

FAKE NEWS DETECTION

Submitted by:

Shubham Mishra

**ACKNOWLEDGMENT**

Below references are used, which helped in the completion of project:

1. <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>
2. <https://www.datacamp.com/community/tutorials/scikit-learn-fake-news>
3. <https://medium.com/swlh/build-your-own-fake-news-classifier-7918f05c2ec7>

**INTRODUCTION**

* Business Problem Framing

Classify whether the news is genuine or not, based upon the raw data, which includes excerpts from news websites like- ‘The New York Times’ or via social media platforms like- Twitter.

This problem has a relativity with real world, since nowadays people are more relying on the social media, as every single click, every single tweet is being noticed, and thus, it’s quiet easier to spread unauthentic information via news alerts on these social media platforms. Sometimes even the big news websites also spread falsify information, in order to increase the TRP’s or clicks on their news pages, to attract the audience, or sometimes celebrities even spread fake news for example- recently during US presidential elections, there were a huge amount of false information, which was spreading over Twitter and Facebook, related to ongoing voting counts. Though some of the times, it’s helpful as well, when the news spread is real, for instance- during coronavirus spread, in earlier stages, some of the websites are providing real-time information, related to patients and hospitals scene, which might have been helpful to people, who were at that time living in the most affected areas.

* Conceptual Background of the Domain Problem

A person should be aware of the fact that, mostly news websites name or their author name would be legit most of the times in their news excerpts, whereas, mostly social media news would have higher chances of being fake, as people create fake accounts and spread unauthentic information. Apart form that, if the data is from Twitter, we can separate the hashtags and @RT and emoticons (to express emotion of tweet, whether sarcastic, angry, sad, happy etc.), to understand more about who has written the particular piece of information, and whether their sentiment is positive or negative, based upon the hashtags used. While, in news websites, hashtags might come, but usually, news headlines are in the form of sentences, without hashtags or smileys.

* Review of Literature

It can be seen from the dataset ‘train\_news.csv’ file that the data has come from variants source of mediums like news websites- The New York Times, Breitbart etc., podcasts, videos, tweets, and other social media form. Apart form that, few of the text excerpts only consisting of smileys.

There are various features that can be generated-

1. Social media platform- Twitter, Facebook, News websites etc.
2. Topic Modelling – to understand what the top 10 topics are, related to this specific news excerpts, e.g., covid- 0.70, hospital- 0.20, china- 0.05 etc, and then, can check which are the top topics related to ‘fake’ and ‘not fake’ news.
3. Location- for some of the news excerpts, location is also given either in ‘headline’ or ‘news’ column.
4. Language- if required, as people write tweets in any language or a new website is for a particular medium audience e.g., in Spanish language.

(Due to restricted amount of time, I couldn’t be able to create these features or perform feature extraction. )

In the above dataset, both ‘fake’ and ‘nor fake’ news excerpts are in equal proportion.

**Analytical Problem Framing**

* Data Sources and their formats

There are total 6 features present:

1. “id” (int): Unique id of each news article.
2. “headline” (Object): Title of the news.
3. “news” (Object): Text of the news article
4. “Unnamed:0” (int): Serial number
5. “written\_by” (Object): Author of the news article
6. “label” (int): It tells whether the news is fake (1) or not fake (0).

Dataset Snapshot-



There are missing values present:

headline 558

written\_by 1957

news 39

* Data Preprocessing Done

Since, I have worked upon ‘news’ feature, to apply statistical models, thus, doesn’t care about missing values in other columns. Deleted the 39 rows, which contains null values.

And the remaining news data left is- 20761 rows.

These are few other Data Preprocessing steps, performed:

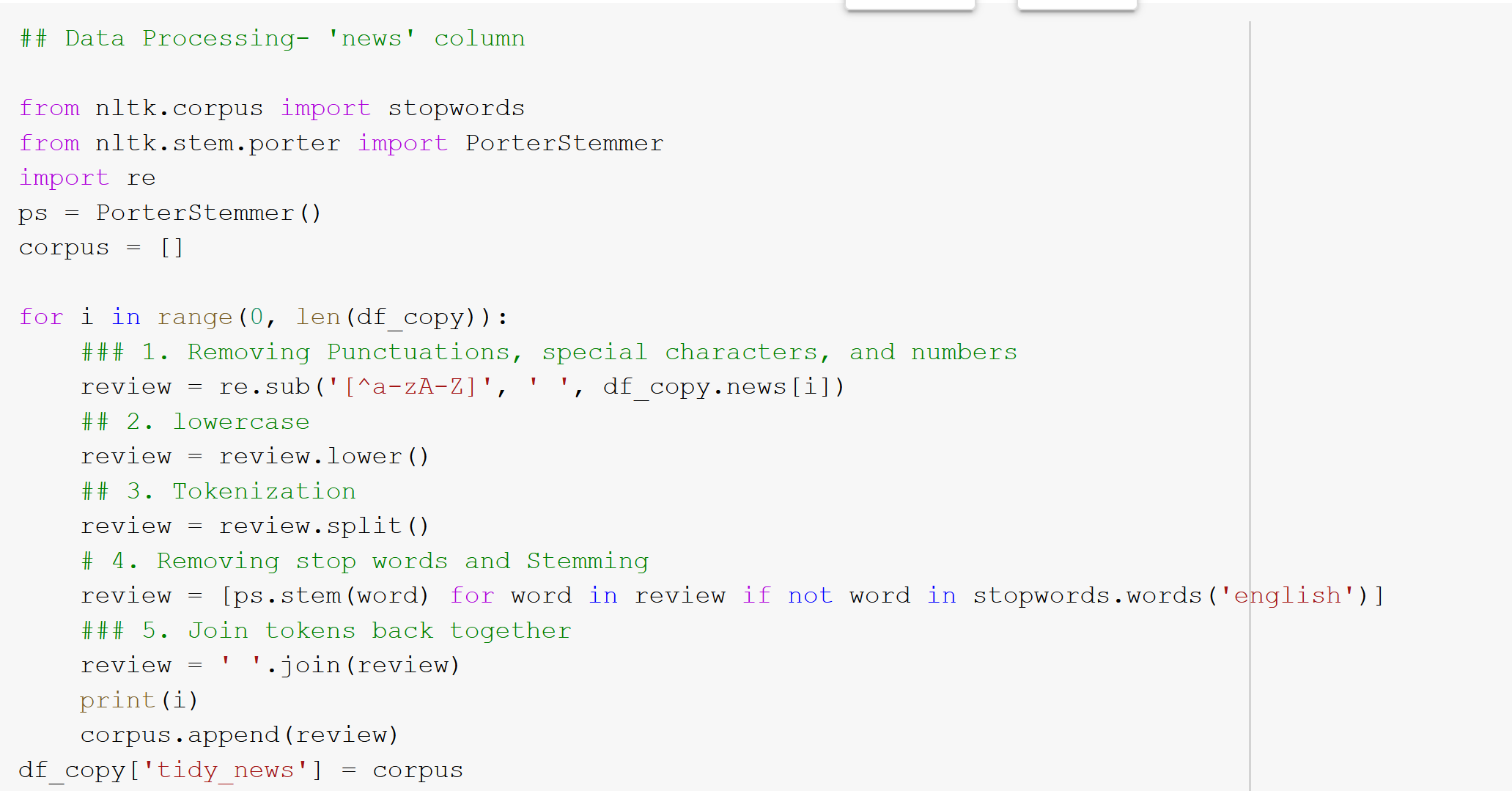
  1. Removing Punctuations, special characters, and numbers

