Real vs Fake news: Machine Learning For Predicting Truthfulness of the News

**ABSTRACT**

The unchecked spreading of fake news is an alarming phenomenon on every social media platform and information outlet. Since online content can have a decisive affect on users’ political, social and business decisions and opinions, the identification and elimination of false information has become a critical prerogative for many online companies.

Natural language processing (NLP) encompasses linguistics, computer science, information engineering, and artificial intelligence. The main application of NLP is to train algorithms to analyze texts and extrapolated actionable insights based on text’s word composition and frequency. Another way of examining a document is analyzing the document’s intrinsic complexity. Text complexity is measured by several linguistic indexes such as Flesch Kincaid and Coleman Liau indexes.

Herein, I tested several machine learning models that leverage NLP techniques, linguistic indexes or both, to predict the news’ truthiness. The best model I came up with, was able to discriminate real from fake news with a balanced accuracy of 0.9836 and AUC score of 0.9998.

**DATA OVERVIEW AND EDA**

**Data overview**

The data utilized in this project consists of two separate csv files, one harboring all the real news (24417), one all the fake ones (23502 tweets) (<https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset>). The two files were compiled by two independent sources and thus could be inherently different. To minimize this suspected external bias, I applied a very aggressive data wrangling. The following features were available for each news:

* Date the news was published. The oldest news in the available dataset was published on March 31st, 2015, the last one on February 19th, 2018.
* Title of the news.
* Text of the news.
* Category of the news (Government News, Middle-east, News, US\_News, left-news, politics, politicsNews or worldnews).

**Data wrangling**

Data wrangling of the dataset required numerous steps which are detailed in the Jupiter notebook available in the GitHub [repository](https://github.com/Gianl-msi/Real-Vs-Fake-News). Herein, I will mention some of most critical steps. I filtered out all news with text shorter than 8 characters. Both real and fake news presented cross site scripts which were completely removed. Next, Facebook, Twitter, Youtube, Bitly and Tmsnrt hyperlinks were converted into 11-character long tokens for further analysis. All the other hyperlinks were removed. Vulgarities were replaced with more politically correct words and verbal contractions were expanded out. At this point text and title of the news were ready for sentences, words, syllables, long words, polysyllabic words and characters counts. Finally, all the linguistic indexes were calculated. Words in the clean text and title were then lemmatized. Standard English stop words were removed during this last step.

Before training and tuning the machine learning model, the number of features was trimmed down to reduce overallvariables collinearity. If two variables had a correlation of > 0.95, one was discarded. Next, variance inflation factor (VIF) was calculated for each remaining feature. Variables were removed until all VIF scores were below 10. Table 1 summarizes the dataset features available for training and testing of the machine learning classifiers.

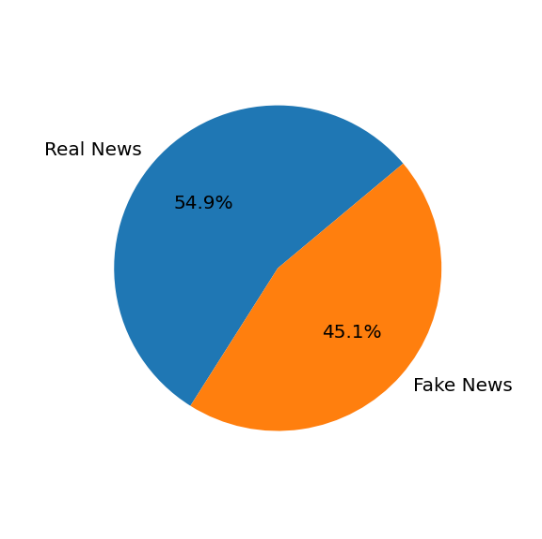
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **News Categories** | **Basic text features** | **Linguistic indexes** | **Processed text** |
| Real (0)  Fake (1) | - Government  - Middle-east news  - News  - US news  - Left news  - Politics  - Politics news  - World news | Title:  - n. sentence  - n. words  - n. long words  Text:  - n. sentences  - n. complex words | Title:  - Flesh Kincaid Grade  - SMOG  - ARI  - Gulpease  Text:  - Flesh Kincaid Grade  - CLI  - Gulpease | - Lemmatized title words  - Lemmatized text words |

