Sidebar:** Package... (LabSession1), MLDataAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, KMeansCluster.scala.
- Code Editor:** The KMeansCluster.scala file contains Scala code for performing k-means clustering on word vectors. It includes imports for sparkConf, sc, sqlContext, Word2Vec, StopWordsRemover, and Tokenizer. The code removes stop words, converts tokens to vectors, splits the data, trains a k-means model, and calculates the WSSSE. Additionally, it includes a prediction count algorithm.
- Console Output:**

```

<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (24 Jan. 2019 00:03:04)
[Stage 24:*****]
[Stage 24:*****] (102 + 6) / 200
[Stage 24:*****] (100 + 4) / 200
+-----+-----+
|prediction|count|
+-----+-----+
|      2|     6|
|      16|     6|
|      5|     5|
|     10|     3|
|      6|     3|
|     17|     3|
|     15|     3|
|     19|     2|
|      9|     2|
|      1|     2|

```
- Right Sidebar:** Quick Access, Outline (showing imports like import declarat, KMeansCluster, main(args: Ar, sparkConf, sc, sqlContext, import dec, output, data, tokenizer, remover, stopWords, word2Vec, model, preparedD, trainingDa, testData, kmeans, kmeansMc, WSSSE, kmeans\_pr).

Figure 18 : KMeansCluster console results Hillary vs Trump & iphone vs android – Predict count algorithm

The screenshot shows a Scala IDE interface with the following details:

- File Explorer:** Shows files like MLDatAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, and KMeansCluster.scala.
- Code Editor:** Displays the KMeansCluster.scala code. The code performs several steps: removing stop words, converting words to vectors, splitting the data into training and testing sets, training a K-Means model, and calculating the WSSSE (Within-Sum-of-Squares Error).
- Console:** Shows the execution results. The output includes a matrix of numbers (16x6) and the calculated WSSSE value: 649M of 854M.
- Outline View:** Shows the class structure and member variables and methods.

Figure 19 : KMeansCluster console results Hillary vs Trump & iphone vs android – Predict count algorithm

The screenshot shows a Scala IDE interface with the following details:

- File Explorer:** Shows files like MLDatAnalysis.scala, TwitterAnalysis.scala, CollectingTweets.scala, FeatureExtractor.scala, and KMeansCluster.scala.
- Code Editor:** Displays the KMeansCluster.scala code, identical to Figure 19 but using the word2vec model.
- Console:** Shows the execution results. The output includes a large table of tweet features and their associated cluster assignments and URLs.
- Outline View:** Shows the class structure and member variables and methods.

country	follower	friends	hashtags	lang	likes	location	name	retweets	source
Other	0	0	[]	en	0	Other	VoteTrump	0	<a href="http://t...@realbo...
Other	0	0	[]	es	0	Other	camila pacheco	0	<a href="http://w...No tien...
Other	0	0	[]	fr	0	Other	sidener loe	0	<a href="http://t...RT @Dokl...
Other	0	3	[android, android...]	de	0	Other	sim one	0	Ich hab...
Other	1	9	[android, android...]	en	0	Other	JoJo-K	0	I've co...
Other	1	28	[android, android...]	ru	0	Other	mihail buzinov	0	Ypoxak...
Other	3	4	[Android, android...]	en	0	Other	Valsan Elena	0	I have ...
Other	4	13	[]	fr	0	Brasilieber!!!	SKOTNICKI SELIN	0	<a href="http://t...RT @Dokl...
Other	6	67	[]	in	0	Other	makgad	0	<a href="http://b...RT @ais...
Other	7	18	[]	en	0	Brooklyn, NY	Galewyn Massey	0	<a href="http://t...@Real10%
Other	12	26	[]	en	0	Other	382 Farm	0	<a href="http://t...No more...
Other	13	231	[]	en	0	Other	Kerri Pinegar	0	<a href="http://t...RT @gam...
Other	14	78	[]	ja	0	JKo	MICO	0	<a href="http://t...RT @luzi...
Other	20	1264	[]	ru	0	Иркутск, Иркутска...	Алина Романова	0	<a href="https://...RT @gam...
Other	36	330	[]	es	0	Other	Mauricio Carmona	0	<a href="http://w...Un pequi...
Other	41	28	[]	tr	0	late 90's	weaboo trash (ipek)	0	<a href="http://t...bilemiyu...
Other	70	90	[]	en	0	Other	RegulationDigital	0	<a href="http://i...Hillary...
Other	80	251	[]	en	0	West Virginia	Bejour	0	<a href="http://t...RT @rea...

