CM-3070

Preliminary Report

Fake News Detection: Using Machine Learning to Combat Disinformation

Template: CM3060 Natural Language Processing

Table of Contents

1. A detailed introduction

Idea and Reasons Behind the Project:

The current time period, which is called the "Information Age," has made knowledge more available to everyone. But the fact that we can easily access information brings a problem that has never been seen before: the spread of "fake news." This widespread spreading of false news or lies has gotten to the point where it's hard for many people to figure out what's real and what's not.

Our project, "Fake News Detection," is meant to tackle this important problem head-on. The main idea behind our project is to build a complex system with advanced machine learning algorithms that can tell the difference between real and fake news with high accuracy. We want to make sure that news in the public sphere is true by making a system that separates real news from fake news automatically.

This project was started for many reasons. Our main goal is to stop fake news from spreading so it doesn't hurt the credibility of journalists and public discourse. At the same time, we want to protect people from any negative affects that could come from being exposed to false information. Also, by giving people a way to judge the reliability of news, we hope to help them make better decisions and create a more intelligent society.

Project Template:

Considering the nature of the problem we're trying to solve and the method we've chosen, the Natural Language Processing (NLP) Project Template seems to be the best fit for our work. Since it focuses on data processing, predictive modelling, and language-centric data analysis, it is the best choice for a problem that is mainly about understanding language. This plan will help us through important steps such as data collection, cleaning, feature extraction, model training and validation, and performance evaluation. Each of these steps is important for building an effective system for detecting fake news.

Similar Projects:

- "LIAR: A Benchmark Dataset for Fake News Detection" [1] is a study by William Yang Wang that aims to build a complete dataset for detecting fake news. It gives a solid basis but doesn't go into detail about recognition methods. This project will try to take advantage of this kind of information while also building and testing different recognition models.

- "FakeNewsNet" [7] is a project by Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu that is a large database for studying fake news on social media. The project is mainly about gathering and organising data, not finding ways to find things. Our project will take advantage of these rich data sources and try to improve identification methods. https://arxiv.org/abs/1809.01286

- "FakerFact" is a web tool and browser plugin that uses machine learning to classify the kind of website. Even though it gives a framework for how trustworthy news is, it doesn't say "fake" or "real." Our project will try to make spotting clearer and more accurate. https://www.fakerfact.org/ [8]

- "Full Fact" uses a combination of AI and human fact-checkers to verify public material. Even though automatic recognition is very successful, the fact that it needs human help shows that it could be better. This project will focus on improving machine learning models so that they don't need as much human help. Reference https://fullfact.org/ [9]

2. Literature Review

In this chapter, our goal is to undertake an extensive analysis of six influential pieces of literature within the realm of fake news detection. Through this endeavour, we aspire to gain a deeper comprehension of their distinct methodologies, achievements, and shortcomings. This understanding will enable us to spot the gaps that our project is designed to fill.

1.“Liar, Liar Pants on Fire”:A New Benchmark Dataset for Fake News Detection [1]

Summary

The LIAR dataset, 12.8K human-labelled short statements from POLITIFACT.COM, trains and develops computer models for false news detection and automated fact-checking. This dataset offers more detailed analysis and model development than earlier datasets. The report covers combining and relabeling duplicate labels to obtain six distinct truthfulness scores after a POLITIFACT.COM editor evaluated each statement in the dataset.

Democratic and Republican speeches and social media postings are included in the LIAR dataset. Speaker information includes party affiliation, current employment, home state, and credit history, as well as past tallies of false comments. This text and metadata are included into a deep learning model to enhance false news identification. Their hybrid convolutional neural network (CNN) outperforms existing text classifiers.

Critique

Pros:

- The LIAR dataset is larger than prior datasets, making it easier to design and test computer models for false news identification.
- Diverse settings, speakers, and information provide opportunity to create and test increasingly complex models that may evaluate statement veracity.
- A hybrid CNN that integrates text and metadata enhances false news identification.

Shortcomings:

- The study mentions metadata but does not thoroughly explain how to use it. The authors concatenate information and text representation to feed CNN, however improved integration approaches may improve outcomes.
- The dataset's vastness is a strength, although its major source, POLITIFACT.COM, may skew it. Diversifying data sources may provide a more balanced and complete dataset.
- Accuracy determines the authors' models. In circumstances with unbalanced classes, accuracy, recall, or F1-score may offer a better insight of the model's performance.
- Finally, the author briefly acknowledges the dataset's potential for NLP tasks including posture categorization, argument mining, and rumour identification but does not explore them. These routes might enhance LIAR dataset value in future study.

