Fake News Detection based on Polynomial equation: A Beginners Approach

Sourabh Namillikonda, Hrishikesh Rendalkar, Pratik Dhumal,

Hemambujam Veeraraghavan

San Jose State University

**Abstract**: With the advent of internet and social media, along with the numerous benefits comes drawbacks. Fake news has been one such malicious threat to the society and given the connectivity of internet, they spread like wild fire. Social media and some news outlets deliberately publish fake new to increase viewership and readership while other spread it for political agenda. This outburst of fake news for various reasons has built out to become an immense social problem and one that requires immediate attention. The criticality of accessing a news in order to differentiate between real news and fake news is huge and it goes without saying is a enormous task as the reach for them is enormous whose outcomes are nothing short of disturbing. With this Alternus Vera project, we intend to work on a fake news dataset to analyze news, its source, the bearer of the news, political affiliation and other major factors that influence the news and come up with a solution to differentiate between fake news and real news. The purpose of this paper is to describe the solution we have come up with to differentiate fake news from real news and filter them out based on:[1]Sensationalism of the news, [2]Credibility of the news source, [3]Political affiliation and [4]The Speaker’s position. These simple but carefully chosen features has helped us with the accuracy while measuring and calculating the fake news quotient and identifying fake news based on the trained dataset. In order to increase the accuracy of the model, we have also used various data enrichments thus making the model accurate and reliable.

**Literature Review:** On researching on the analysis of fake news, we were able to come across a number of scholarly papers on analyzing, decoding and accessing fake news. On one such paper, the authors have observed that even though the advent of fake news was for marketing and advertising of products, they have now turned other major purposes[1]. And this is now a major concern that has to be handled by the Information Security department. While other authors have also sighted on how initially this fake news tread started off to attract web traffic to their websites, while now this has turned into activities involving creation of chaos amongst masses to hackers who send malicious codes in Clickbaits, to name a few[2].

On the other hand, some researchers have also shed light on the positive implications of fake news on the society. For instance, they have cited cases where the stock price of the companies has increased substantially with the fake news making its rounds among people. As more and more the news spread, the value of the stock kept increasing – even though the profit is short lived and momentary, the company has been made to look good on papers[3].

With Natural Language Processing and Machine learning libraries, along with some attention to details from the readers, fake news has proven to be detectable. In this effort to understand and predict fake news, we have focused on a lightweight machine learning model based on a dataset to access, analyze and predict fake news based on high-level feature titles.

**Data Source exploration:** Using the dataset available on the internet (Liar, Liar), we started our exploration on fake news analysis. We started off with understanding the dataset in terms of the labels and other critical data present in it. On careful attention we were able to identify the following labels that might be useful for the project:

*1: the ID; 2: the label; 3: the statement;*

*4: the subject(s); 5: the speaker;*

*6: the speaker's job title; 7: the state info;*

*8: the party affiliation; 9-13: the total credit history count, including the current statement; 9: barely true counts;*

*10: false counts; 11: half true counts;*

*12: mostly true counts; 13: pants on fire counts;*

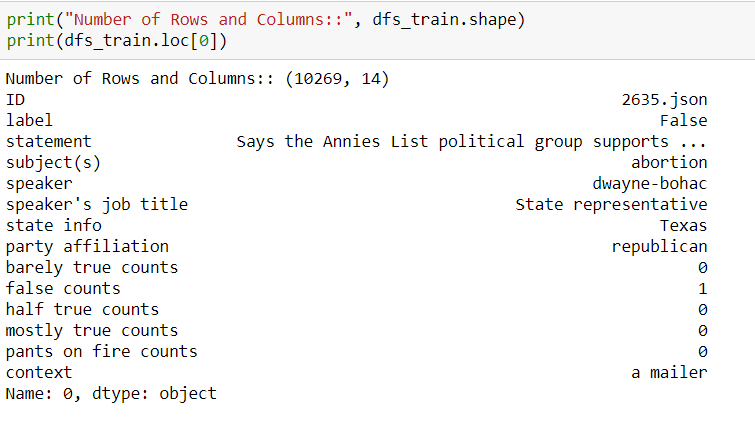
*14: the context.*

With these above-mentioned labels, as a team we were able to divide and conquer the fake news dataset. This paper focuses on:

* **Sensationalism of the news**
* **Credibility of the news source**
* **Political affiliation**
* **The Speaker’s position**

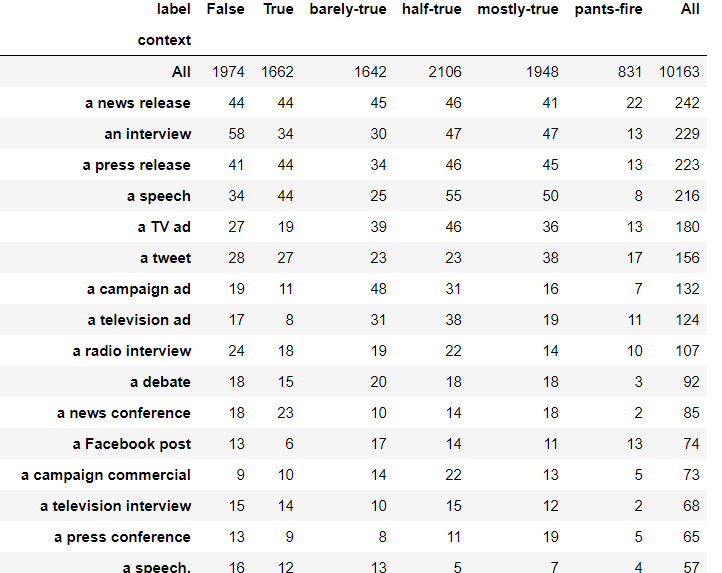
These components decipher a given news into four ways of looking at it and thus adding an in-depth analysis which pays back in accessing the credibility of the news.

*Pre-processing:* With the identification of the data source, we proceed to the pre-processing it for the analysis. The dataset contains approximately 10000 rows in the data set with 14 columns to them. We have removed the characters in order to enhance the runtime of the code and not having to deal with Not a Number (NaN) kind of errors. The sample visualization of the dataset that has been pre-processed looks as below:

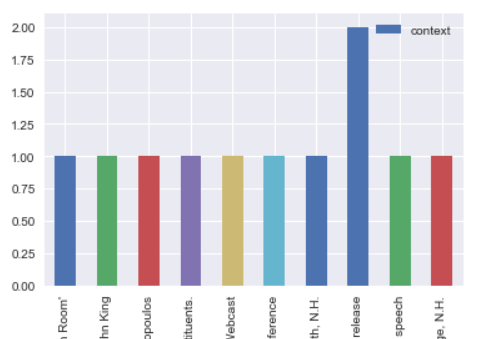


*Visualization:* With the pre-processed dataset, we do visualizations to understand the dataset and the correlation between the columns against one another, better. This step plays a crucial role in getting the nuances of the data present with relation to another.

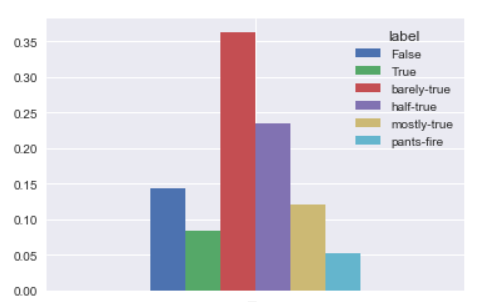
1. *Cross-Tabulation:*



1. *Bar Chart:*



1. *Column based label analysis:*



These visualizations servers as the proof of an initial probe giving us directions with regards to algorithms and other machine learning techniques that might turn out to be fruitful in order to build a proper model.

**Individual contributions:** Based on the above data explorations, we as a team decided to divide and conquer fake news.

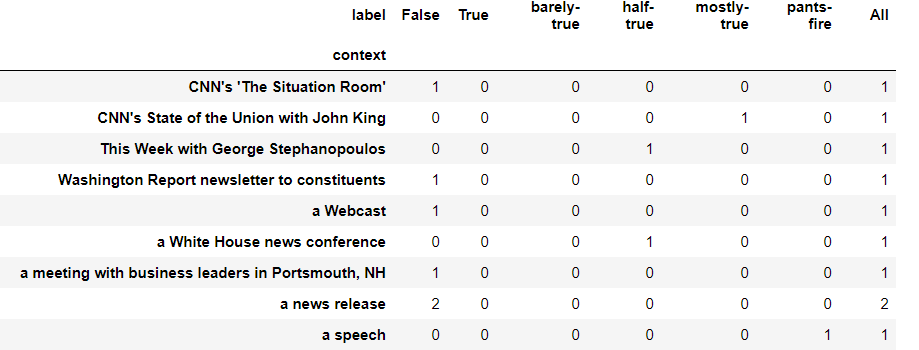
The dataset was viewed from different angles to ensure the important factors that affect the dataset to swing between fake and real news were identified and divided amongst the team. This approach helped us with different polynomial equations for each factor, building towards a model:

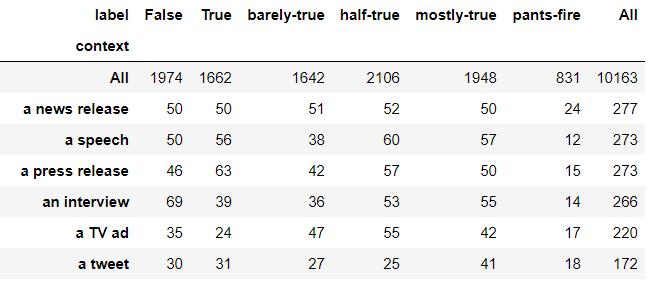
|  |  |
| --- | --- |
| **Feature** | **Author** |
| Credibility | Hemambujam Veeraraghavan |
| Sensationalism | Sourabh Namillikonda |
| Political affiliation | Hrishikesh Rendalkar |
| Bias | Pratik  Dhumal |

**Topic 1: Credibility**

Credibility of a news source plays a huge role in accessing and analysing fake news. In the given dataset, the “context” column has been chosen to be the area of interest. This label hold data of the news source, which gives an insight to the exact origin of the news. The tracking down of the news source first point of news analysis and help narrow down the genuity of the news itself.

