# Big data Project Report
# Topic: Analysis of News Dataset

By,
PES2201800769 - Sanketh B.K.
PES2201800276 - Tejeshwar U
PES2201800282 - Sumukha

Date:9-12-2020

## Introduction

We choose news dataset as online data in terms of articles, newspapers are evolving and producing news rapidly. We choose to segregate the topics as it might be the future towards the next evolution of AI in reasoning the incident with the help of previously occurred incidents or published news which either could be Hypothetically assumed or predicted earlier.

Our main goal is to
1. Cluster similar news articles.
2. Extract important topics from a large corpus of news articles.

Our data consists of 200k+ rows and 15 columns (description of each column has been given below)

Dataset has 15 columns , details are given below

```
root
 |-- id: decimal(38,18) (nullable = tr
 |-- title: string (nullable = true)
 |-- author: string (nullable = true)
 |-- date: string (nullable = true)
 |-- content: string (nullable = true)
 |-- year: integer (nullable = true)
 |-- month: integer (nullable = true)
 |-- publication: string (nullable = t
 |-- category: string (nullable = true
 |-- digital: integer (nullable = true
 |-- section: string (nullable = true)
 |-- url: string (nullable = true)
```

## Pre-processing

We will drop all the unwanted columns and we are concatenating the title and content column.

We want to combine title and contents of title and content column also we need to maintain id's of documents for later identification

Setting up the Data preprocessing Pipeline

1. Convert the raw text into sparknlp's Document type using DocumentAssembler
2. SentenceDetector - splits the document into sentences.
3. Tokenizer - splits a document into tokens (words).
4. Normalizer - Removes all dirty characters from text following a regex pattern and transforms words based on a provided dictionary
5. StopWordsRemover - Removes all the stop words.
6. Lemmatizer - converts words to their base forms. eg: studying , studies, studied all are converted to study.

## Before Preprocessing:

```
Row(id=Decimal('1.000000000000000000'), title='Agent Cooper in Twin Peaks is the audience: once delighted, now disintegrati
ng', author='\nTasha Robinson\n', date='2017-05-31', content='      And never more so than in Showtime’s new series revival
Some spoilers ahead through episode 4 of season 3 of Twin Peaks. On May 21st, Showtime brought back David Lynch’s groundbre
aking TV series Twin Peaks, and fulfilled a prophecy in the process. In the second season finale, back in 1991, the spirit
of series-defining murder victim Laura Palmer told FBI special agent and series protagonist Dale Cooper, “I’ll see you agai
n in 25 years.” That clip plays again in the first episode of Lynch’s Twin Peaks revival, as a reminder that decades have i
n fact gone by, Laura’s promise has been carried out, and a series canceled mid-story is back on the air.A lot has changed
in 25 years. The original cast members, who are mostly back on board, have all aged heavily and visibly. Many of the charac
ters have moved on in life, getting new jobs, forming families, or taking up new obsessions. But in the opening episode, Da
le Cooper was still where the show left him in 1991: trapped in the spirit domain known as the Black Lodge, at the mercy of
incomprehensible forces that behave in erratic, alien ways. In other words, he’s just like anyone who’s actually watching T
win Peaks. As the third season began, the audience was also stuck back in 1991, waiting to see whether we were going to mov
e on from the show’s many long-gestating cliffhangers. And we were also at the mercy of the incomprehensible force that is
David Lynch, with his erratic, alien storytelling methods.All protagonists are mediators who help tell audiences how to int
erpret the narrative around them, but Dale Cooper is something else entirely: he is the audience, stumbling through Lynch’s
obscure vision, and mutating along with it. The show’s tone, budget, and format have all changed with the 2017 revival, and
Cooper, too, has changed — more so than any other character on the show. But he’s changed in ways we should recognize. The
y’re the same ways we’ve changed, as Twin Peaks has progressed from era-defining hit to weird cable art-experiment. The old
series followed Agent Cooper through the painful, awkward process of maturing as an agent and a man. And his development ha
ppened in parallel with the maturing of a TV audience that had to learn how to follow a new kind of story.All protagonists
mediate their stories, but Dale Cooper is something else entirelyToday, viewers have more sophisticated expectations than t
hey did in the 1990s. They expect long arcs, slow character development, and mysteries that may take hours of air time to e
xplore, let alone to decode. But Showtime’s new series is still leaving viewers curious, frustrated, baffled, and without t
he tools to translate what they’re seeing — which is exactly what Agent Cooper seems to be feeling right now as well. Consi
der Cooper as he started Twin Peaks back in 1990, as a fresh-faced, perky outsider walking into a mid-sized mountain commun
ity with no idea what to expect. Like the audience, he was entranced by the town of Twin Peaks — its quirky people, its une
xpected pleasures, the sheer vividness of everything around him. Yes, he was there to solve a murder, but he was endlessly
confident about his ability to tackle any case. The audience felt the same natural confidence. We knew how TV murder myster
ies went. We thought we knew exactly what to expect from that end of the story: some procedural details, some red herrings,
some drama, and eventually a solution.              Image: ABC  But like Cooper, we were seduced and distracted by Twin P
```

