**Twitter sentimental analysis using pySpark DataFrame**

Sentiment of a text means weather a given text is a positive or negative sentence. Twitter sentimental analysis is used to identify whether a comment is positive or negative. In this project we are using Unsupervised Machine learning methods to classify the texts.

Here to check the accuracy , We count the number of positive and negative words in the sentence and if the count of positive words in the sentence is greater than the count of negative words in the sentence then the sentence is considered as a positive sentence and same idea for classifying negative sentence.

Lets see the steps we will cover in order to build a Unsupervised classifier.

1. Loading data.
2. Data cleaning.
3. Feature Extraction
4. Passing it to KMEANS, Bisecting KMEANS.
5. Confusion matrix

**1. Loading data**

First we will load the data from an already existing csv file. For this we have 2 csv files

1. train.csv - contains sl.no , user name, comments and labels

2. test.csv – contains sl.no and comments

Load both files into PySpark RDD

**2.Data Cleaning**

In this part we will extract the necessory data from the RDD (In this case we will extract the comments)

In order to extract we will use different transformations on the RDD.

**3.Feature Extraction**

Here as we told earlier we use the count of positive and negative words as features to pass it to the model. In order to find the count we have 2 files containing positive words and negative words . We use that to find the count by mapping a list comprehension to the RDD of comments.

Here we also find the label in order to check the accuracy.

(**Both label and feature as save together as a Label Point**)

For faster deployment purpose we save the result as LIBSVM using saveAsLibSVMFile() function so we don’t have to re run this whole transformations in each iterations.

**4. Kmeans and Bisecting Kmeans**

Now we will read the LibSVM file using MLUtils.LoadLibSVM() and then we will vectorize using Vector.dense and use the vectorized rdd to perform Kmeans And Bisecting Kmeans

**5.Evaluation**

Evaluate the model using confusion matrix , RMSE or MSE

Pre-req

1. RDD
2. Feature Extraction
3. LabeledPoints
4. List Comprehension

api - tutCQh30lNDK5s2gWxRpTLt8L

api key secret – Gcshd9CC2SlLVPe519WxCsYhDC09SluF8HKdHGEfaDKmiLkvH4

bearer tokken - AAAAAAAAAAAAAAAAAAAAAGp5JgEAAAAAKxqmlg9zmSb%2FbaLyC%2Bg8zCREjJc%3D9eoPEMWP6TOhm52fgS0pxAKB0151rP9v83egbkF7Sfy0Rfrv0l