2. **Fake news detection based on news content and social contexts: a transformer-based approach** [2]

Summary

The work presents a deep learning-based false news detection method that uses news content and social context. The authors provide a framework including news, social situations, and detecting modules. They focus on early detection and label shortage to identify bogus news. The researchers modelled the detecting module's Transformer after BART's architecture. Encoder blocks learn representations and decoder blocks anticipate future behaviour based on prior observations. The researchers used a successful weak supervision labelling technique to address label scarcity.

Previous study has shown the importance of news content and social circumstances in false news identification. Its transformer-based design successfully integrates these two forms of information for prediction, offering a novel approach. The BART-inspired model and weak supervision labelling system used in this technique are a major advance in false news identification.

Critique

- Innovation and thoroughness are the study's merits. A BART-inspired Transformer model can identify bogus news early by predicting based on historical observations. The model's use of news content and social circumstances enriches false news detection. A poor supervision labelling strategy also solves this field's label scarcity problem.

Shortcomings

- The research has certain drawbacks. The false news detection technique uses NELA-GT-19, a dataset of 260 news sources. The study may not apply to other languages, target audiences, or new languages not in the dataset.
- Previous study has revealed that Media Bias Fact Check ground truth source-level labels affect downstream observations. Future study should test models utilising false and mainstream news ground facts.
- Weak supervision for model training limits the study. This has cut model development time, although poorly marked data may restrict generalizability.
- Finally, to match Fakeddit users' interactions with the NELA-GT-19 dataset's news chronology, the research only uses a tiny subset of user profiles. The research's test situations restrict the study's applicability.

Improvement Ideas

- Research could Add additional news sources, languages, and audiences to the study dataset.
- Use diverse ground truth labels in model assessment to achieve wide applicability.

- Balance poor supervision in model training with alternative strategies to promote generalizability.
- To guarantee model applicability beyond the test situations, use comprehensive data sets.

3. Fake News detection Using Machine Learning [3]

Summary

"Fake News Detection Using Machine Learning"[3] by Nihel Fatima Baarir and Abdelhamid Djeffal examines the rising problem of false news and provides a machine learning approach to identify it. The authors integrate and preprocess two datasets of false and authentic news to extract significant elements. They use bag of words, N-grams, and TF-IDF for feature extraction and the SVM classifier.

The authors tested feature extraction settings and methods to achieve 100% dataset recognition. They study how text, author, source, and date factors affect false news detection accuracy. The report recommends further research with greater datasets and model updates.

Critique:

The study contributes to false news identification using machine learning. Paper strengths include:

- Comprehensive approach: Bag of words, N-grams, and TF-IDF for feature extraction and SVM for classification. This holistic examination of news content helps false news identification.
- Social contexts: The article incorporates source, author, and date to identify bogus news. The suggested approach may improve detection by using social signals.
- Experimental evaluation: The writers test the system. They present thorough findings and assess how different characteristics and strategies affect fake news detection accuracy. This empirical assessment supports their conclusions.

There are several areas for improvement:

- Social circumstances are important, but the research doesn't explain how they're used or how they affect detection. More social cue analysis might improve the article.
- Dataset limitations: The authors blend two datasets, although the size and sources may restrict generalizability. A bigger, more diversified dataset from multiple fields and sources would improve the suggested system's validity and application.
- Comparative analysis: The research does not compare current false news detecting systems. Comparing the suggested system to other cutting-edge methods would help evaluate its performance and benefits.
- False positives and negatives: The study discusses accuracy rates but not false positives and negatives. Given the ramifications of misclassifying news as false or genuine, it's crucial to examine and assess the system's accuracy, recall, and F1 score.

In conclusion, the article contributes to false news identification, but fixing its flaws will enhance its conclusions. Improve the research's quality and effect by discussing social circumstances, using bigger and more varied datasets, comparing with current approaches, and analysing the system's false positives and negatives.

4 .Fake News Detection [4]

Summary:

The Naive Bayes classifier is used to identify bogus news on Facebook. The authors develop a technique that analyses news item titles and content to detect if a Facebook post is false. Web scraping updates the dataset with current news stories, improving the model's false news detection. The findings show that Naive Bayes-based machine learning can identify bogus news.

Critique:

Pros:

- Clear Problem Statement: The paper tackles social media false news concerns and presents a clear problem statement.