. . .

## During Preprocessing:

1. Urls are not specified in any row.

```
+--------------------+--------------------+----------------+----------+--------------------+----+-----+-----------+-------
-+-------+-------+----+
|                  id|               title|          author|      date|             content|year|month|publication|categor
y|digital|section| url|
+--------------------+--------------------+----------------+----------+--------------------+----+-----+-----------+-------
-+-------+-------+----+
|1.000000000000000000|Agent Cooper in T...|
Tasha Robinson
|2017-05-31|      And never m...|2017|    5|      Verge|Longform|      1|   null|null|
|2.000000000000000000|   AI, the humanity!|
Sam Byford
|2017-05-30|      AlphaGo’s v...|2017|    5|      Verge|Longform|      1|   null|null|
|3.000000000000000000|   The Viral Machine|
Kaitlyn Tiffany
|2017-05-25|      Super Delux...|2017|    5|      Verge|Longform|      1|   null|null|
|4.000000000000000000|How Anker is beat...|
Nick Statt
|2017-05-22|      Steven Yang...|2017|    5|      Verge|Longform|      1|   null|null|
|5.000000000000000000|Tour Black Panthe...|
Kwame Opam
|2017-05-15|      Ahead of Bl...|2017|    5|      Verge|Longform|      1|   null|null|
+--------------------+--------------------+----------------+----------+--------------------+----+-----+-----------+-------
-+-------+-------+----+
only showing top 5 rows
```

2. Removing all columns except id, title and content. Further dropping all rows which contain null values.

```
root
 |-- id: decimal(38,18) (nullable = true)
 |-- title: string (nullable = true)
 |-- content: string (nullable = true)
```

3. We merge title and text to create a column called text and clean text using methods like Lemmatization and add usefull words to a column called finished_lemma.

```
+-----+--------------------+--------------------+
|   id|                text|      finished_lemma|
+-----+--------------------+--------------------+
|73471|Patriots Day Is B...|[Patriots, Day, B...|
|73472|A Break in the Se...|[Break, Search, O...|
|73474|Obama's Ingenious...|[Obama's, Ingenio...|
|73475|Donald Trump Meet...|[Donald, Trump, M...|
|73476|Trump: 'I Think' ...|[Trump:, 'I, Thin...|
|73477|Seth Meyers Quest...|[Seth, Meyers, Qu...|
|73478|Obama Frames His ...|[Obama, Frames, E...|
|73479|The Trump Adminis...|[Trump, Administr...|
|73484|The Longstanding ...|[Longstanding, Cr...|
|73485|The Atlantic Dail...|[Atlantic, Daily:...|
+-----+--------------------+--------------------+
only showing top 10 rows
```

4. Considering the first hundred words in the finished_lemma drop other words and column named text.

```
+-----+--------------------+
|   id|         final_words|
+-----+--------------------+
|73471|[Patriots, Day, B...|
|73472|[Break, Search, O...|
|73474|[Obama's, Ingenio...|
|73475|[Donald, Trump, M...|
|73476|[Trump:, Think', ...|
|73477|[Seth, Meyers, Qu...|
|73478|[Obama, Frames, E...|
|73479|[Trump, Administr...|
|73484|[Longstanding, Cr...|
|73485|[Atlantic, Daily:...|
|73486|[Pledge, Pentagon...|
|73487|[Atlantic, Politi...|
|73488|[Contradictions, ...|
|73489|[Sanctions, Skept...|
|73490|[Baltimore, Polic...|
|73491|[Young, Pope, Ima...|
|73492|[DeVos, Hearings:...|
|73493|[Donald, Trump, S...|
|73494|['One-Stop, Shop'...|
|73495|[Rich, Students, ...|
+-----+--------------------+
only showing top 20 rows
```

## Making a countvectorizer

Countvectorizer is a sparse matrix compression technique which works by creating a vocabulary of words and giving numerical indexes to them. The numerical index is used to replace the actual word.

