Case Study:Fake News Identifier

May 4, 2024

Abstract

The goal of the project is to try and identify fake news in a given dataset. In the world we live in today, there is a surplus of information and most of it is false. It is a necessity to be able to identify the false information as misleading. This was our inspiration for this project. We have chosen to use the dataset Liar for this task. The dataset has statements that have been manually labeled as true statements, false, half-truths, mostly-true, barely-true, and pats-fire. We explore the data, preprocess it, and apply the following classification algorithms: Logistic Regression, Decision Tree, AdaBoost Classifier with base Decision Tree, XGBoost Classifier, Voting Classifier.

1 Introduction

1.0.1 Motivation and application

The motivation behind our project was the surplus of information that we encounter daily in our lives. Social media and the internet have given voice to many untrust-worthy sources and it is becoming very hard to tell if certain news are true or not. A perfect example demonstrating the importance of verifiable news sources is the coronavirus pandemic. When the virus started to spread all over the world a surplus of fake information started flooding the internet. The virus was something new that was happening to the world and the lack of information gave the right opportunity to the creation of misinformation. This is one perfect example of why we need a fake news identifier. The big tech companies couldn't verify all the news out there individually, instead, they came up with an algorithmic solution that would flag some information as fake. A fake news identifier can be a very useful tool, and it can find application in many fields such as economics, politics, general public safety and much more.

1.1 Literature review

The work "Fake News Detection" by Amey Kasbe and Akshay Jain is a conference paper from 2018. This study applies Naïve Bayes classification model for finding fake news in social media as an application. Social media is the platform which has been sharing false information rapidly. There is so much information which is falsely wriiten on the social media or there is no legit source for it. In paper, the proposed approach was classifying postings as REAL or FAKE based on the frequency and kind of terms used, explains how Naive Bayes is utilized to forecast the post's legality. They used 11,000 news stories in their dataset, which has been classified as genuine.

A document named "DEFEND: Explainable Fake News Detection" investigates the creation of an explainable model that can identify fake news, especially on social media. This paper, written by Kai Shu and associates, solves the crucial requirement for transparency in the identification of false news by outlining the reasoning behind the designation of some news as fraudulent. In this research, a model called "DEFEND" was introduced, which detects and explains fake news by using user comments along with the content of news stories. A sentence-comment co-attention network in this model helps identify noteworthy sentences and valuable user comments that clarify and explain why a news item is considered false. Detailed testing demonstrates that DEFEND outperforms several advanced methods in detecting fake news.

In order to fight the rapidly spreading misinformation through blogs, social media posts, and online news outlets, Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea have published a paper titled "Automatic Detection of Fake News." This paper focuses on developing computational tools to identify fake news content online. In order to ease the detection of false news, this work represents two new datasets that span seven news categories. Additionally, it conducted a few experiments to investigate linguistic differences between fake and real news material. The authors outline the manual and crowdsourcing procedures they used to ensure a wide range of domains and accurate notes in the creation of these datasets. To create efficient false news detectors, they examine the linguistic characteristics of both true and fake news, with up to 78

The objective of the research "Supervised Learning for Fake News Detection" by Julio C. S. Reis and associates is to create methods for identifying false information on social media by examining various elements that have been taken from news reports. This comprises elements from the source, the news content, and the social media context in which it was posted. The researchers employed a sizable dataset of BuzzFeed news stories about the 2016 US election that had been margined by reporters and was elevated with Facebook users' comments, shares, and responses. Along with bringing some new features, they implemented a range variety of features that have been recommended by earlier studies. These attributes were divided into three categories: textual attributes (like language processing methods), news sources (like reliability and credibility).

The authors Kai Shu, Suhang Wang, and Huan Liu's paper "Beyond News Contents: The Role of Social Context for Fake News Detection" addresses how social context can improve the identification of fake news on social media, with an emphasis on the connections between news publishers, news items, and users. The study represents the TriFN framework, a tri-relationship embedding model that evaluates the credibility of users and publisher partiality to determine the accuracy of news. The framework seeks to offer a highly developed and successful method of detecting fake news by examining the circulating patterns and user characteristics of the news. Its effectiveness is demonstrated by notable gains in test results compared to conventional content-based approaches using real-world datasets.