Figure 20 : KMeansCluster console results Hillary vs Trump & iphone vs android – WordToVec algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

```
// Remove frequently appearing words that have no meaning
val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
val stopWordsRemoved = remover.transform(tokenized)

// Convert this to a vector form
val word2Vec = new Word2Vec().setInputCol("filtered_words").setOutputCol("features").setVectorSize(5).setMinCount(1)
val model = word2Vec.fit(stopWordsRemoved)
val preparedData = model.transform(stopWordsRemoved)

// Split Training and Testing data in the ratio 70:30
```

Problems Tasks Console

```
<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (24 jan. 2019 00:03:04)
```

	source	text	time_zone	words	filtered_words	features	prediction
a href="http://t...	@realDonaldTrump ...			Other [@realdonaldtrump ...	[@realdonaldtrump ...	[-0.0097515524498...]	9
a href="http://w...	No tiene sentido ...			Other [no, tiene, sentido,  ...	[no, tiene, sentido,  ...	[0.01587041504681...	16
a href="http://t...	RT @DokPhone: En ...			Other [rt, @dokphone,  ...	[rt, @dokphone,  ...	[0.01661467115627...	2
	Ich habe 15,905 G...			Other [ich, habe, 15,90... ...	[ich, habe, 15,90... ...	[-0.0417262377217...	12
	I've collected 75...			Other [i've, collected,... ...	[i've, collected,... ...	[-0.07107618054522...	17
	Ypoxakai собран - 1...			Other [ypoxakai, собран, ... ...	[ypoxakai, собран, ... ...	[-0.0722111819422...	17
	I have completed ...			Other [i, have, complet... ...	[i, have, complet... ...	[-0.0211961798919...	6
a href="http://t...	RT @DokPhone: En ...			Other [rt, @dokphone,  ...	[rt, @dokphone,  ...	[0.01661467115627...	2
a href="http://b...	RT @aisyahbusya:...			Other [rt, @aisyahbusya ...	[rt, @aisyahbusya ...	[0.00959786009448...	2
a href="http://t...	@RealGMHHillary A...	Pacific Time (US ...)		Other [@realm4hillary ...	[@realm4hillary ...	[0.01142085572064...	5
a href="http://t...	No more passport ...			Other [no, more, passpo... ...	[no, more, passpo... ...	[0.00966683486476...	5
a href="http://t...	RT @egamingacademy...			Other [rt, @egamingacade... ...	[rt, @egamingacade... ...	[0.00725762922173...	16
a href="http://t...	RT @luzabs: カラオケ...			Other [rt, @luzabs:, カ... [rt, @luzabs:, カ... [0.05901318478087...	[0.05901318478087...	18	
a href="http://t...	RT @games4mob: Cx...			Other [rt, @games4mob:,  ...	[rt, @games4mob:,  ...	[0.03955241696288...	3
a href="http://w...	Un pequeño esfuer...			Other [un, pequeño, esf... ...	[un, pequeño, esf... ...	[0.03228493008848...	16
a href="http://t...	bilemiyorum trump...	Istanbul		Other [bilemiyorum, tru... ...	[bilemiyorum, tru... ...	[0.02071435544639...	9
a href="http://i...	Hillary Clinton J...			Other [hillary, clinton ...	[hillary, clinton ...	[0.01159335594857...	16
a href="http://t...	RT @realdonaldTru...			Other [rt, @realdonaldt... [rt, @realdonaldt... [0.03647670143150...	[0.03647670143150...	10	
	Ypoxakai собран - 1...			Other [ypoxakai, собран, ... ...	[ypoxakai, собран, ... ...	[-0.0722033732454...	17
a href="http://i...	iPhone 6 (4.7") 6...			Other [iphone, 6, (4.7"... ...	[iphone, 6, (4.7"... ...	[0.00252322555304...	2

