K-Means Clustering and Latent Dirichlet Allocation (LDA) using Databricks Spark ML and Microsoft Azure ML Studio: An Analysis of Stack 0verflow for R Language Question and Answers

Gregory D. Mamoyac, Tejas Agara Chandrakumar, Nitesh Kamboj, Julia Stachurska
Department of Information Systems, California State University
Los Angeles
e-mail : gmamoya@calstatela.edu, afnu3@calstatela.edu, nkamboj@calstatela.edu, jstachu@calstatela.edu

Abstract: This project explores the analysis of K-Means Clustering and Latent Dirichlet Allocation (LDA) on Microsoft Azure ML and Databricks Community Spark ML.

Initially different parameters were used to optimize the different technology platforms because both platforms differ in library and module availability. However, after finding initial results, the project utilizes similar parameters in order to find better comparison. The project final results of K-Means Clustering and Latent Dirichlet Allocation (LDA).

The dataset is found on Stack Overflow. The project clusters tags/topics from words found in the Question and Answer Body text of the .csv files. The cluster groups help use predict "if an answer submission is acceptable to the question" of R Language.

1. Introduction

Our idea is to train, test, and evaluate ML models to predict "if an answer submission is acceptable to the question". This analysis will be done using K-Means Clustering and Latent Dirichlet Allocation (LDA) on Microsoft Azure ML and Databricks Community Spark ML.

The analysis surrounds the Q&A website called Stack Overflow. Stack Overflow is a question and answer site for professional and enthusiast programmers. It's built and run by you as part of the Stack Exchange network of Q&A sites.

The data consists of R Language questions from Stack Overflow and R Language answers from Stack Overflow. The data of the question, answer, and tags is found on https://www.kaggle.com/stackoverflow/rquestions and the datasize is 2 GB.

2. Experimental Specifications

The experimental specifications available through the two platforms are as follows. Parameters settings will be discussed later in the paper.

2.1 Databricks Community

For Databricks Community, the specifications are as follows:

- Execution: Single Node
- Max Storage 6 GB Memory
- Databricks Runtime Version 4.0
- Includes Apache Spark 2.3.0, Scala 2.11

2.2 Microsoft Azure ML Studio

For Microsoft Azure ML Studio, the specifications are as follows:

- Execution: Single Node
- Max storage space: 10 GB Memory
- Compute Resource Type: The Machine Learning service is a multitenant service

3. K-Means Clustering

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. Data points are clustered based on feature similarity. The number of centroid is standardized at three(3).

Note that according to Microsoft Azure ML, Cross Validation module cannot be used for the nature of clustering.

> The **Cross-Validate Model** module takes as input a labeled dataset, together with an untrained classification or regression model. It divides the dataset into some number of subsets (*folds*), builds a model on each fold, and then returns a set of accuracy statistics for each fold. [1].

Unlike Regression, clustering takes the robustness of the data to congregate similar values. Therefore, Cross-Validation could not be used because the analysis is done on the entire set and there are no folds.

3.1 Databricks Community

On Databricks Community, the model consisted of two centroids that were used in to build the model by setting knum=3. This means that three cluster centers are created based the table features. The StringIndexer was used to index the features in order to create integer values from the Body column's terms/words. The column IsAcceptedAnswer was a significant feature in conjunction with Body. Finally, ClusteringEvaluator was imported and called to evaluate the three clusters.

3.1.1 Cluster Centers

Cluster Centers:			
1.16456934e+06	1.15970409e+07	6.06907086e+00	3.91494089e-01
3.83991607e+06	4.04048463e+07	1.34801940e+00	4.68644121e-01
2.39544180e+06	2.70155044e+07	2.32995207e+00	4.57974784e-01

Table 1. Cluster Centers. The ClusterEvaluator model created the evaluations of the three cluster centers.

3.1.2 Clustering Results Based for K-Means Clustering on Databricks

Cluster1

Tag	OwnerUserId	CreationDate	ParentId	Score	IsAcceptedAnswer_Indexed	Body
dataframe	687713	6/8/11	41831156	8	0.0	<p>From the the h...
dataframe	324364	6/8/11	41831156	4	0.0	"<p>Perhaps it's ...
quantmod	5493656	10/31/15	33455050	1	1.0	<p>If you don't u...
labels	5513586	11/1/15	33467171	0	0.0	<p>So, the only t...
rounding	5513586	11/1/15	33467171	0	0.0	<p>So, the only t...

only showing top 5 rows

Cluster2

Tag	OwnerUserId	CreationDate	ParentId	Score	IsAcceptedAnswer_Indexed	Body
bioinformatics	2854638	10/7/13	19223440	1	0.0	<p>You can use th...
data-visualization	2840385	10/7/13	19228659	1	0.0	"<p>I would use t...
plot	2840286	10/7/13	19229870	2	1.0	<p>The problem is...
sorting	2840286	10/7/13	19229870	2	1.0	<p>The problem is...
reshape	2809684	10/8/13	19236881	0	0.0	<p>Loading is eas...