By using language and network analysis techniques, Niall J. Conroy, Victoria L. Rubin, and Yimin Chen's paper "Automatic Deception Detection: Methods for Finding Fake News" explores the use of technology to detect false information. Network analysis looks at relationships and the flow of information inside networks to find misinformation, whereas linguistic cue techniques use tools like the Stanford Parser to evaluate text for patterns of fraudulent news. In order to improve the success of detection systems, the research proposes a hybrid strategy that combines various techniques. The authors argue that combining multiple detection techniques could significantly increase the accuracy of identifying fake news, addressing a crucial need in preserving the integrity of information in the digital age as information spreads quickly on social media and complicates traditional fact-checking.

1.2 Methodology

For the case study, we used the Liar dataset containing 14 attributes, and a total of 12788 instances. The dataset classifies statements as true statements, false, half-truths, mostly true, barely-true, and pats-fire. In order to convert the labeling into binary we classified the following statements that were labeled as false, barely-true, and pants-fire as false, and the least was classified as true. In the given dataset we used three classifiers using the majority voting rule: (1) Decision Tree, (2) Gaussian Naïve Bayes, and (3) Logistic Regression. Then compare the accuracy of the fused model with (4) AdaBoost Ensemble with Decision Trees as the base learner, and (5) Decision Tree, as well as with the XG Boost Classifier and Logistic Regression. Below you will find the logical order that we followed when working on the project:

- Exploring the data by creating the following visualizations :
  - Label Distribution
  - Feature distribution

    - Top 10 Most Used Words in Truthful Statements
    - Top 10 Most Used Words in Deceptive Statements (Without Stopwords and Punctuation) including true statements, mostly-true statements and half-true statement
    - Top 10 Speakers with the Most True Statements
    - Top 10 Speakers with the Most False Statements

- Preprocessing the data through Tokenization, Stop-words removal and Lemmatization. As well as text vectorization: TF-IDF, Bag of Words, N-gram Models, Hashing Vectorizer.

- Implementing the classification algorithms

2 Understanding the data

2.1 Exploring the data

Before implementing the classification methods is a good practice to understand the data you are working with, and make the necessary changes to get the best output. Some of the changes may include cleaning it, checking for missing data, replacing some values, and naming the features. All of the preparations of the data depend on the dataset that we decide to work on.

Important elements from the dataset:

- The Dataset consists of 14 features 12788 samples.

- The data consists of 9 object types, and 4 float64 type features.

- The object type named ”Label” takes 6 values, true statements, false, half-truths, mostly true, barely-true, and pats-fire.

- There are some missing values in the dataset.

Features of the dataset:

- Column 1: the ID of the statement ([ID].json).

- Column 2: the label.

- Column 3: the statement.

- Column 4: the subject(s).

- Column 5: the speaker.

- Column 6: the speaker’s job title.

- Column 7: the state info.

- Column 8: the party affiliation.

- Columns 9-13: the total credit history count, including the current statement.

- 9: barely true counts.

- 10: false counts.

- 11: half true counts.

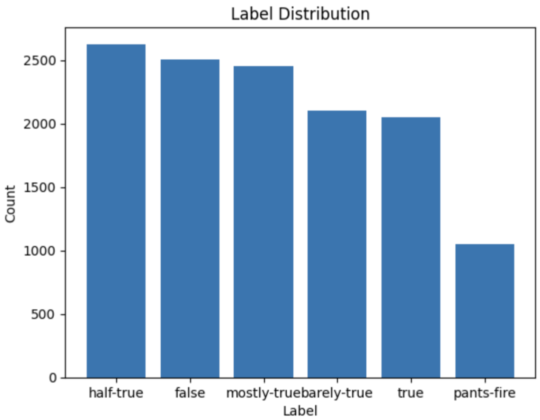
- 12: mostly true counts.

- 13: pants on fire counts.

