SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

Fake News detection

Juan Gil de Gómez Pérez, Oriol Vall, Reem Hageali

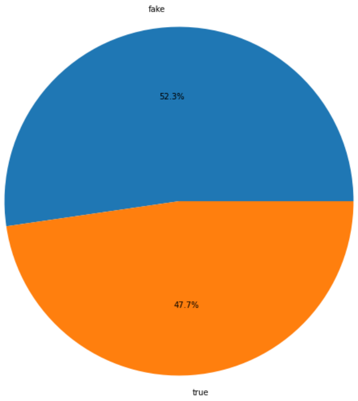
6/5/2022

Emerging Topics

TASK 1:

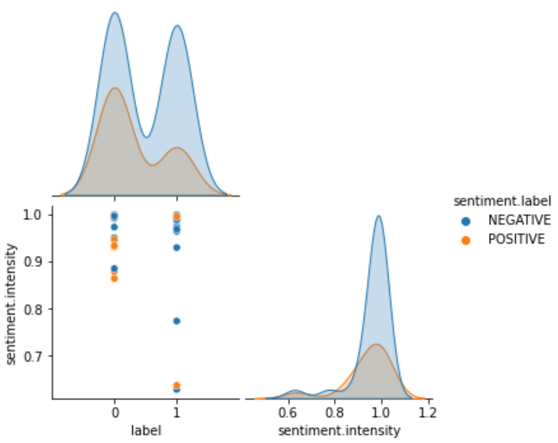
Dataset:

The data set is called “fake and true news dataset”. It was obtained from kaggle (community where you can get notebooks, datasets…). It consists of 5 variables: the title of the news, the text body of the news, the subject (news, worldnews….), the date it was published and the label of whether it was fake or real news. The first thing that we looked at in the dataset was whether the dataset was balanced or not. This was important for the selection of the pretrained model and also to choose the performance metric. In this case as we can see that it is balanced we have a little bit more freedom with the model and the performance metric.



After looking at the count of cases, we created sentiments regarding the text of the news. And then compared them to the labels of either fake or true. The main takeaway from this graph is that for positive sentiments is that positive sentiments lead more to fake news while negative sentiment does not have that much of an impact.

```
<seaborn.axisgrid.PairGrid at 0x7fa28b98e850>
```



After doing more EDA we chose the model

Model: distilbert-base-uncased:

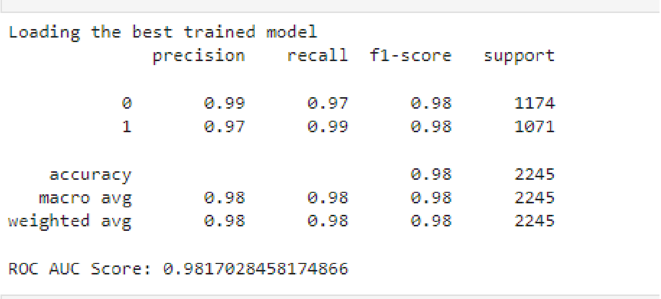
Distilber is uncased meaning that capital letters do not make a difference.

Also the Distilbert-base model is an improved version of another model (Bert base). Improved by being smaller and faster. Distilbert-base-uncased model was evaluated on its performance on raw datasets, meaning that there were no preprocessing like stopwords tokenizations… Being able to generate labels through an automatic process using the BERT base model. It was trained with the premise of polishing the following matters:

- Cosine embedding loss: hidden states generation was one of the purposes regarding the training of this model, doing so as good as the BERT base model.
- Distillation loss: BERT base model returns around the same probabilities as this model.
- MLM (Masked language modelling): this methodology is used a lot, as well as in the base model BERT. What this does is that it takes a percentage of words the text is trying to analyse, after taking those words the algorithm tries to predict the omitted words. This allows the model to learn more about the nature of the sentence as you can get a bidirectional view and representation of the text.

Key takeaways: Learns the same amount of information and has the same power as its base model, but it is faster and simpler. It also operates better for downstreams tasks or inferences.

Metrics of performance:

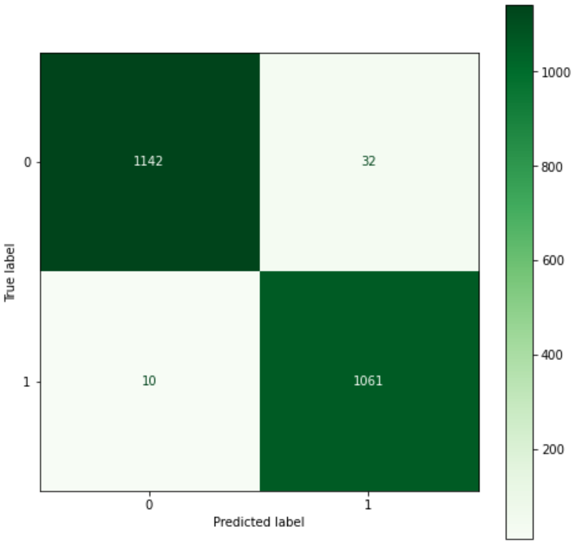


```
Loading the best trained model
              precision    recall  f1-score   support

           0       0.99      0.97      0.98      1174
           1       0.97      0.99      0.98      1071

    accuracy                           0.98      2245
   macro avg       0.98      0.98      0.98      2245
weighted avg       0.98      0.98      0.98      2245

ROC AUC Score: 0.9817028458174866
```



We trained over the train X and train Y set. After doing so we predicted over the test X set and compared it to the test Y set. We can see that the model is performing extremely well. In ALL the metrics (precision, f1-score, ROC…). Looking at the confusion matrix is still performing really well but we can see that it may be a little bit worse since it missed classified positives into negatives more times than it did for false negatives.

