CLASSIFIER FOR FAKE NEWS RECOGNITION USING MULTINOMIAL NAIVE BAYES

PAVAN MANCHIKATLA

OUTLINE

I. INTRODUCTION

II. DATASET

III. MULTINOMAL NAÏVE BAYES CLASSIFIER

IV. FEATURE SELECTION

V. PREDICTIONS AND VALIDATING MODEL

VI. RESULTS

VII. CONCLUSION

INTRODUCTION

- Fake news are 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 in order to detect and classify fake news.

DATASET

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
1	Health care reform legislation is likely to mandate free sex change surgeries.	health-care
2	The economic turnaround started at the end of my term.	economy,jobs
5	The Chicago Bears have had more starting quarterbacks in the last 10 years than the total number of tenured (UW) faculty fired during the last two decades.	education
0	Jim Dunnam has not lived in the district he represents for years now.	candidates-biography
2	I'm the only person on this stage who has worked actively just last year passing, along with Russ Feingold, some of the toughest ethics reform since Watergate.	ethics

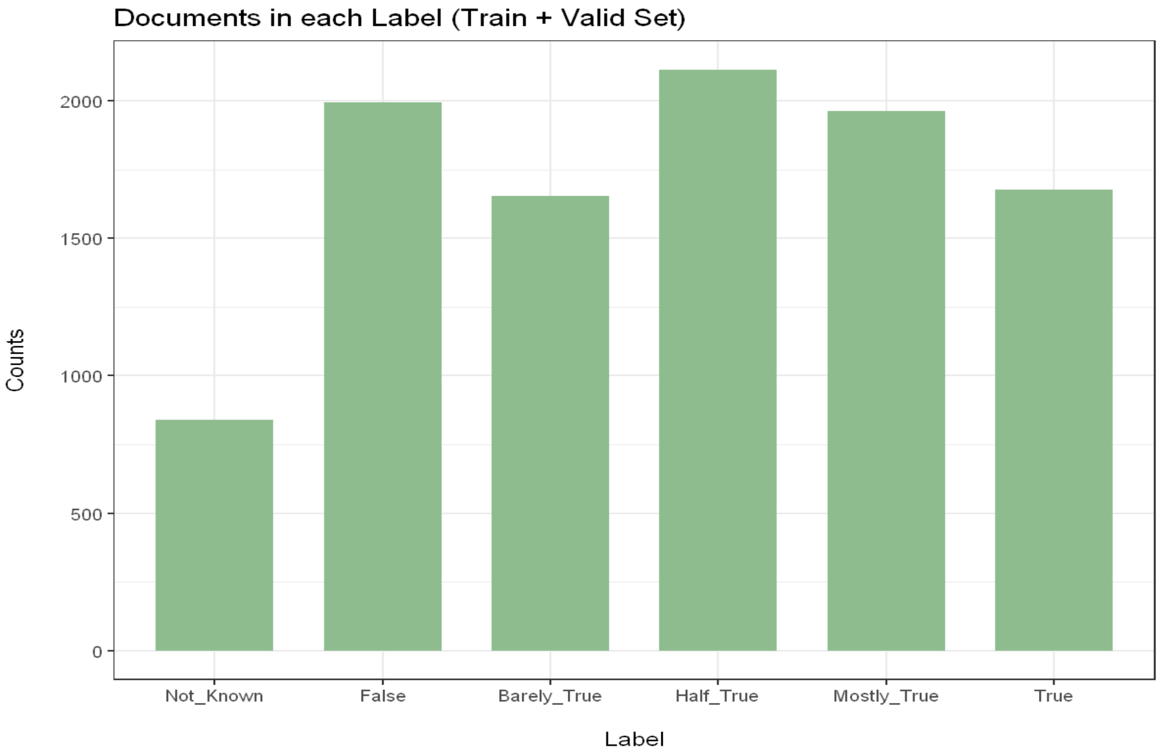
MODEL AND DATASET

Being able to distinguish fake contents form real news is today one of the most serious challenges facing the news industry. Naive Bayes classifiers are powerful algorithms that are used for text data analysis and are connected to classification tasks of text in multiple classes. 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.

- We have one dataset consisting of labelled documents.
- We need to build classifiers that are able to correctly predict the label of a given document.
- Naïve Bayes algorithm is a basic, yet effective model for a text classification model.



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).