**Table 1 – Final Dataset Specification.**

**Explorative data analysis**

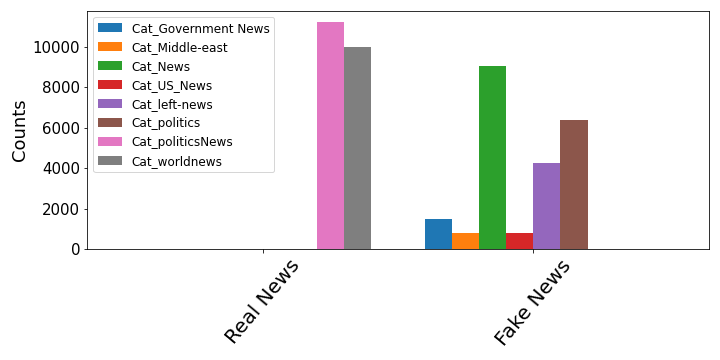
The final dataset included 38614 observations evenly distributed between real (55%) and fake (45%) (Exhibit 1) news which confirms that the two classes are well balanced.

**Exhibit 1. Dataset Classes distribution**



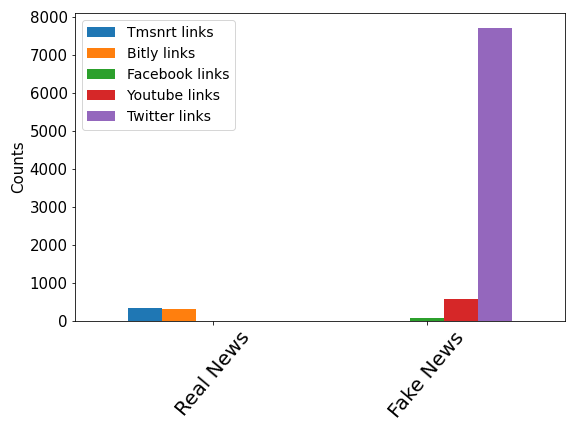
Above I mentioned that every news was assigned to one or more of the following categories: Cat\_Government News, Cat\_Middle-east, Cat\_News, Cat\_US\_News, Cat\_left-news, Cat\_politics, Cat\_politicsNews, Cat\_worldnews. The criteria utilized for this classification were not known. When data are grouped in real and fake tweets (Exhibit 2), real news belong only to either politics-news or world-news categories. Meanwhile, all the fake news were assigned to the remaining six categories. This dichotomic categorization is clearly reflected the different sources from real and fake news were gathered from. This classification will be not be included in the training and testing of the ML classifier.

**Exhibit 2. Real and fake new distribution among the eight categories.**



During data wrangling I introduced artificial tokens to track and analyze specific hyperlinks embedded in the text. Their frequency is shown in Exhibit 3. Tmsnrt and Bitly hyperlinks are found only in the text of real news. Conversely, Facebook, Twitter and Youtube hyperlinks are presently exclusively in fake news. This dichotomy may be the result of the different sources fake and real news came from, further supporting the intuition that two datasets may have some intrinsic dissimilarities. However, since these links are tokenized, they can be easily removed in the successive tuning steps of the ML model to assess their contribution to the overall prediction.

