Case Study : Spam Detection Domain: Telecom

There is a telecom operator forum in which cell phone users make public claims about SMS spam messages. The dataset contains a public set of SMS labeled messages that have been collected for mobile phone spam research. The sample collection is composed by 5,574 English, real and non-encoded messages, tagged according to being legitimate (ham) or spam.

Below is the sample dataset:

- Ham: What you doing? how are you?
- Spam: Sunshine Quiz! Win a super Sony DVD recorder if you can name the capital of Australia? Text MQUIZ to 82277

Tasks:

As a big data consultant, you are provided the sample dataset to generate the word cloud using Spark MLlib You have to load this dataset in the HDFS and perform:

1. Extract words from the SMS message
2. Removed stop words.
3. Modify the stop words to include your custom words such as '-'
4. Create the features from SMS message using CountVectorizer
5. Split the data into train and test - decide on a strategy
6. Use logistic regression and check the accuracy
7. Try to use a Random Forest classifier and see if it increases the accuracy.
8. Introduce bi-gram and tri-gram and note the change in accuracy.
9. Decide on a strategy and generate a data pipeline.

Solution:

Major Steps of the Machine Learning Algorithm :

1. Tokenization is the process of taking text (such as a sentence) and breaking it into individual terms (usually words). A simple Tokenizer class provides this functionality. RegexTokenizer allows more advanced tokenization based on regular expression (regex) matching.

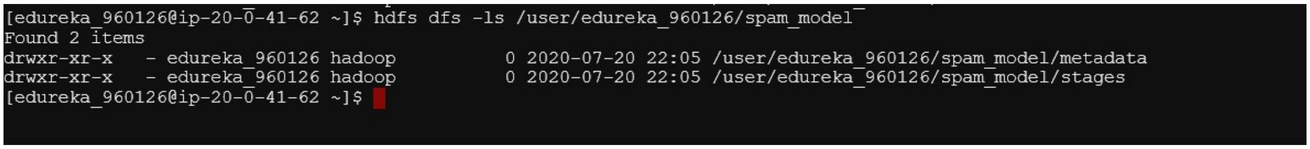
2. Removing of the Stop Words. Stop words are words which should be excluded from the input, typically because the words appear frequently and don't carry as much meaning

3. CountVectorizer and CountVectorizerModel aim to help convert a collection of text documents to vectors of token counts which are then used for training and testing of the classification models like Logistic and Random Forest regression

4. N Grams(2, 3) were being generated from the cleaned data and then vectorized using CountVectorizer and then went through same process as the tokenized words output

5. Results of accuracy with tokenized output and N grams were same(0.5)

6. Finally Model was stored in the HDFS as shown below.



```
[edureka_960126@ip-20-0-41-62 ~]$ hdfs dfs -ls /user/edureka_960126/spam_model
Found 2 items
drwxr-xr-x   - edureka_960126 hadoop          0 2020-07-20 22:05 /user/edureka_960126/spam_model/metadata
drwxr-xr-x   - edureka_960126 hadoop          0 2020-07-20 22:05 /user/edureka_960126/spam_model/stages
[edureka_960126@ip-20-0-41-62 ~]$
```

Note that the – Jupyter Notebook with all code , outputs and the comments has been provided with this submission.