After the generic analysis of the dataset in the previous section, we dig deep into each of the selected component. Below is the initial analysis of the dataset with respect to the “context” of the news source:

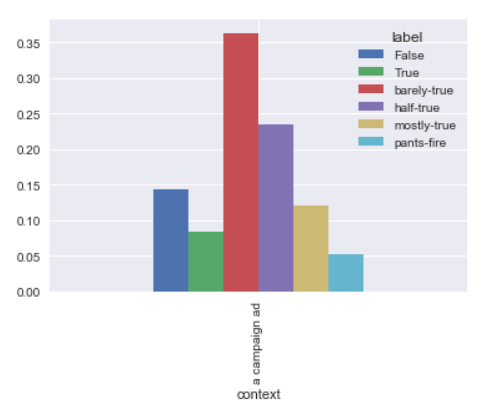




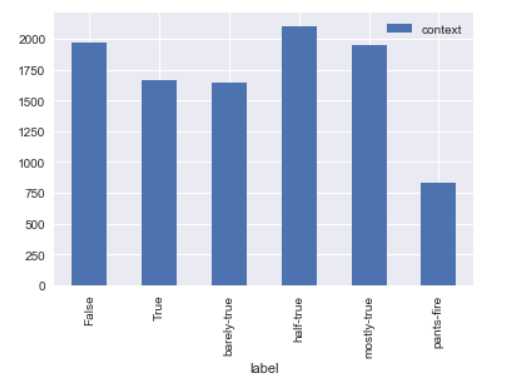
With the above cross tabulation, we infer the total number of

* False
* True
* Barely-True
* Half-True
* Mostly-True
* Pants-fire

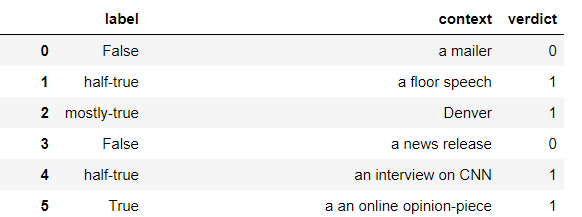
They give clear insight on the credibility of the news sources in the dataset along with their respective numerical counts. This gives a fair idea on the testing data set that is to be used for this component analysis. A bar chart on the same data that has been used for cross tabulation resulted in a chart which was further drilled down to depict just one context from the column – “a campaign ad” to get a closer analysis on it over a bar chart:



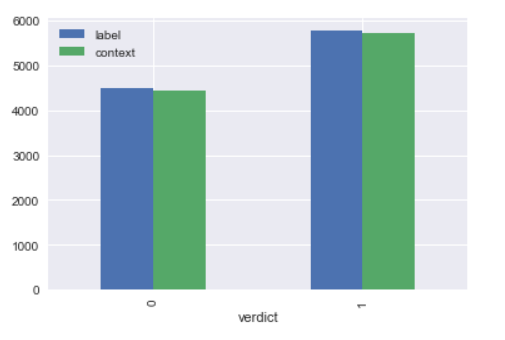
While the below gives the cumulative analysis of the labels for context from which a ball park count of the credibility of the news sources int eh training dataset can be reviewed:



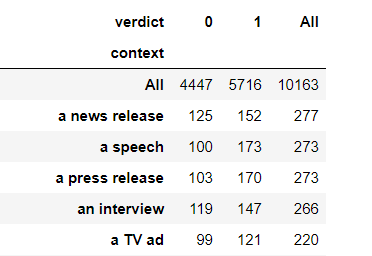
For making a clearer classification, combine False, barely-true, pants-fire as lie and half-true, mostly-true and True as truth into a separate column called “verdict” gives us the reliability of the news:



On a comparative scale, the below graph weighs the odds of a verdict being false against a label. This gives a different, fresher prospective of the dataset:

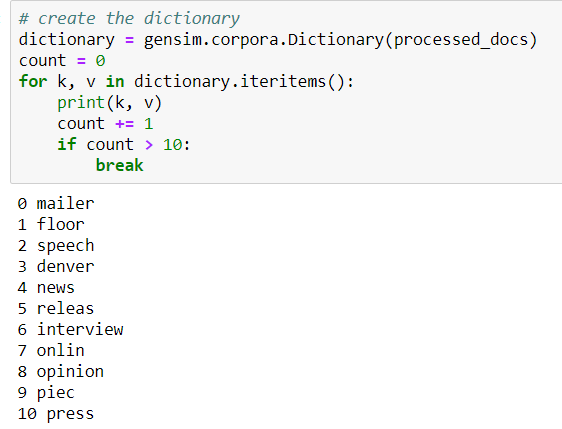


Based on the above graph, if we are to perform cross tabulation, then this gives a clear picture on the reliability of the news source given we have the verdict column to decide on:

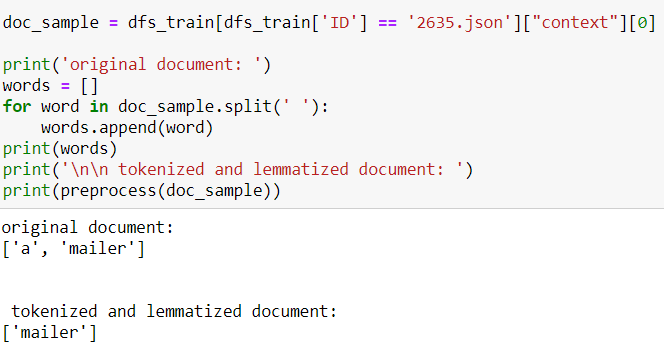


***Natural Language Processing:*** Using the nltk library, data preprocessing has been performed. Topic Modelling is a statistical model for finding nonconcrete topics that are present in the dataset. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions[4].

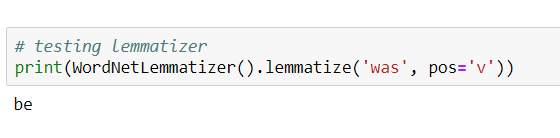
* **Tokenization:** Split text to sentences and sentences to words. Switch into lowercase for words and remove punctuation.



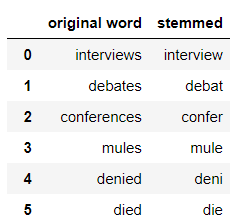
* Remove words that have fewer than 3 characters.
* Removed stop words from the dataset.



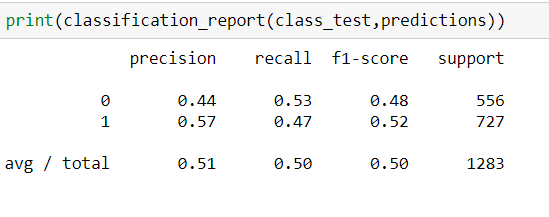
* **Lemmatization** — words in third person are changed to first person and verbs in past and future tenses are changed into present.



* **Stemming** — words are reduced to their root form.



***Naive Bayesian classifier:*** Based on the dataset, the below list the precision, recall for the context. The average precision is about 51% as shown below:

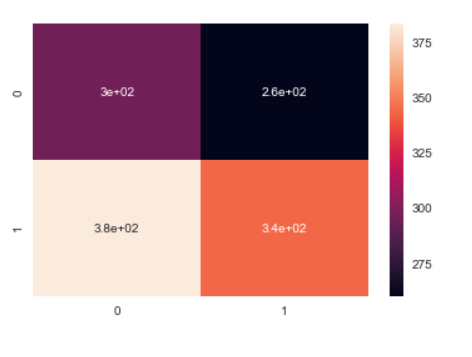


***Polynomial equation for the factor:***

Based on the NB calculated above, a polynomial equation for the credibility of the news source can be calculated using:

*Credibility factor = 0.4\*(prediction from NB)*

A heat map can be plotted on the basis of the outcome form the Naïve Bayesian classifier which is found below:



***Topic 2: Sensationalism***

Sensationalism was the factor selected by me and based on some research, following are some of the core factors in sensationalism that can be important in determining how sensationalism effects the truth component in a data.

Sensationalism basically targets human emotions instincts. Sensationalism is used primarily to get ‘unwholesome emotional responses’ and grab the attention of a reader.

In a general sense it is considered undesirable as it motivates a move away from base facts.

Sensationalism has been properly studied for traditional news sources but requires further inspection in unconventional news like social media and internet-based sources. This is important as unconventional sources are rapidly gaining importance as major source of information dissemination.