**Exhibit 3. Hyperlinks frequency in real and fake news.**

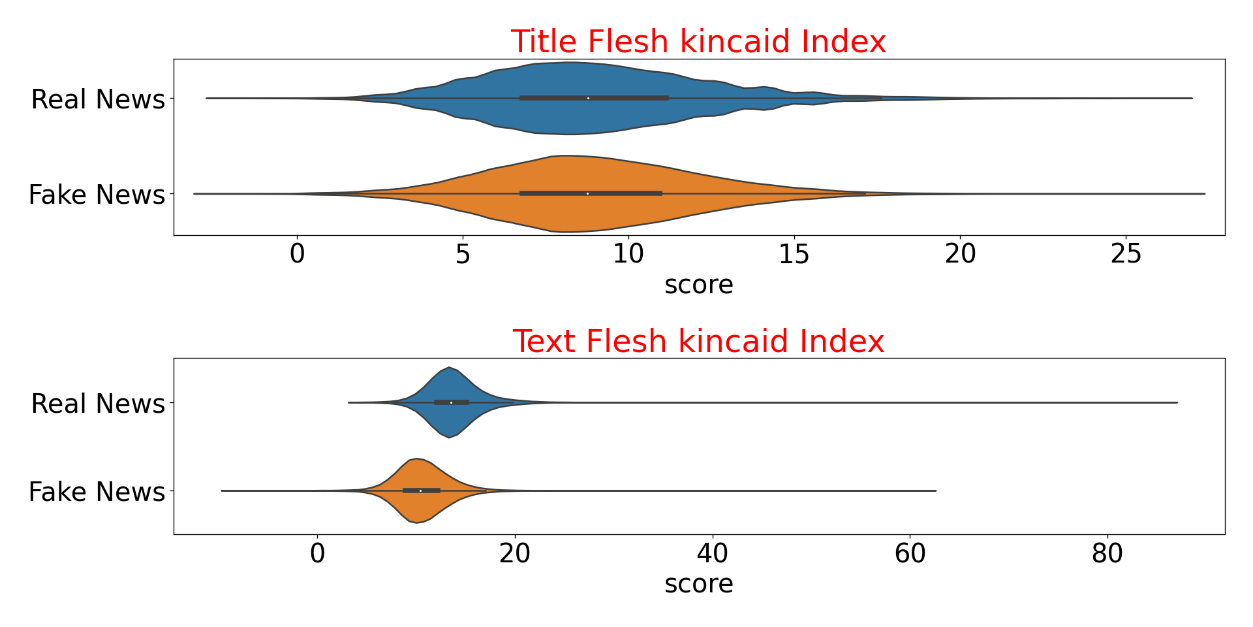


Linguistic indexes analysis

**This FILE contains all the details of the exploratory data analysis. Below, I reported three of the most relevant findings.**

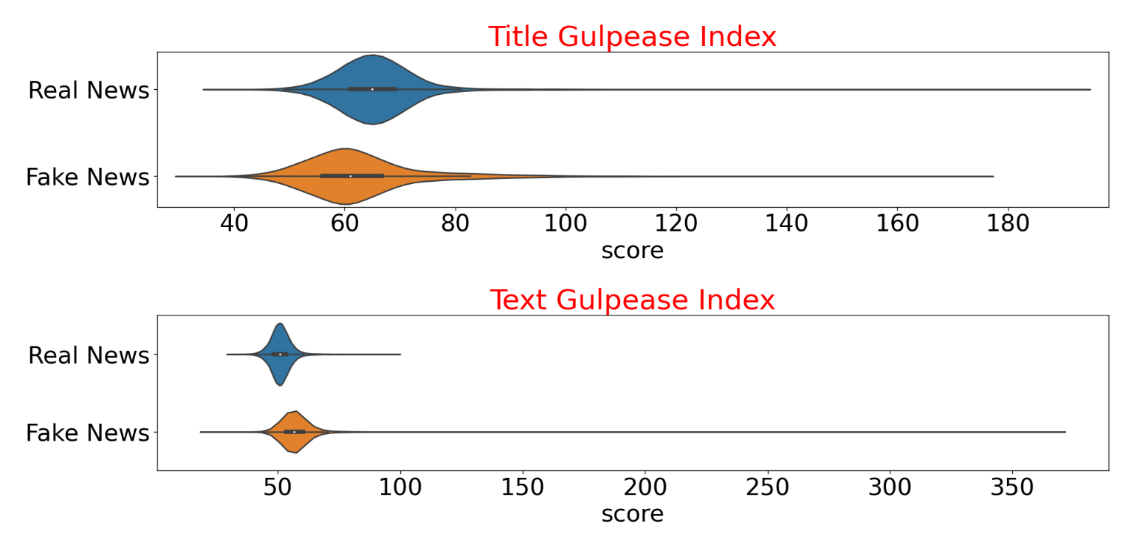
a) Flesh Kincaid index. This readability index indicates how difficult a passage in English is to understand. It is calculated using the number of words, sentences and syllables (Appendix). The Flesh Kincaid index of real and fake tweet titles is similar (Exhibit 4) (p~0.0813). However, when the text is analyzed, real tweets have higher Flesh Kincaid score and, thus, more complex texts (p<0.01). The text of several fake tweet has very small, even negative, Flesh Kincaid index.

**Exhibit 4. Flesh Kincaid index of title and text of real and fake news.**



b) Gulpease index. The Gulpease score is a readability index normalized on the Italian language. The title of real news has higher Gulpease score than title of fake news (p<0.01). This trend is inverted when the text is analyzed: Gulpease score is higher in the text of fake news (Exhibit 5).

**Exhibit 5. Gulpease score of title and text of real and fake news.**



c) Most common words in the news.

Exhibit 6shows the 20 most frequent words in real (top) and fake (bottom) news.The words *trump* and *say* are highly mentioned in both fake and real news. The word *reuters* is present in every real news. Tracking back to the source of the dataset of the real news, led me to discover that all the real news were indeed collected by the Reuterswebsite. This finding will be considered during testing of the optimization of the final predictive model.