- Appropriate Methodology: The Naive Bayes classifier has been frequently used in comparable research for text classification problems.

- Web Scraping Integration: Web scraping to refresh the dataset with latest news items improves the model's adaptability.

- The Area Under the Curve (AUC) statistic is used to assess the model's accuracy.

Flaws and Improvements:

- The research utilised 11,000 news stories. Adding more articles from different sources may increase the model's performance and generalizability.

- The research does not compare the suggested false news detection algorithm to other methods. Comparative study would reveal the model's efficacy compared to other methodologies.
- Biases and Dataset Quality: Journalists rated the dataset "REAL" or "FAKE." However, dataset biases and labelling quality must be addressed.

- Ethical Considerations: The article does not address the possible censoring or labelling bias of a false news detection system. Automated content filtering affects society and ethics.

- The article uses the Naive Bayes classifier and web scraping to identify bogus news, although it might be improved. Future study should increase the dataset, undertake comparative assessments, correct biases, and evaluate the ethical implications of such systems.

5.**Detecting COVID-19 Fake News Using Deep Learning** [5]

Summary

"Detecting COVID-19 Fake News Using Deep Learning"[5] handles COVID-19 pandemic false news. False information may weaken genuine facts, according to the authors. The Jaccard index between the title and text, polarity, and adjective frequency are used in a modified LSTM model to identify bogus news. The model is trained on 300 false and 300 actual news pieces, attaining an overall accuracy of 0.91 with F1 scores of 0.89 and 0.92 for real and fake news, respectively.

The study addresses false news detection research using CNNs, RNNs, and LSTMs. TraceMiner, FAKEDETECTOR, and others use network dispersion and embedding to classify bogus news. The authors suggest using their modified LSTM model to handle COVID-19-related false news.

The authors explain their COVID-19-related dataset. They got the pieces from trustworthy fact-checkers and false news sites. The dataset is diverse in sources and subjects. The study describes the data collecting procedure, including human checking of publications to confirm their falsity.

A naïve Bayes classifier, three-layer neural network, and LSTM are baselines in the techniques section. Their customised LSTM outperforms these baselines. They show how adding false news' linguistic and stylistic qualities improves the model.

Results and discussion show the model's accuracy and F1 scores. The authors address extending the dataset, adding characteristics, and identifying bogus news on social media. Their study combats the "second pandemic" of bogus news during the COVID-19 epidemic, they conclude.

Critique:

Pros:

- Relevant and Timely Topic: Fake news about the COVID-19 epidemic has had serious effects for public health and safety.

- Novelty: The authors offer a modified LSTM model with COVID-19 false news detecting capabilities. It enhances field research.

- Dataset Collection: The writers utilise credible fact-checking sites and manually verify articles to ensure the inclusion of verifiably fake news.

- Performance assessment: The authors compare their model to many baselines and present extensive assessment criteria to show its superiority.

- Future Directions: The report suggests expanding the dataset, adding characteristics, and extending the algorithm to identify social media false news.

Flaws and Improvements:

- The authors recognise that their tiny dataset may restrict the model's generalizability. The dataset should contain additional articles from varied sources in future research.

- Lack of External Validation: The study cites manual verification and cross-referencing with trustworthy media sources but does not explicitly validate the dataset or model's predictions against a wider ground truth.

- Feature Selection: The study adds features to increase model performance, although the reasoning and selection procedure should be challenged. Explaining why particular characteristics were picked and their efficacy is helpful.

- Interpretability: Deep learning models like LSTMs lack interpretability. The study might assist identify false news by discussing the model's decision-making process and linguistic and stylistic clues.

- The material aids COVID-19 false news detection. It's new and suggests improvements. Addressing these issues will improve the study and its effect.

6.Fake news detection in the Hindi news dataset [6]

Summary

Machine learning is used to identify Hindi false news. Naive Bayes, logistic regression, and LSTM are used to preprocess, feature extract, and classify a Hindi news dataset from diverse sources. LSTM performs best at 92.36%.

The literature review covers false news detecting methods. In the era of social media, false news is important, and social bots, trolls, and cyborgs spread it. For false news identification, the paper emphasises Naive Bayes, logistic regression, support vector machines, and multilayer perceptrons.

Critique:

- The document improves Hindi false news identification. It solves a major issue and uses machine learning to identify bogus news with great accuracy. LSTM, a deep learning algorithm, works well.