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

$$\hat{P}(c) = \frac{N_c}{N},$$

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}},$$

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

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c).$$

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c)].$$

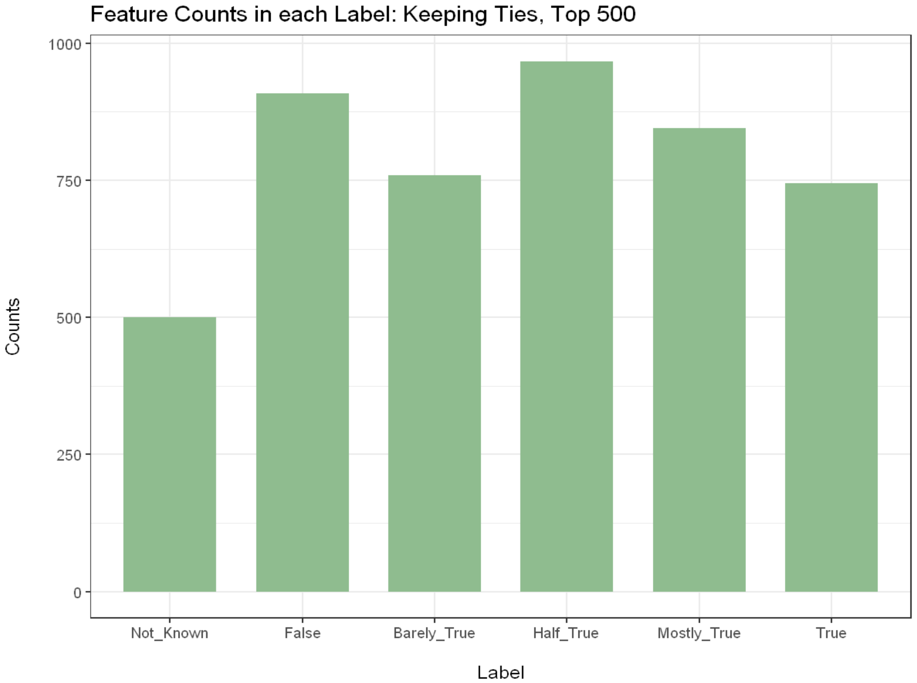
- **Laplace Smoothing:** To prevent the product of conditional probability from going to zero due to absence of a word, we add 1 to all term counts.

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B},$$

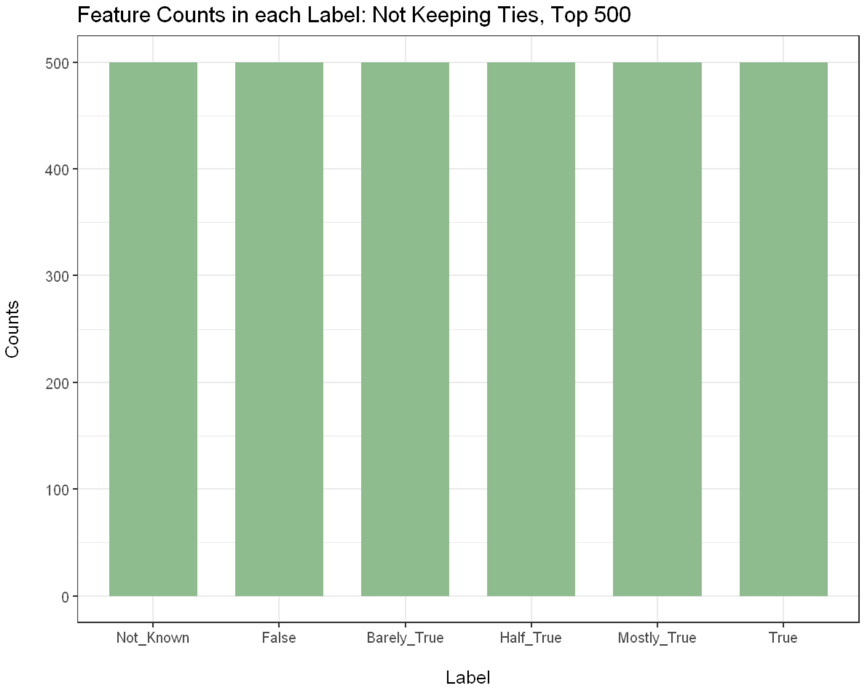
FEATURE SELECTION

- **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.

