**CLASSIFIER FOR FAKE NEWS RECOGNITION USING MULTINOMIAL NAIVE BAYES**

by

Pavan Manchikatla

STAT 52900  
Applied Decision Theory and Bayesian Statistics

**Contents**

* Introduction ……………………………………………………………………………………………… 4
* Data Preprocessing …………………………………………………………………………………… 4
* Dataset Description …………………………………………………………………………….. 4
* Dataset Sample …………………………………………………………………………………… 4
* Data Preprocessing ……………………………………………………………………………… 5
* Feature Selection ………………………………………………………………………………… 6
* Multinomial Naive Bayes Classifier …………………………………………………………… 7
* Assumption: Bag of words Model ………………………………………………………… 8
* Building and Training Model ……………………………………………………………………… 9
* Comparing the features on various Metrics ………………………………………………. 11
* Results ………………………………………………………………………………………………………. 15
* Conclusion …………………………………………………………………………………………………. 17
* References …………………………………………………………………………………………………. 18

**List of Figures**

* Segregation of labels ……………………………………………………………………………….6
* Feature Counts in each Label (Keeping Ties) ……………………………………………7
* Feature Counts in each Label (Not Keeping Ties) ……………………………………..7
* General Approach Used ……………………………………………………………………………9
* Valid vs Invalid Documents (Remove Percentage = 0.25) ………………………….10
* Valid vs Invalid Documents (Remove Percentage = 0.5) ……………………………11
* Valid vs Invalid Documents (Remove Percentage = 0.75) ………………………….11
* Comparsion of Features when Stemming is Applied ………………………………..12
* Comparsion of Features when Stemming and Lemmatization is Applied ….12
* Comparsion of Features when Stemming is Applied and Feature is not applied …………………………………………………………………………………………………….13
* Comparsion of Features when Stemming and Feature is Applied …………….13
* Micro Accuracy vs Number of Features ……………………………………………………14
* Macro Accuracy vs Number of Features …………………………………………………..14
* Macro F-1 Score vs Number of Features …………………………………………………..14
* F1 Score vs Number of Features ………………………………………………………………15
* Accuracy vs Number of Features ……………………………………………………………..15
* Heatmap: Validation Set ………………………………………………………………………….16
* Precision, Recall,Accuracy and F1 Score on Validation Set ……………………….16
* Number of labels in Test Data After Prediction ………………………………………..17

**Abstract**

In this project titled "Classifier for Fake News Recognition using Multinomial Naive Bayes" I model the task of fake news detection using a Multinomial Naïve Bayes classifier within a Bayesian framework. The focus is on categorizing text data accurately, considering the inherent variability and diverse nature of news data. My approach simplifies the process of classification, resulting in a straightforward procedure for predicting whether a news piece is fake or real. This method is easily adaptable by any organization aiming to automate the detection of fake news. I demonstrate the effectiveness of this model using a variety of datasets, highlighting the impact of different preprocessing techniques and feature selections on the model's performance. The results, detailed in this presentation, illustrate the practical application of the Multinomial Naïve Bayes classifier in real-world scenarios, offering an efficient solution for combating the spread of misinformation.

**1.Introduction**

* Fake news is defined by the New York Times as a made-up story with an intention to deceive”, with the intent to confuse or deceive people. They are everywhere in our daily life and come especially from social media platforms and applications in the online world.
* The goal of the project is to construct a text classification model using multinomial naive bayes algorithm to implement a Multinomial Naive Bayes classifier in R and test its performances to detect and classify fake news.

**2. Data Preprocessing**

2.1. Dataset Description

The suggested data set is available on Kaggle. Possible suggested labels for classifying the text are the following:

* True - 5
* Not-Known - 4
* Mostly-True - 3
* Half-True - 2
* False - 1
* Barely-True – 0

The Kaggle dataset consists of a training set with 10,240 instances and a test set with 1,267 instances.