- The paper's extensive literature review gives the study context. It discusses false news' effects, contributors, and detecting methods. This clarifies the work's importance.

However, there are some improvements.

- First, the study needs a more complete discussion of dataset acquisition. How sources were chosen, data quality was assured, and biases in the dataset would be good to know.

- The study also doesn't explore the approach's shortcomings and biases. The dataset's possible biases, such as the news sources' representativeness or the imbalance between false and actual news pieces, should be examined. Addressing these constraints would boost credibility and generalizability.

- The article may also illuminate model interpretability. While accuracy is vital, knowing the elements and attributes that influence model categorization is useful. This would detect Hindi false news tendencies.

- Finally, the study might explore false news detection ethics. As false news detection algorithms are used, freedom of expression and algorithmic bias or censorship must be considered.

- The research improves Hindi false news identification and sheds light on machine learning approaches. However, fixing the issues and doing more in-depth analysis will improve the study.

In conclusion, the above literature sources explain contemporary fake news detection methods. They also reveal this domain's limits. We can use these gaps to build a better fake news detection system via our project.

3. Project Design

Project Synopsis:

Our project aims to make a machine learning model that can tell the difference between real and fake news. People who use digital tools to read news often are the ones who will benefit the most. The goal is to give these people a reliable way to tell the difference between real news and false information. We want to stop the spread of fake news by putting in place an automatic, accurate, and effective system.

Reasons for Design Choices:

This project's design choices are based on how to use machine learning methods, which are well-known for their ability to find trends and make predictions. Their success in text sorting jobs makes them perfect for finding fake news. By combining Natural Language Processing (NLP) with machine learning, we can analyse the news material and look for signs of fake news.

Project Framework:

The project is broken up into the following main steps:

- **Data Collection and Preprocessing**: We will collect a large set of news stories that have been marked as "real" or "fake." The data will be preprocessed to get rid of noise and unnecessary information, such as punctuation, special characters, and stopwords. We'll also use normalisation methods for text, like stemming and lemmatization.

- **Extraction of Features:** After data preparation, we will use methods like TF-IDF, Bag of Words, and word embeddings to extract features from the cleaned-up text. Our machine learning model will use these traits as inputs.

- **Model Training**: The collected features will be used to train our model with different machine learning methods, such as logistic regression, SVM, random forest, and neural networks. Based on these traits, the model will learn to tell the difference between fake and real news.

- **Performance Evaluation**: The accuracy, precision, memory, and F1-score will be used to judge how well the model works. This step will help us figure out which model will work best for our job.

Methods and technologies :-

The most important technologies in this project are:

- **Python**: We'll use Python because it has a lot of tools for data analysis, machine learning, and handling natural language.

- **Natural Language Processing (NLP)**: NLP methods will be used for text preparation, feature extraction, and possibly mood analysis. In this situation, libraries like NLTK, SpaCy, and Gensim will be very helpful.

- **Machine Learning Libraries**: We will use Scikit-learn, TensorFlow, and maybe PyTorch to build and evaluate models. There are a lot of functions for machine learning jobs in these packages.

Project Timeline

A detailed project plan will be made, showing when each part of the project will happen. This schedule will make sure that the project goes according to plan and meets its goals. Tools for managing projects, such as github projects / notion, will be used to show the schedule and track the project's progress.

Task	Start Date	End Date
Model Design and Development	July 15, 2023	August 14, 2023

Model Testing	August 15, 2023	August 22, 2023
Model Evaluation	August 23, 2023	August 30, 2023
Model Deployment	August 31, 2023	September 7, 2023
Performance Monitoring	September 8, 2023	September 10, 2023

Strategy for Testing and Evaluating

The efficiency of the end model will be checked with a test sample that the model has never seen before. We will use different measurements to figure out how well the model works, such as:

- Accuracy: This number measures how often the model makes accurate predictions.
- Precision: This measure figures out how many true positive guesses there are out of all the positive predictions.
- Recall (Sensitivity): This metric measures how many true positives the model finds out of all the real positives it finds.
- F1-score: The F1-score is the harmonic mean of accuracy and memory. When the class distribution isn't even, it gives a fair measure of how well someone knows the material.
- A confusion grid will also be used to get a more clear and thorough look at how well the model works. It will show how many guesses were right, wrong, true positive, true negative, and fake positive.

4. Feature Prototype

This section talks about the first steps of making the text preparation and feature extraction method, which is an important part of our project. This feature is important to the success of the machine learning model because it turns raw news data into organised, useful information that helps the model learn.