KEEPING TIES:



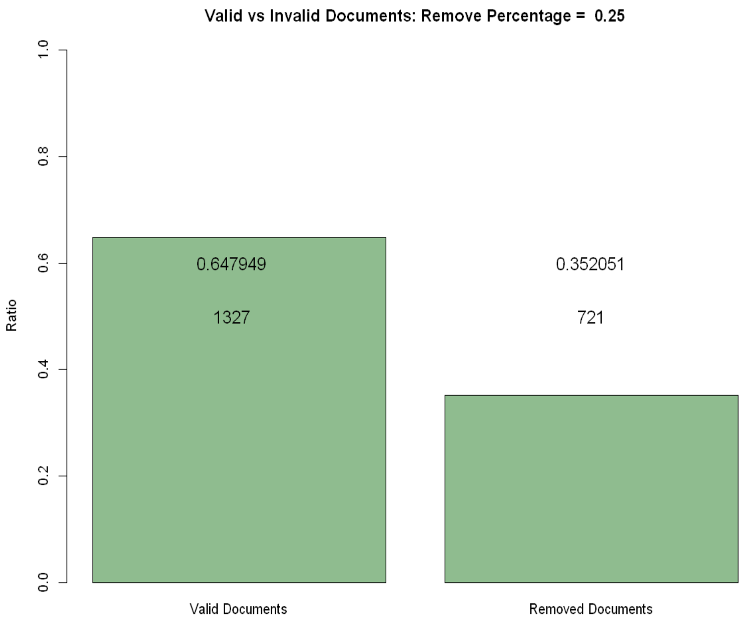
REMOVING TIES:

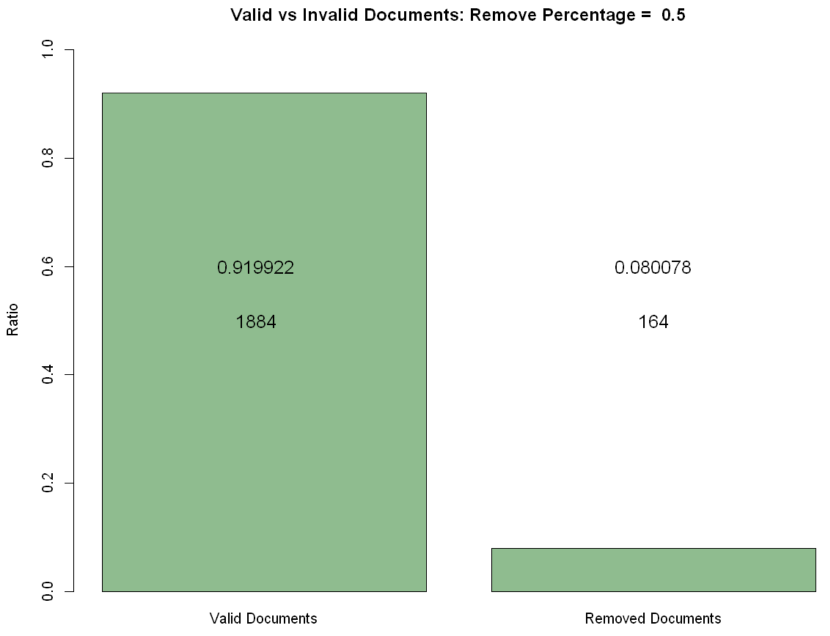


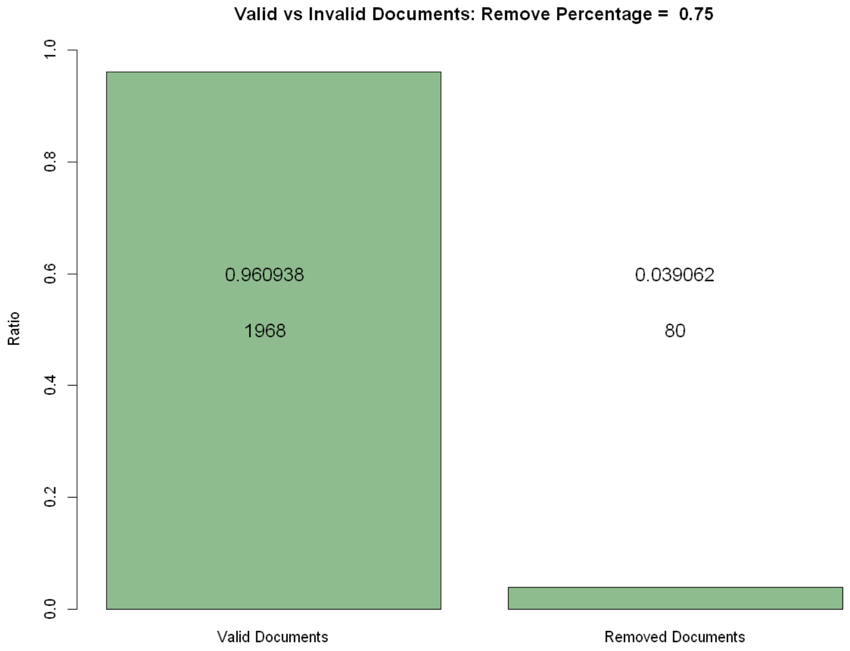
PREDICTIONS AND VALIDATING MODEL

○ 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.


