2023

Fake News Detection using Natural Language Processing

Fabiolla Mayrink Costa
CCT College Dublin

Rael Guimaraes
CCT College Dublin

CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

Module Title:	Problem Solving for Industry
Assessment Title:	Capstone Group Project
Lecturer Name:	Muhammad Iqbal
Student Full Name:	Fabiolla Mayrink Costa Rael Guimaraes
Student Number:	2019226 2019216
Assessment Due Date:	19th May 2023
Date of Submission:	19th May 2023

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

Introduction

Nowadays with the advance of technologies we have vast access to any sort of information. We are able to use our phone/computer to access the news of any part of the world. It is great to keep us informed about everything that is happening around the world. It is also a powerful tool used for companies while making strategic business decisions.

The biggest issue is that technology can and is being used to manipulate people/companies by propagating fake news. Fake news can mislead people's perceptions while forming opinions on a determined subject. It can also have a big impact on a company during the strategic decision-making process.

Our fake news detection model will allow people/companies to identify whether the news/information is real or not. With the model, people can form opinions on any subject without being manipulated by fake news. In the same way, companies can make better market decisions based on the analyses of real news/information instead of being influenced by fake news.

Our fake news detection model will use NLP (Natural Language Processing) to predict if the news is real or fake. See below for a more detailed explanation of how our model will predict the news by analysing phrase patterns.

Word count

The total word count for this document not including the cover letter, titles, citations and references is 4566 words. The word count of each topic is:

- Introduction – 206
- Fake news detection model – 348
- Data understanding – 677
- Data preparation – 980
- Modelling – 866
- Evaluation of the model – 599
- Deployment of the model – 399
- Conclusion – 527.

Fake News Detection Model

Fake news can have a huge impact on society. The impact can be even bigger on the business environment. Fake news can manipulate people in order to beneficiate companies, politics and others. In a business environment, it can mislead companies while analysing information causing them to take wrong strategic decisions. Fake news is a problem that we are all facing at the moment, in one way or another. Actions need to be taken in order to prevent it from happening. And that is where we come into action. After extensive research, we came up with the best solution which is to develop a model that will allow us to analyse and identify fake news.

Some “fake news" is published on satire sites that are usually clearly labelled as satire. However, when people share articles without reading beyond the headline, a story that was supposed to be a parody can end up being taken as the truth (Haskins, 2023).

In order to prevent people from being manipulated and companies from taking wrong strategic decisions we are developing a model which will predict if the given news is fake or real. The Fake News Detection Model will use NLP (Natural Language Processing) to analyse the given data and classify if it is real or fake based on word patterns. NLP is a part of Artificial intelligence (AI) that gives computers the ability to understand text and spoken words in the similar way humans do. It combines computational linguistics rules-based human language models with statistical, machine learning and deep learning models.

NLP models work by finding relationships between the constituent parts of language — for example, the letters, words, and sentences found in a text dataset. NLP architectures use various methods for data pre-processing, feature extraction, and modelling (DeepLearning.AI, 2023).

There are many advantages to the implementation of NLP for the fake news detection model. But the most important ones are its performance of large-scale analysis, its ability to automate processes in real-time, and because the NLP algorithm can be tailored to any company's needs and criteria, industry-specific language, sarcasm and misused words.

Data Understanding

In order to develop our fake news detection model we were required to find suitable datasets to work with. With some research, we were able to find 2 datasets that met all the necessary requirements. The datasets are "True.csv" and "Fake.csv". The datasets were found on the Kaggle website at https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset.

The "True.csv" dataset is a dataset of articles which are considered to be real news. It consists of 4 columns and 21417 rows (entries). The columns are:

- **title**- as the name suggests the title column has the article title. It has an object as its data type.
- **text** – the text column has the article body. Its data type is an object.
- **subject** – the subject column has the category of the article (politics, sports, world news, and so on). Its data type is an object.
- **date** – the date column has the date that the article was posted. And its data type is also an object.

In order to have a better understanding of the data we performed an initial data exploration. In the exploration of the "True.csv" dataset, we were able to see that it does not have any null value, meaning that actions to replace/eliminate the null value would not be required. During the exploration of this dataset we found out that it had 206 duplicated entries and before proceeding with the exploration we removed them. In the exploration of the "True.csv" dataset, we saw that there are only 2 categories, politics news and world news. The largest category is "politic news". It is unusual to have only 2 categories of news, but we assume that it is due to the short period of data collection. The final aspect explored in this initial exploration was the date range of the collected data, which is approximately 1 year and 5 months. The collected articles go from April 1st 2016 to September 9th 2017.

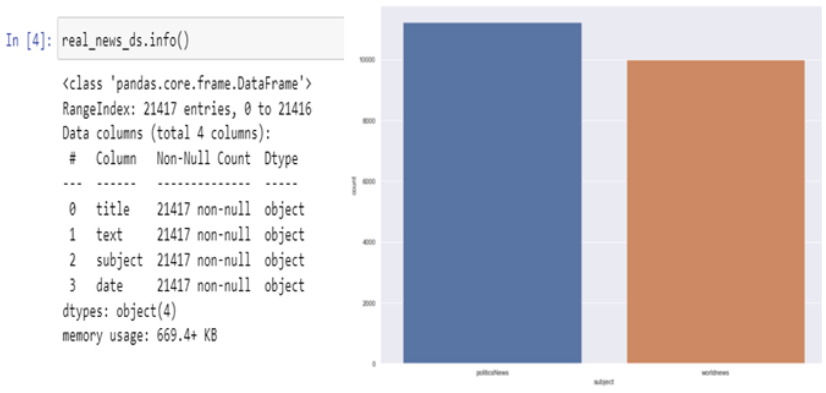

Figure 1

The “Fake.csv” dataset is a dataset of articles which are considered to be “fake” news. It consists of 4 columns and 23481 rows (entries). The column structure of the “Fake.csv” dataset is identical to the column structure of the “True.csv” dataset, with 4 columns which are title, text, subject and date. All columns have an object as data type. For more detailed information see the “True.csv” column structure above.