2.2. Dataset Sample

|  |  |  |
| --- | --- | --- |
| Labels | Text | **Text\_Tag** |
| 1 | Says the Annies List political group supports third-trimester abortions on demand. | abortion |
| 2 | When did the decline of coal start? It started when natural gas took off that started to begin in (President George W.) Bushs administration. | energy,history,job-accomplishments |
| 3 | Hillary Clinton agrees with John McCain "by voting to give George Bush the benefit of the doubt on Iran." | foreign-policy |

A graph of a bar chart

Description automatically generated with medium confidence

Figure 1: Segregation of labels

2.3. Data Preprocessing

* **CHALLENGE:** Remove inconsequential words and simplify the data
* **METHODS USED:**
  + Remove Stop-Words, Punctuations, Numbers, Whitespaces
  + Lowering Text
  + Normalization using Stemming, Lemmatization
    - Stemming: (‘running’, ‘runner’, ‘runs’) -> (‘run’, ‘run’, ‘run’)
    - Lemmatization: (‘ate’, ‘ran’, ‘running’) -> (‘eat’, ‘run’, ‘run’)
  + Tokenization (with each word as token)

2.4. Feature Selection

* **CHALLENGE:** Large vocabulary containing irrelevant words
* **FEATURE SELECTION:** Selects a subset of vocabulary such that only relevant terms of each class are used. It decreases the size of vocabulary and removes noise features, reducing the overfitting that may occur.
  + Assigns utility measure to each combination of (term, class) and selects the top n terms per class.
  + We use Mutual Information measure, which gives a sense of how relevant and related a word is to a class.
* **Dealing with Tie Cases:** For terms with same score, we have the option of keeping the ties vs removing the ties.

A graph with green bars

Description automatically generated

Figure 2: Feature Counts in each Label (Keeping Ties)

A graph of a bar chart

Description automatically generated

Figure 3: Feature Counts in each Label (Not Keeping Ties)

**3. MULTINOMIAL NAÏVE BAYES CLASSIFIER**

* In a basic sense, if we have a text document, we can utilize the Bayes’ Theorem to make a prediction on it.
* The prior is the probability assigned to a label or a class. This can either be taken as a uniform probability throughout all classes or can be assigned with respect to each class occurrence in the training set.
* The likelihood is the probability of having the document given the class. Since a document is made of words, it can be broken down into individual words (terms/features/tokens).

A black and white math symbol

Description automatically generated with medium confidence A black and white math equation

Description automatically generated A mathematical equation with numbers and symbols

Description automatically generated

* To make a classification, we select the class with the highest probability.

A black text with black letters

Description automatically generated with medium confidence A black text on a white background

Description automatically generated

* **Laplace Smoothing:** To prevent the product of conditional probability from going to zero due to absence of a word, we add 1 to all term counts.

A black and white math equation

Description automatically generated

**3.1. ASSUMPTIONS: BAG OF WORDS MODEL**

* **Conditional Independence:** Words are independent of each other given the class, and do not influence each other
* **Positional Independence:** Words occurring in different positions of a document have the same conditional probability.
* **Utilizing a SQL approach with dplyr library:** Since text classification can deal with large datasets, we decided using a SQL approach would be optimal where using methods like select, filter, join, etc. would be efficient.
* This has the added advantage that it has easy translatability into other frameworks like SQL, Spark (SparkR), etc.

Figure 4: General Approach Used

**4. Building and Training Model**

* We use the cleaned, tokenized dataset that has been reduced after feature selection.
* Assign Priors and Conditional Probabilities to each term in each class.
* Use Laplace Smoothing on conditional probabilities.

A black and white math equation

Description automatically generated A black and white math equation

Description automatically generated

* We use the tibbles with conditional probabilities and priors to get the corresponding probability in each word in our validation dataset. Get the log probability for each label in each document, choosing the one with the largest value.
* **Removing Documents with high number of new words:** If a document contains a high percentage (say 75%) of words outside of our vocabulary, they are ignored since the predictions made on them would not be reliable.