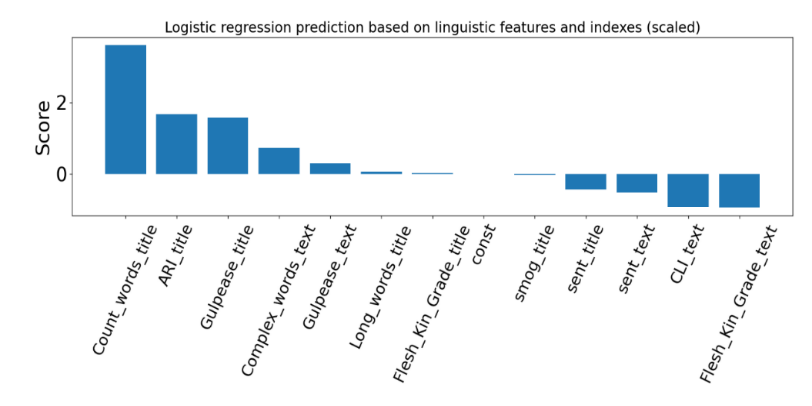
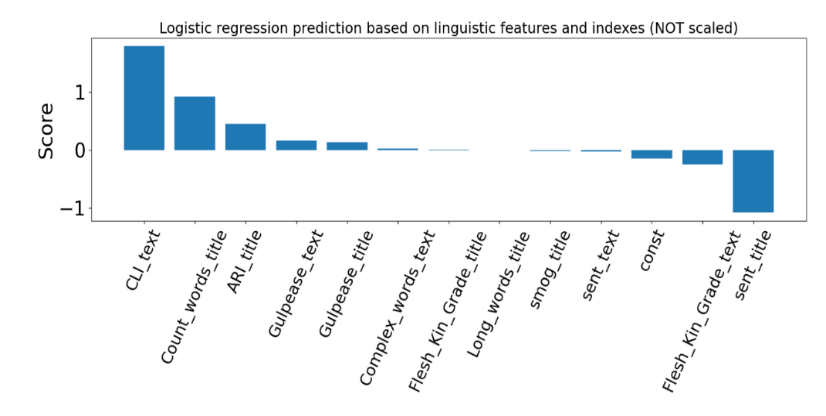
**Exhibit 6 – Most frequent words in the news.** The number of times the most common words are present in the news are reported on the left y-axis. The right y-axis shows the percentage of news having that word.

**MACHINE LEARNING MODELING**

**Models employed.** To predict the truthfulness of the news several models were implemented. Some models only utilized the linguistic characteristics of the text, some only the lemmatized words in the text, others both features. For details and the scripts please follow this [link](https://github.com/Gianl-msi/Real-Vs-Fake-News).

Model -1. Logistic Regressor with linguistic scores, text features

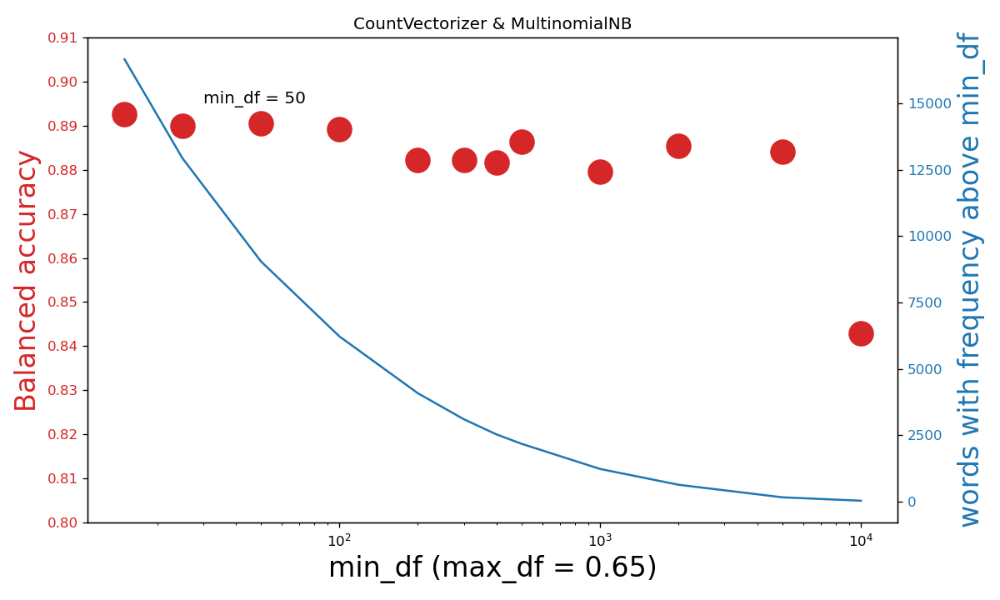
This is was one the first models to be deployed and its main goal was to highlight which linguistic scores and text features with the highest predictive power on the truthiness of the news. Even though this model was the worst performant (balanced accuracy = 0.760, AUC = 0.939) among all the models tested, it helped single out which linguistic feature best separate real from fake news: number of words in the title, the Flesh Kincaid index of the text (Exhibit 7, top chart), CLI index of the text and the number of sentences in the title (Exhibit 7, bottom chart).