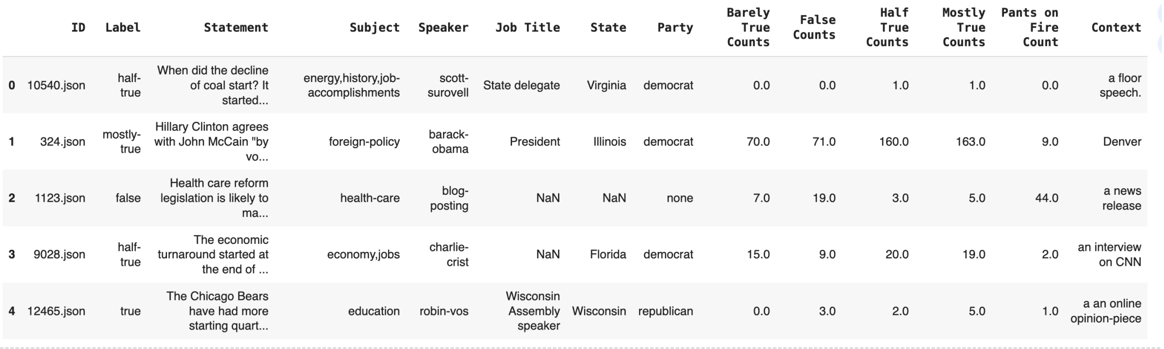
- Column 14: the context (venue/location of the speech or statement).

2.2 Exploring the dataset

The figure below visualizes the distribution of labels within a dataset, where the x-axis represents the different labels and the y-axis represents the count of each label.

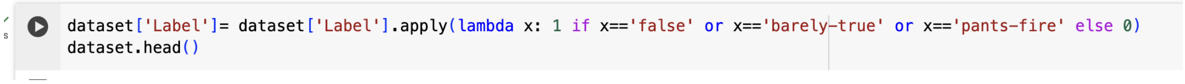


The figure below represents how the dataset looks



	ID	Label	Statement	Subject	Speaker	Job Title	State	Party	Barely True Counts	False Counts	Half True Counts	Mostly True Counts	Pants on Fire Count	Context
0	10540.json	half-true	When did the decline of coal start? It started...	energy,history,job-accomplishments	scott-surovell	State delegate	Virginia	democrat	0.0	0.0	1.0	1.0	0.0	a floor speech.
1	324.json	mostly-true	Hillary Clinton agrees with John McCain "by vo...	foreign-policy	barack-obama	President	Illinois	democrat	70.0	71.0	160.0	163.0	9.0	Denver
2	1123.json	false	Health care reform legislation is likely to ma...	health-care	blog-posting	NaN	NaN	none	7.0	19.0	3.0	5.0	44.0	a news release
3	9028.json	half-true	The economic turnaround started at the end of ...	economy,jobs	charlie-crist	NaN	Florida	democrat	15.0	9.0	20.0	19.0	2.0	an interview on CNN
4	12465.json	true	The Chicago Bears have had more starting quart...	education	robin-vos	Wisconsin Assembly speaker	Wisconsin	republican	0.0	3.0	2.0	5.0	1.0	a an online opinion-piece

We have assigned 1 to 'false'statements including the barely true and pants-fire and 0 to the rest in the dataset because it makes it easier for us to work with having them as integers. The replacement is seen in the following code:

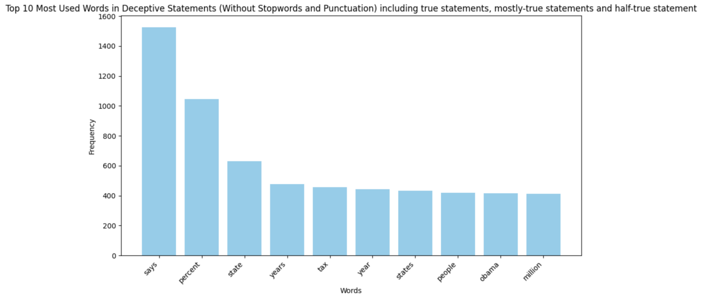


```
dataset['Label']= dataset['Label'].apply(lambda x: 1 if x=='false' or x=='barely-true' or x=='pants-fire' else 0)
dataset.head()
```

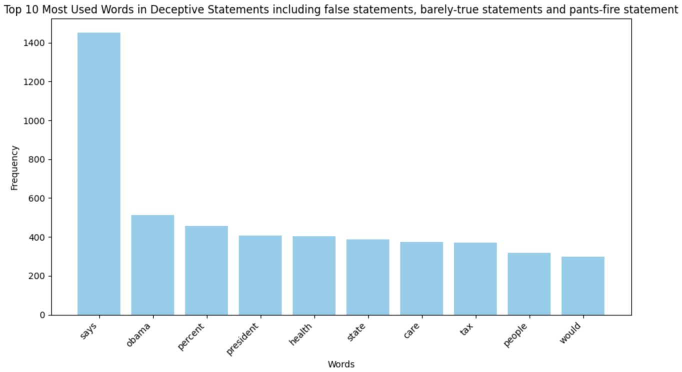
Here we can see the changes that happened to the "Label" column in the dataset.