2. lowercase

3. Tokenization

4. Removing stop words and Stemming

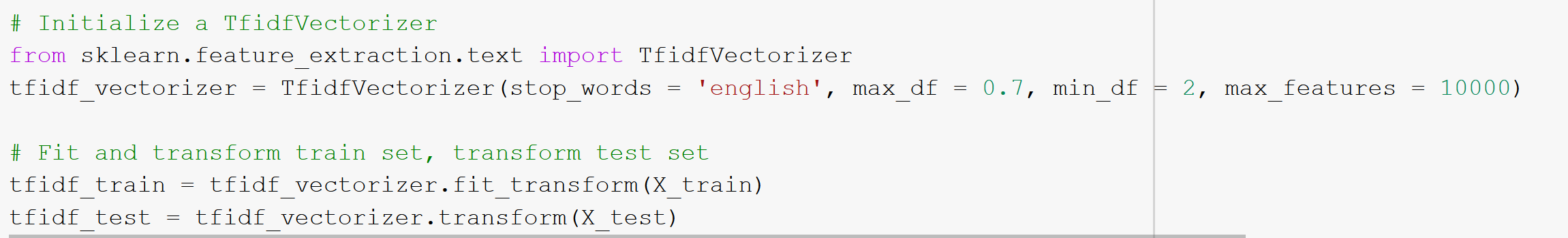
5. Join tokens back together

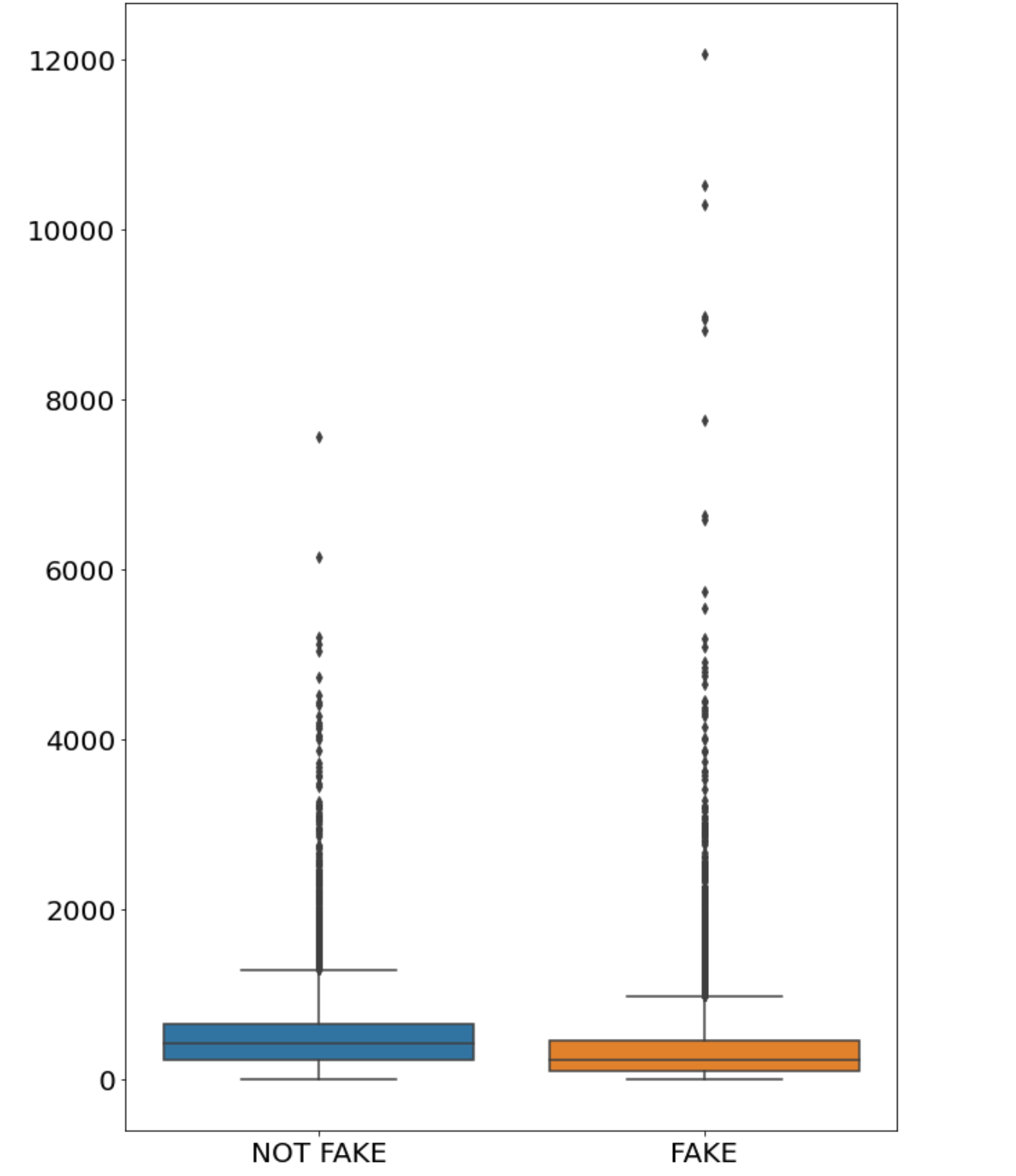


1. Transform tokens to vectors, using TF-IDF Vectorizer.

TFIDF works by penalising the common words by assigning them lower weights while giving importance to words which are rare

in the entire corpus but appear in good numbers in a few documents. Also, I took max\_features as 1000, as by tweaking it to higher values too, there are not enough changes in the validation output of Statistical models.





From the above boxplot, it can be seen that in both ‘Fake’ and ‘Not Fake’ classes, most of the values lies under length= 1000, while rest are the outliers.

* Hardware and Software Requirements and Tools Used

Python

NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, Tensorflow 2.0, Keras, Tensorboard.