**Exhibit 7 – Logistic regression coefficients for the linguistic indexes and text features.**

Model - 2. Multinomial Naïve Bayes with CountVectorized lemmatized text

Next, I investigated the predictive power of the lemmatized words in the text of the news. The first parameter I adjusted was the lower cut-off for document frequency for the vectorizer (min\_df). While lower values for min\_df tend to improve prediction accuracy, it also is computational expensive as the matrix representing the frequency of the words is much larger. Exhibit 8 depicts how model accuracy changes with different values of the minimum word frequency.

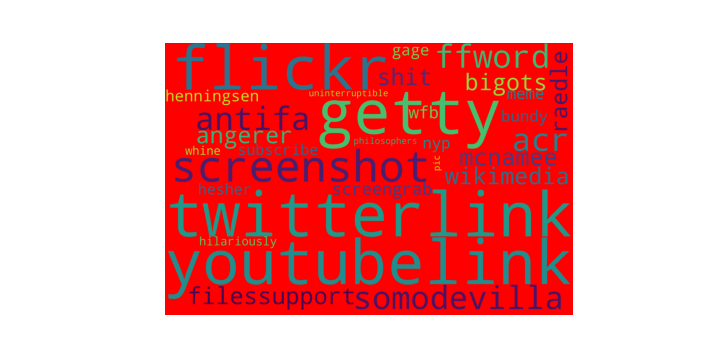
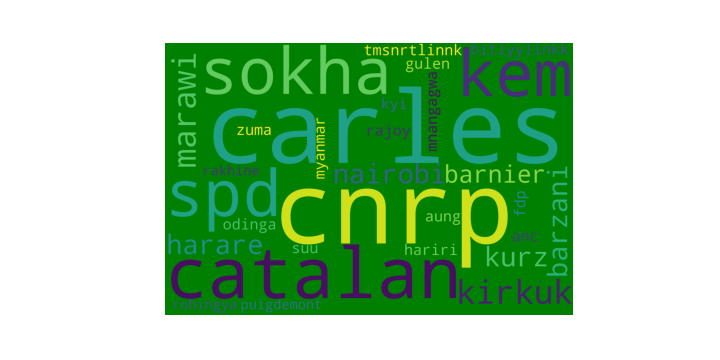


**Exhibit 8. Model 2 performance with increasing document frequency.**

Even though the maximum balanced accuracy is reached when each word appears in at least 15 documents, I selected the value of 50 for min\_df for the reason hinted above: speed. The accuracy when min\_df is 50 is just slightly lower than the balanced accuracy obtained with min\_df is 15 (0.8926 vs 0.8905), but the prediction is much faster.

Next, I extrapolated the 60 most predictive words for real and fake news (Exhibit 9). Vulgarities, twitter/youtube links and flickr/getty related words (*getty*, *screenshot*, *flickr*, *getty,* *raedle*, *angerer*) are enriched in the fake tweets. On the other hand, real tweets are characterized by the presence of bitly/tmsnrt links and name of non-mainstream politicians (*Odinga* - Kenyan politician, *Rajoy* - Spanish politician, *Hariri* - Lebanese politician, *Gulen* - Turkish Islamic scholar) and name of cities and countries rarely mentioned by the mainstream media (*Kirkuk* - city in Iraq, *Nairobi* -Kenya’s capital, *Marawi* - Islamic city, *Harare* - capital of Zimbabwe, *Rakhine* - state in Myanmar). Several acronyms appear in both real and fake tweets. Their meaning is reported in the appendix. Furthermore, if a picture is embedded in the text of the news, it is more likely to be fake. As It can be noted from the word cloud analysis, words associated with the photography field frequent in text of fake news (*getty*, *somodevilla*, *angerer*, *screengrab*, *raedle*, *wikimedia*). Finally, vulgarities (*f\*\*k,* *s\*\*t*, *bigots*) are almost exclusively only found in fake news.