Sometimes, certain words or phrases seem to go hand in hand with either telling the truth or telling a lie. For this same reason, we will analyze the most frequent words used in this database when people were saying the truth and when they were lying. By really getting into how people talk when they're lying compared to when they're being real, we can start to understand more about their habits. This kind of study helps psychologists and behavior experts, giving them a leg up in spotting when someone's trying to lie.

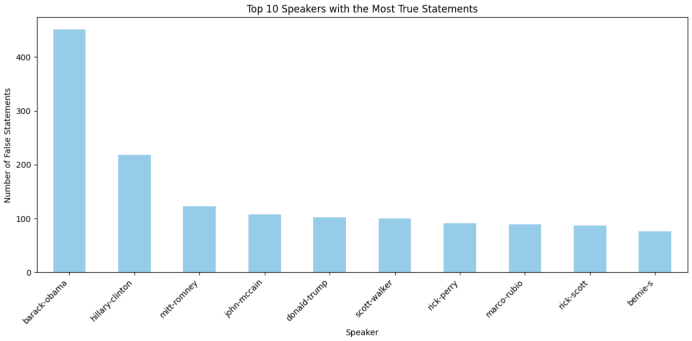
We have analyzed what are the most frequent words used when telling the truth



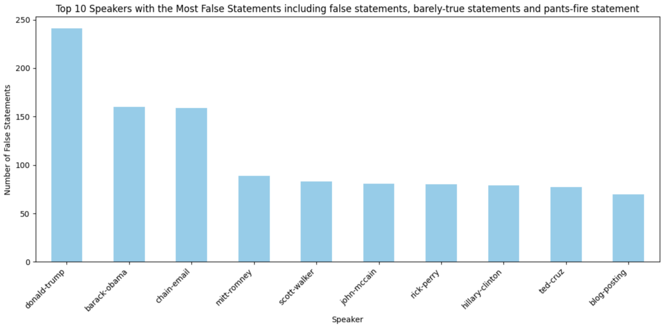
We have analyzed what are the most frequent words used when telling false statements



We have analyzed the most frequent speakers that have given more true statements



We have analyzed the most frequent speakers that have given more false statements

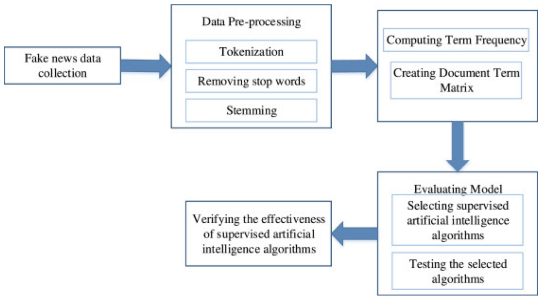


3 Leveraging Text Preprocessing Techniques for Fake News Classification

Methodology:

- Tokenization: The dataset is being tokenized using the word_tokenize function from the NLTK library. here the text is broken down into individual tokens.
- Stopword Removal: remove all the words that don't give us any important insights like is, are, have, etc.
- Lemmatization: is performed to reduce each word to its base or root form. This is done in order to normalize the text and reduce lexical variations.
- TF-IDF Vectorization: assigns weights to words based on their frequency in a document relative to the entire corpus, highlighting significant terms.