Initial data exploration was also performed on the “Fake.csv” dataset for a better understanding of the data. During the exploration, we saw that the “Fake.csv” does not have null values. In the exploration, we spotted that the “Fake.csv” dataset had only 3 duplicated entries, and before proceeding we removed them. In the exploration of the “Fake.csv” dataset, we saw that there are 6 categories, news, politics, government news, left-news, US news and Middle-east. The largest category is news.

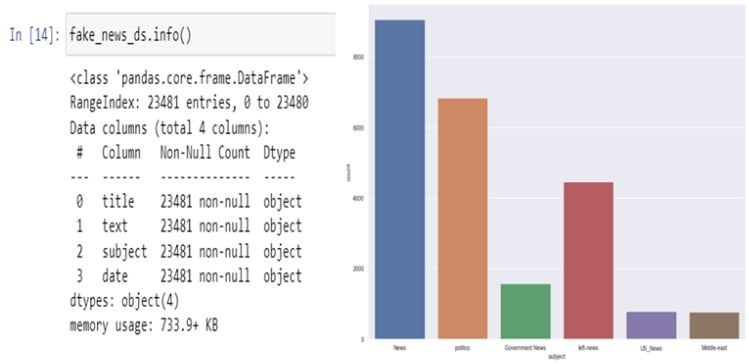

Figure 2

When we tried to perform operations to analyse the range of the collected data we got malformed information. Instead of getting the last date of the collected article, we got a hyperlink. It means that there are malformed entries in the date column. To have a better understanding of the data malformation we extracted the day, month and year aspects of the date columns into individual columns. With this extraction, we were able to look at them individually. Once the date aspects were extracted we checked for null values on the new columns. In this analysis, we found 39 entries without the collected article’s year and 1 entry without the day. We removed those entries as they were not complete. Now that we only have complete entries we checked the unique values of the new columns. In this process, we found out that there were malformation data on the date column. There were hyperlinks included in it. We removed the malformed entries using their index id. Now that we eliminated the malformed entries we concluded the initial data exploration by analysing the date range of the collected data. The data range of the collected data is approximately 2 years and 5 months. The collected articles go from April 1st 2015 to September 9th 2017.

Data Preparation

When the initial data analysis was concluded we had a better understanding of the data we are working with. Now we know important aspects of the data, such as data type, date range of the data collection, type of articles, etc. Knowing this information is very relevant to the implementation and evaluation of the model. As you probably noticed we performed some operations for an initial data cleaning while analysing it. We removed the duplicated entries and also removed incomplete/malformed entries. There was no entry containing null values in the datasets, meaning that we did not need to remove it.

With the initial cleaning already performed, we started to prepare the data for the fake news detection model implementation. The first action taken in the process of preparing the data was to combine/merge the “True.csv” and the “Fake.csv” dataset. In order to be able to identify which is the data from the “True.csv” dataset and which is from the “Fake.csv” dataset we created a new column on each called label. On the label column, we gave a value of 1 to the “True.csv” dataset and a value of 0 to the “Fake.csv” dataset. Once concluded, we merged the 2 datasets into a new one called “combined_news_ds”. The next step taken in the data preparation process was to remove the noisy data. Noisy data is data which is not relevant to the model that will be implemented. Noisy data can affect the results of the implemented model.

Noisy data can adversely affect the results of any data analysis and skew conclusions if not handled properly. Statistical analysis is sometimes used to weed the noise out of noisy data (Javatpoint, 2021).

The columns ‘day’, ‘month’ and ‘year’ were removed as they will not be used in the model implementation. These columns were only created in order to find and remove the malformed entry on the date column of the “Fake.csv” dataset. After removing the noisy data we performed another data analysis in the “combined_news_ds” dataset. The new analysis was performed in order to assure that the data was not modified during the merging process. Assuring that the merging did not create any malformed data.

During the analysis of the data, we could see that there was no malformed data created during the merging process. There was no null value or duplicated entries created. We could also see that the categories of each dataset were also not affected by the merging process. It is still having 6 different categories for “fake news” and 2 categories for “true news” (as shown in figure 3).