TASK 2:

Our system can be better if we take more steps to process the data and fine-tune the stages of the model. In this case, we used the DistilBERT model because it helps computers understand unclear language. When we used the DistilBertForSequenceClassification, we could have used the DistilBertModel to make a copy of the base DistilBERT model without adding any task-specific heads on top. Then, we could have added our own classification heads during the fine-tuning process to help the system tell the difference between real and fake. During the first stages of training, we would want to keep our model from changing the DistilBERT's pre-trained weights. This is when our model is learning how to use the weights for our new classification layers. If we want to keep the DistilBERT's pre-trained weights from being changed, we can set layer.trainable = False for each of its layers. Then, when the model's performance converges, we can set the argument back to True to unfreeze the weights again. DistilBERT's embedding layer can be topped with a classification head. Word-level understanding is provided by the sequence tokens, and we could have used the DistilBERT's sentence-level understanding of the sequence by looking at the classification (CLS) token. It stores a sentence-level embedding for the first token in the sequence. Starting out, we could have just added a single dense output layer with sigmoid activation function on top of the classification token's sentence-level embedding. This would have given us a baseline for how well our model would work at first. As a last step, we can build the model with the adam optimizer's learning rate and the loss function set to focal loss instead of binary cross-entropy because our dataset has a lot of different types of things. For the weights of the classification layer, we can start with random weights until the model's performance improves. It would also be a good idea to add more dense and drop-out layers to the surface. It is possible to unfreeze the embedding layer and fine-tune all weights with a lower learning rate after we have trained the classification layers. This way, we don't change the pre-trained weights too much.

After reading *A Survey of Fake News*, we also could have taken a style-based approach by combining latent textual features commonly used in news text embedding that can be conducted at word, sentence and document level. For a machine learning approach we could have used support vector machines. An SVM is a supervised machine learning method used for binary classification. It creates a decision boundary line with maximum margins to the support vectors. The goal in SVM is to separate fake news data with hyperplanes and extend it to non-linear boundaries. Preprocessing the data: converting the titles and the text into lowercase characters and tokenizing them, punctuation removed, only alphanumeric characters remain. To optimise the accuracy of the model we would have to tune many parameters.

- Different kernel types: Linear, Radial Basis Function, Sigmoid, Polynomial
- Testing various values for gamma

There are many different ways to improve the performance of the system, and there are many different models that can achieve a better result depending on the data.

Task 3

When creating a database it is very important to understand the purpose for which it will be used. Obviously, without knowing the value and usefulness of a dataset it is very difficult to come up with different variables that may be of interest to future tasks or analytics operations. For the problem at hand, generating a dataset to classify fake news from truthful ones we need to make very clear what is fake news and what is not. Since messing with the definition of the variables and labels will have a huge impact on the conclusions extracted from the analysis. What do we understand for Fake News? A news article that is intentionally and verifiably false [Allcott and Gentzkow 2017; Shu et al. 2017]. Once the purpose and meaning of the target variable, in other words, the objective of the study or analysis, is clear it becomes time to start compiling data. We could take two approaches for this step. Either using an already existing dataset and updating it with more recent and relevant data points or starting from scratch. In the case of starting from scratch, it is crucial to understand the first point mentioned in the previous paragraph.

In the situation of a FakeNews classifier it is most probable that a web scraping procedure will need to take place to compile the reviews and articles that will form the dataset. Now what will be the objectives when creating the dataset? Well, it will be very important to have a balanced target variable with similar or same amount of Fake and Real news. In addition, variables that will be useful and impactful for the model creation. For example, the text, the title, date … Variables that will be easily extractable from the article or review. In a situation, let's say we were to create a dataset, we would use for example 10 300 fake news articles and 10 300 legitimate news articles (a decent size dataset is necessary to have accurate and reliable modelling results). Now, the tricky factor here is the following: who gets to decide what is a fake news article and a credible one? Well that is where the fact-checking websites come into play. To create the dataset we will be scrapping the news from fact-checking websites, where the articles are labelled by human experts. Another approach to labelling news articles as Fake could be to use Style-based fake-news detection, which relies on rather quantifiable features, like characters, sentiment…

With this we move into a new key when creating a dataset. What variables to gather. As mentioned before the most common and easy to gather are text, title, date… which are all done by scrapping the article. But in order to maximise the potential and value of the model some extra variables can be added and generated. Depending on the way in addressing the problem we will need different variables. If we are planning on using a Style-based fake-news detection like mentioned before, we will need to add variables like sentiment for example, whether the review is positive or negative, quantity and complexity of the article as number of characters, uncertainty terms… Also if we are tackling a Sourced-based fake-news detection it will be crucial to have a source variable, that is from what newspaper or writer is the source coming from. One can detect fake news by assessing the credibility of its source, where credibility is often defined in the sense of quality and believability. And finally, Propagation-based fake-news detection is detecting fake news from a propagation-based perspective, meaning checking for the spreadability or impact of a fake article.

Once the dataset is created, the next step is to make sure it doesn't become obsolete, since that will make it lose its value and purpose. Therefore to ensure that the dataset is kept updated it would be very interesting to create and deploy ScraPy Spiders. A useful technology for web scraping. By having them regularly deployed, let’s say once in a month we ensure the dataset is constantly updated and from there automate the process of creating and modifying the necessary variables mentioned in the previous paragraph.

All in all, creating a dataset depends purely on the purpose for which is being created, to make sure that it has everything needed for the problem to be solved.

Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy*, *1*(1), e9.