BBM469

Data Intensive Applications Laboratory

Report of Assignment-3

Topic: Clustering and Classification with Spark
Group Name: Budgeregiar- Group 6

Name of Student-1: Yusuf Efe Kalaycı

ID of Student-1: 21827517
Name of Student-2: Mert Tazeoğlu

ID of Student-2: 21946606

Instructors:

- Fuat Akal
- Tuğba Erdoğan

Purpose

In this assignment we implement clustering and classification methods with using pyspark. Our aim is geting familiar with spark environment.

The dataset has 16 feature columns with continuous values and the target label is cathegorical. There are 7 categories for target label. So we need to implement with MultiClass pyspark environments.There are 13611 rows in this data.

Data Understanding

The dataset has 16 feature columns with continuous values and the target label is cathegorical. There are 7 categories for target label. So we need to implement with MultiClass pyspark environments.There are 13611 rows in this data

To work with classification models we needed to convert Class column to integer. We shuffled the data for cross validation. Since dataset doesn't have any NA values we didn't handle that situation. We normalized a copy of organized data to compare the results.

Data Preparation

To work with classification models we needed to convert Class column to integer. We shuffled the data for cross validation. Since dataset doesn't have any NA values we didn't handle that situation. We normalized a copy of organized data to compare the results.

We chose Kmeans as a clustering model since the features are continuous values. We thought KMeans will fit well.

For the organized data clustering rand score is 0.8038 and for the normalized data rand score is 0.9082. These values are quite good and it's expected to see clustering with normalized data gives better results. Because numeric values of dataset have very large scale.

Class	count
CALI	1630
SEKER	2027
SIRA	2636
HOROZ	1928
BOMBAY	522
BARBUNYA	1322
DERMASON	3546

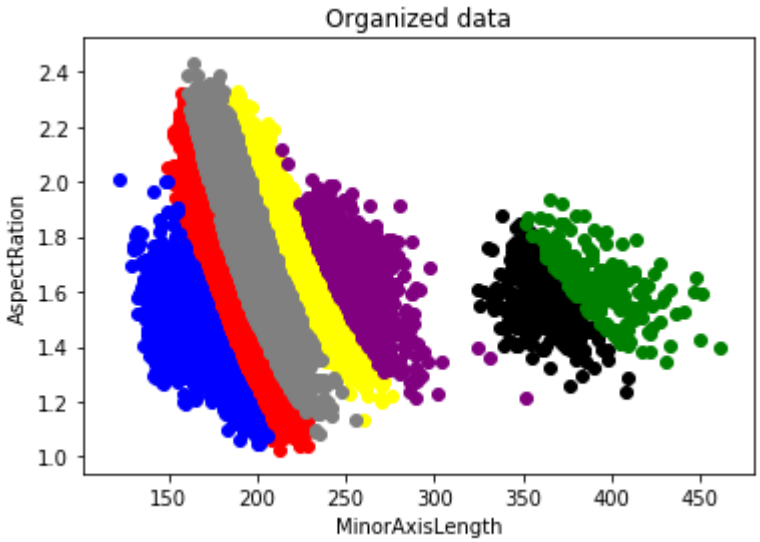
ClassInt	count
0	2027
6	3546
5	2636
1	1322
3	1630
2	522
4	1928

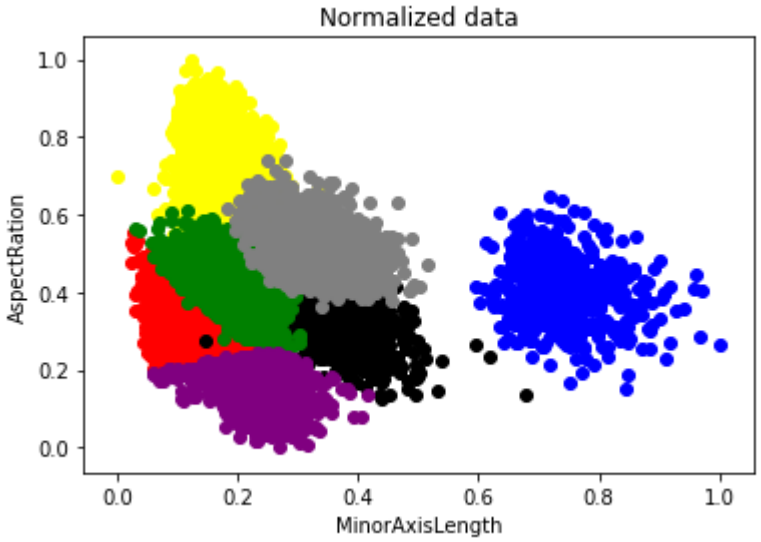
This is the tranformation we implement for the Class label.