How Important Is the Feature?

In any Natural Language Processing (NLP) project, the step of text preparation and feature extraction is very important. Since our raw data is in the form of unorganised text, it needs to be cleaned up and pre-processed before it can be turned into numbers. Then, our machine learning models can understand and make sense of these numbers, which makes learning easier. This process builds a strong basis for the model's accuracy and speed, which is important to the model's goal of spotting fake news.

How to put something into action:

For text preparation and figuring out what the text is about, our prototype uses Python tools like NLTK and Scikit-learn.

Text Preprocessing: The first step is to normalise the text by turning it all into lowercase. This makes sure that everything is the same. The next step is to get rid of punctuation, special characters, and stop words, which are common words like "and," "is," and "the" that don't add much to the sense of the text as a whole. The last step in this process is called lemmatization or stemming. This is when words are broken down to their roots. For example, the word "running" would become "run."

Feature Extraction: The cleaned text is turned into a number format after editing. For this change, we use the TF-IDF (Term Frequency-Inverse Document Frequency) method. The TF-IDF method shows how important a word is to a group or database of documents.

Dataset Usage and Balancing

Fake news corpus dataset :- https://github.com/several27/FakeNewsCorpus

Before getting into how the prototype code works, it's important to know that we only used a small part of a much larger dataset to test the prototype's performance. This method was needed because memory was limited, and it made sure that the programme would run faster, which is helpful during the testing process.

Also, it's important to point out that the collection is balanced. In the real world, records tend to be out of balance, with one type of data being much more common than the other. This difference can cause the model to be more accurate at predicting the majority class. To prevent this, we put in place a method for balancing datasets. We found the smallest number of classes and picked the same number of cases from each class to make sure the data was spread out fairly.

Explaining the Prototype Code in depth

The sample code can be broken down into several steps: data loading and preparation, text cleaning and lemmatizing, dataset balance, TF-IDF vectorization, and finally, applying the logistic regression model.

- Data Loading and Preprocessing: The data comes from a CSV file and is put into a pandas DataFrame. We only use a small part of the data. For faster processing, we only use the first million rows. Then, we get rid of columns that aren't important and keep only the "content" and "type" columns. We also get rid of news types that don't fit either the "fake" or "real" categories.

- Text Cleaning and Lemmatization: Next, the prototype cleans up and lemmatizes the news material. During the cleaning process, non-word characters are taken out, text is changed to lowercase, single letters are taken out, and any extra spaces are turned into single spaces. When the WordNetLemmatizer from the NLTK library does lemmatization, it cuts words down to their base or root form. For example, "running" becomes "run."

- Getting the Dataset in Balance: As we've already said, we then get the Dataset in Balance. To make a fair list, we figure out the minimum number of classes and pick the same number of cases from each class.

- TF-IDF Vectorization: The cleaned and lemmatized text is turned into number vectors using the Scikit-learn library's TfidfVectorizer. Based on the term frequency-inverse document frequency, these vectors show how important each word is in relation to the whole collection.

- Logistic Regression Model: The vectorized training dataset is used to run the logistic regression model from Scikit-learn on. The model is used to make predictions on the test set after it has been trained.

- Performance Metrics: The accuracy score and the uncertainty matrix are used to measure the model's performance.