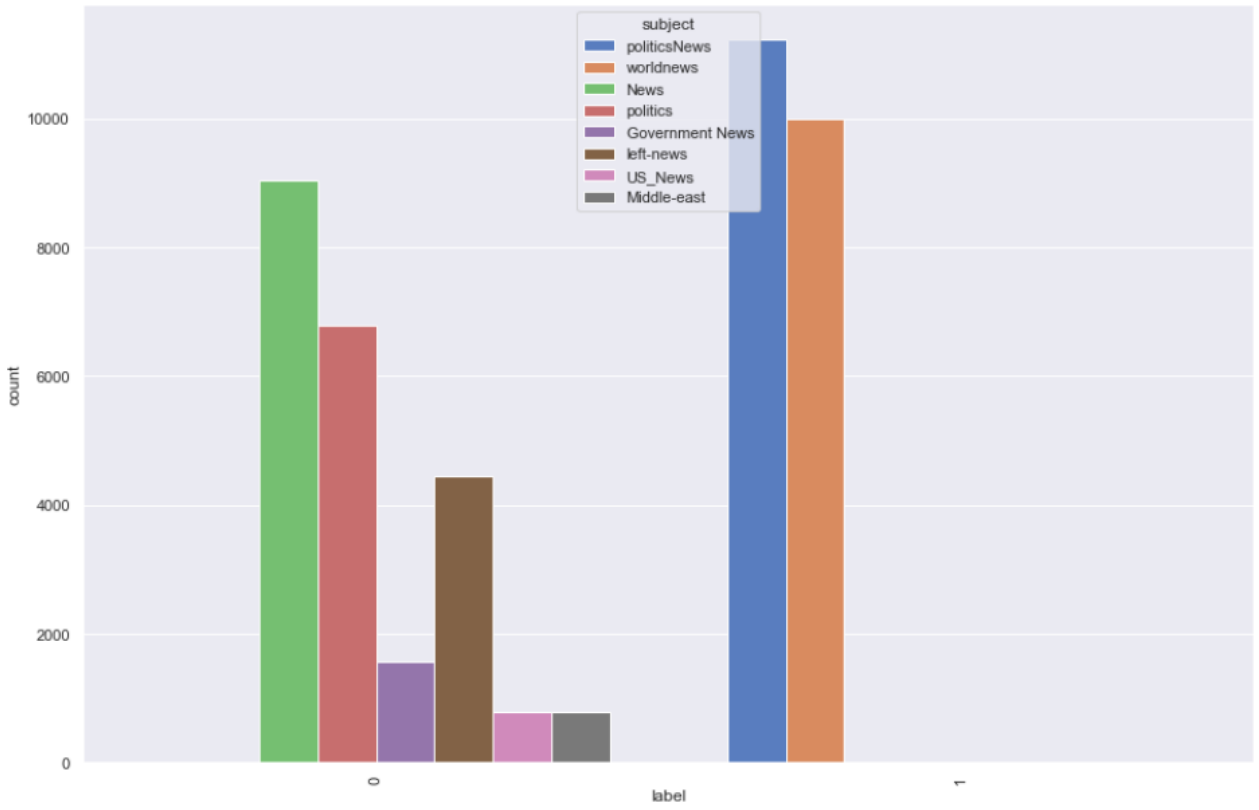

Figure 3

In order to plot the article's collection over the months, we converted the data type of the date column from object to DateTime. After converting the data type we were able to plot the data. As shown in Figure 4 on the plot we are able to see that the "fake news" data were collected more evenly while the "real news" had a large increase in the collection from July 2017.

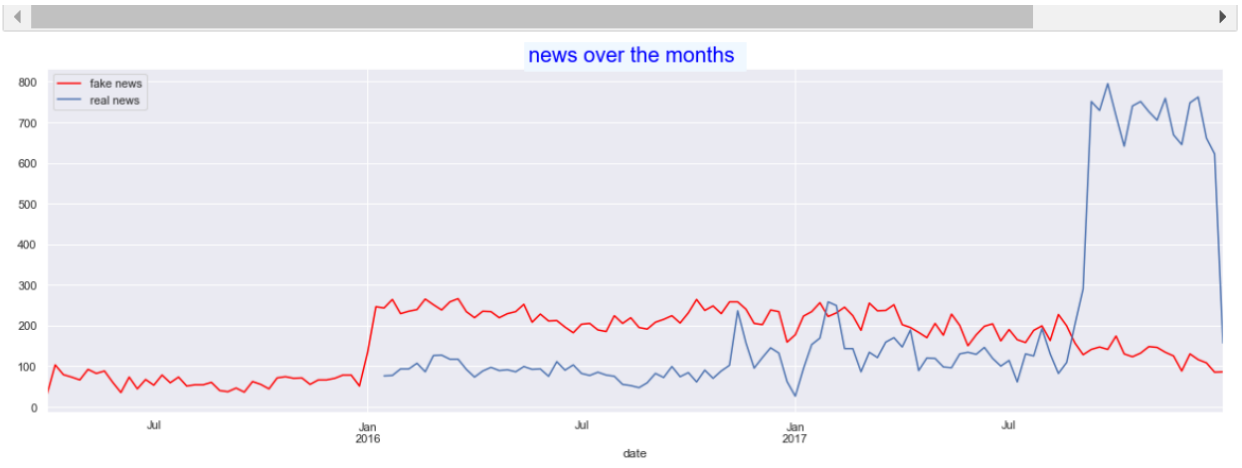

Figure 4

The next step taken in the preparation of the data was to concatenate the 'title' and the 'text' column of the "combined_news_ds". The new concatenated data was placed in the 'text' column. Once it was concluded we removed the remaining noisy data from the dataset. The noisy data removed were the 'title', 'subject', and 'date' columns. Now we only have the relevant data for the fake news detection model implementation, the 'text' (which is now a combination of text and title) and the 'label' columns.

We displayed the most used words on the "real" and "fake" news. The word that seems to be used most is "Trump" or "Donald Trump" and "United States". It means that the majority of the articles ("real" and "fake") are related to "Donald Trump" and "United States". It is important to mention that the stop words were removed from the plot. The stop words are the common words in a language (the, is, and, a, an, etc).

The last aspect to analyse before the model implementation was the balance between the "real" and "fake" data. In this analysis, we identified that the data is not completely balanced. There is a larger amount of samples for the "fake" news data than the "real" news data (see Figure 5). In order to balance the data we are going to apply the re-sampling technique. We used the oversampling technique to be more precise.