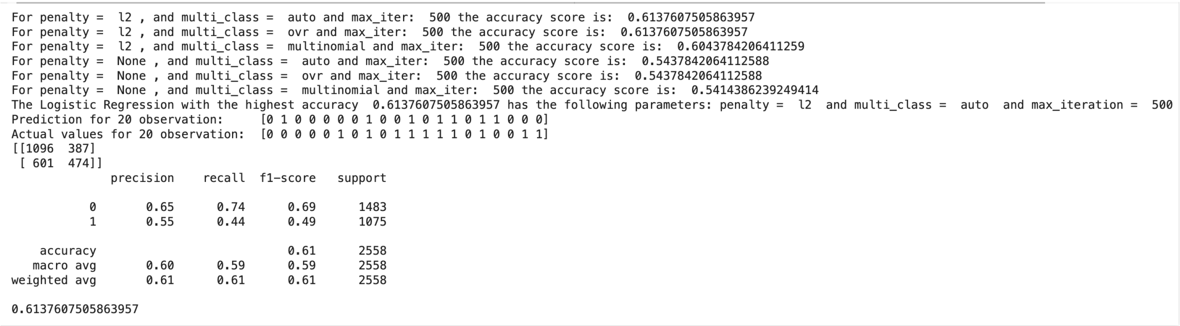
- Bag of Words (BoW) Vectorization: basically counts the word frequency and it a simple but powerful technique for classification.
- N-gram Models: grab groups of words that stick together, giving us more context and making our text representation richer. In our case we decided to use both unigram and bigram.
- Hashing Vectorization: implements a hashing function to map words to a fixed-length vector space.


Fake news data collection
Data Pre-processing
Tokenization
Removing stop words
Stemming
Computing Term Frequency
Creating Document Term Matrix
Evaluating Model
Selecting supervised artificial intelligence algorithms
Testing the selected algorithms
Verifying the effectiveness of supervised artificial intelligence algorithms

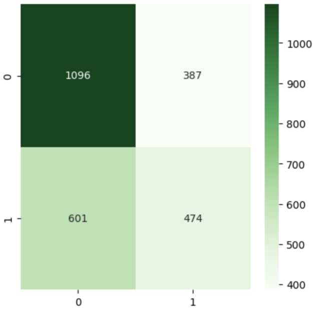
4 Implementing the classification algorithms

4.1 Logistic Regression

In this project, our goal was to explore various classification algorithms and evaluate their performance for text classification tasks using the Liar dataset. We applied the Logistic Regression model to different datasets. The four different datasets had different feature extraction methodologies such as those mentioned above: TF-IDF, bag of words, N-grams, and Hashing Vectorization. Of all these 4 methods of extraction, the one that gave the highest accuracy was TF-ID. We got the following accuracy:

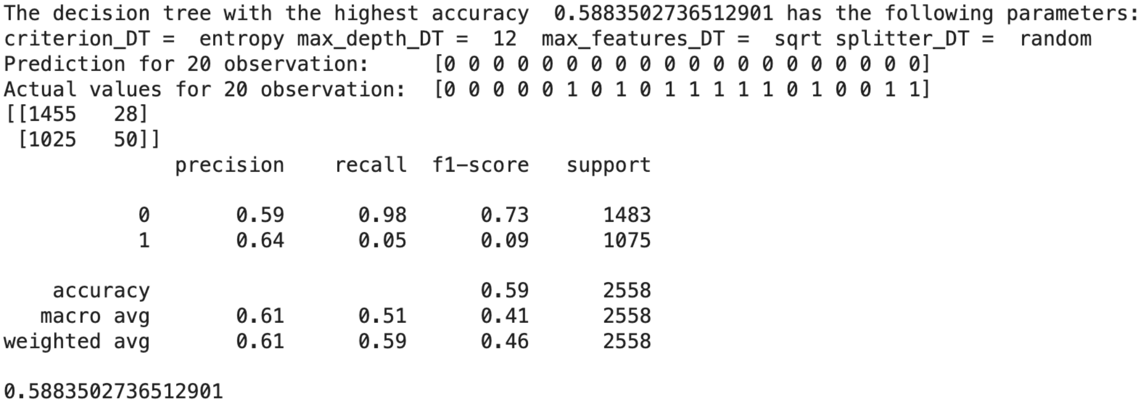


```
For penalty =  l2 , and multi_class =  auto and max_iter:  500 the accuracy score is:  0.6137607505863957
For penalty =  l2 , and multi_class =  ovr and max_iter:  500 the accuracy score is:  0.6137607505863957
For penalty =  l2 , and multi_class =  multinomial and max_iter:  500 the accuracy score is:  0.6043784206411259
For penalty =  None , and multi_class =  auto and max_iter:  500 the accuracy score is:  0.5437842064112588
For penalty =  None , and multi_class =  ovr and max_iter:  500 the accuracy score is:  0.5437842064112588
For penalty =  None , and multi_class =  multinomial and max_iter:  500 the accuracy score is:  0.5414386239249414
The Logistic Regression with the highest accuracy  0.6137607505863957 has the following parameters: penalty =  l2  and multi_class =  auto  and max_iteration =  500
Prediction for 20 observation:     [0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 0 0 0]
Actual values for 20 observation:  [0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1]
[[1096  387]
 [ 601  474]]
              precision    recall  f1-score   support

           0       0.65      0.74      0.69      1483
           1       0.55      0.44      0.49      1075

    accuracy                           0.61      2558
   macro avg       0.60      0.59      0.59      2558
weighted avg       0.61      0.61      0.61      2558

0.6137607505863957
```