Figure 1. Clustering Results Based for K-Means Clustering on Databricks. These are the results for the "Tags" created through clustering for Clusters 1 and 2. The "Tags" are the terms created from the Body column text. This table shows each of the clusters of the similar points from each of the three centroids.

3.1.3 Prediction Results Based for K-Means Clustering on Databricks

features	prediction
[3259.0,79709.0,-...	0
[3259.0,79709.0,-...	0
[3259.0,79709.0,-...	0
[3259.0,79709.0,-...	0
[6043.0,79709.0,9...	0
[6043.0,79709.0,9...	0
[6043.0,79709.0,9...	0
[6043.0,79709.0,9...	0
[8002.0,79709.0,0...	0
[8002.0,79709.0,0...	0
[8002.0,79709.0,0...	0
[8002.0,79709.0,0...	0
[14257.0,79709.0,...	0
[14257.0,79709.0,...	0
[14257.0,79709.0,...	0
[14257.0,79709.0,...	0
[14928.0,79709.0,...	0
[14928.0,79709.0,...	0

Figure 2. Prediction Results Based for K-Means Clustering on Databricks. The prediction model clustered non-unique features like Month, Day, Year, Tag, OwnerUserID, and Score. The following results uses these key features to model "if the answer is acceptable" based on actual and predicted. The Squared Euclidean Distance= 0.7247.

3.2 MicroSoft Azure ML

For Microsoft Azure ML, the modules were set to include columns: Id, Tag, OwnerUserId, ParentId, Score, IsAcceptedAnswer, Body, Month, Day, and Year. The algorithm module centered on K Means Clustering with Single Parameter, Three Centroids, Euclidean Metric, Iterations: 100, and Initialization: Kmeans++Fast for a faster processing.

Three centroids were used in K-Means Clustering Module to build the model. Each of the cluster centers provided metric for distance from the center. The next graph was visualized on PCA1 and PCA 2 Axis Principal Component 1 axis is the combined set of features that captures the most variance in the model Principal Component 2 axis represents some combined set of features that is orthogonal to the first component and that adds the next most information to the chart.

3.2.1 Separation of Clusters for K-Means Clustering on Azure ML

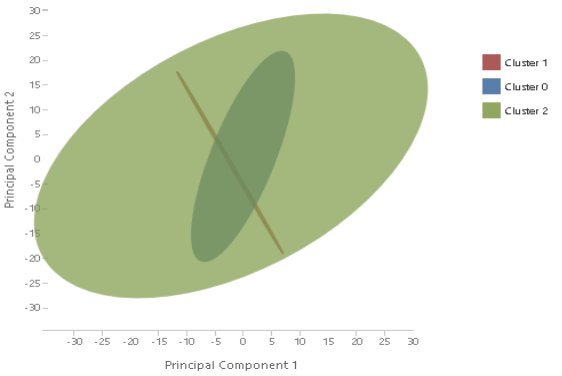

Figure 3. Separation of Clusters for K-Means Clustering on Azure ML. The figure measures the Average Distance from Cluster Center. Cluster 0 (Blue) = 2.39367. Cluster 1 (Red) = 0.97166. Cluster 2 (Green) = 1.41954. The Squared Euclidean Distance =1.028835 and the lower the value, the closer the prediction is to the actual value.

4. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is an unsupervised learning model. As a part of natural language, it is a generative statistical model that uses clustering. It allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.These observations will help us predict "if an answer submission is acceptable to the question". The number of clustered topic is standardized at five(5).

4.1 Databricks Community

The data is being broken into 5 centroids called topics (k=5). The features are clustered in conjuction with the preprocessed Body column text through Natural Language Processing analysis. In addition, StopWords are removed and cleaned from the Body column Text. Features are used like Tag: approximation and IsAcceptedAnswer: FALSE. The model is breaking the content of body columns in 5 topics.

4.1.1 Mean Percentage Error for Latent Dirichlet Allocation (LDA) on Databricks

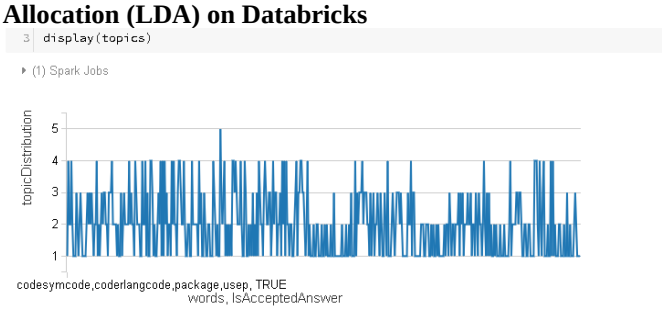

Figure 4. Mean Percentage Error for Latent Dirichlet Allocation (LDA) on Databricks. The closer the percentage error of zero the closer the predicted topic is to the actual topic value after clustering 5 topics. The figure shows topic distribution with a Mean Error Percentage= 2.0%.