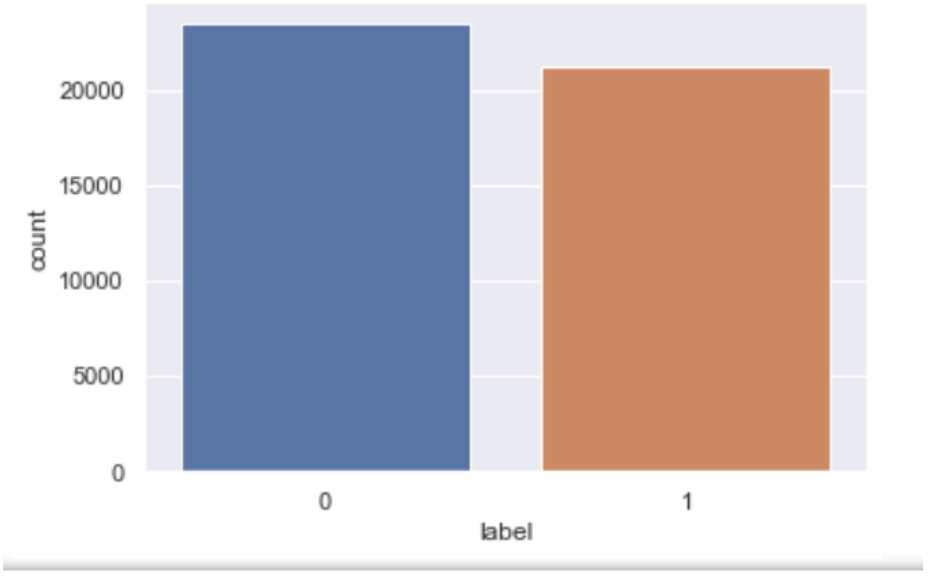

Figure 5

To balance the "real" and "fake" data we used the resampling technique. There are 2 ways of resampling the data, oversample which duplicates samples from the minority class and undersample which deletes samples from the majority class. As mentioned above we used the oversampling technique. The oversampling technique selects samples from the minority class randomly and duplicates it. The reason we decided to use oversample instead of undersample is the fact that by deleting random samples on the

majority class my result on losing valuable samples. Since our model will analyse word patterns to identify 'fake' news it is better to duplicate samples from the minority class than lose samples from the majority class. To apply the oversampling technique we first created 2 temporary datasets, one for the 'real' news and one for the 'fake' news. The second step was to resample the "real_news" data, having the number of samples (n_samples) set to the same amount as the "fake_news" samples. And also having the number of reproducible samples (random_states) set to 50. It means that 50 samples will be randomly used to be duplicated. The last step was to concatenate the temporary 'true' and 'fake' datasets back into the "combined_news_ds" dataset. Now we have our data balanced and ready for the fake news detection model implementation (see Figure 6). In the following pages, we will be explaining how the process for the model implementation was.

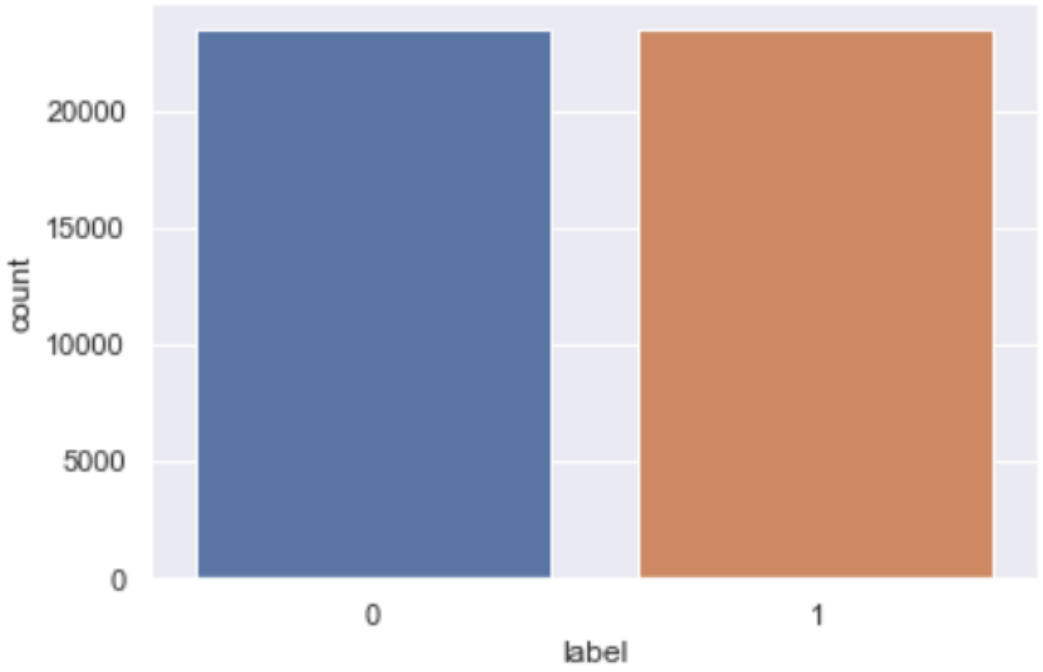

Figure 6

Modelling

The first step taken in order to choose the best model to be implemented was to split the data into training and test data. The training data will be used to train the model while the test data will be used to test the performance of the model. We split the data leaving 80% in the training set and 20% in the testing set. In order to choose the best model to be implemented we tried 3 most used models for NLP. The used models are SVM (linear SVC), Naive Bayes (multinomial naive Bayes) and Logistic Regression. It is a good mixture of linear (Logistic Regression) and nonlinear (SVM, Naive Bayes) models.

- **Support Vector Machine (SVM) –** is used to solve linear and non-linear problems. It is used for classification and regression problems. The SVM model creates a line/hyperplane in order to separate the data into classes. It maps the data to a high-dimensional feature space in order to categorize the data points. The data is transformed in a way that the separator between them can be drawn as a hyperplane. It uses the characteristics of new data to predict the group it belongs.