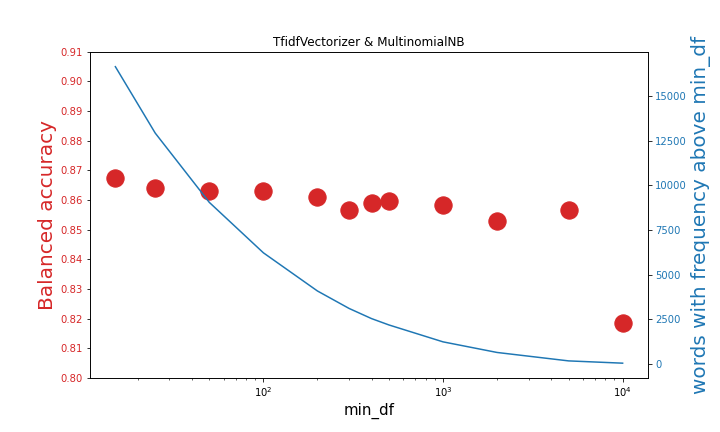
**Exhibit 9 – Word cloud for the words with highest predict power for real (left) and fake (right) news.**



To improve the performance of the model, different n-grams for the CountVectorizer were assessed. At the same time multiple values of the smoothing parameter alpha were evaluated. The model reached the highest balanced accuracy of 0.9467 (AUC = 0.9954) with n-grams of 2, 3 and alpha = 0.01.

Model – 2. Multinomial Naïve Bayes with TFIDFVectorized lemmatized text.

The lemmatized words of the text of the news were fed into the TFIDFVectorizer and modeled with Multinomial Naïve Bayes. I found again that the highest accuracy is reached with the lowest min\_df (Exhibit 10) but I decided to select the value of 50 for a faster prediction.



**Exhibit 10. Model 3 performance with increasing document frequency.**

Next, different n-grams and values of the smoothing parameter alpha were evaluated. While the overall performance increased, this model was still not able to outperform the best performant model 2. Exhibit 11 graphs the performance of model 2 (MultinomialNB with CountVectorizer) and model 3 (MultinomialNB with TFIDFVectorizer) side to side. The arrow indicates the most accurate model.



**Exhibit 11. Model 2 & 3 balanced accuracy comparison with different n-grams.**

Models 4 & 5. Logistic Regressor with lemmatized words, linguistic scores, basic text features

Next, I concatenated the vectorized lemmatized words with the linguistic features and trained a logistic regressor algorithm. Model 4 and 5 differs by which vectorizer was implemented. In model 4 I used the CountVectorizer, in model 5 the TFIDFVectorizer. For both vectorizers I set min\_df to 50 and n\_gram range to 2,3. Different value for the regularization hyperparameter C of the logistic regressor were tested. Among the two models, model 4 is the highest performant.

Models 6 & 7. Random Forest Classifier with lemmatized words, linguistic scores, basic text features

Finally, a random forest classifier was implemented. As input, I plugged in the vectorized lemmatized words concatenated with the linguistic features. For model 6 I used the CountVectorizer, in model 7 theTFIDFVectorizer. When the CountVectorizer is utilized the random forest classifier performs slightly better, which perfect aligns with the previous observations.

**Models comparison.**

Table 2 summarizes the characteristics and performance of the 7 tested models. Model 4 has the highest balanced accuracy and the second highest AUC score. Meanwhile, model 5 delivered the highest AUC score and second highest balanced accuracy. Both models undoubtedly produce very satisfying prediction. I choose model 4 for next few analysis.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Classifier | Extraction Tecnique | Linguistic scores & text features | Balance Accuracy | AUC  Score |
| Model 1 | Logistic Regressor  (C=0.1) | None | YES | 0.7524 | 0.938 |
| Model 2 | MultinomialNB  (alpha =0.01) | CountVectorizer  (min\_df = 50, max\_df =0.65, ngram\_range = 2,3) | NO | 0.9467 | 0.9954 |
| Model 3 | MultinomialNB  (alpha =0.01) | TFIDFVectorizer  (min\_df = 50, max\_df =0.65, ngram\_range = 2,3) | NO | 0.9385 | 0.9956 |
| Model 4 | Logistic Regressor  (C=10) | CountVectorizer  (min\_df = 50, max\_df =0.65, ngram\_range = 2,3) | YES | 0.9836 | 0.9993 |
| Model 5 | Logistic Regressor  (C=200) | TFIDFVectorizer  (min\_df = 50, max\_df =0.65, ngram\_range = 2,3) | YES | 0.9831 | 0.9997 |
| Model 6 | Random Forest (n\_estimators = 100, min\_samples\_leaf = 1 ) | CountVectorizer  (min\_df = 50, max\_df =0.65, ngram\_range = 2,3) | YES | 0.9744 | 0.9989 |
| Model 7 | Random Forest (n\_estimators = 100, min\_samples\_leaf = 1 ) | TFIDFVectorizer  (min\_df = 50, max\_df =0.65, ngram\_range = 2,3) | YES | 0.9714 | 0.9988 |