4.2 MicroSoft Azure ML

Microsoft Azure ML utilizes the Latent Dirichlet Allocation (LDA) module. HTML tags were removed using Python script. In addition, missing values were processed with "Clean Missing Text" module. The Total number of topics= 5 and Split was done with 50-50.

4.2.1 Evaluation for Latent Dirichlet Allocation (LDA) on Azure ML

Figure 5. Evaluation for Latent Dirichlet Allocation (LDA) on Azure ML. Latent Dirichlet Allocation (LDA) was achieve by clustering 5 topics from the Body features through the LDA Module and Poisson Regression.

4.2.2 Python Scripting For Removing HTML Tags

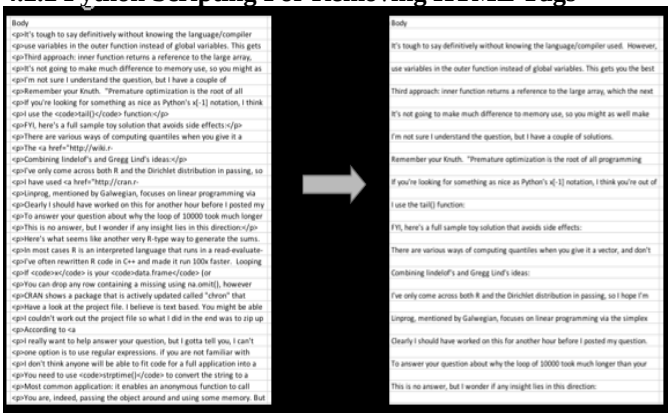

Figure 6. Python Scripting for Removing HTML Tags.
The Body was preprocessed for use in NLP functions. This allowed for the NLP to work correctly without frequency interference of HTML protocol.

4.2.3 Error Histogram for Latent Dirichlet Allocation (LDA) on Azure ML

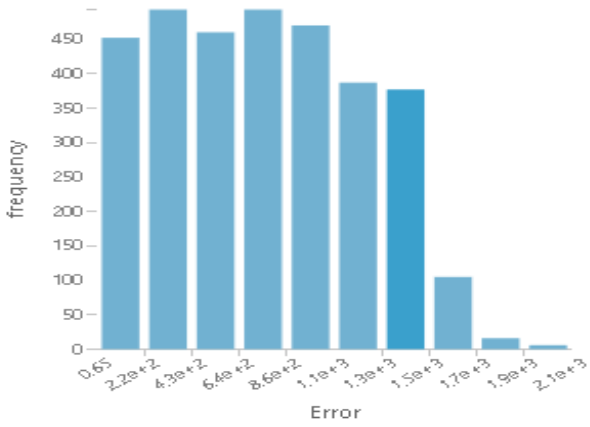

Figure 7. Error Histogram for Latent Dirichlet Allocation (LDA) on Azure ML. This error histogram plots compute the error values as the difference between target values and predicted values into 5 Topics/Centroids. The higher the frequency the closer to zero error occurs.
The Mean Error Percentage= 3.18%.

5. Conclusion

Initially, the clustering parameters across the two platforms, Databricks Community and Microsoft Azure ML Studio, were optimized according to each platformdue to the availability of packaged resources. After this error was discovered, the clustering parameters were correctly standardized with three(3) centroids for K-Means Clustering and five(5) topics for Latent Dirichlet Allocation (LDA). This way comparison between the two platforms could be more easily understood.

Databricks Community

- K-Means Clustering
  - Squared Euclidean Distance= 0.7247
  - Centroids=3
- Latent Dirichlet Allocation (LDA)
  - Mean Error Percentage=2.0%
  - Topic Clusters=5

Microsoft Azure ML

- K-Means Clustering
  - Squared Euclidean Distance =1.028835
  - Centroids=3
- Latent Dirichlet Allocation (LDA)
  - Mean Error Percentage= 3.18%
  - Topic Clusters=5

Per the points above, Databricks Community performed better in K-Means Clustering with Square Euclidean Distance= .7247 over Microsoft Azure ML's Square Euclidean Distance=1.028835. Furthermore, Latent

Dirichlet Allocation (LDA) with Databricks Community also performed better Mean Error Percentage =2.0% over Microsoft Azure ML's Mean Error Percentage =3.18%.

Considering similar parameters, Databricks Community with Spark ML technology performed better than Microsoft Azure ML for both models. For future study, it is important to note that free and student versions of the platforms were used. Fortunately, sampling allowed for faster processing while demonstrating sufficient results. For higher big data size, subscriptions or professional versions should be acquired.

References

[1] Martens, J. (n.d.). Azure Machine Learning Studio Algorithm and Module Reference. Retrieved April/May, 2018, from https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference

[2] Community,D.B. Welcome to Databricks. Documentation. Retrieved May 7, 2018, from https://docs.databricks.com/