Taking a look at the prototype

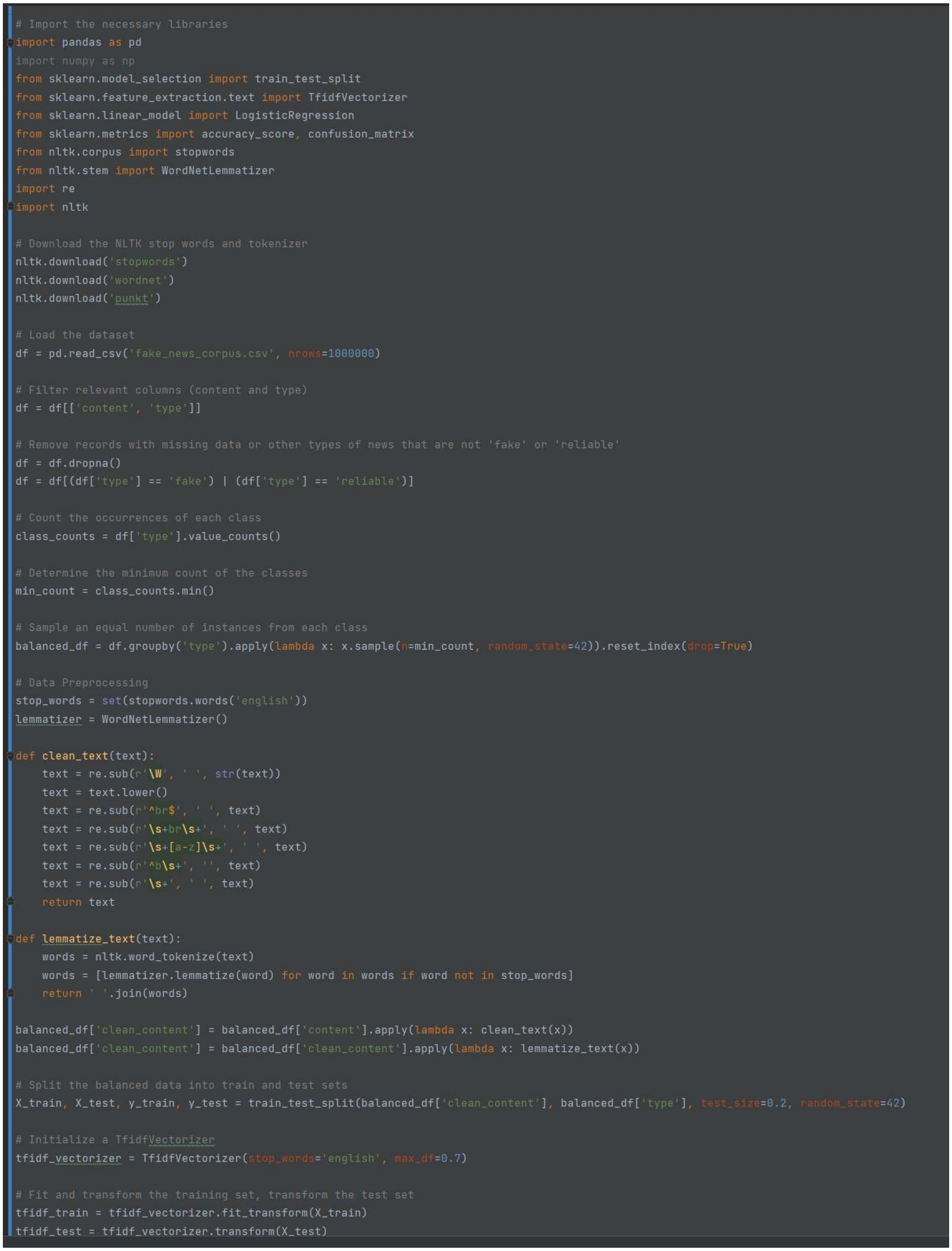
By looking at the quality of the final features, we can tell how well our text preparation and feature extraction system works. We assume that the preparation step will get rid of anything that isn't needed and break things down to their simplest form. The TF-IDF numbers, on the other hand, should correctly show how important each word is in the whole collection.

We expect to get a good classification after putting these traits into a basic machine learning model like logistic regression. This result suggests that the features help people tell the difference between real news and fake news. Still, we wouldn't do a full review until later, when we train and test our fully built machine-learning model using these features.