**Table 2 – ML model summaries.**

Accuracy of model 4 is very high, but not 100 %. The confusion matrix shows that, most of the misclassifications (61) are false negatives (41). Below I provide an excerpt of the a fake news which was classified as real by the classifier:

*“the people have spoken and it is clear that americans are 100% fed up! with the sanctuary city policy and illegal immigration. embattled san francisco sheriff ross mirkarimi convincingly lost his bid for re-election tuesday after spending months in the national spotlight as the face of his city s controversial sanctuary city policy on illegal immigration. the murder of kate steinle brought to the forefront the sanctuary city policy in san francisco. the refusal to acknowledge the illegality of this policy and the treatment of the steinle family could not be more shameful….*

Assuming that no mistakes were made during compiling of the fake news dataset, what stands out is that the text above could easily belong to real news. There is nothing in the text that would lead the reader to suspect fakeness. There are no vulgarities, no excess of punctuation, the grammar is correct, and the document even conveys a sound message. This is a puzzling observation which will require further studies.

Next, I analyzed the false positives, e.i. real news classified as fake. Below I reported one of the 20 misclassified real news:

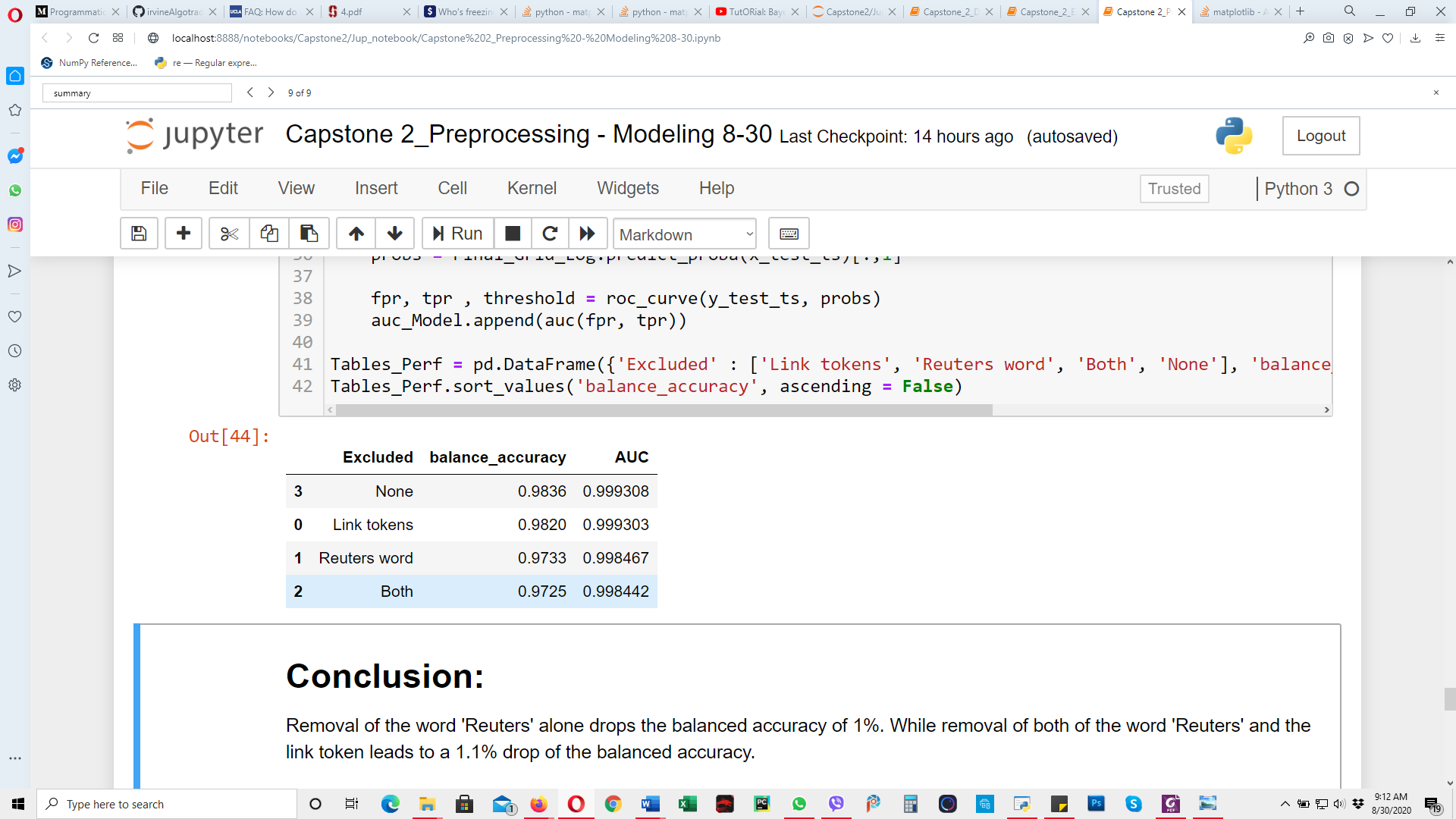
*“(reuters) - drugmaker valeant pharmaceuticals international inc's (vrx. to) new chief executive, joseph papa, will receive a base salary of $1. 5 million, the company said in a filing on wednesday. valeant said on monday former perrigo co plc (prgo. n) head papa would replace michael pearson as its ceo. papa will also receive a cash payment of $8 million to make up for the equity-based compensation he forfeited in connection with the termination of his employment with perrigo. valeant said in march pearson was leaving the company, just three weeks after returning from a two-month medical leave. pearson's base salary for full-year 2014 was $2 million and his total compensation amounted to $10. 3 million, according to a regulatory filing. (1. usa. gov/1aapxim)”*