Valid vs Invalid Documents: Remove Percentage = 0.25


Valid vs Invalid Documents: Remove Percentage = 0.5

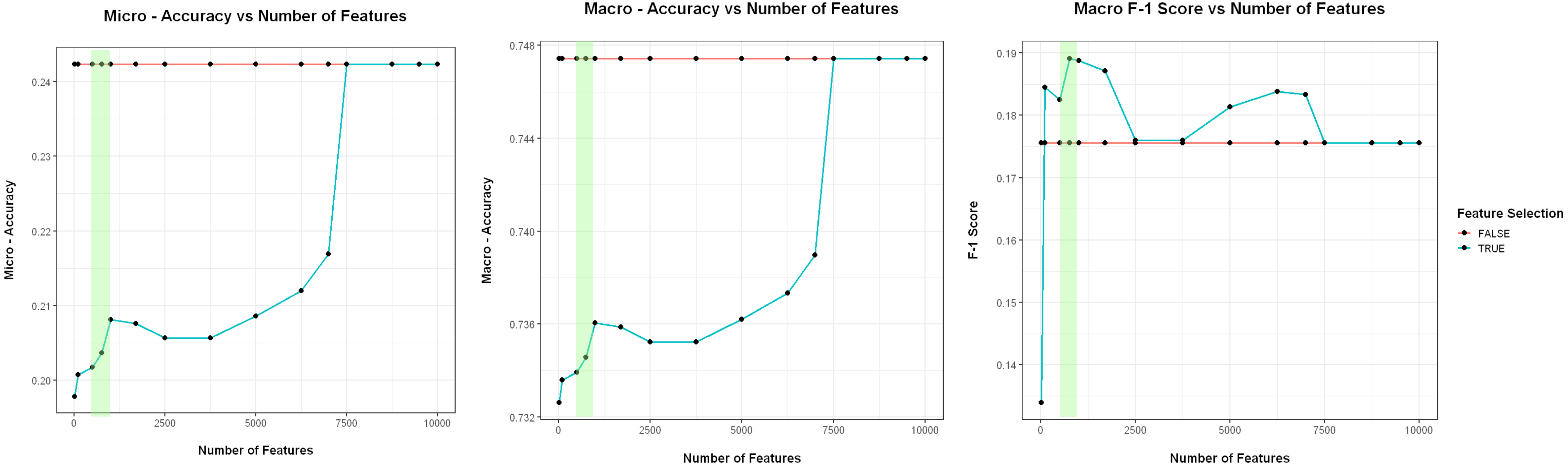

Valid vs Invalid Documents: Remove Percentage = 0.75

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)