Here are some of the major factors that were identified as a measure to quantify sensationalism:

• Timeliness: If story focuses on a recent event. E.g. breaking news

• Impact and prominence: If famous or public figures are involved and/or if the story impacts people in a geographic area.

• Conflict and controversy: If it involve confrontation and controversy.

• Human interest: If it is tailored for a specific interest.

• Proximity: Local or global impact

• Unusualness: How unique, odd, and singular the story is.

• Usefulness: If useful, practical, or educational information and only salient points are given e.g. bullet points are easier to focus on.

Thus, we identified some features which can play an important role in determining how sensational an article is.

We aim to quantify this based on training data and get a meaningful result of sensationalism which will help us identify fake news.

**Implementation**:

To extract sensationalism various columns were added to the original data set. Columns like statement length, capital letters in statement, punctuation, profane words and occurrence sensational terms was calculated. The final two required a separate data set for estimation profane words were used from Pattern library word list and sensational topics were included from “Analyst's Desktop Binder of Homeland Security”.

These known sensational topics along with other factors were good in proving our assumptions. Following are the conclusions that we were hoping to draw and were actually shown by the data:

1. On an average sensational story has:

• more number of punctuation like (!,?)

• more profanity

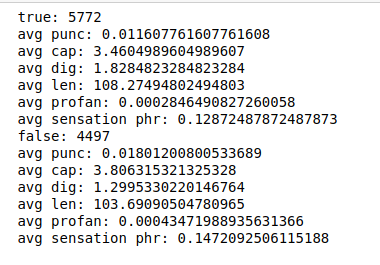
• more sensational phrases

• more capital letters

1. Whereas true stories are:

• more specific with higher digits

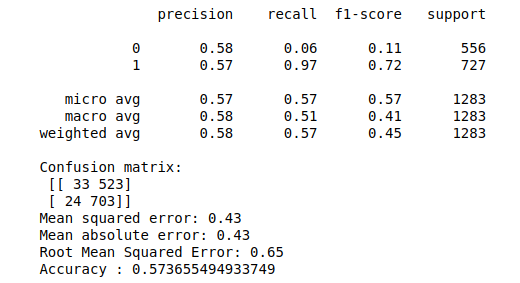
• longer with more details



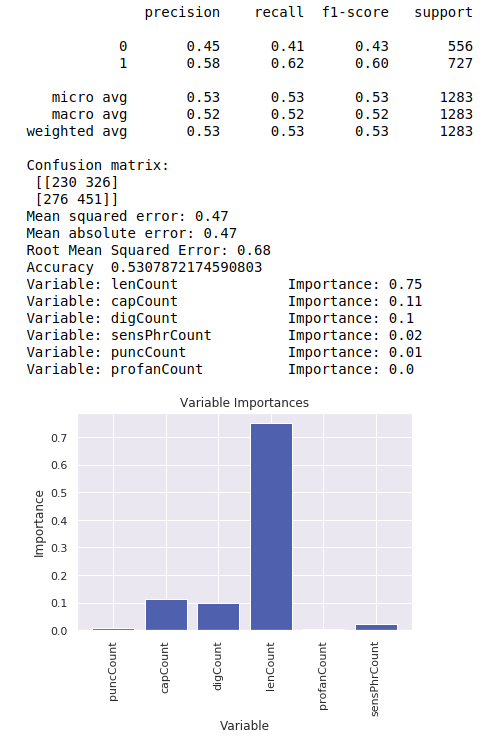
Using some of these features we wanted to build a model to judge sensationalism and further the truth factor in the statements. So we applied various models and got the following results:

**Decision tree:** 52%

**Naive Bayesian:**

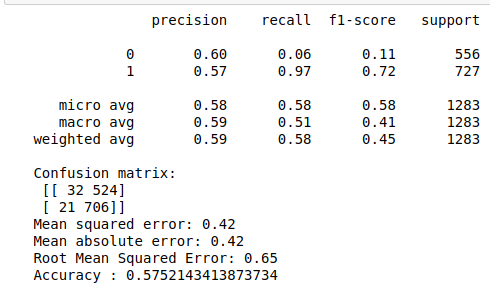


**Random Forest:**



Next to get a complicated factor a Doc2Vec algorithm was used using a separate data set (data enrichment) which had a training data containing sensational news articles.

The data was tagged, and a 100-epoch model was made. Using this new model our original data set was predicted and a new issensational column was added to improve the results. After this operation we improved the results:



Our expectation is that if we get better training data closer to the original data that we are testing we will significantly improve our results. This is indicated by the fact that even the use of unrelated data set improved the overall results although not by much.

We understand that sensationalism alone is not enough to determine if a news is fake or not, but it does play a role. We wanted to assign a 30% importance to sensationalism among the 4 factors that we are using to classify data as true or not.

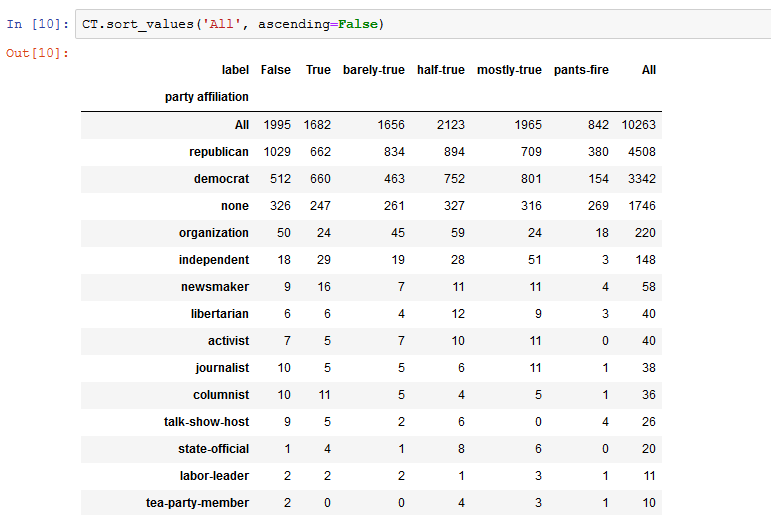
*Sensationalism factor = 0.3\*(prediction from NB)*

Further, sensationalism may not be just linearly related to the overall equation. We can conclude that it may depend on the results of other factors like source credibility. It can be assumed that if credibility is high then sensationalism factor will compound the effect that the news is explosively true. But if credibility is low then high sensationalism will most likely be fake news.

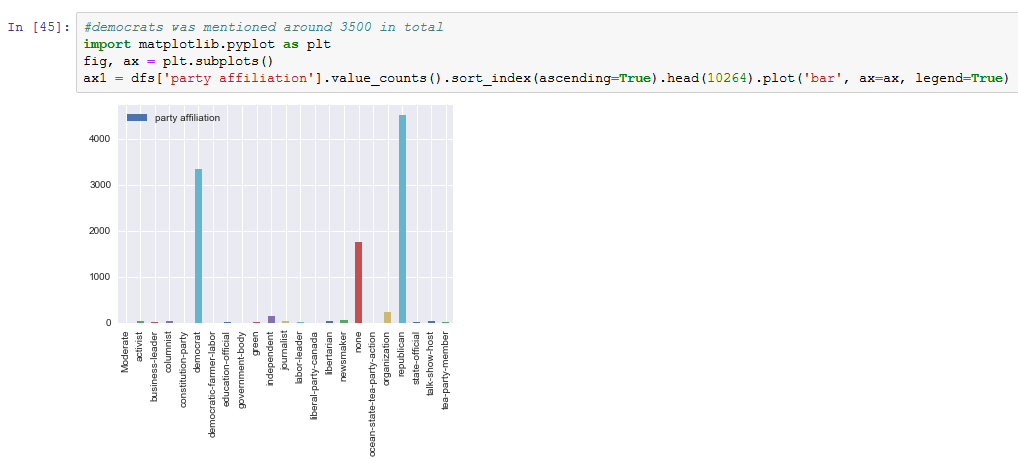
***Topic 3: political affiliation:***

Political affiliation is nothing but having official connection to a news by a party. To begin with we had to understand the dataset. So, we had divided the dataset into test and train. The dataset had number of columns and every column was interrelated with each other. So, we decided to visualize the dataset to understand it better.

To establish the relationship between columns such as fake, true, barely-true etc. with party affiliation. We can see in below figure the obtained result:

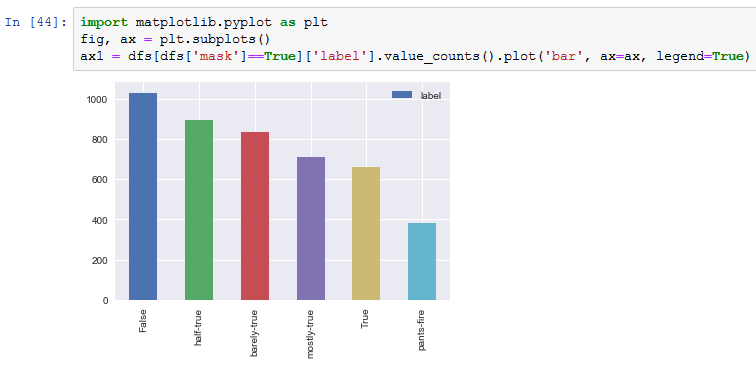


Now using matplot library I tried to plot the graph to see how many times each party is mentioned in the news. From the graph I could infer that Republican and democrat are the two parties with highest occurrences. We can see that Republicans were maximum time in the news followed by Democrats.