- **Multinomial Naive Bayes** – is a model that uses Bayes Theorem. It is a specific instance of Naive Bayes that uses multinomial distribution for each of the features. It guesses the tag of the data using the Bayes Theorem, then calculates each tag's probability of a given sample and outputs the tag with the greatest chance.

- **Logistic Regression** – uses mathematics in order to find the relationship between 2 data features. Logistic Regression uses the relationship of the value of one feature based on the other.

It is not possible to directly use text as an input to the classifier models. It is necessary to convert the text into numbers. In order to convert the text into numbers we used Count Vectorizer and TF-IDF (Term Frequency-Inverse Document Frequency) Transformer.

Count Vectorizer gets a collection of documents and tokenizes it in order to form a vocabulary of unique words. Count Vectorizer can also use this vocabulary to encode new documents. In simple words count vectorizer converts the text data into numerical data.

TF-IDF Transformer is used to eliminate words which do not have any meaning on their own but are often used in every document, such as 'the', 'in',' a', 'an', 'on', etc. These words need to be eliminated because they can reduce the performance of the model. TF-IDF provides frequency scores to words by headlining the most frequent ones in a document, but not across the documents. TfidfVectorizer tokenizes documents, learn the vocabulary of it and inverse its frequency scores. It also encodes new documents. As we will be using the CountVectorizer we will be using TfidfVectorizer to calculate the inverse document frequencies and start encoding the documents.

To resume, in the model implementation, we are using count Vectorizer to tokenize (transform into numerical value) the text and count the word frequency, while the TF-IDF transformer normalises the data by eliminating meaningless words which are often used. As we need to chain the CountVectorizer and TfidfTransformer along with the model, we will be using a pipeline object. A pipeline object is used to automate machine learning workflow. It permits several transformers to be chained together. The data flow of the pipeline as the name suggests is from the start to the end. The output of each transformer in the pipeline is used as the input of the next. There are 2 main methods in the pipeline. The first is the 'fit_transform' which is called for each transformer. It is also called each time each time the result of a transformer is used in the next one. And the other main method of the pipeline is the 'fit_predict' which is called in case the pipeline ends with an estimator.

In order to identify which model would give the best results we created an array and added the 3 models to be analysed. We implemented a loop where the data would pass through the pipeline for the text conversion before getting to the model. On the pipeline the data goes first through the CountVectorizer where it gets converted to a numerical value, when concluded it goes through the Tfidf transformer where it is normalised by removing meaningless information, and finally it goes to the model. When the data had finished the pipeline process we used it to train the model. Once the model had been trained we test it and analyze the score. As mentioned before, this process is done for the 3 models.

We are using the test harness technique to get the model's accuracy. On the test harness, we are using stratified 10-fold cross-validation. It splits the data into 10 parts, trains on 9 and tests on 1. This process repeats for all combinations of train-test splits. Stratified assures that each split of the data used in the model training has the same distribution of the sample by class. The random seed is set via the random_state argument to a fixed number. It ensures that each model is evaluated on the same splits of the training data. To evaluate the models we are using a metric of accuracy (see Figure 7).

```
results = []
names = []

for name, model in models:

    pipe = Pipeline([
    ('Vectorizer', CountVectorizer()),
    ('Transformer', TfidfTransformer()),
    ('model', model)
    ])

    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(pipe, X_train, y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

Figure 7

Evaluation of the Model

After analyzing the results of each implemented model we got to the conclusion that the best model to be used in our fake news detection model implementation was the Linear SVC model (see Figure 8).