**Model/s Development and Evaluation**

* Testing of Identified Approaches (Algorithms)

I have used the following algorithms for training and testing of dataset:

Passive Aggressive Model, XGBoost model , LSTM and Naïve Bayes model, with Hyperparameter tuning.

* Models used-

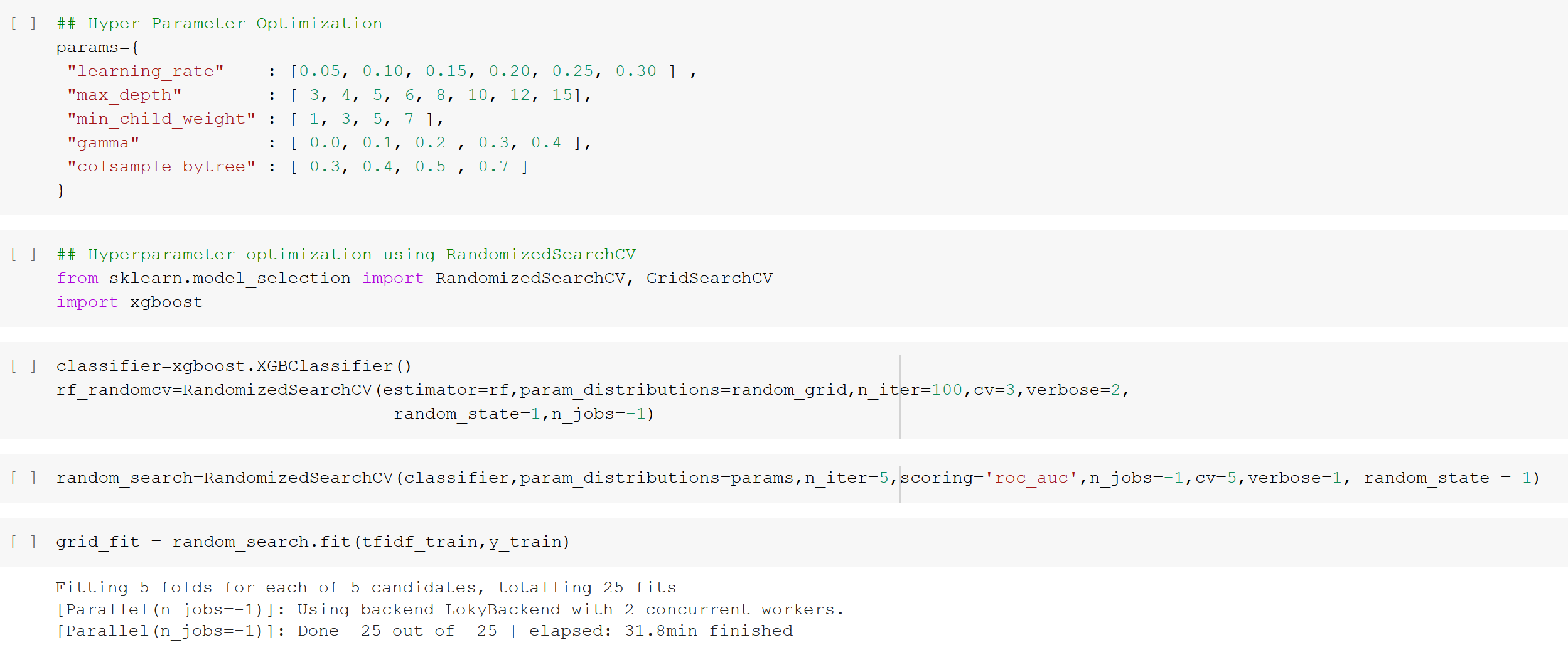
1. Passive Aggressive Model-



1. XGBoost Model-
2. w/o Hyperparameter Tuning-

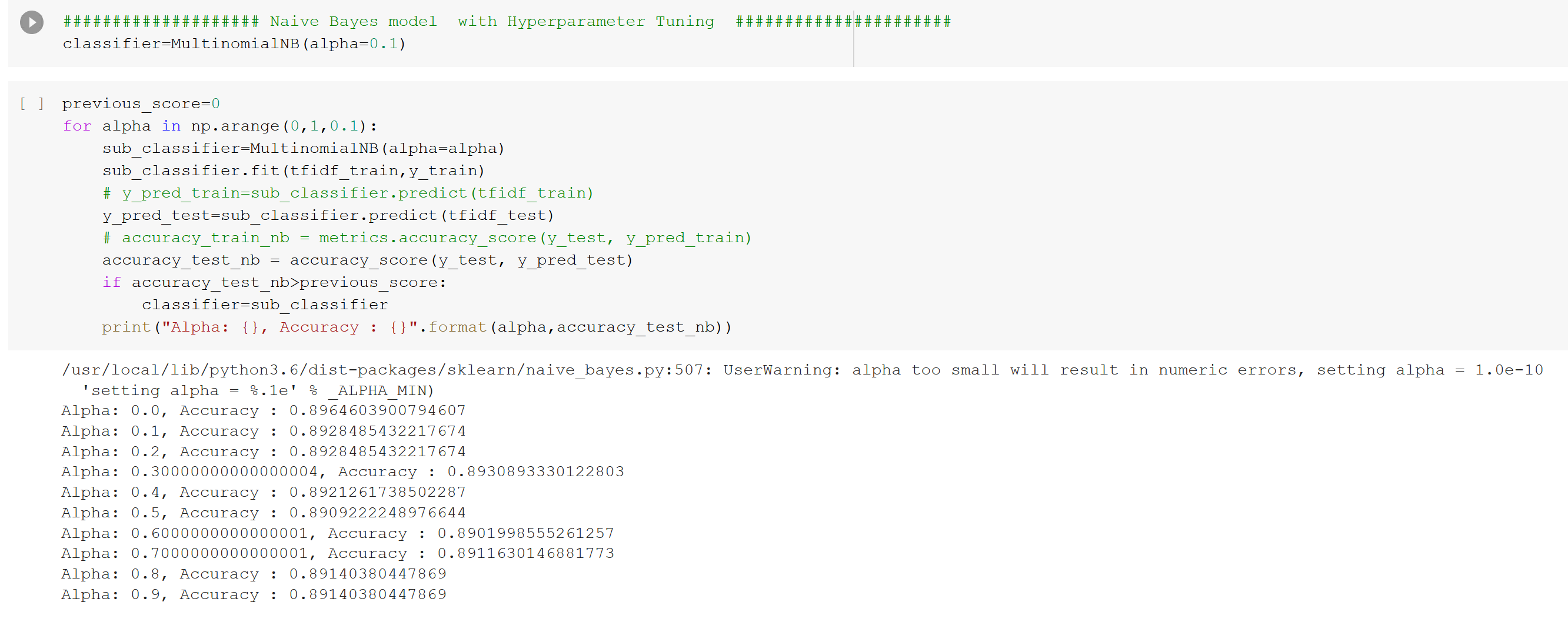


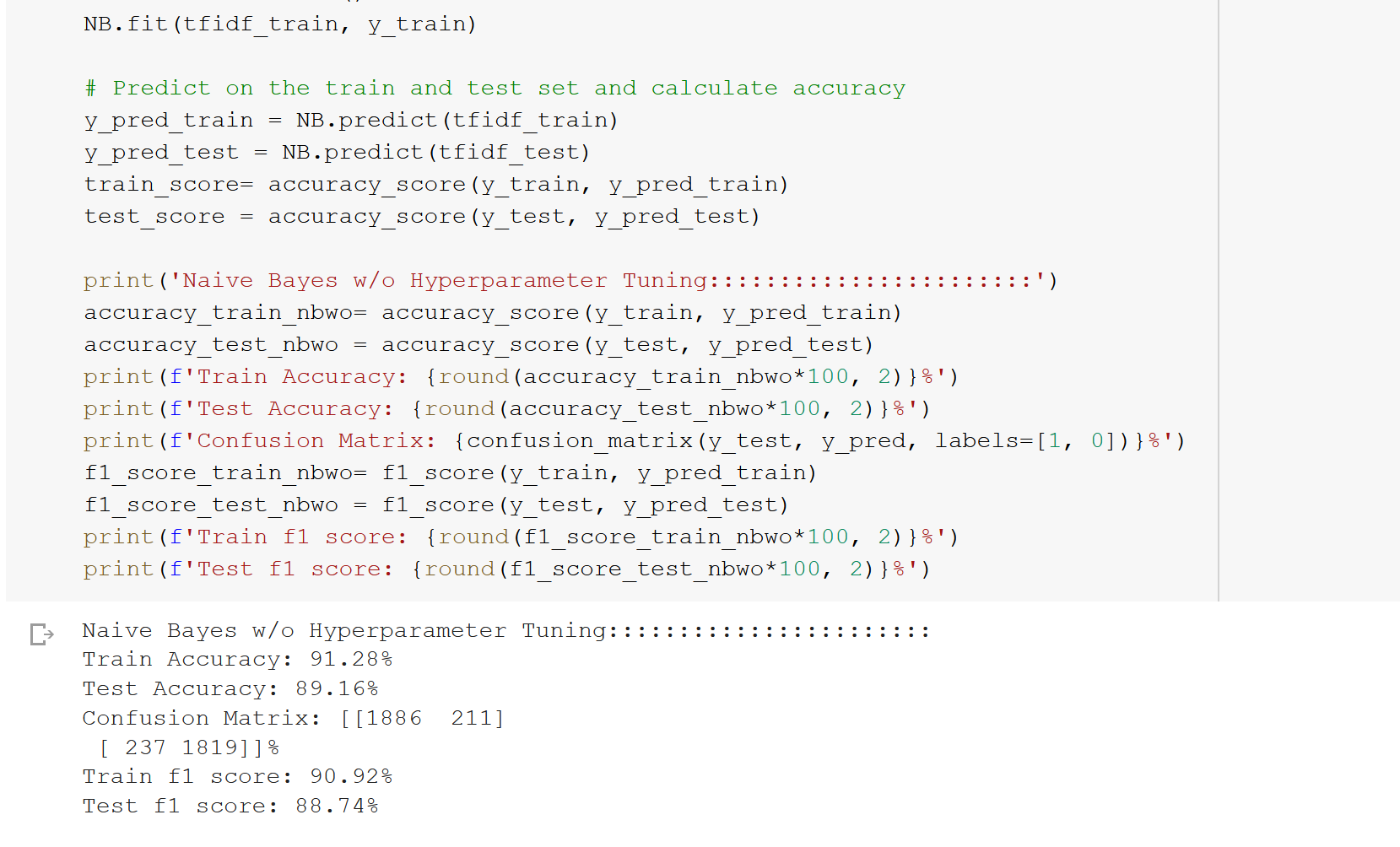
1. with Hyperparameter Tuning-