Micro - Accuracy vs Number of Features

Macro - Accuracy vs Number of Features

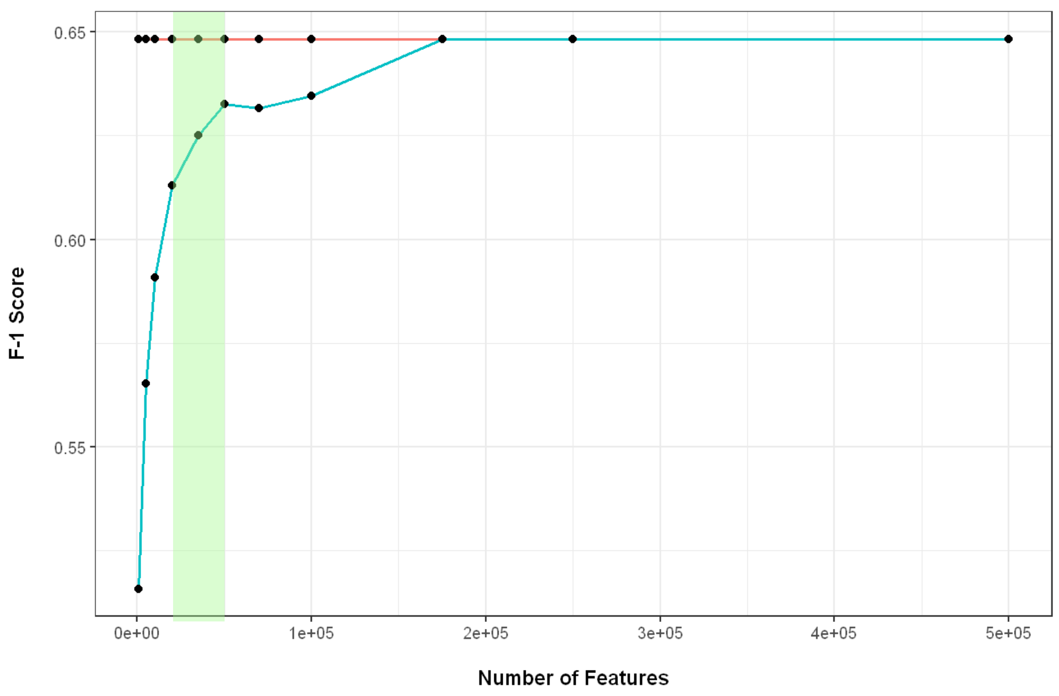
Macro F-1 Score vs Number of Features

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

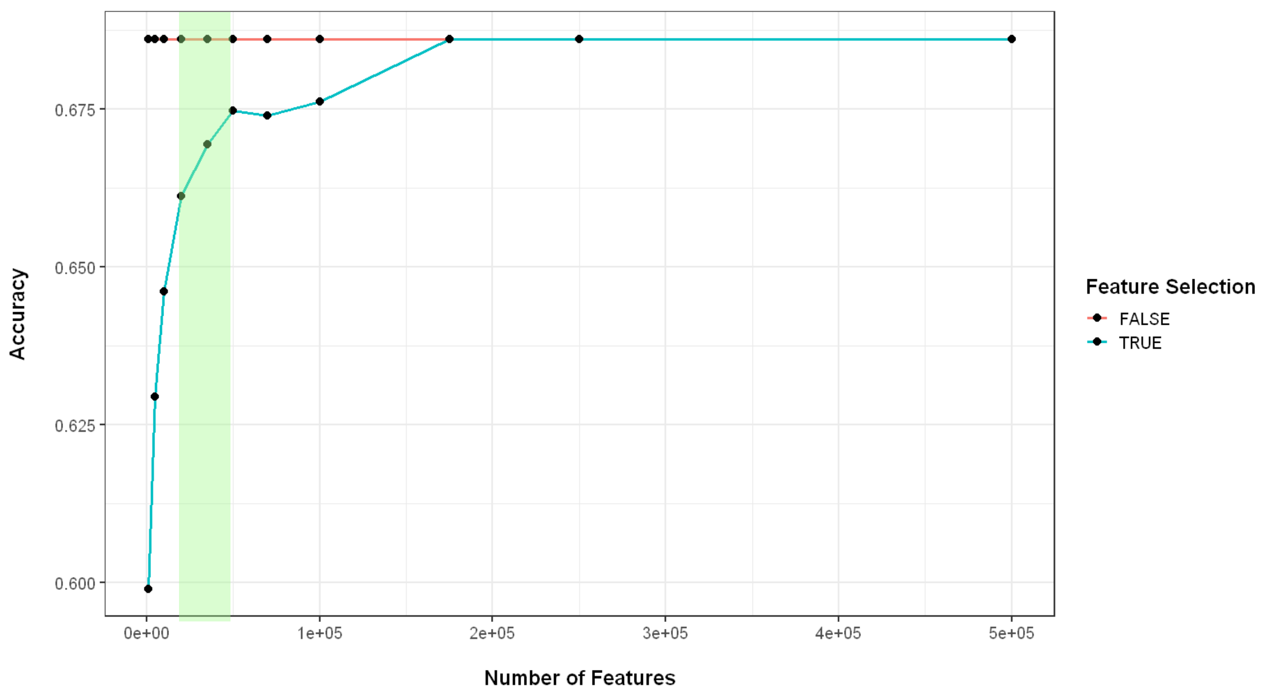
STEM: 1, LEM: 0, FEAT: 1, TIES: 0

TOP_K OPTIMAL: 20000 (28%), 35000 (49%), 50000 (70%)

(Out of 142525 features in vocabulary)