Clustering

For the organized data clustering rand score is 0.8038 and for the normalized data rand score is 0.9082. These values are quite good and it's expected to see clustering with normalized data gives better results. Because numeric values of dataset have very large scale.

These are the scatter plots for Columns AspectRation and MinorAxisLength. We chose them while trying columns. This difference looks good to us.




Normalized data
AspectRation
MinorAxisLength
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Classification

We chose Random Forest Classifier for classification model with maxDepth=10. We used k-fold cross validation and made table for different k values.

Our table for Organized dataset

	Accuracy	Recall	Precision	F1	Confusion Matrix
k=3	0.9487	0.9487	0.9492	0.9488	[[1956 3 0 0 0 43 25] [4 1221 0 70 5 22 0] [0 1 521 0 0 0 0] [1 28 0 1573 19 9 0] [0 4 0 23 1847 42 12] [10 5 0 3 19 2410 189] [20 0 0 0 1 139 3386]]
k=4	0.9487	0.9487	0.9492	0.9488	[[1956 3 0 0 0 43 25] [4 1221 0 70 5 22 0] [0 1 521 0 0 0 0] [1 28 0 1573 19 9 0] [0 4 0 23 1847 42 12] [10 5 0 3 19 2410 189] [20 0 0 0 1 139 3386]]
k=5	0.9487	0.9487	0.9492	0.9488	[[1956 3 0 0 0 43 25] [4 1221 0 70 5 22 0] [0 1 521 0 0 0 0] [1 28 0 1573 19 9 0] [0 4 0 23 1847 42 12] [10 5 0 3 19 2410 189] [20 0 0 0 1 139 3386]]
k=7	0.9487	0.9487	0.9492	0.9488	[[1956 3 0 0 0 43 25] [4 1221 0 70 5 22 0] [0 1 521 0 0 0 0] [1 28 0 1573 19 9 0] [0 4 0 23 1847 42 12] [10 5 0 3 19 2410 189] [20 0 0 0 1 139 3386]]

Our table for Normalized dataset. These tables shown us all metrics with different k values are same. This is expected because Random Forest Classification is a tree-like algorithm.

	Accuracy	Recall	Precision	F1	Confusion Matrix
=3	0.9475	0.9475	0.9480	0.9476	[[1948 3 0 1 0 48 27] [4 1217 0 68 4 29 0] [0 1 521 0 0 0 0] [1 35 0 1563 20 11 0] [0 4 0 20 1847 46 11] [9 2 0 2 17 2414 192] [21 0 0 0 1 137 3387]]
=4	0.9475	0.9475	0.9480	0.9476	[[1948 3 0 1 0 48 27] [4 1217 0 68 4 29 0] [0 1 521 0 0 0 0] [1 35 0 1563 20 11 0] [0 4 0 20 1847 46 11] [9 2 0 2 17 2414 192] [21 0 0 0 1 137 3387]]
=5	0.9475	0.9475	0.9480	0.9476	[[1948 3 0 1 0 48 27] [4 1217 0 68 4 29 0] [0 1 521 0 0 0 0] [1 35 0 1563 20 11 0] [0 4 0 20 1847 46 11] [9 2 0 2 17 2414 192] [21 0 0 0 1 137 3387]]
=7	0.9475	0.9475	0.9480	0.9476	[[1948 3 0 1 0 48 27] [4 1217 0 68 4 29 0] [0 1 521 0 0 0 0] [1 35 0 1563 20 11 0] [0 4 0 20 1847 46 11] [9 2 0 2 17 2414 192] [21 0 0 0 1 137 3387]]

These tables shown us all metrics with different k values are same. This is expected because Random Forest Classification is a tree-like algorithm.

Normalized and organized datasets metrics are too close to each other. This is also expected because of the Random Forest Classification structure. We have accuracy around 0.947 which is quite good.

References

https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

https://spark.apache.org/docs/latest/api/java/index.html?org/apache/spark/ml/tuning/CrossValidatorModel.html

https://spark.apache.org/docs/latest/ml-tuning.html

https://piazza.com/class_profile/get_resource/kz2hotg9swd1gf/l1v4fv8mmyt3y5

https://www.youtube.com/watch?v=9SfO9KhjklK