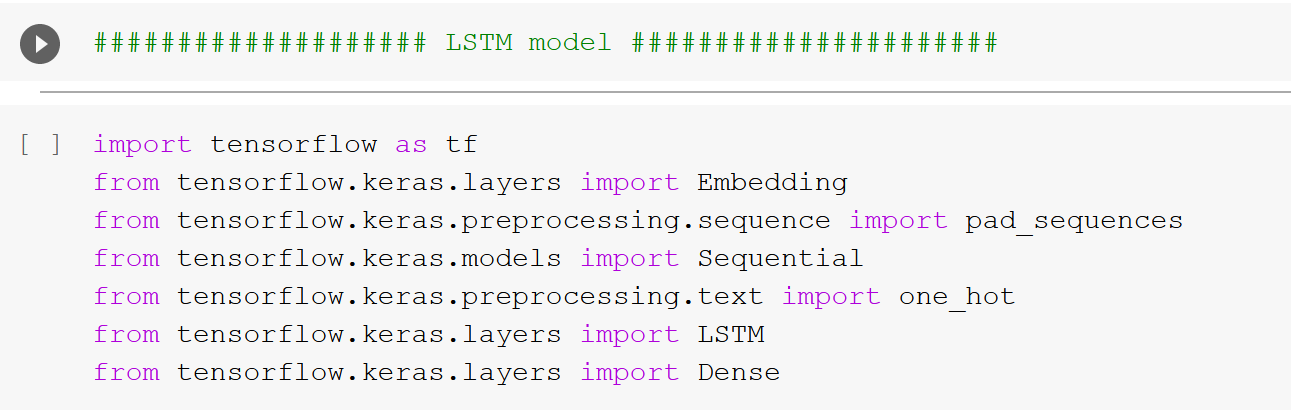


1. Naïve Bayes Model (with hyperparameter tuning)-

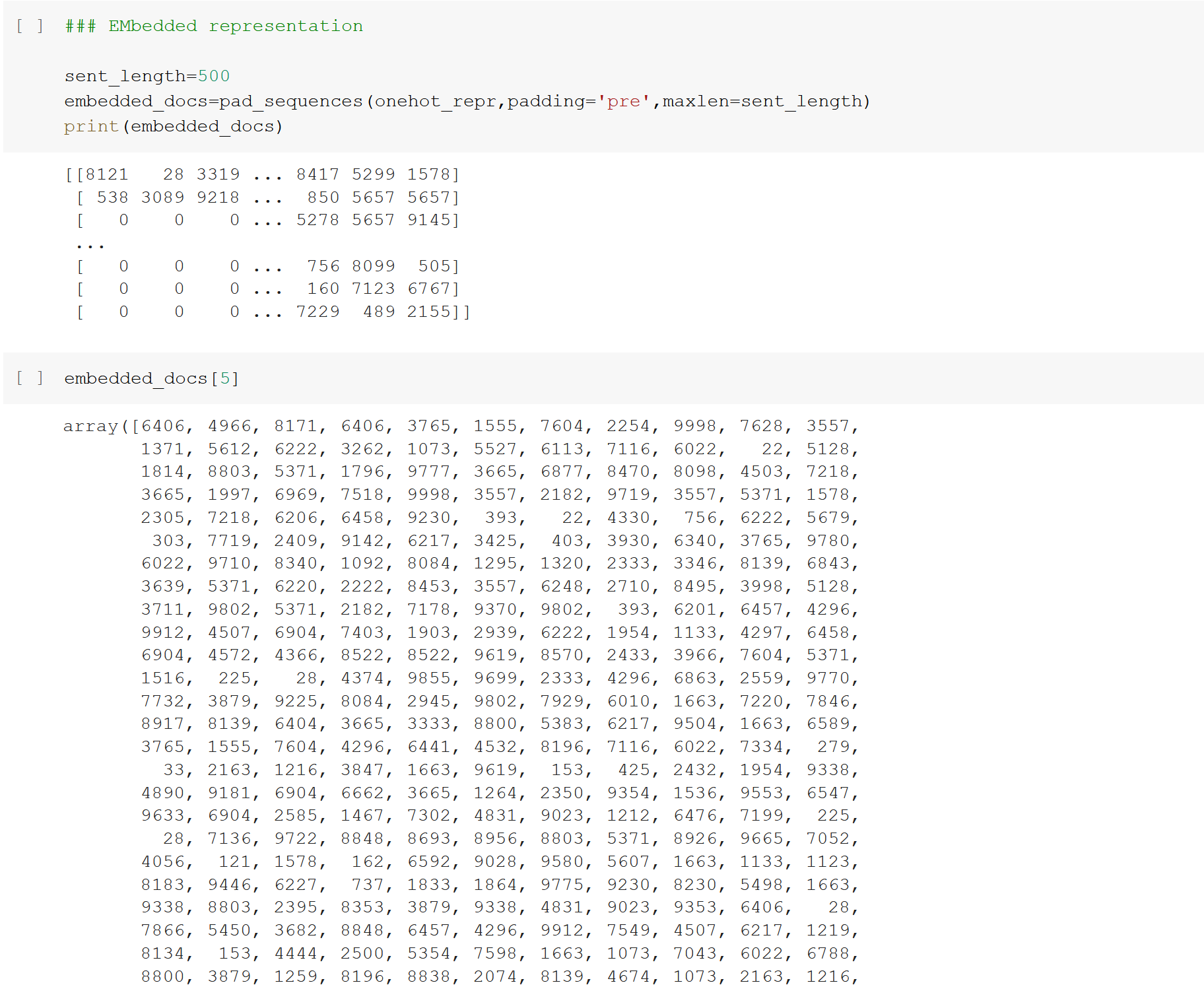


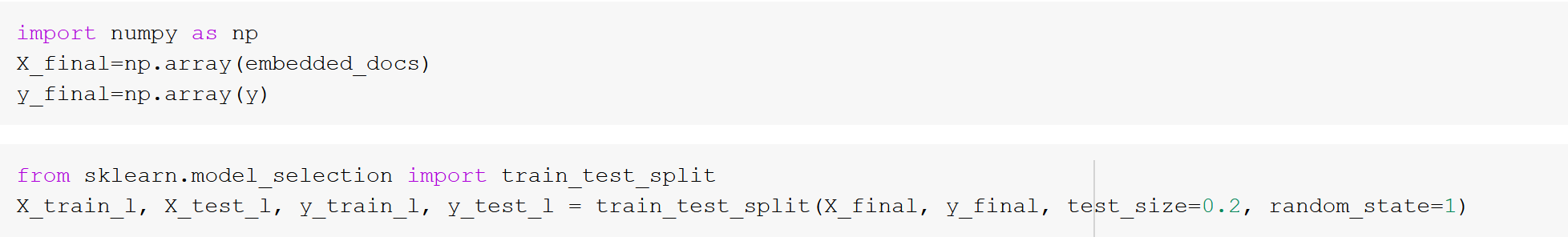


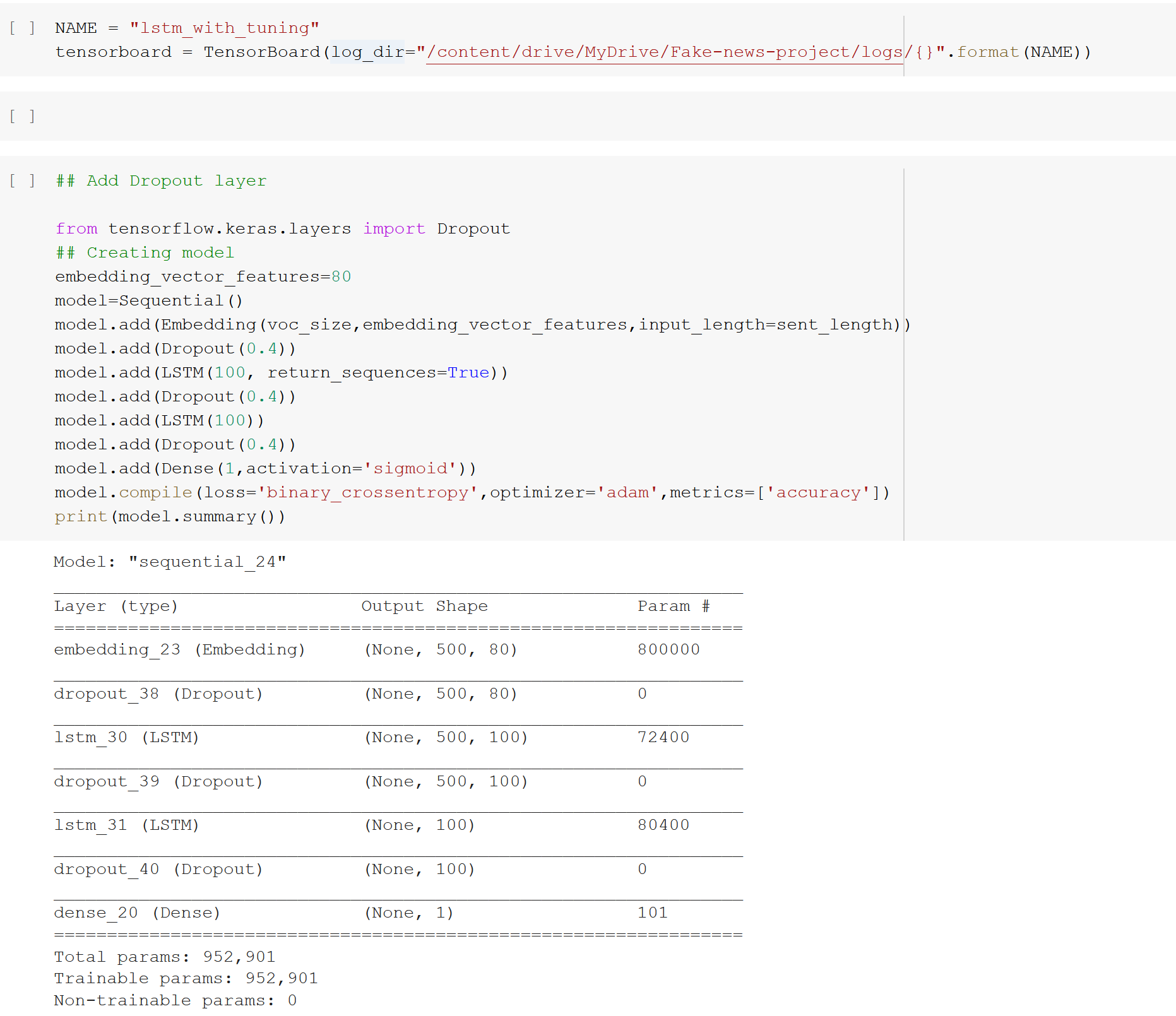
1. LSTM model (with hyperparameter tuning)-