F-1 Score vs Number of Features

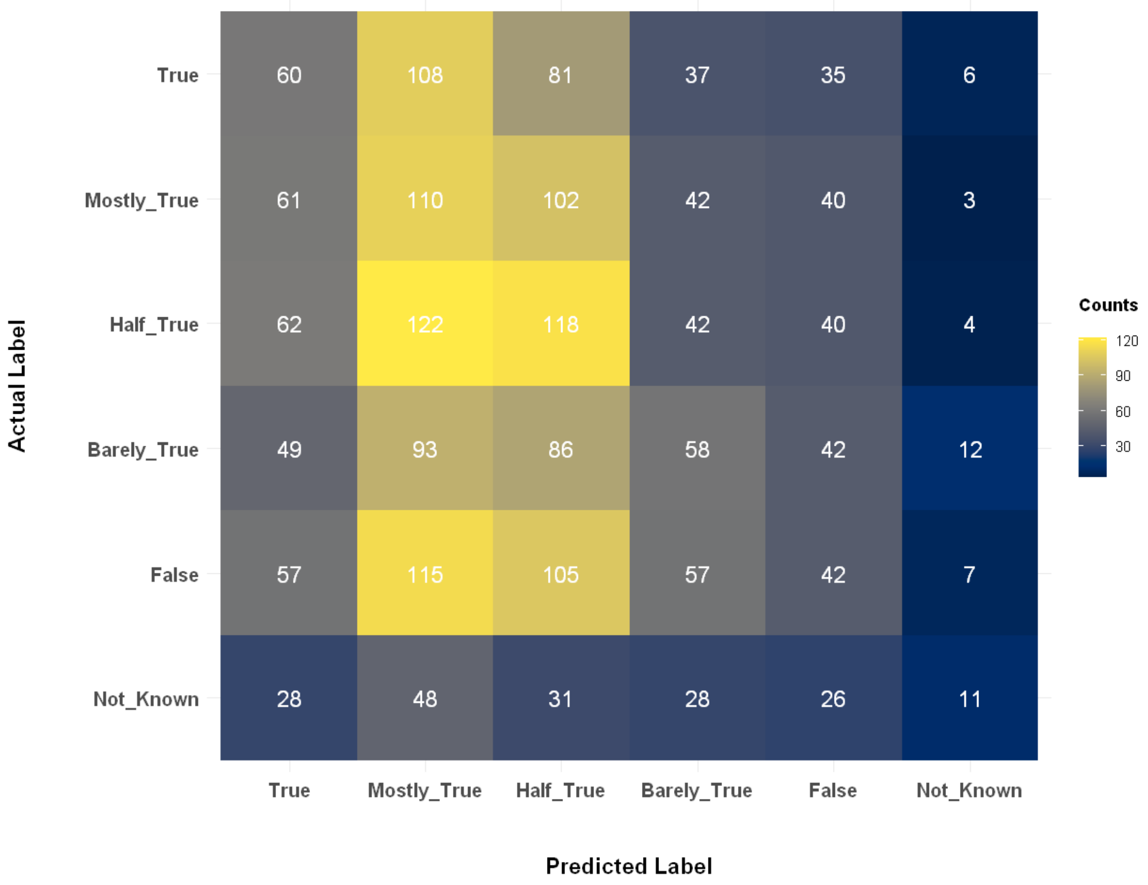


Accuracy vs Number of Features

RESULTS (SIX CLASSES)

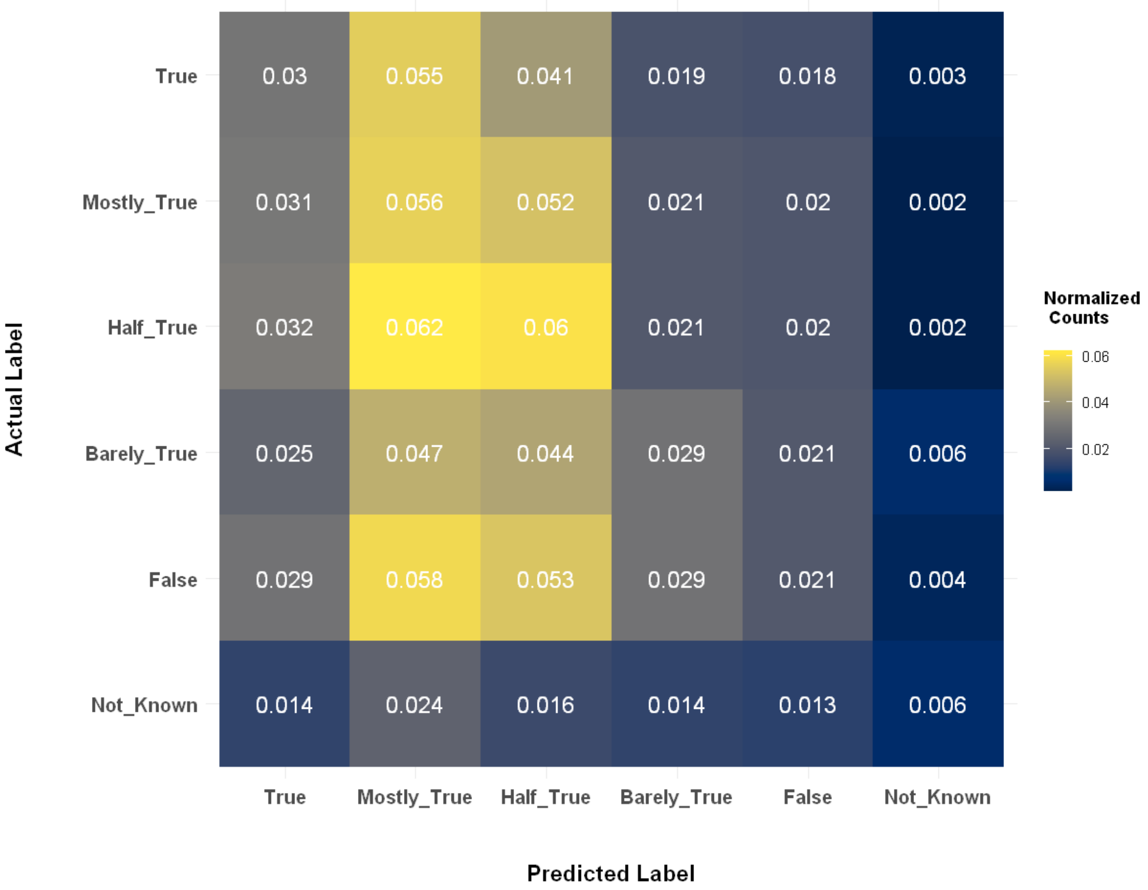
- We have optimal set of parameters: (Stem: 1, Lem: 0, Feat: 1, Ties: 0, Top_k: 500)