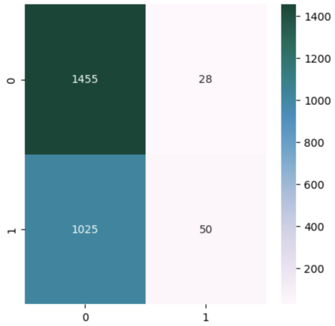


4.2 Decision Tree

We did hyperparameter tunning for the decision tree model as well, which included including criterion, splitter, max features, and max depth. These parameters are systematically varied through nested loops to explore different configurations. For each configuration, the decision tree classifier is trained on the different datasets. Of all these 4 methods of extraction, the one that gave the highest accuracy was TF-ID. Same as above. We got the following accuracy:

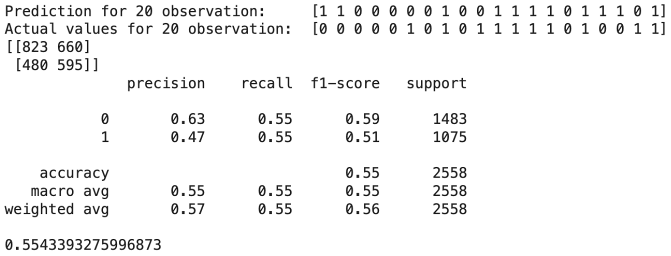


```
The decision tree with the highest accuracy  0.5883502736512901 has the following parameters:
criterion_DT =  entropy max_depth_DT =  12  max_features_DT =  sqrt splitter_DT =  random
Prediction for 20 observation:     [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Actual values for 20 observation:  [0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1]
[[1455   28]
 [1025   50]]
              precision    recall  f1-score   support

           0       0.59      0.98      0.73      1483
           1       0.64      0.05      0.09      1075

    accuracy                           0.59      2558
   macro avg       0.61      0.51      0.41      2558
weighted avg       0.61      0.59      0.46      2558

0.5883502736512901
```

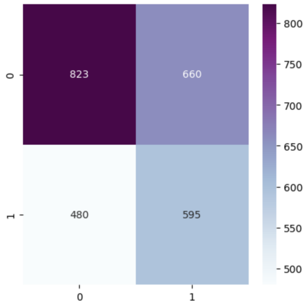


4.3 AdaBoost Classifier with base Decision Tree

For each configuration, the AdaBoost classifier is trained on the different datasets. Of all these 4 methods of extraction, the one that gave the highest accuracy was TF-ID. Same as above. We got the following accuracy:

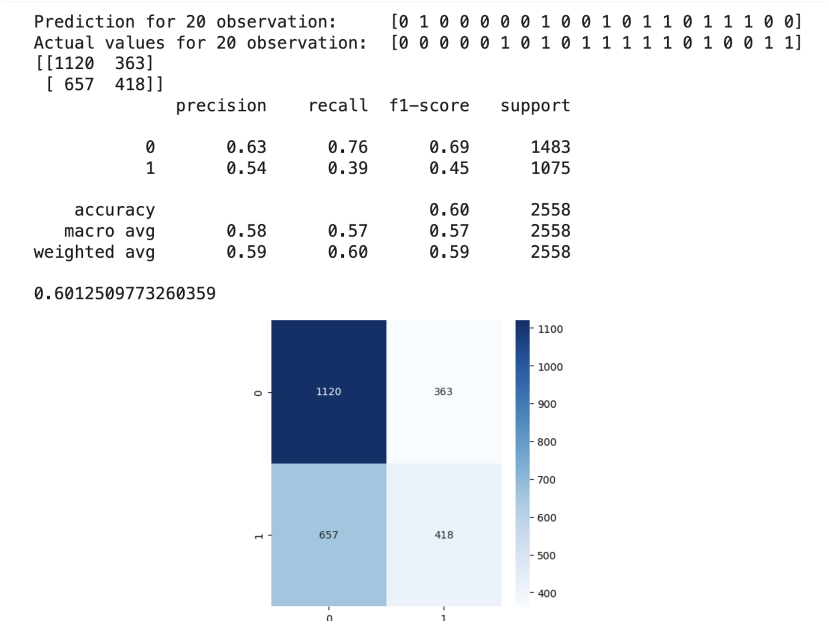


```
Prediction for 20 observation:     [1 1 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 1 0 1]
Actual values for 20 observation:  [0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1]
[[823 660]
 [480 595]]
              precision    recall  f1-score   support

           0       0.63      0.55      0.59      1483
           1       0.47      0.55      0.51      1075

    accuracy                           0.55      2558
   macro avg       0.55      0.55      0.55      2558
weighted avg       0.57      0.55      0.56      2558

0.5543393275996873
```



4.4 Voting Classifier

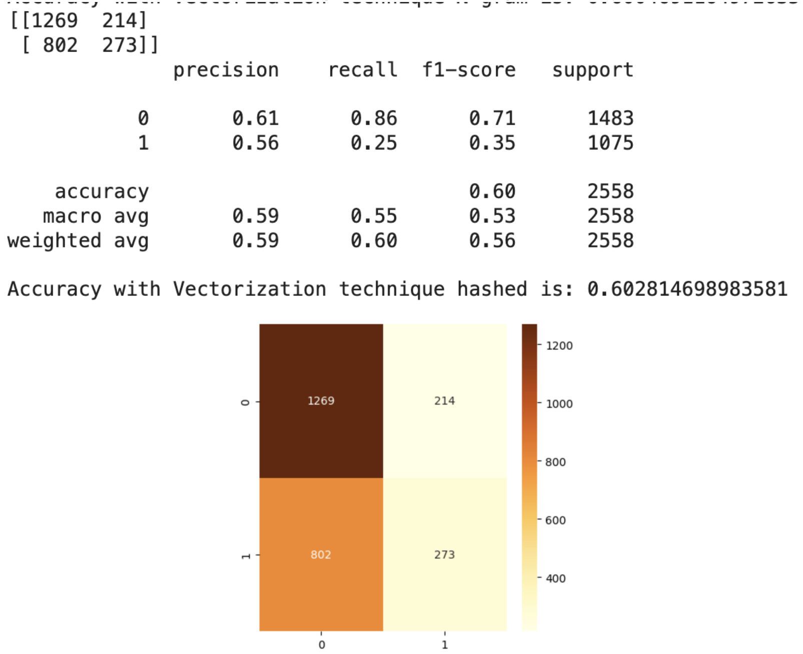
For each configuration, the Voting Classifier is trained on the different datasets. Of all these 4 methods of extraction, the one that gave the highest accuracy was Hashing Vectorizer. Same as above. We got the following accuracy:



```
Prediction for 20 observation:     [0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0]
Actual values for 20 observation:  [0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1]
[[1120  363]
 [ 657  418]]
              precision    recall  f1-score   support

           0       0.63      0.76      0.69      1483
           1       0.54      0.39      0.45      1075

    accuracy                           0.60      2558
   macro avg       0.58      0.57      0.57      2558
weighted avg       0.59      0.60      0.59      2558

0.6012509773260359
```

4.5 XGBoost Classifier

For each configuration, the XGBoost Classifier is trained on the different datasets. Of all these 4 methods of extraction, the one that gave the highest accuracy was Hashing Vectorizer. Same as above. We got the following accuracy:



```
[[1269  214]
 [ 802  273]]
              precision    recall  f1-score   support

           0       0.61      0.86      0.71      1483
           1       0.56      0.25      0.35      1075

    accuracy                           0.60      2558
   macro avg       0.59      0.55      0.53      2558
weighted avg       0.59      0.60      0.56      2558

Accuracy with Vectorization technique hashed is: 0.602814698983581
```