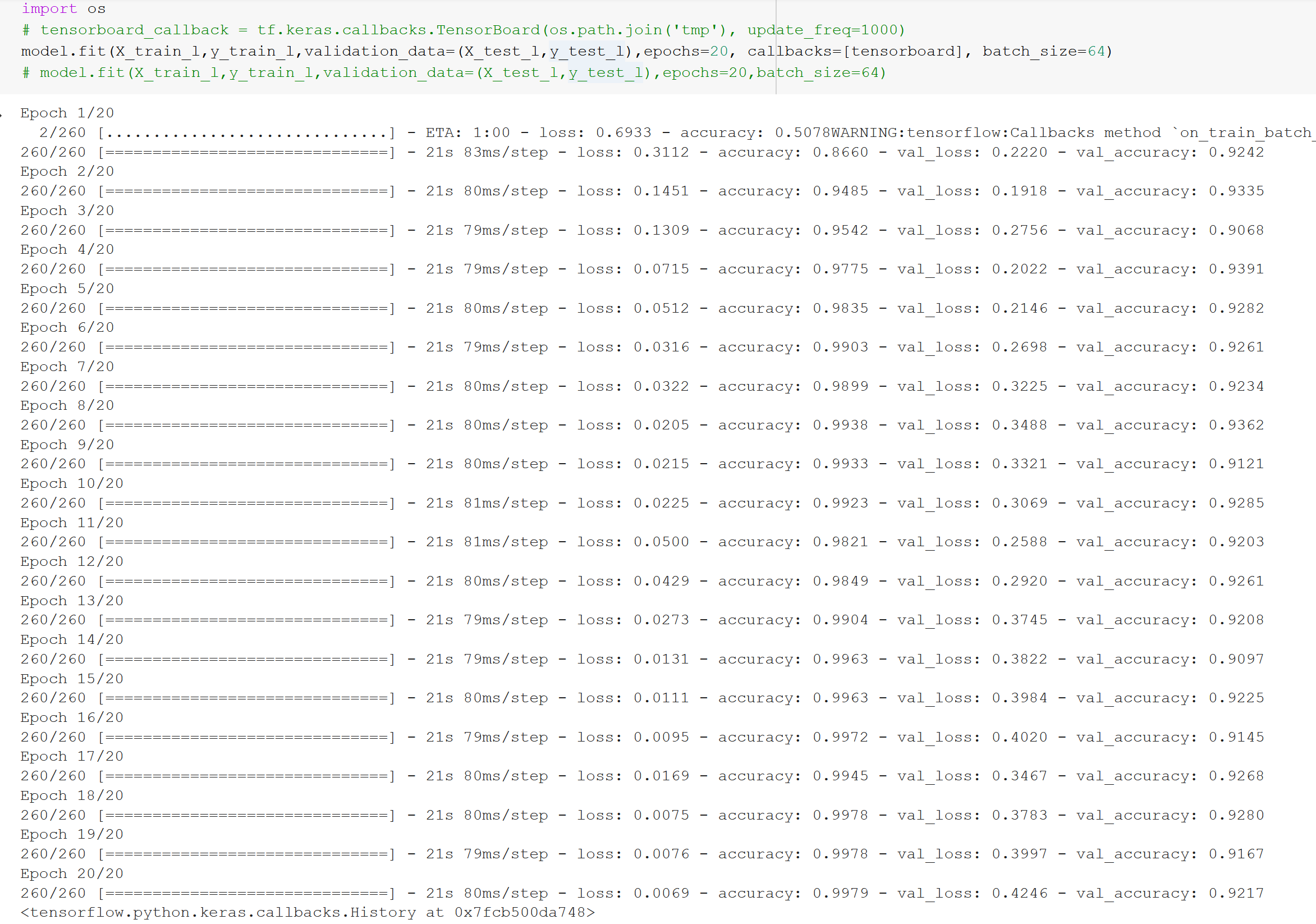










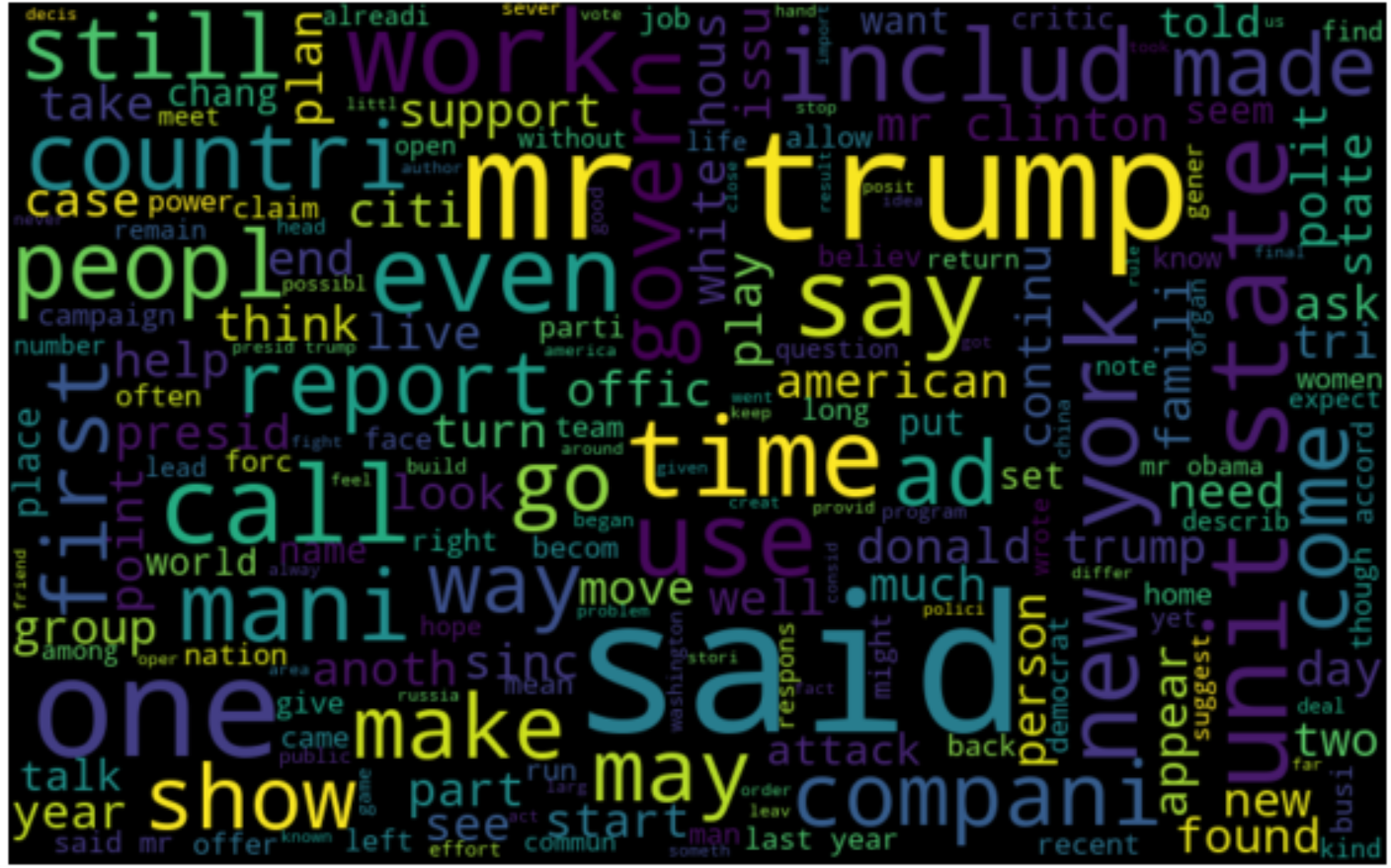


* Key Metrics for success in solving problem under consideration

I have used ‘accuracy’ score and ‘f-1 score’ metrics for solving problems. Though, problem tells to retrieve less number of false negatives (since, should remove ‘fake’ news).