Figure 21 : KMeansCluster console results Hillary vs Trump & iphone vs android – WordToVec algorithm

workspace - LabSession1/src/main-scala/KMeansCluster.scala - Scala IDE

```
// Remove frequently appearing words that have no meaning
val remover = new StopWordsRemover().setInputCol("words").setOutputCol("filtered_words")
val stopWordsRemoved = remover.transform(tokenized)
```

Problems Tasks Console

```
<terminated> KMeansCluster$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (24 jan. 2019 00:03:04)
```

	source	text	time_zone	words	filtered_words	features	prediction
a href="http://t...	@realDonaldTrump ...			Other [@realdonaldtrump ...	[@realdonaldtrump ...	[-0.0097515524498...]	9
a href="http://w...	No tiene sentido ...			Other [no, tiene, sentido,  ...	[no, tiene, sentido,  ...	[0.01587041504681...	16
a href="http://t...	RT @DokPhone: En ...			Other [rt, @dokphone,  ...	[rt, @dokphone,  ...	[0.01661467115627...	2
	Ich habe 15,905 G...			Other [ich, habe, 15,90... ...	[ich, habe, 15,90... ...	[-0.0417262377217...	12
	I've collected 75...			Other [i've, collected,... ...	[i've, collected,... ...	[-0.07107618054522...	17
	Ypoxakai собран - 1...			Other [ypoxakai, собран, ... ...	[ypoxakai, собран, ... ...	[-0.0722111819422...	17
	I have completed ...			Other [i, have, complet... ...	[i, have, complet... ...	[-0.0211961798919...	6
a href="http://t...	RT @DokPhone: En ...			Other [rt, @dokphone,  ...	[rt, @dokphone,  ...	[0.01661467115627...	2
a href="http://b...	RT @aisyahbusya:...			Other [rt, @aisyahbusya ...	[rt, @aisyahbusya ...	[0.00959786009448...	2
a href="http://t...	@RealGMHHillary A...	Pacific Time (US ...)		Other [@realm4hillary ...	[@realm4hillary ...	[0.01142085572064...	5
a href="http://t...	No more passport ...			Other [no, more, passpo... ...	[no, more, passpo... ...	[0.00966683486476...	5
a href="http://t...	RT @egamingacademy...			Other [rt, @egamingacade... ...	[rt, @egamingacade... ...	[0.00725762922173...	16
a href="http://t...	RT @luzabs: カラオケ...			Other [rt, @luzabs:, カ... [rt, @luzabs:, カ... [0.05901318478087...	[0.05901318478087...	18	
a href="http://t...	RT @games4mob: Cx...			Other [rt, @games4mob:,  ...	[rt, @games4mob:,  ...	[0.03955241696288...	3
a href="http://w...	Un pequeño esfuer...			Other [un, pequeño, esf... ...	[un, pequeño, esf... ...	[0.03228493008848...	16
a href="http://t...	bilemiyorum trump...	Istanbul		Other [bilemiyorum, tru... ...	[bilemiyorum, tru... ...	[0.02071435544639...	9
a href="http://i...	Hillary Clinton J...			Other [hillary, clinton ...	[hillary, clinton ...	[0.01159335594857...	16
a href="http://t...	RT @realdonaldTru...			Other [rt, @realdonaldt... [rt, @realdonaldt... [0.03647670143150...	[0.03647670143150...	10	
	Ypoxakai собран - 1...			Other [ypoxakai, собран, ... ...	[ypoxakai, собран, ... ...	[-0.0722033732454...	17
a href="http://i...	iPhone 6 (4.7") 6...			Other [iphone, 6, (4.7"... ...	[iphone, 6, (4.7"... ...	[0.00252322555304...	2

Figure 22 : KMeansCluster console results Hillary vs Trump & iphone vs android – WordToVec algorithm