```
SVC: 0.995296 (0.001133)
NB: 0.938682 (0.005377)
LR: 0.987764 (0.001487)
```

Figure 8

To a better analysis of the models we are plotting the evaluation results in order to compare the spread and the mean accuracy of each model. As each model was evaluated 10 times because of the 10 fold-cross validation we have a population of accuracy measures to be plotted. To have a clear visualization we used a box and whisker plot as shown in the Figure 9.

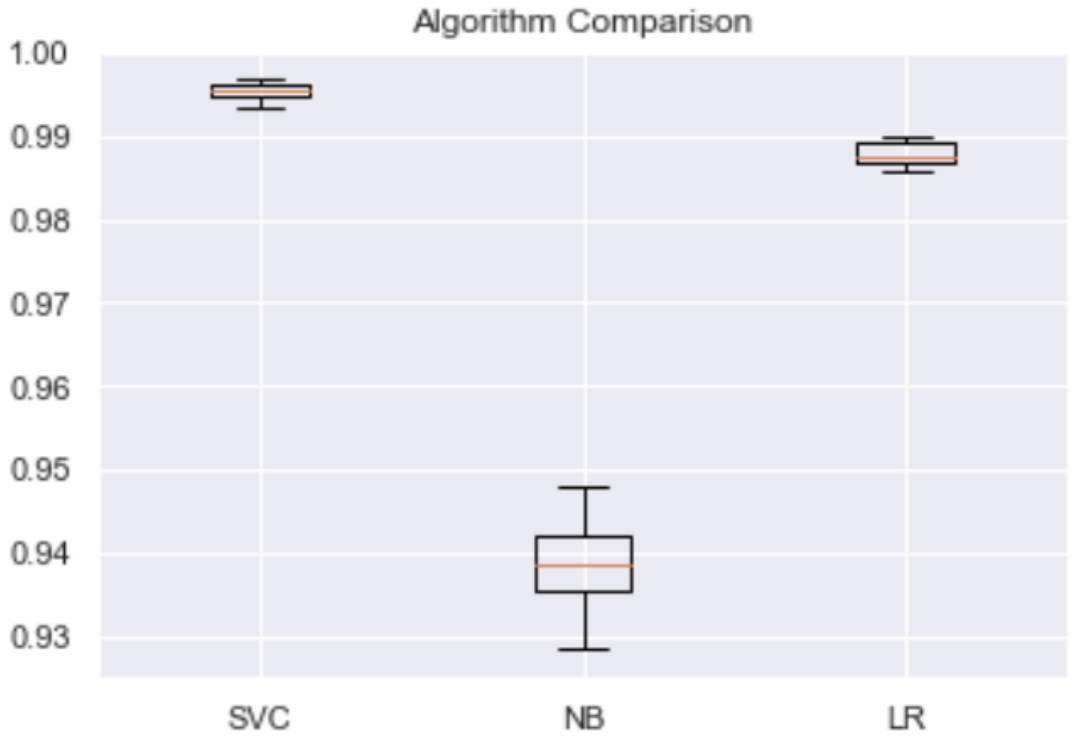

Figure 9

The image shows that the Linear SVC evaluation results have a shorter spread than the others. It means that its results are more concentrated, it does not spread much. It has a higher mean accuracy, which is above 99%. The Linear SVC (SVC) had its results slightly better than the others and we decided that it would be the model used for our implementation. Linear SVC has better results because it has better performance with unstructured and semi-unstructured data, such as text and images. It makes Linear SVC better than Logistic Regression in our context. Linear SVC has better results for linear and non-linear classification. It makes it better than the Naive Bayes model.

The first step taken in the model implementation was to define the variables. We defined the 'text' field as an independent variable ('X') and the 'label' field was defined as a dependent variable ('y'). The second step was to split the data into training and testing samples. It was divided in the same way as the modelling process, 80% for training and 20% for testing. The following step was to pass the data through the pipeline for converting the text into numerical value as explained above. Finally, we trained the model. Once the model was trained we tested it. The model gave us an accuracy of around 99% and precision results of around 99%. It is considered to be an excellent result.

Achieving very high accuracy could indicate that the model is being overfitted. To assure that it was not the case we are comparing the training accuracy with the testing accuracy. As you can see in the Figure 10 our model is not being overfitted as the difference between the training and testing accuracy is very small, having the training accuracy slightly higher than the testing accuracy.

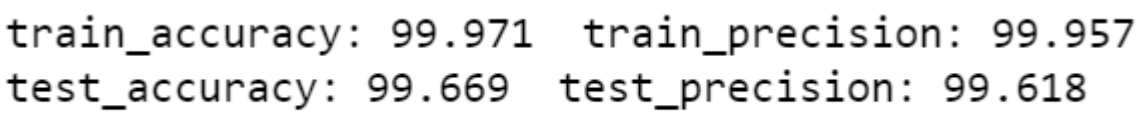

Figure 10

In machine learning, it's typical for the training accuracy to be a bit higher than the testing accuracy. This is because the model uses the training data to make predictions, so it's expected to perform slightly better on the training data. However, if the difference between the training and testing accuracy is too significant, this could indicate a problem. You generally want the difference between the training and testing accuracy to be as small as possible. If the difference is too significant, it could mean your model is not performing well on new data and needs improvement (Kaplan, 2023).

Analyzing the confusion matrix (see image below) we can see that our model correctly predicted 4693 True Positive (TP) samples. It means that 4693 samples were predicted positive, and it is true. On the

analysis we can see that our model correctly predicted 4200 True Negative samples (TN), meaning that 4200 samples were predicted negative, and it is true. Our model also incorrectly predicted 18 False Positive (FP). It means that 18 samples were predicted positive but it is false. Lastly, our model incorrectly predicted 18 False Negative (FN) samples, meaning that 18 samples were predicted negative but it is false (see figure 11).

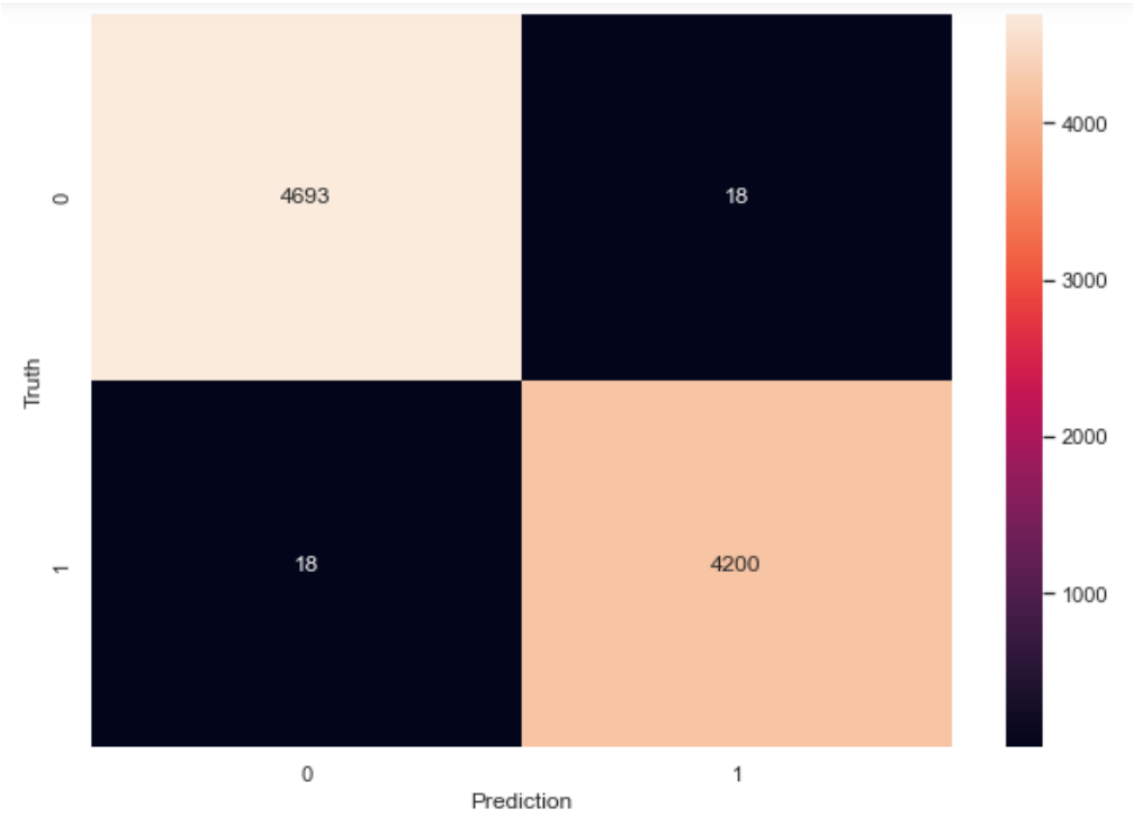

Figure 11

Deployment of the Model

For the deployment of our fake news detection model we intend to create a web application where the users will enter either the URL of a news article or copy and paste the article itself. The model will analyse the input and returns its prediction, whether the article is true or fake. Our clients will need to subscribe to our web application in order to use the fake news detection model. A fee will be charged for the subscription. The prices will vary according to the client profile. Companies will have higher fees than regular clients as they are expected to use the application more often. The access will be limited based on the paid fee. If a client reaches the model usage limit they will be able to purchase more if wanted. Our clients are expected to be companies that want to assure if the news is real or fake as it can have an influence on their business plan, and also people in general who do not want to be manipulated by untrue news.

Deploying a machine learning model is relatively easy and it might prevent us from realising how difficult it is to maintain the model. It can also be very expensive. The deployment plan that we intend to use is to first deploy the model and constantly work on feeding it with up-to-date data and constantly training the model in order to avoid the model staleness.

Model staleness test. The model is defined as *stale* if the trained model does not include up-to-date data and/or does not satisfy the business impact requirements. Stale models can affect the quality of prediction in intelligent software (INNOQ, 2023).