* Visualizations

**Plot of ‘not fake’ news:**



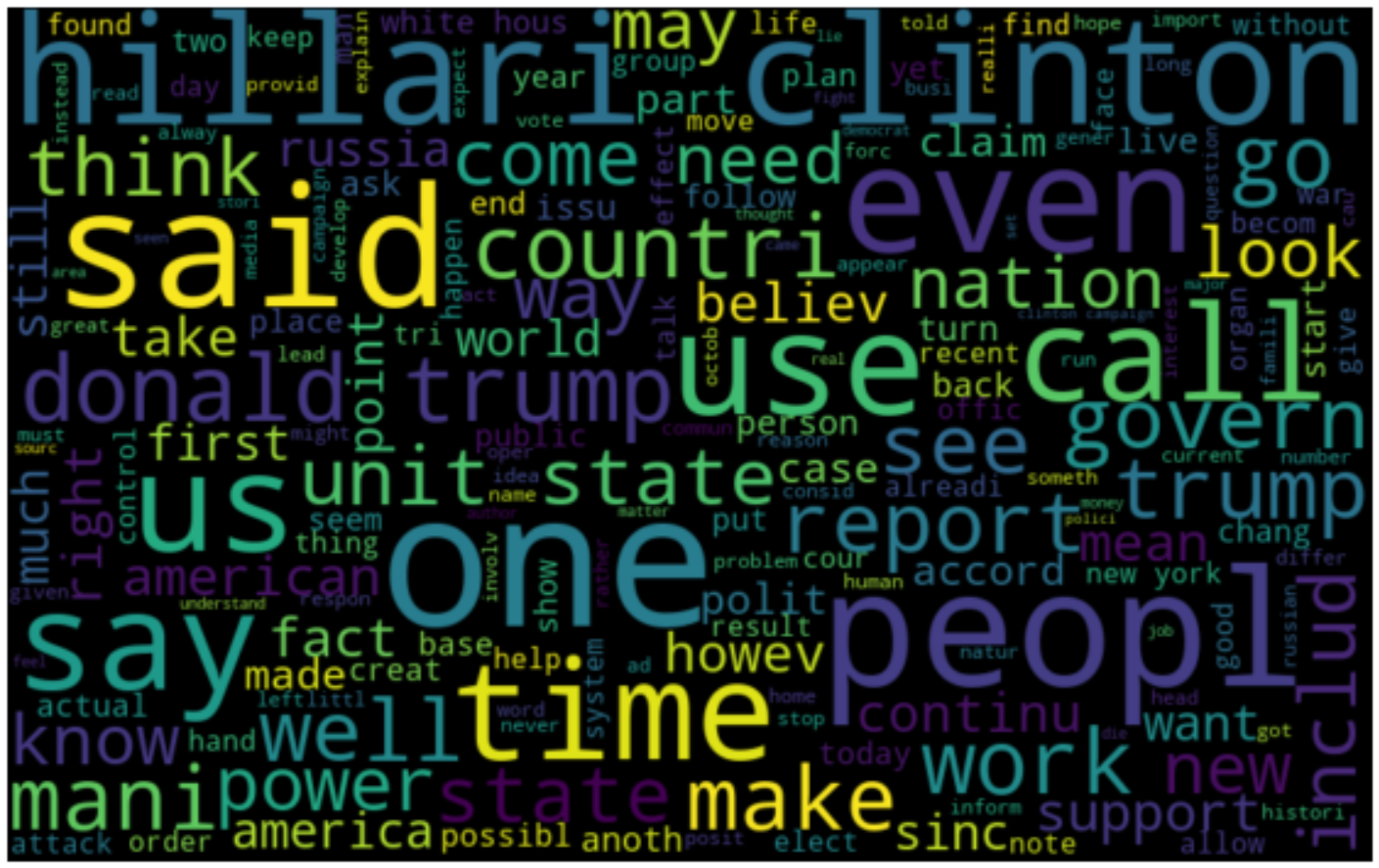
From the above Word Cloud of ‘not fake’ classified news, it can be depicted that these words are displayed mostly: trump, work, companies, American, new York etc.