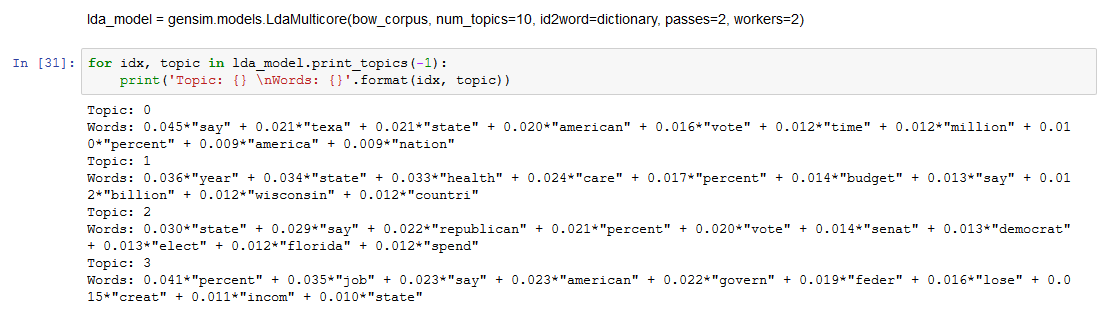
The more we visualize, more clarity we get into the dataset. I tried plotting the graph for barely-true values against their count in the entire dataset using seaborn library. I also tried plotting mostly true count and false count using matplot library. But the I could not gain any insight from that.

Using NumPy and matplot I tried plotting the graph for republicans from party affiliation column. Basically, I wanted to visualize which parties are associated with which labels. For example, the number of false news associated with Republicans and so on. The below graph shows the actual plotting:



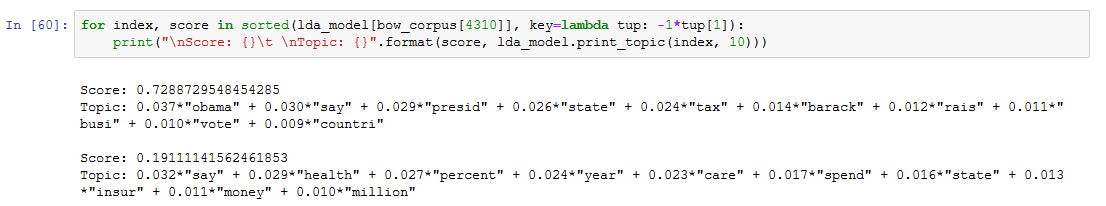
In the group part I implemented LDA using bag of words. So, to start with I had to research on what is LDA, what are its uses and how can we incorporate in our dataset. I referred to lecture slides and some online material. From what I read LDA is generative probabilistic model. So, in layman’s term it is a collection of documents which is composed of different words. So basically, it is used for topic modelling. It is used to identify text in a particular document.

Before applying LDA on a dataset we should perform preprocessing on the data. Preprocessing includes tokenization, removing stop-words, lemmatization and stemming. What I did was identified number words occurring in a particular document.

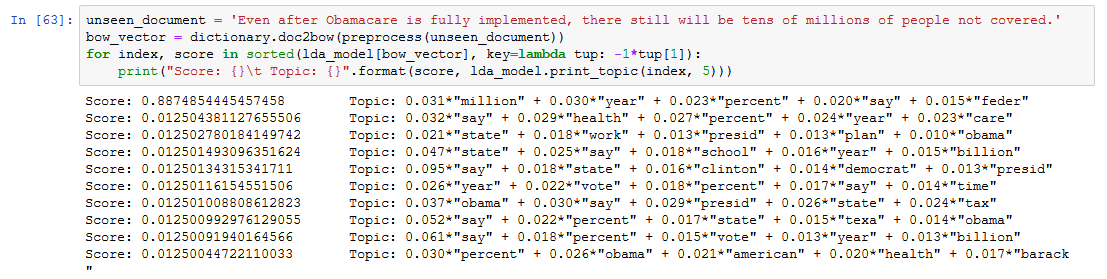


So, in the figure above we can see that weights of the words in a given topic.

Then I tried performance evaluation using LDA bag of words. So, from that we can infer how much our document is related to the given topic. For example, the below document is around 72% relevant.

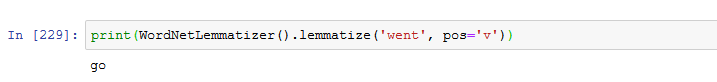


Then I tested the model on an unseen document. Basically, it is done in order to test how accurate our built model fits into the given topic. So, I got the result saying that the unseen document is around 88% related to the topic.

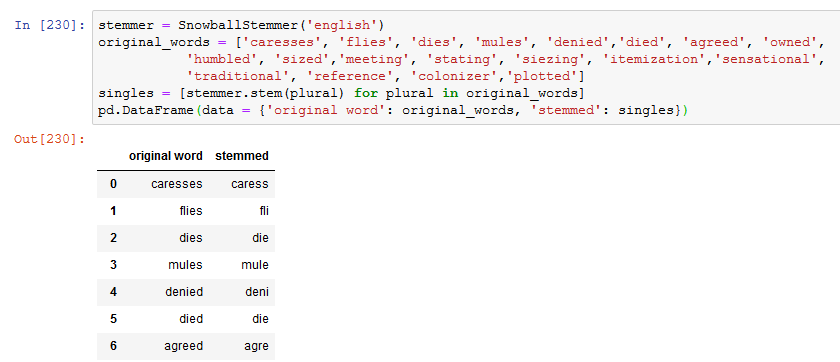


It was very interesting to know how LDA works and how the topic relevance is calculated. We are working on formulating polynomial equation.

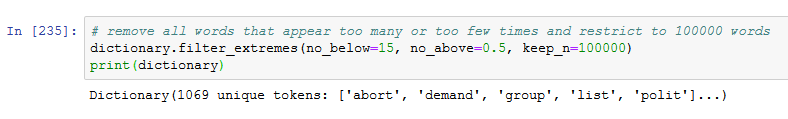
I have applied Lemmatizer which will help in simplifying our sentences without changing the meaning of the sentence. For e.g. word went becomes go.



Then I have applied stemmer. Stemmer helps in removing suffixes from the word. This makes the word compact to process. For e.g. owned becomes own.



We have applied above two methods on the statement column of the dataset. We have also created a dictionary eliminating words which appear too frequent or too less.

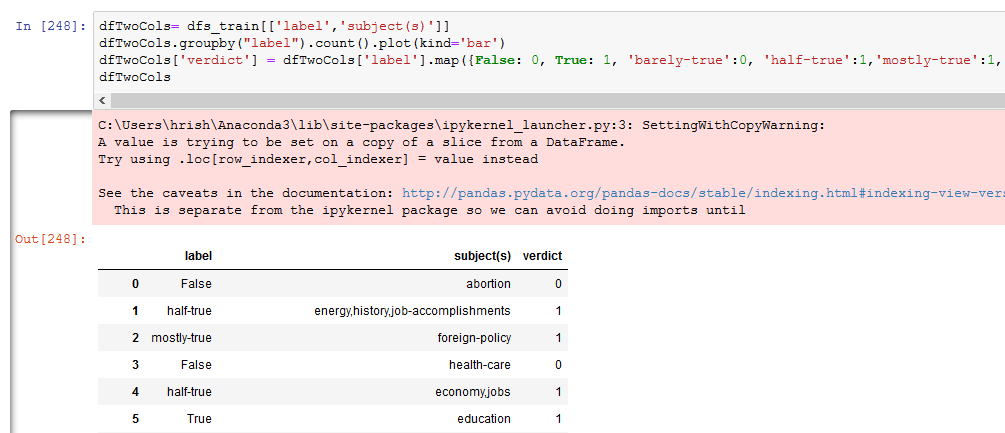


Then I have assigned weights to the labels. This help with the polynomial equation for the factor:

*Political affiliation*

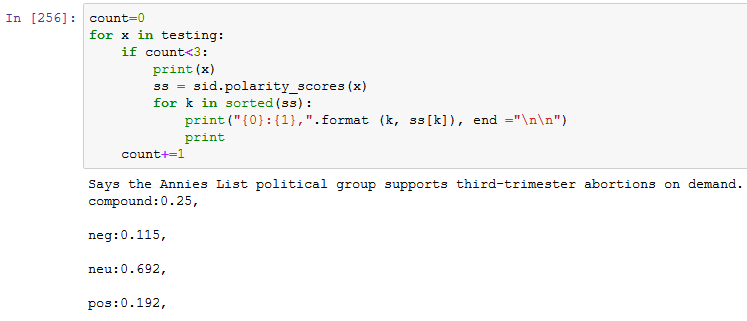
*factor = 0.2\*(prediction from NB)*

This makes the data processing easier.