The reader would immediately notice the presence of the word *reuters* which is a hallmark of every real news. One other thing that stands out is the shortness of news: it has indeed a small Flesh Kincaid score, 8.8 (the average text Flesh Kincaid score for real and fake news is 13.8 and 10.6 respectively). Exhibit 12 shows the distribution of text Flesh Kincaid scores for real and fake news. Every single vertical green line is a misclassified real news. From this simple analysis is appears the shortness and, thus simplicity of the text may have contributed to the misclassification of a subset of real news. While this observation is not enough to establish a cause-effect relationship, it suggests that real news with shorter, simple text are more prone to misclassification than real news with longer, complex real text.



**Exhibit 12. Flesh Kincaid Score for the misclassified real news**

Next, I investigated how important the word *Reuters* and the tokens I introduced during data wrangling are for the model prediction. As mentioned above that word *Reuters* appears in every real news while the tokens frequency distributions among real and fake news are fully dichotomic. The word *Reuters* alone is responsible for 1% of the total balanced accuracy of the model. Meanwhile, the hyperlink tokens contribute to just less 0.02% to it (Table 3).



**Table 3. Weight of the word Reuters and hyperlink token in the final prediction.**

**CONCLUSION AND FUTURE DIRECTION**

The unchecked spreading of fake information has become a real problem for social media platforms and online information outlets, and it has prompted data science and machine learning communities to develop precise tools for identification and successive removal of untrue information. Herein, I created and tested several models for the accurate discrimination of real news from fake news. These modeIs leverage NLP techniques as well as linguistic complexity and indexes of title and text of the news. One important limitation of this study is that real and fake news were collected from two different sources, suggesting that difference in collection and labelling may have favorited prediction in a biased way. Despite this issue, there is space for improving the model and thus, numbing this suspected bias. For instance, the document frequency parameter min\_df could be lowered to include rare words and to solve the computational time, the script could be run on a faster machine. One more feature that could be added is the amount of punctuation in the text of the document. Fake news’ text tends to have an excess of exclamation marks, asterisks and other punctuation marks. The text could also be analyzed for spelling and grammar errors. Finally, a more sophisticated vectorizer such as word2vec could be implemented to maintain and analyze words’ semantic in text and title of the news.

In conclusion I developed a high performant machine learning model for the discrimination of real from fake news. While my model may have some limitations, alternative courses of action are available to improve its robustness, reliability and predictive power.