A green squares with black numbers

Description automatically generated

Figure 5: Valid vs Invalid Documents (Remove Percentage = 0.25)

A comparison of a green rectangle with black numbers

Description automatically generated

Figure 6: Valid vs Invalid Documents (Remove Percentage = 0.5)

A comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of a comparison of

Description automatically generated

Figure 7: Valid vs Invalid Documents (Remove Percentage = 0.75)

**5. Comparing the Features on Various Metrics**

Normalization: Stemming vs lemmatization

STEM: 1, LEM: 0, FEAT: 0 (TIMING 1.2 seconds)

A group of green bars

Description automatically generated with medium confidence

Figure 8: Comparsion of Features when Stemming is Applied

STEM: 1, LEM: 1, FEAT: 0 (TIMING 5 minutes)

A group of green bars

Description automatically generated with medium confidence

Figure 9: Comparsion of Features when Stemming and Lemmatization is Applied

FEATURE SELECTION: WITH VS WITHOUT

STEM: 1, LEM: 0, FEAT: 0 (Timing : 1.2 seconds)

A group of green bars

Description automatically generated with medium confidence

Figure 10: Comparsion of Features when Stemming is Applied and Feature is not applied

STEM: 1, LEM: 0, FEAT: 1 (Timing : 2.4 seconds)

A group of green and white bars

Description automatically generated with medium confidence

Figure 11: Comparsion of Features when Stemming and Feature is Applied

NUMBER OF FEATURES: ACCURACY, F-1 SCORE ACROSS VALUES

STEM: 1, LEM: 0, FEAT: 1, TIES: 0

TOP\_K OPTIMAL: 100 (8%), 500 (41%), 750 (60%)

( Out of 7417 features in vocabulary )

A graph of a number of features

Description automatically generated A graph with a line graph

Description automatically generated Fig 11: Micro Accuracy vs Number of Features Fig 12: Macro Accuracy vs Number of Features

A graph with a number of features

Description automatically generated

Fig 13: Macro F-1 Score vs Number of Features

STEM: 1, LEM: 0, FEAT: 1, TIES: 0

TOP\_K OPTIMAL: 20000 (28%), 35000 (49%), 50000 (70%)

( Out of 142525 features in vocabulary )

A graph with a number of features

Description automatically generated

Fig 14: F1 Score vs Number of Features

A graph with a line and a blue line

Description automatically generated with medium confidence

Fig 15: Accuracy vs Number of Features

**6. Results**

* The optimal set of parameters: ( Stem: 1, Lem: 0, Feat: 1, Ties: 0, Top\_k: 500)

A yellow and blue squares with white text

Description automatically generated

Fig 16: Heatmap: Validation Set

A group of green and white bars

Description automatically generated with medium confidence

Fig 17: Precision, Recall,Accuracy and F1 Score on Validation Set

**Running on the Test Set:**

A graph of a bar chart

Description automatically generated

Fig 18: Number of labels in Test Data After Prediction

**7. Conclusion**

The analysis of the model's performance reveals a promising foundation for accurately classifying statements across a spectrum of truthfulness. The model demonstrates a strong capability to distinguish between the most truthful statements ("Mostly\_True" and "True") and less truthful ones, as evidenced by higher precision and F1-scores in these categories. This indicates a solid understanding of key features that determine the veracity of statements.

Furthermore, the model's performance metrics indicate that there is a beneficial effect of feature inclusion up to a certain point. The rapid improvement in both accuracy and F1-score as the number of features increases up to around 50,000 underscores the model's capacity to leverage a substantial number of features effectively, which is a significant positive aspect of its current design.

The leveling off of performance gains beyond this point of feature inclusion suggests that the model has successfully captured the most salient patterns within the data, achieving an efficient balance between complexity and performance. This efficiency is an excellent indicator of the model's scalability and robustness.

**8. References**

[1] C. D. Manning, Chapter 13, Text Classification and Naive Bayes, in Introduction to Information Retrieval, Cambridge University Press, 2008.

[2] Fake News Content Detection, KAGGLE data set: https://www.kaggle.com/datasets/anmolkumar/fake-news-content-detection?select=train.csv

[3] Fake News: build a system to identify unreliable news articles https://www.kaggle.com/ competitions/fake-news/data?select=train.csv