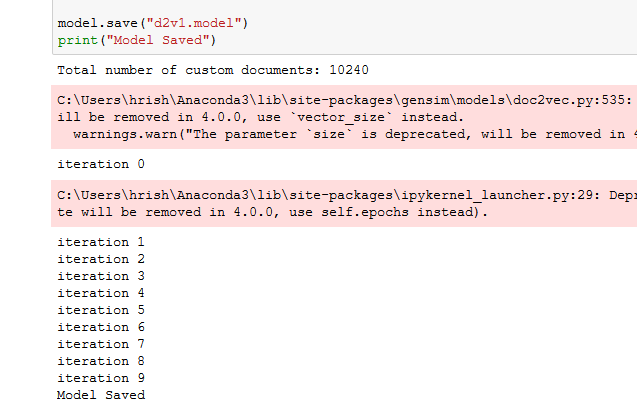


Additionally, we cleaned and tokenized the document.

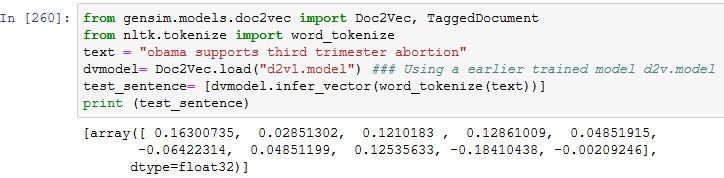
Using nltk Vader library I tried to calculate the frequencies of positive/negative and neutral words in an article. Then we have used algorithm to train the dataset and calculate sentiment analysis.



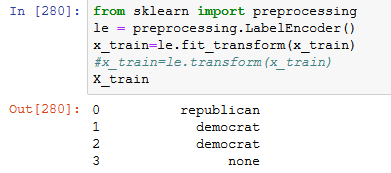
Then I have applied doc2vec to generate custom embeddings of word vectors. So I performed 10 iterations to train my model. More the number of iterations more accurate is your model.



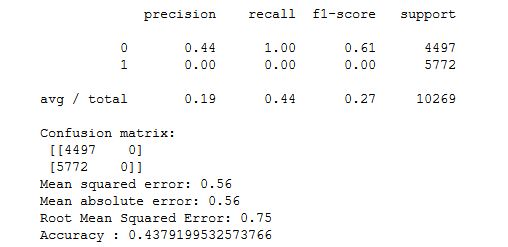
Word embeddings for a sample text gave me the output as follows:



Then I am training my model using verdict column which we have introduced as a part of data enrichment. So here also we need some data cleaning as the we are taking party affiliation as our x\_train. We need to remove ‘nan’ to process the document. We are using label encoder to properly transform our data to integer. LabelEncoder assigns a number to every word in the column.



After that to check the model effectiveness I applied random forest and logistic regression on top of the vector I got. I got less accuracy for both of around 43%.



This states that the model is not very accurate. Political affiliation on the statements column and combined with the verdict gives better result but it cannot be the only criteria for deciding the inclination of the political party. So, I can assign around 20% of weightage to political affiliation in the overall decision-making process. So, the equation would be 0.2\*prediction by naïve Bayes/random forest.

***Topic 4: Bias***

The below are the list of steps covered in the process of understanding the effect of bias on the dataset:

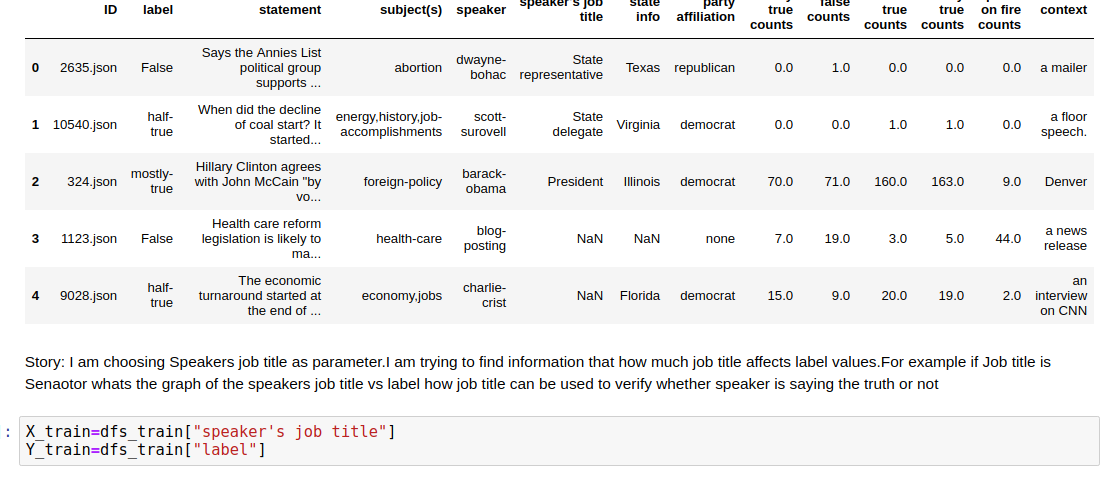
1. Preprocessing
2. Visualization
3. Lemmatization
4. Stemming
5. Spell Check
6. Stop words removing
7. Dictionary
8. Doc2Vec
9. LDA
10. TF-IDF
11. LDA using TF-IDF
12. Sentiment analysis
13. Rating
14. Naive Bayes
15. Data enrichment
16. Vectorization