In order to overcome the high expenses issue we are planning to deploy our model on a smaller scale and expand according to the demand. It is accounted for personal, storage and any other expenses related to the model deployment. At the start, we will be the only ones responsible for the data collection and maintenance of the model, because the cost of a larger team can be very high. But we are open to expanding our team if the demand increases. The same idea goes for the storage size we intend to start with small storage which will have a low cost and expand it as necessary due to the increase of the data used to feed/train the model.

Conclusion

With the advance of technology we have access to world news in the palm of our hands. It is great, but it has also increased the amount of untrue (fake) news, which is propagated with the intention of beneficiating someone or some organization. New applications/technologies are required in order to prevent it from happening.

In order to develop a fake news prediction model we were required to do extensive research to find the best techniques to be used on the model implementation. It was also required in order to find datasets that met the requirements for the model training and testing.

Before implementing any model is necessary to have a clear understanding of the data that we are working with. It is always necessary to analyse the data before any preparation. Once we understand the data we are working with we prepare it for the model implementation.

Usually, the classification models works only with numerical data, meaning that if we want to do a text classification we might need to use techniques which will convert the text data into numerical data such as Count Vectorizer and TF-IDF Transformer. We might also need to use methods which will facilitate the use of the techniques such as pipelines.

To identify which model implementation would give us better results we were required to implement different models and analyse their outcome. Choosing a model will always depend on what you are trying to accomplish, the model which works better for our task might not be the best one for your project. The only way to identify it is to experiment with different ones and analyse its outcome.

In order to find the best model to implement we analysed the accuracy and precision of 3 different models. The analyses showed us that the best model to be implemented was the Linear SVC. It gave us an accuracy of around 99% and a precision of around 99%. It is considered to be an excellent result. The main reason why the Linear SVC model got better results than the other models is that it performs better with unstructured and semi-unstructured data (text and images). It also has high-accuracy results for linear and nonlinear problems.

To conclude we can say that throughout the development of this project, we were able to understand the importance of following the right steps in a model implementation which are the data collection, data analysis, data preparation and model implementation. Without properly following the steps we might end up with a model having low or even untrue results based on the poor quality of data it is being fed with. It is also fundamental to check if the implemented model is not being overfitted or underfitted as it can lead to untrue accuracy. We also concluded that although it seems easy to deploy a machine-learning model, it is not. Deployment of a machine learning model can be very difficult and

expensive to maintain. The wiser solution is to deploy your model on a small scale and slowly expand as required. Otherwise, you might end up having high expenses and an outdated model, leading to the failure of your model deployment.

Appendix 1

This project was concluded with a group effort where we divided the workload equally between the 2 integrants of the group. Throughout de project development we were able to understand how important it is to have good communication among the group integrants in order to have a great workflow.

With the conclusion of this project, we are confident to say that we have gained many skills such as communication, teamwork and time management. We have also gained technical skills related to machine learning and data science such as data analysis, data preparation, count Vectorizer and TF-IDF Transformer techniques, pipeline method and model implementation.

We faced many challenges during the project development. It is difficult to learn new technology while implementing it. It required a lot of research. But we are happy with the final outcome of the project. As mentioned, learning while implementing the technology was difficult, but the main challenge faced during this project was the time shortage. It has been challenging to complete the project while working, attending classes and developing projects from other subjects. But we were able to improve our time management skills. It was gratifying.

For a better understanding of the roles and responsibilities of each team member please see the following pages.

Video Presentation Link

Here is the provided link for our video presentation:

https://drive.google.com/drive/folders/1nQqN0lpUpacHAK4yq8FJibNMIyDBallc?usp=sharing

GitHub

Here is the link to access our GitHub repository:

https://github.com/rael-guimaraes/Fake_news_prediction

Appendix 2

In order to develop the fake news detection model project we divided the workload equally between the 2 integrants of the group. It is also important to mention that despite having the work divided we were always in constant communication as all parts of the project are fully connected. And our schedule about our meetings and progress can be found on BaseCamp.

Group integrants' roles and responsibilities