Heatmap: Validation Set



Actual Label \ Predicted Label	True	Mostly_True	Half_True	Barely_True	False	Not_Known
True	60	108	81	37	35	6
Mostly_True	61	110	102	42	40	3
Half_True	62	122	118	42	40	4
Barely_True	49	93	86	58	42	12
False	57	115	105	57	42	7
Not_Known	28	48	31	28	26	11

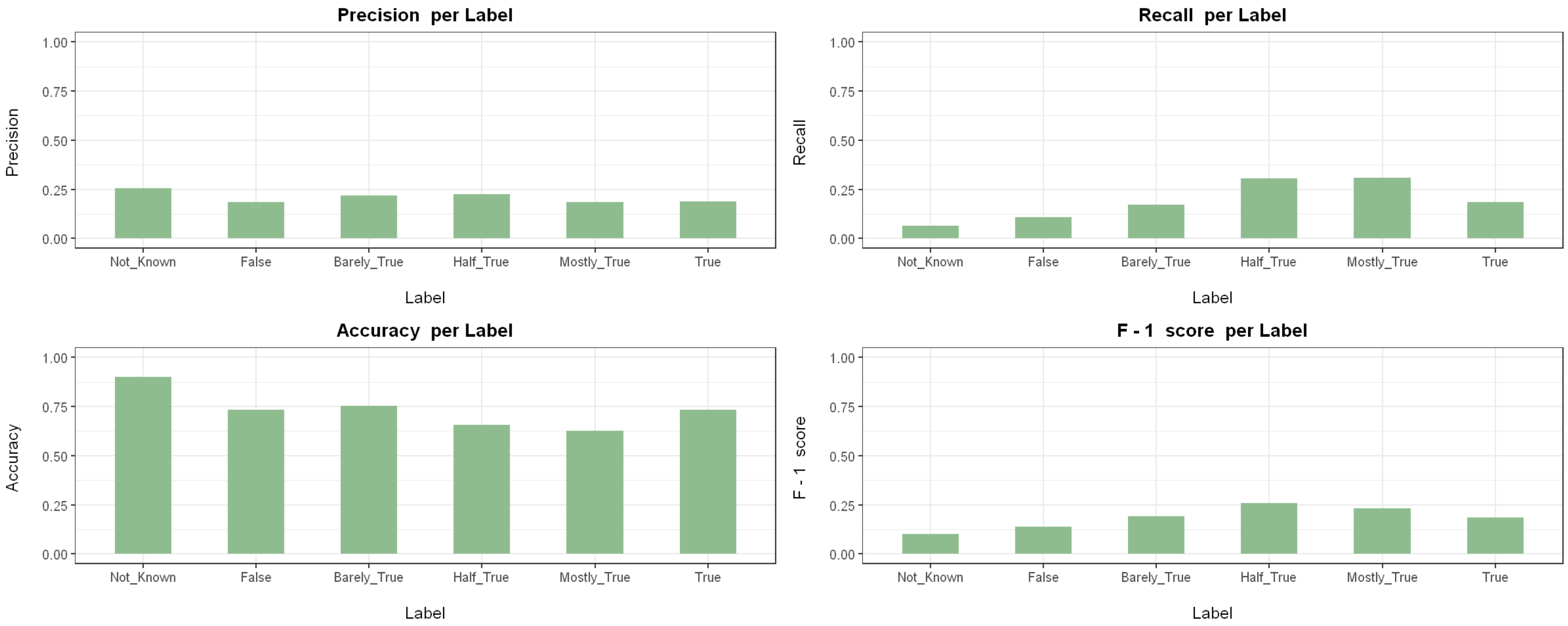
Heatmap: Validation Set



Actual Label \ Predicted Label	True	Mostly_True	Half_True	Barely_True	False	Not_Known
True	0.03	0.055	0.041	0.019	0.018	0.003
Mostly_True	0.031	0.056	0.052	0.021	0.02	0.002
Half_True	0.032	0.062	0.06	0.021	0.02	0.002
Barely_True	0.025	0.047	0.044	0.029	0.021	0.006
False	0.029	0.058	0.053	0.029	0.021	0.004
Not_Known	0.014	0.024	0.016	0.014	0.013	0.006