**Factor: Bias**

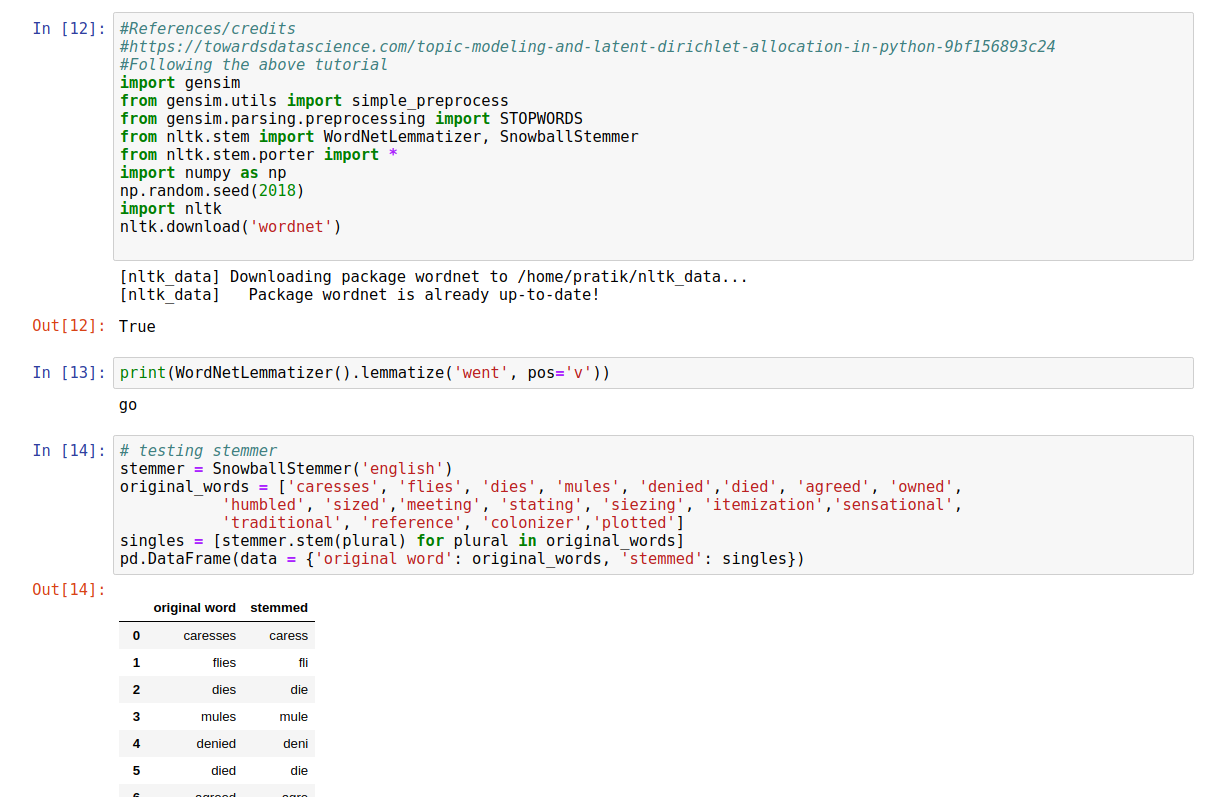
**Factor accuracy:0.3**

**Preprocessing:** It involves choosing the appropriate column, the speakers job title.

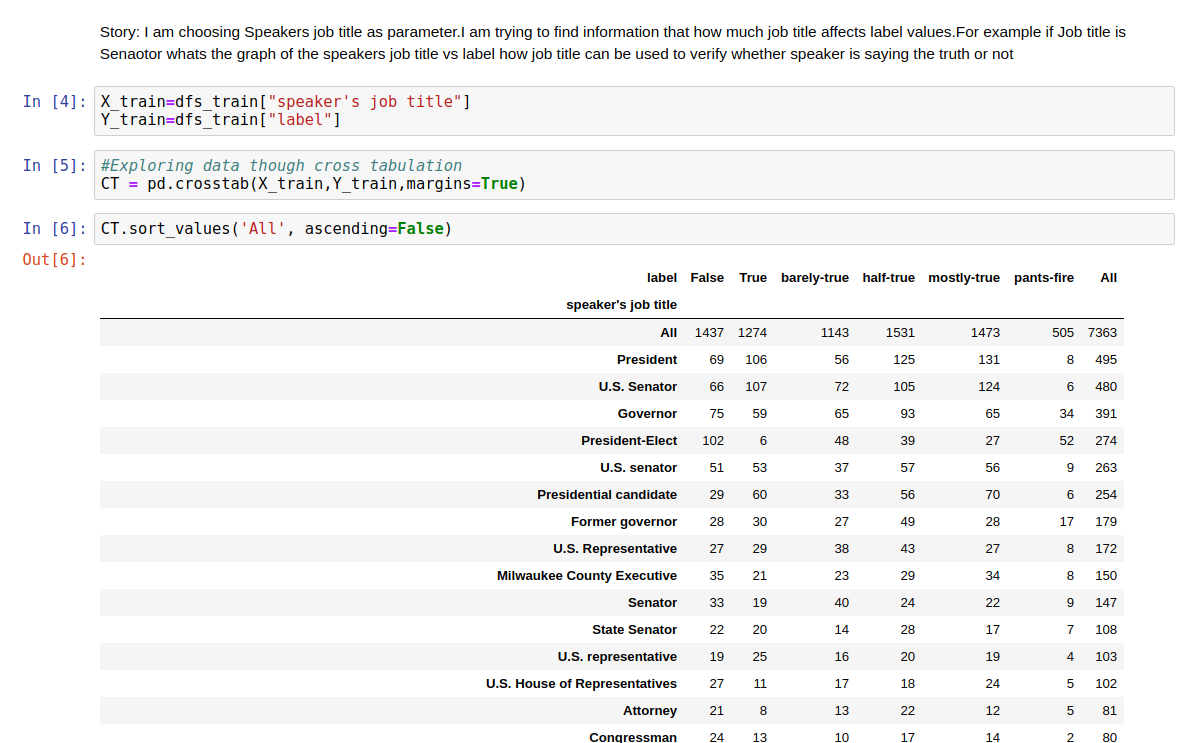
Preprocessing-1

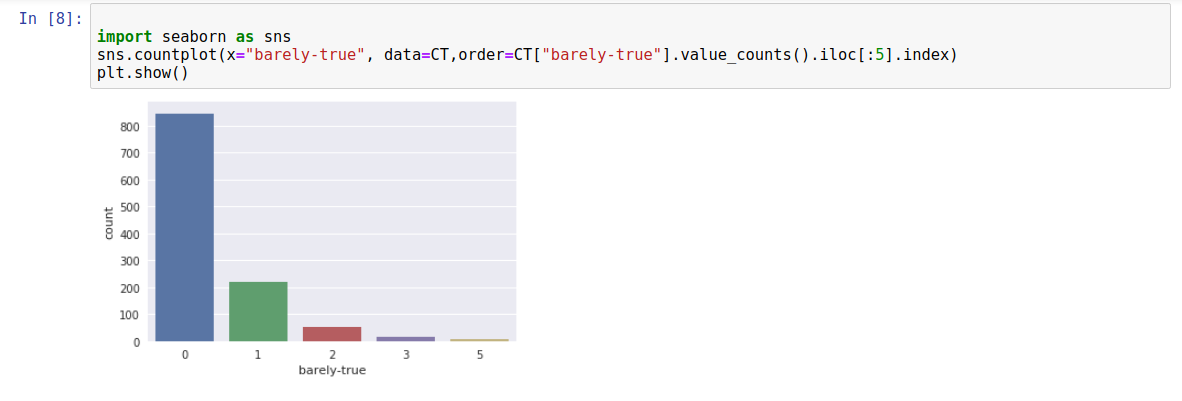


Preprocessing-2



2. **Visualization** in terms of cross tabulation.

Plotting graphs of speaker’s job title and the value of labels which are True, False, Barely true and pants on fire.

Plotting value for one specific user such U.S Senator and his graph against label values to review his credibility.

**Stemming and Lemmatization**: Lemmatization means to convert words into its original format. Stemming means converting word such dies in its original format such as die which is required in NLP to process document.

**Removing stop words**

In NLP the stop words such as a, an the which are insignificant we are removing theta to process the document. Below is the combined definition of lemmatization, stemming and stop words removal which takes entire text and outputs formatted text.

**Dictionary:** We have created our own dictionary to store the words which can be used in applying TF-iDF and LDA algorithms. We are also checking the frequency count of each word in the dictionary.

**TF-IDF and LDA:**

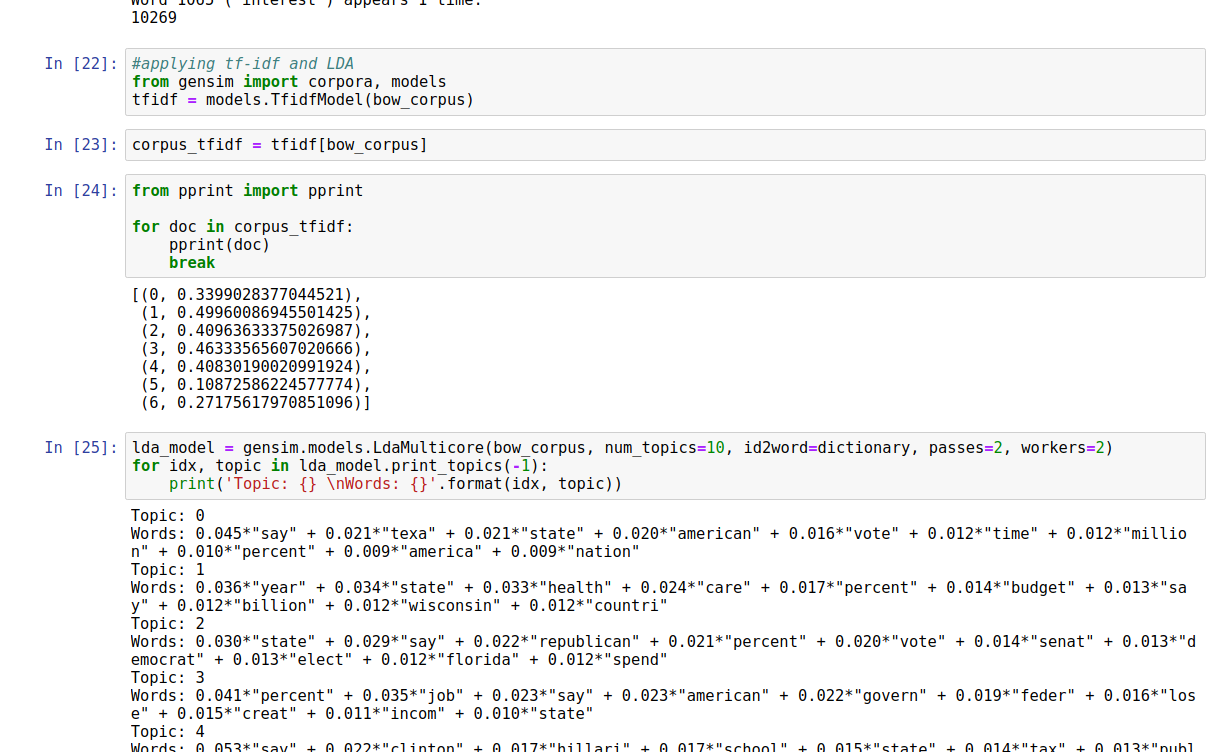
TF-IDF is an algorithm it has 2 subparts:

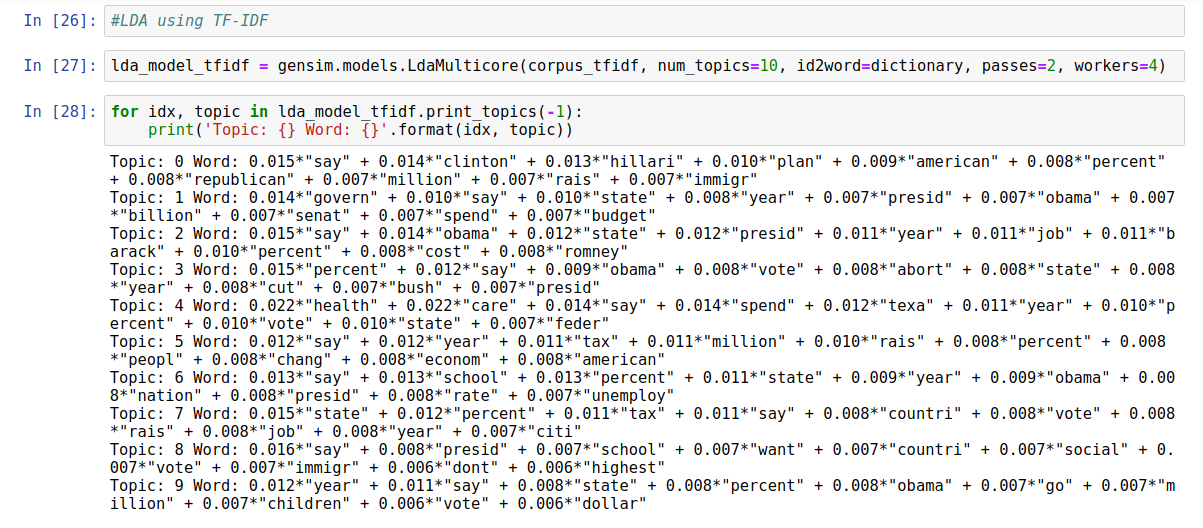
1. Terms frequency

2. Inverse data frequency.

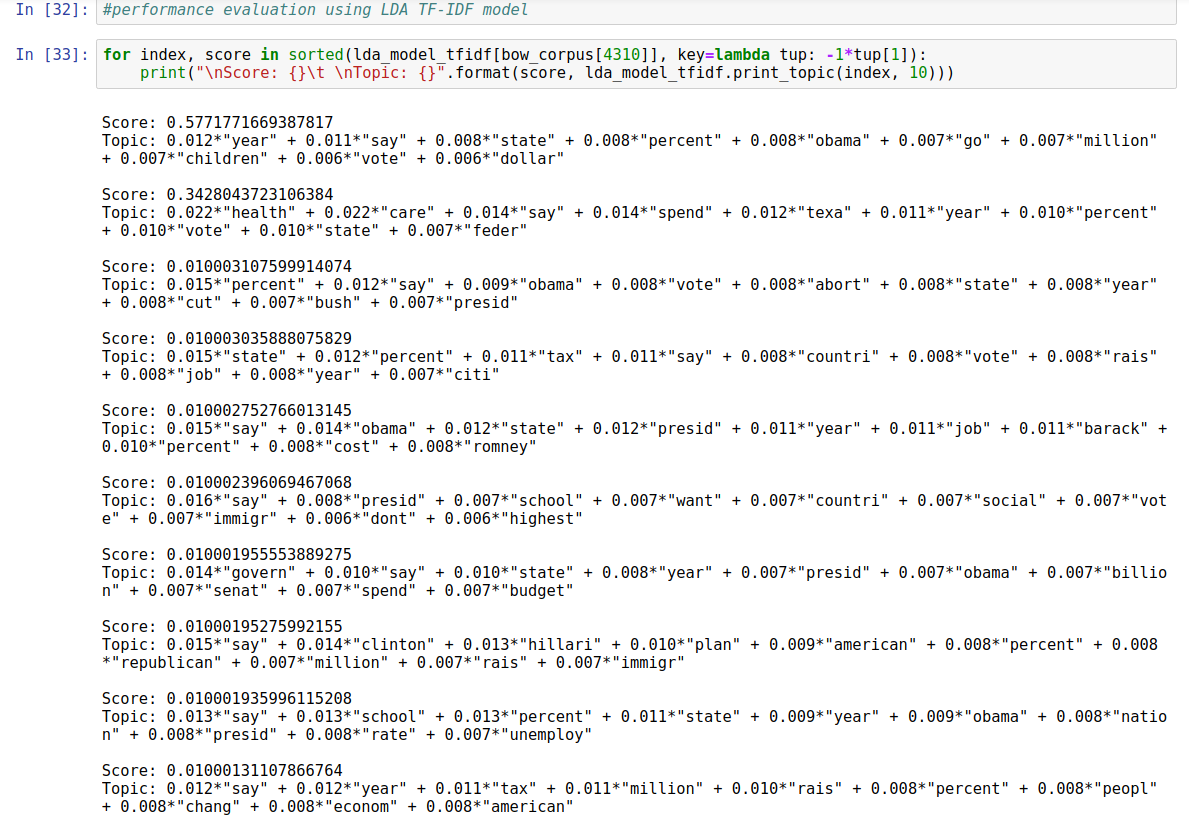
TF gives us the frequency of the word in each document while IDF give us the weight of all rare words present in the document.

In LDA we divide the document on the basic of topics and classify words as per topic. Here we are checking the frequency of each word and comparing it with normal LDA model.