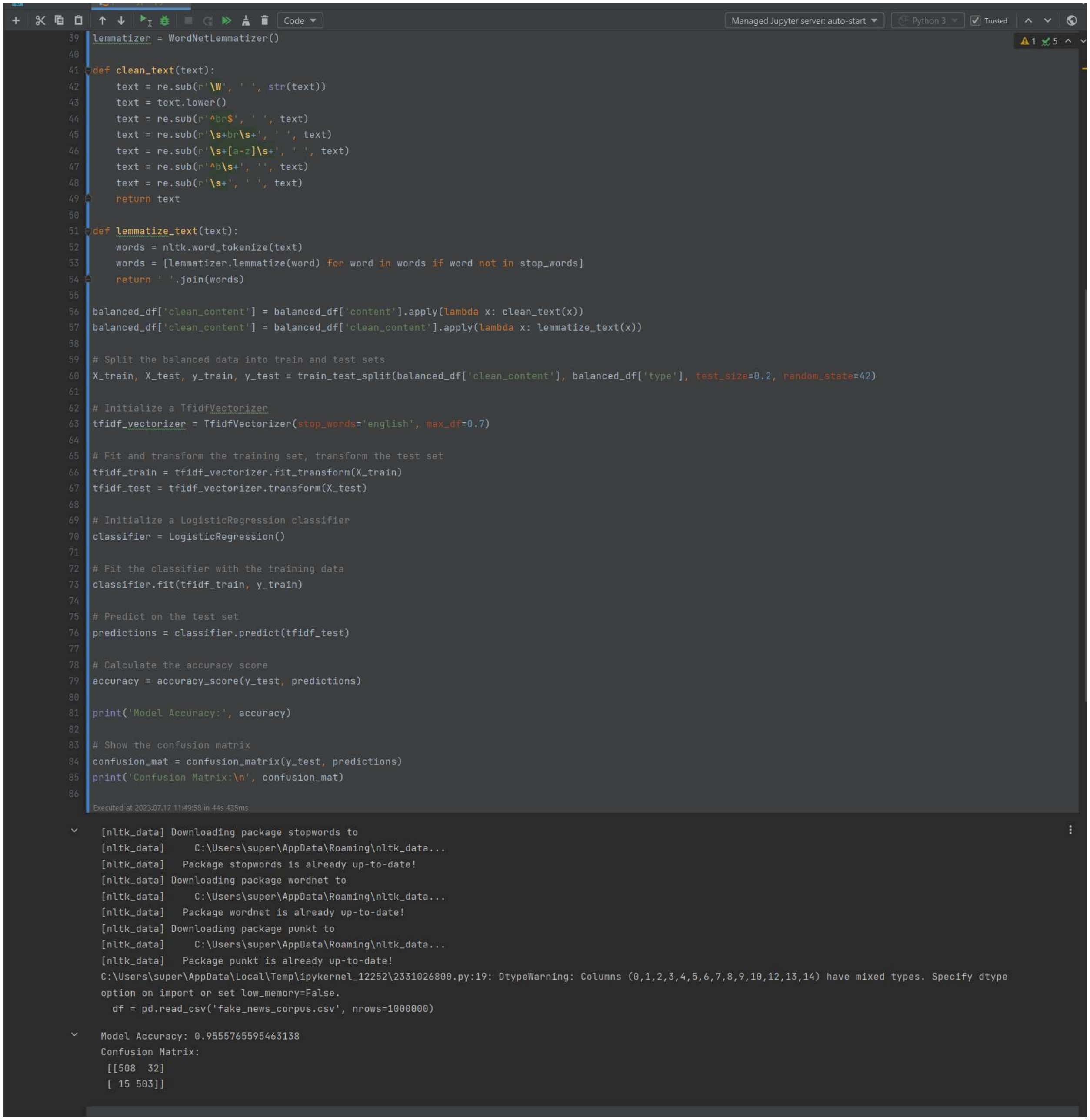
It's important to note that we only used a small part of a bigger dataset to test the performance of the prototype. This helps with large datasets that might not fit in memory. We also made sure that the information is fair so that our model doesn't favour the class with the most people in it. This was done by randomly picking the same number of cases from each class. This created a fair sample, which can help a model work better.