RESULTS (SIX CLASSES)

- We have optimal set of parameters: (Stem: 1, Lem: 0, Feat: 1, Ties: 0, Top_k: 500)


Precision per Label

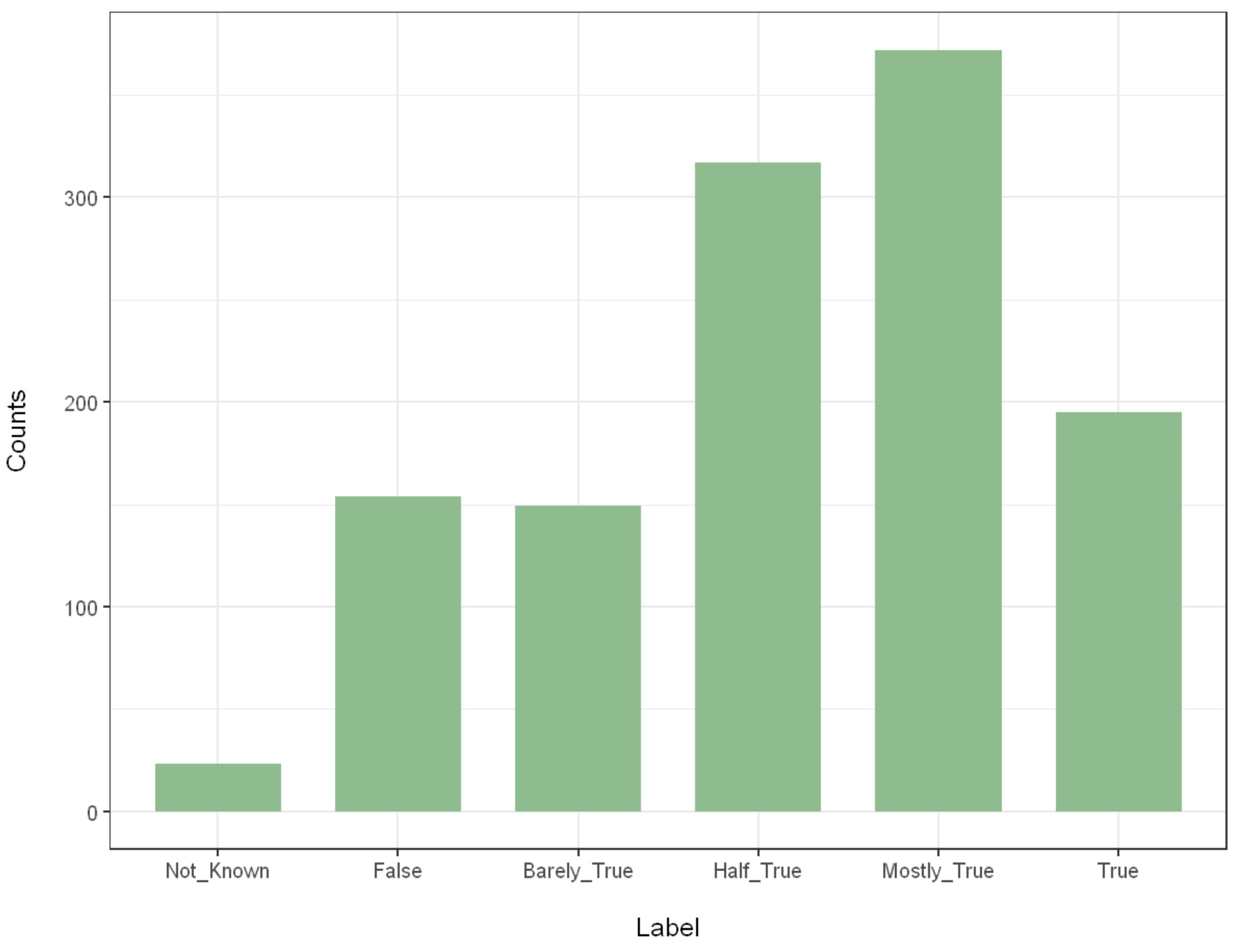
Recall per Label

Accuracy per Label

F - 1 score per Label

RESULTS

- **Running on the Test Set:**



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.

REFERENCE

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

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

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

THANK YOU