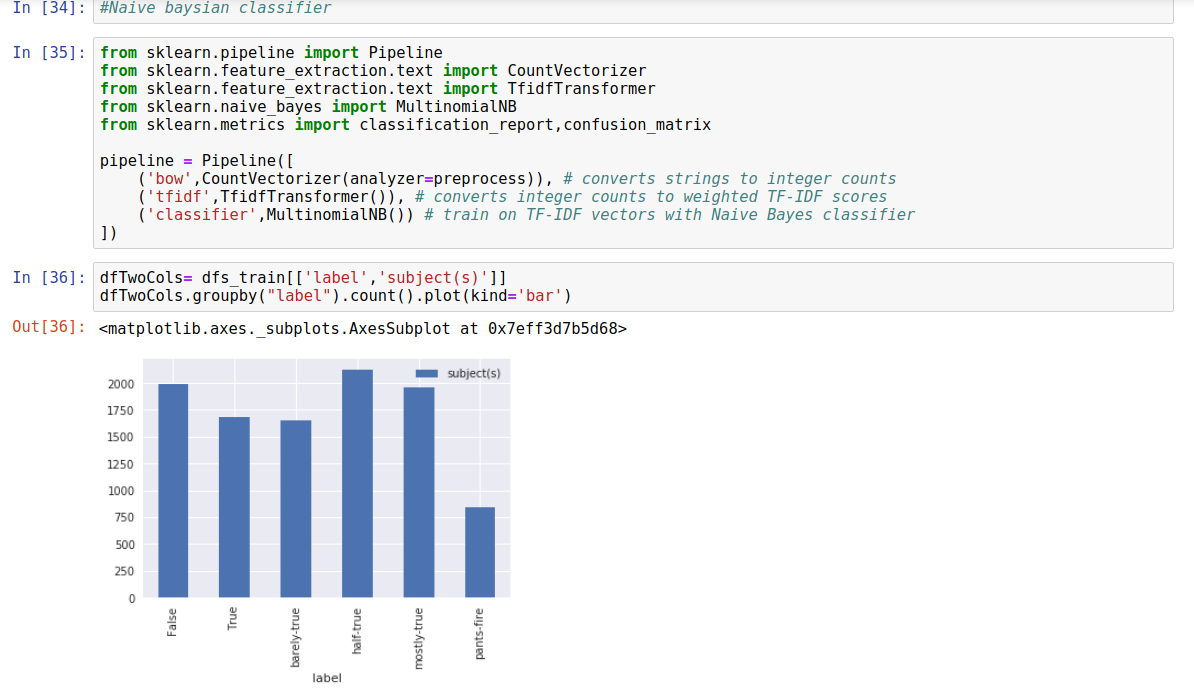
****

**TF-IDF using LDA:** Here we are applying TF-IDF using LDA.

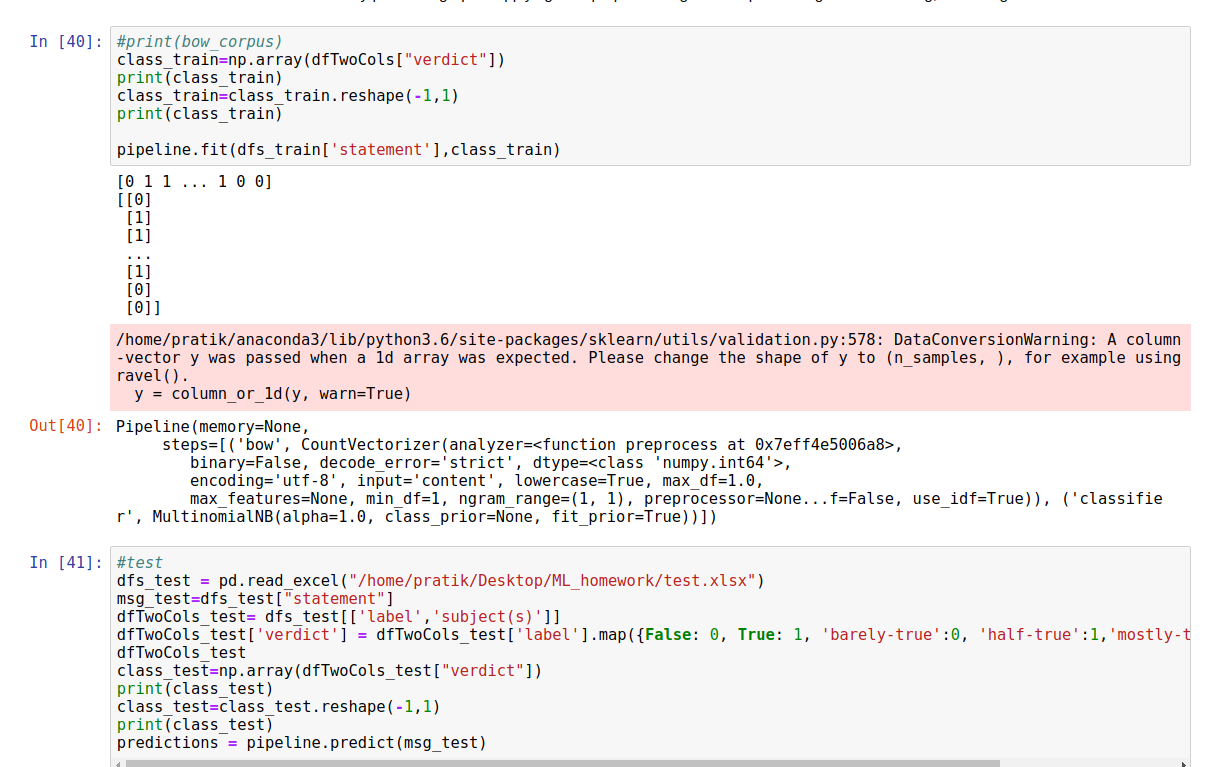
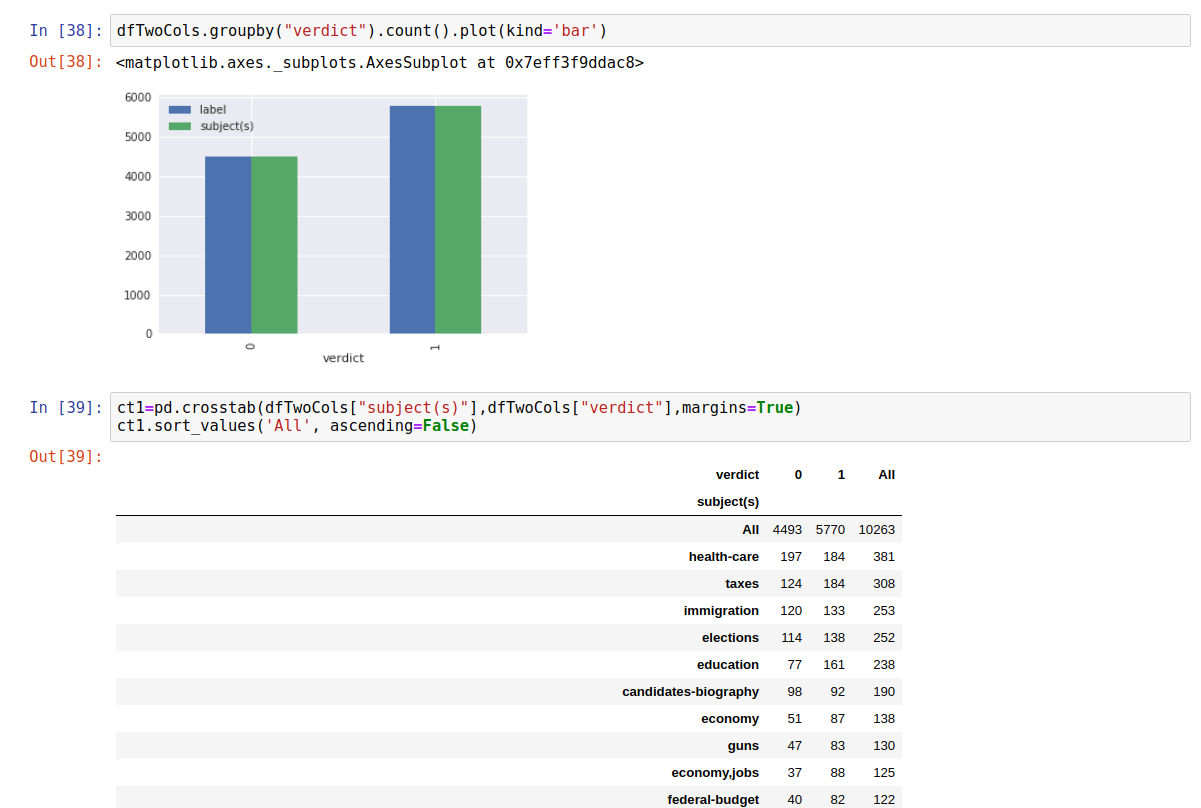
**Performance Evaluation of TF-IDF:** Here we are evaluating the performance the performance of our model.

****

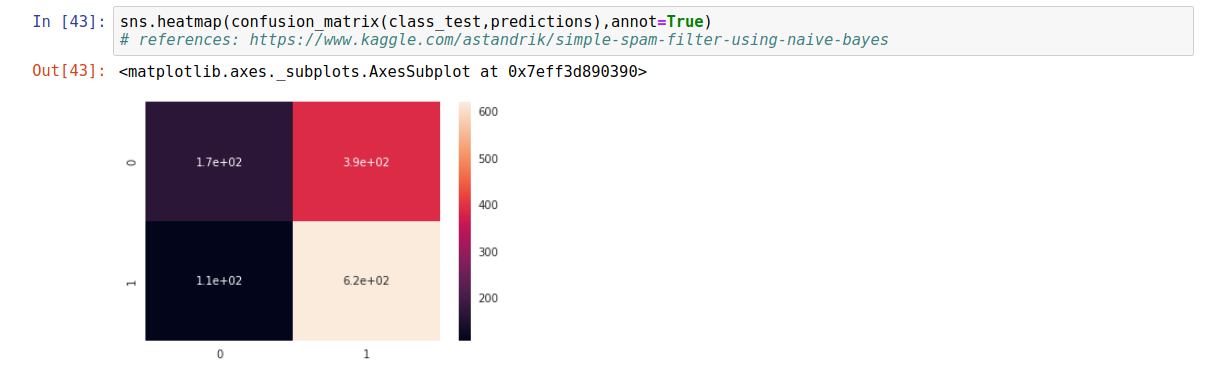
**Naive Bayes Classification:** Here we are training TF-IDF vectors using Naive Bayes Classifier.

****

**Data Visualization for Verdicts**

****

**Heatmap:** Heat map of confusion matrix**.**

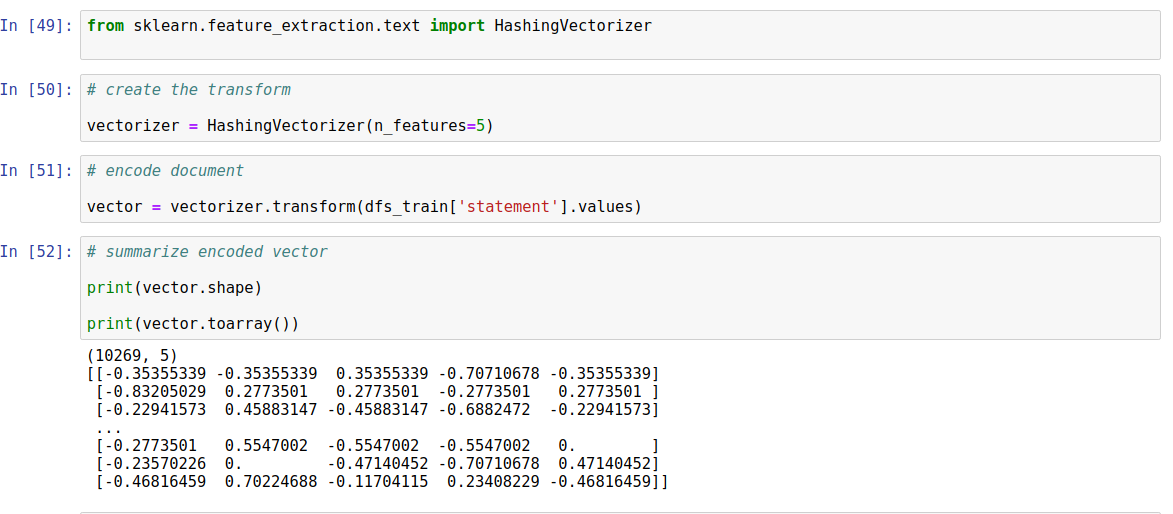
****

**Data enrichment and sentiment analysis**

Based on the above calculation, we have come up with the polynomial equation for bias as:

*Bias factor = 0.1\*(prediction from NB)*

**Vectorization:** To encode the document and convert it into vector for further analysis.

****

**Conclusion:** Thus, based on the individual contributions, we were able to form a polynomial function to identify fake news. The function is of the form (a1x1 + a2x2), where,

a -> constants based on regressive testing

x -> factors identified to be affecting news

**Polynomial Function**

**F(n) = { 0.4\*Source**

**+ 0.3\*Sensationalism**

**+ 0.2\*Party affiliation**

**+ 0.1\*Bias }**

The accuracy of the model is about 67% and we intend to work towards the betterment of the model in due course of time.

**References:**

[1] Rubin, V.L., Chen, Y., Conroy, N.J., 2015. **Deception detection for news:** **Three types of fakes**, in: Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community, American Society for Information Science, Silver Springs,

MD, USA. pp. 83:1–83:4. URL:

<http://dl.acm.org/citation.cfm?id=2857070.2857153>

[2] l Messabi, K., Aldwairi, M., Al Yousif, A., Thoban, A., Belqasmi, F., 2018. **Malware detection using dns records and domain name features**”,

in: International Conference on Future Networks and Distributed Systems (ICFNDS), ACM. URL:

<https://doi.org/10.1145/3231053.3231082>

[3] Hassid, J., 2011. Four models of the fourth estate: A typology of contemporary Chinese journalists. The China Quarterly 208, 813832.

doi: 10.1017/S0305741011001019

[4] Data topic modelling using nltk library:

<https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24>

[5] Giuliani-Hoffman, Francesca (November 3, 2017). ["'F\*\*\* News' should be replaced by these words, Claire Wardle says"](https://money.cnn.com/2017/11/03/media/claire-wardle-fake-news-reliable-sources-podcast/index.html). *Money.CNN*. Retrieved November 24, 2018.

[6]<https://journals-sagepub-com.libaccess.sjlibrary.org/doi/abs/10.1177/1464884916683549>

[7]<https://medium.com/@mishra.thedeepak/doc2vec-simple-implementation-example-df2afbbfbad5>

[8]<https://www.scribd.com/doc/82701103/Analyst-Desktop-Binder-REDACTED>

[9]<https://github.com/clips/news-audit/blob/master/SensationalismClassifier/SensationalismClassifier.py>