As previously mentioned the workload was equally divided between the 2 participants of the group. See below the roles and responsibilities of each team member.

- **Fabiolla Mayrink Costa** – Fabiolla's role was a software engineer and project manager. In the report, Fabiolla was responsible for the development of the business understanding, data understanding and modelling. In the model implementation, she was responsible for the data analysis, and modelling. Fabiolla was also responsible for the poster presentation and document formatting.
- **Rael Guimaraes** – Rael's role was a software engineer and project manager. In the report, Rael was responsible for the development of the data preparation, evaluation and deployment. In the model implementation, he was responsible for the data preparation and evaluation. Rael was also responsible for setting the meeting time and document formatting.

Since the group had only 2 members the roles are similar. Both members were involved in the technology research, model implementation and report development.

The report conclusion was the product of a group effort. We organised a long meeting where we discussed the outcomes obtained during the project development in order to form our conclusion. The same goes for the model deployment. The strategy for the model deployment is the product of a group discussion, we analysed what would be the best strategy for the model deployment. After a long research, we defined our deployment strategy which we are confident to be the most appropriate for what we intend to accomplish.

When each member of the group concluded each part of the work it was exchanged, allowing each member to analyse and give suggestions for improvement. Once the project was concluded we organized a meeting to discuss the project as a whole and make the final adjustments to improve it.

The project was uploaded to a GitHub repository and does not have many versions updates. The reason for it is that the project was entirely developed locally and only uploaded once it was concluded. It can be found at https://github.com/rael-guimaraes/Fake_news_prediction.

References

Adachi, F.de P. (2021) *Deploying a fake news detector web application with Google Cloud Run and flask*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/deploying-a-fake-news-detector-web-application-with-google-cloud-run-and-flask-eb750cce986d (Accessed: April 25, 2023).

Bisaillon, C. (2020) *Fake and real news dataset*, *Kaggle*. Available at: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset (Accessed: April 25, 2023).

Borcan, M. (2020) *TF-IDF explained and python sklearn implementation*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/tf-idf-explained-and-python-sklearn-implementation-b020c5e83275 (Accessed: April 25, 2023).

DeepLearning.AI (2023) *Natural language processing (NLP) - A complete guide*, *(NLP) [A Complete Guide]*. Available at: https://www.deeplearning.ai/resources/natural-language-processing/ (Accessed: April 10, 2023).

Ganesan, K. (2023) *What are stop words?*, *Kavita Ganesan, PhD*. Available at: https://kavita-ganesan.com/what-are-stop-words/#.ZEeRBXbMI6k (Accessed: April 25, 2023).

Haskins, J. (2023) *Fake news: What laws are designed to protect*, *LegalZoom*. Legalzoom.com. Available at: https://www.legalzoom.com/articles/fake-news-what-laws-are-designed-to-protect (Accessed: April 10, 2023).

IBM (2023) *What is natural language processing?*, *IBM*. Available at: https://www.ibm.com/topics/natural-language-processing (Accessed: April 25, 2023).

INNOQ (2023) *ML-ops.org*, *ML Ops: Machine Learning Operations*. Available at: https://ml-ops.org/content/mlops-principles#:~:text=Model%20staleness%20test.,of%20prediction%20in%20intelligent%20software. (Accessed: April 25, 2023).

Jain, P. (2021) *Basics of countvectorizer*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/basics-of-countvectorizer-e26677900f9c#:~:text=Countvectorizer%20is%20a%20method%20to%20convert%20text%20to%20numerical%20data. (Accessed: April 25, 2023).

Javatpoint (2021) *What is noise in data mining - javatpoint*, *www.javatpoint.com*. Available at: https://www.javatpoint.com/what-is-noise-in-data-mining (Accessed: April 25, 2023).

Kaplan, D. (2023) *Machine learning: High training accuracy and low test accuracy " EML*, *EML*. Available at: https://enjoymachinelearning.com/blog/machine-learning-high-training-accuracy-and-low-test-accuracy/#:~:text=you’re%20using.-,Should%20Training%20Accuracy%20Be%20Higher%20Than%20Testing%20Accuracy%3F,better%20on%20the%20training%20data. (Accessed: 11 May 2023).

Mazumder, S. (2022) *5 techniques to handle imbalanced data for a classification problem*, *Analytics Vidhya*. Available at:

https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/ (Accessed: April 25, 2023).

Narkhede, S. (2018) *Understanding confusion matrix*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (Accessed: April 25, 2023).

Oshikawa, R., Qian, J. and Wang, W.Y. (2020) *A survey on natural language processing for fake news detection*, *ACL Anthology*. Available at: https://aclanthology.org/2020.lrec-1.747/ (Accessed: April 25, 2023).

TechTarget (2017) *What is support Vector Machine (SVM)?: Definition from TechTarget*, *WhatIs.com*. TechTarget. Available at: https://www.techtarget.com/whatis/definition/support-vector-machine-SVM#:~:text=A%20support%20vector%20machine%20(SVM,which%20are%20labeled%20for%20classification. (Accessed: April 25, 2023).