Countvectorizer is a matrix with all unique words in rows and all documents in columns with each cell representing the frequency of that word in that document. Since this is a sparse matrix , we use library functions which are optimized to store sparse matrices.

Countvectorizer first creates vocabulary of "n" words and then uses numerical indexes of those inplace of actual words themselves which saves a lot of space.

This count vectorizer can be used to run multiple NLP algorithms on our document.

## LDA - Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. and

We have to input the number of topics it categorizes some relevant keywords into each topic. This could be useful in cases like : we want to know about all the important events happening in 2018 in India from a large bunch of news articles.

Eg output:

- topic 1 : President, parliament, prime minister, election, budget.
- topic 2 : Cricket, World cup, India, lost, ...

Output for our dataset “:

```
topic: 0
************************
['student', 'school', 'college', 'University', 'percent', 'high', 'year', 'education', 'new', 'black']
************************
topic: 1
************************
['Please', 'ads.', 'write', 'stories,', 'follow', 'continue', 'step', 'block', 'great', 'display']
************************
topic: 2
************************
['Cohen', 'discussions,', 'Fueling', 'Capsule”', 'Activists', '“Trump', 'Trump?”', 'transition', '“How', 'Much']
************************
topic: 3
************************
['Trump', 'want', 'say', 'one', 'receive', 'update', 'partner', 'make', 'sponsors.', 'Donald']
************************
topic: 4
************************
['generic', 'Affleck', 'FDA', 'petition', 'pharmaceutical', 'Flynn’s', 'Cohen', 'frequent', 'mechanical', 'Florid
a']
************************
topic: 5
************************
['Sims', 'David', 'Stewart', 'film', 'Week', 'Spencer', 'Samantha', 'Bee', 'Kornhaber', 'new']
************************
topic: 6
************************
['police', 'officer', 'shoot', 'Police', 'kill', 'update', 'receive', 'gun', 'say', 'partner']
************************
topic: 7
************************
['Saudi', 'Arabia', 'Iran', 'Iranian', 'Tehran', 'Reed', 'Shiite', 'execution', 'relation', 'Iran’s']
************************
topic: 8
************************
['Apple', 'coal', 'carbon', 'iPhone', 'Carl’s', 'Brownback', 'Kansas', 'Las', 'felon', 'dioxide']
************************
topic: 9
************************
['song', 'reader', 'hello@theatlantic.', 'love', 'like', 'cover', 'film', 'music', 'please', 'Track']
************************
```

## K-Means Clustering

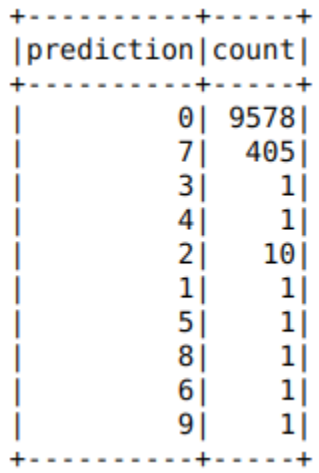
K-Means is used to cluster similar documents. Using countvectorizer is not effective for k-means so we are using Tf-idf values instead of countvectorizer.

TF - term frequency : a word has higher TF value if it appears more frequently in the document,

IDF - Inverse Document Frequency : A word has lower inverse document frequency if it appears in more than one document.

Pyspark’s K-Means produces skewed clusters so we have implemented both pyspark and sklearn’s k-means.

We have taken 10,000 rows and we want to divide it into 10 clusters, the results of pyspark's k-means and sklearn's k-means are shown below.



| prediction | count |
|---|---|
| 0 | 9578 |
| 7 | 405 |
| 3 | 1 |
| 4 | 1 |
| 2 | 10 |
| 1 | 1 |
| 5 | 1 |
| 8 | 1 |
| 6 | 1 |
| 9 | 1 |

pyspark

| | |
|---|---|
| 1 | 7832 |
| 2 | 1743 |
| 3 | 405 |
| 5 | 6 |
| 7 | 5 |
| 6 | 4 |
| 4 | 2 |
| 9 | 1 |
| 8 | 1 |
| 0 | 1 |

sklearn

Future Scope: Preprocessing can be improved by word embedding. We can also improve output of K-means by using Latent Semantic Analysis.