Conclusion

In conclusion, this prototype, which uses a part of a larger dataset and makes sure that the dataset is balanced, does important things to find fake news, such as cleaning the data, lemmatizing it, vectorizing it, and modelling it. Even though it's just a pilot, it gives a good start for building a more complete model for finding fake news.



```
# Import the necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
import nltk

# Download the NLTK stop words and tokenizer
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')

# Load the dataset
df = pd.read_csv('fake_news_corpus.csv', nrows=1000000)

# Filter relevant columns (content and type)
df = df[['content', 'type']]

# Remove records with missing data or other types of news that are not 'fake' or 'reliable'
df = df.dropna()
df = df[(df['type'] == 'fake') | (df['type'] == 'reliable')]

# Count the occurrences of each class
class_counts = df['type'].value_counts()

# Determine the minimum count of the classes
min_count = class_counts.min()

# Sample an equal number of instances from each class
balanced_df = df.groupby('type').apply(lambda x: x.sample(n=min_count, random_state=42)).reset_index(drop=True)

# Data Preprocessing
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def clean_text(text):
    text = re.sub(r'\W', ' ', str(text))
    text = text.lower()
    text = re.sub(r'^br$', ' ', text)
    text = re.sub(r'\s+br\s+', ' ', text)
    text = re.sub(r'\s+[a-z]\s+', ' ', text)
    text = re.sub(r'^b\s+', '', text)
    text = re.sub(r'\s+', ' ', text)
    return text

def lemmatize_text(text):
    words = nltk.word_tokenize(text)
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return ' '.join(words)

balanced_df['clean_content'] = balanced_df['content'].apply(lambda x: clean_text(x))
balanced_df['clean_content'] = balanced_df['clean_content'].apply(lambda x: lemmatize_text(x))

# Split the balanced data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(balanced_df['clean_content'], balanced_df['type'], test_size=0.2, random_state=42)

# Initialize a TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)

# Fit and transform the training set, transform the test set
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)
```



```python
lemmatizer = WordNetLemmatizer()

def clean_text(text):
    text = re.sub(r'\W', ' ', str(text))
    text = text.lower()
    text = re.sub(r'^br$', ' ', text)
    text = re.sub(r'\s+br\s+', ' ', text)
    text = re.sub(r'\s+[a-z]\s+', ' ', text)
    text = re.sub(r'^b\s+', '', text)
    text = re.sub(r'\s+', ' ', text)
    return text

def lemmatize_text(text):
    words = nltk.word_tokenize(text)
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return ' '.join(words)

balanced_df['clean_content'] = balanced_df['content'].apply(lambda x: clean_text(x))
balanced_df['clean_content'] = balanced_df['clean_content'].apply(lambda x: lemmatize_text(x))

# Split the balanced data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(balanced_df['clean_content'], balanced_df['type'], test_size=0.2, random_state=42)

# Initialize a TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)

# Fit and transform the training set, transform the test set
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)

# Initialize a LogisticRegression classifier
classifier = LogisticRegression()

# Fit the classifier with the training data
classifier.fit(tfidf_train, y_train)

# Predict on the test set
predictions = classifier.predict(tfidf_test)

# Calculate the accuracy score
accuracy = accuracy_score(y_test, predictions)

print('Model Accuracy:', accuracy)

# Show the confusion matrix
confusion_mat = confusion_matrix(y_test, predictions)
print('Confusion Matrix:\n', confusion_mat)

```

Executed at 2023.07.17 11:49:58 in 44s 435ms

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\super\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\super\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\super\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
C:\Users\super\AppData\Local\Temp\ipykernel_12252\2331026800.py:19: DtypeWarning: Columns (0,1,2,3,4,5,6,7,8,9,10,12,13,14) have mixed types. Specify dtype
option on import or set low_memory=False.
  df = pd.read_csv('fake_news_corpus.csv', nrows=1000000)
```

```
Model Accuracy: 0.9555765595463138
Confusion Matrix:
 [[508  32]
 [ 15 503]]
```

References:

1. [1]William Yang Wang. 2017. “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection. *arxiv* (May 2017). Retrieved from https://arxiv.org/abs/1705.00648

2. [2]Shaina Raza and Chen Ding. 2022. Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics* 13, 4 (January 2022), 335–362. DOI:https://doi.org/10.1007/s41060-021-00302-z

3. [3]Nihel Fatima Baarir and Abdelhamid Djeffal. 2021. Fake News detection Using Machine Learning. *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)* (February 2021). DOI:https://doi.org/10.1109/ihsh51661.2021.9378748

4. [4]Akshay Jain and Amey Kasbe. 2018. Fake News Detection. *2018 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)* (February 2018). DOI:https://doi.org/10.1109/sceecs.2018.8546944

5. [5]Tukrel, A., Wolfe, A., and Yau, K. 2020. Detecting COVID-19 Fake News Using Deep Learning. http://cs230.stanford.edu/projects_spring_2020/reports/38868289.pdf

6. [6]Sudhanshu Kumar and Thoudam Doren Singh. 2022. Fake news detection on Hindi news dataset. *Global Transitions Proceedings* 3, 1 (June 2022), 289–297. DOI:https://doi.org/10.1016/j.gltp.2022.03.014

7. [7]Shu, Kai, et al. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. ACM Trans. Big Data 8, 3, Article 15 (September 2020), 171-188. DOI:https://doi.org/10.48550/arXiv.1809.01286

8. [8]FakerFact: Fake News Detection for the Modern Age. FakerFact: Fake News Detection for the Modern Age. Retrieved from https://community.ibm.com/community/user/ai-datascience/blogs/michael-mansour1/2020/06/15/fakerfact-fake-news-detection-for-the-modern-age

9. [9]Full Fact - Full Fact is the UK’s independent fact checking organisation. Full Fact. Retrieved from https://fullfact.org