5 Conclusions and Problems

5.1 Conclusions

After applying various classification algorithms we came to the conclusion that logistic regression combined with TF-IDF achieves the highest accuracy among all configurations. There is not significant difference in accuracy between the different classification algorithms, it moves between them with a 0.5 difference in accuracy. With an accuracy of 61.38%, logistic regression with TF-IDF demonstrates its effectiveness in classifying between true and false statements in the Liar dataset.

For the Logistic Regression, we also tried to do hyperparameter tunning.

The logistic regression model with the highest accuracy is configured with the following parameters:

- Penalty: L2
- Multi-class strategy: Auto
- Maximum iterations: 500

We explored other metrics besides accuracy in evaluating the performance of the algorithms. For r the logistic regression model, we find that it performs reasonably well in predicting both classes (true and false statements). However, there is room for improvement, especially in terms of recall for class 1 (false statements), which is relatively lower compared to class 0 (true statements).

In conclusion, logistic regression with TF-IDF preprocessing gave us the highest accuracy but with a small difference from the other. If we were to go back and try again the project we would probably change the fact that we were only concentrated on the 'statement' feature, but instead we would give importance to other features as well. Maybe that would have impacted in getting a higher accuracy.

5.2 Problems

- Complexity of the Task. Before we took over the task of classifying news as true or fack, we really were not prepared for the difficulty of the task. The task turned out to be more challenging than expected and in the hindsight, we could have done some things differently, such as we could have tried to implement some deep learning algorithms. We might have gotten a higher accuracy than what we got.
- Limited Information: As mentioned above we had limited information for the task that we were undertaking and our lack of knowledge in implementing deep learning algorithms was a barrier to great more successful outputs.
- Computational Resources: Because of the nature of the project all the algorithms take a significant amount of time to be run. Time was a restriction for us,
- Project Management and Timeline: Setting realistic goals was a small problem for us. We could have managed that part better. The goals that we planned for were greater than our current skills.

6 Contributions

Team member

- Oksana Dura 1316268
- Paavankumar Deepak Chaudhari: 1329931
- Niyati HirenKumar Bhatt: 1339467
- Sai Surya Vadde: 1338915
- Sakhamuri Greeshma Sriram: 1330380

  The report was written in LaTex by Oksana, while everyone contributed to its content.

Contributions

- Exploring the data by creating the following visualizations (Worked by: Oksana, Niyati,Greeshma):
  - Label Distribution
  - Feature distribution
  - Top 10 Most Used Words in Truthful Statements

  - Top 10 Most Used Words in Deceptive Statements (Without Stopwords and Punctuation) including true statements, mostly-true statements and half-true statement
  - Top 10 Speakers with the Most True Statements
  - Top 10 Speakers with the Most False Statements

- Preprocessing the data through Tokenization, Stop-words removal and Lemmatization. As well as text vectorization: TF-IDF, Bag of Words, N-gram Models, Hashing Vectorizer. (Worked by Paavankumar, Sai)

- Implementing the classification algorithms (Worked by Oksana and Paavankumar)

7 References

- Gottfried Jeffrey and Shearer Elisa. 2016. News use across social media platforms 2016. In Pew Research Center Reports. http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '03, pages 423–430. https://doi.org/10.3115/1075096.1075150.

- Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, and Can Candan. 2016. caret: Classification and Regression Training. R package version 6.0-70. https://CRAN.R-project.org/package=caret.

- K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newslett., vol. 19, no. 1, pp. 22–36, 2017.

- C. Shao, G. L. Ciampaglia, O. Varol, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," 2017, arXiv:1707.07592.

- S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018.

- Essay: The Advantages and Disadvantages of the Internet. Available: https://www.ukessays.com/essays/media/the disadvantages-of-internet-media-essay.

- Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake News Detection on Social Media using Geometric Deep Learning. arXiv preprint arXiv:1902.06673 (2019).

- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)

- Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some Like it Hoax: Automated Fake News Detection in Social Networks. arXiv preprint arXiv:1704.07506 (2017).