**Plot of ‘fake’ news:**

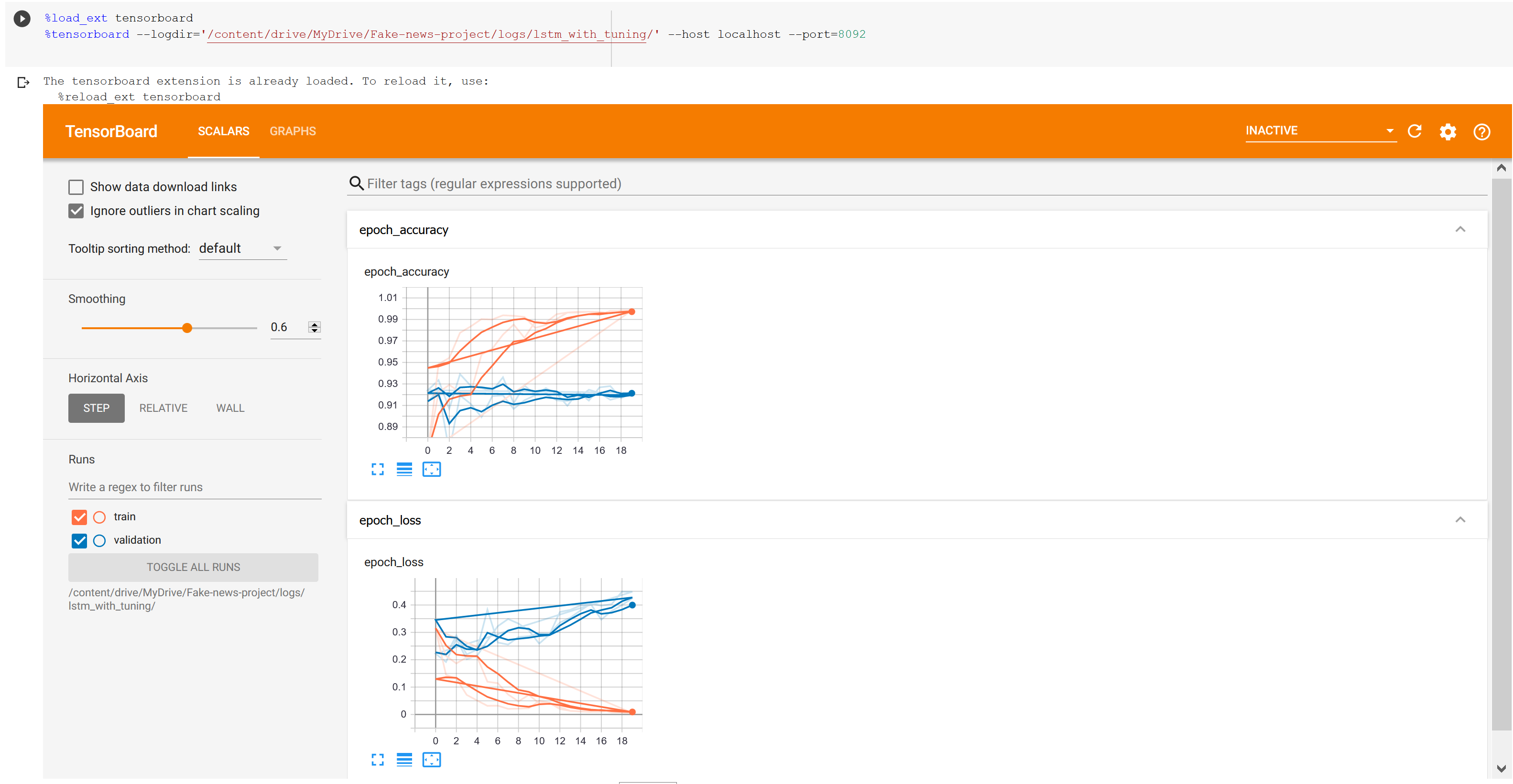
Whereas From the below Word Cloud of ‘fake’ classified news, it can be depicted that these words are displayed mostly: hilary Clinton, people, Donald trump etc.

Though in both ‘not fake’ and ‘fake’ news, ‘trump’ word is mostly used, as being the president of United States, he is most popular. And also, there are various falsified rumoured news always spreading around him, due to his popularity.

Also, from the above word clouds, there are lots of cleaning which can be done. E.g., ‘said’,’made’, ‘come’, ‘call’, ‘see’, ‘work’ etc. can be removed.

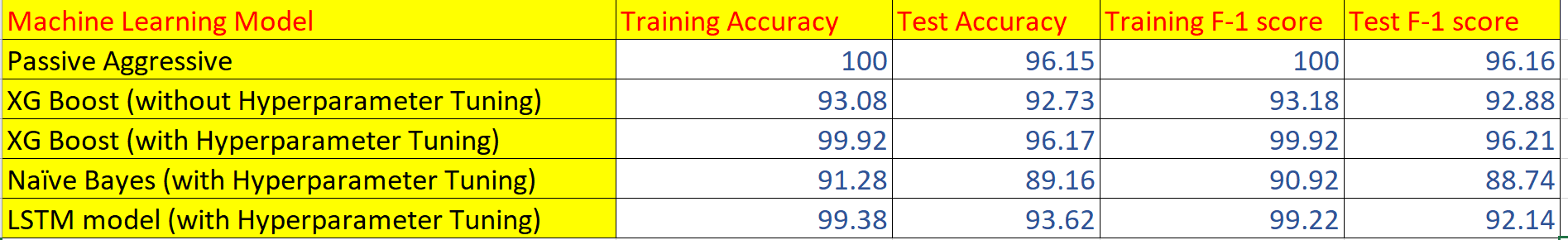


Tensorboard results for train and test datasets ‘accuracy’ and ‘loss’ of LSTM models**-**



Here, Blue color line depicts, Validation dataset, and Red color line depicts, Training dataset.

* Interpretation of the Results



After applying Machine Learning modelling, it can be seen that ‘Passive Aggressive’ and ‘XGBoost’ model, with hyperparameter tuning are performing best, with around 96% accuracy for both ‘accuarcy\_score’ and ‘F-1 score’ metrics.

**CONCLUSION**

* Key Findings and Conclusions of the Study

Though maximum sentence length of ‘news’ is around 7000 words. But there is not much information gain if we take sentence\_length as 1000 instead.

Also, ‘Passive Aggressive’ and ‘XGBoost’ models works best on the test datasets with around 96% ‘accuracy’ and ‘F-1 score’. ‘Passive Aggressive’ modle is quite helpful as it is best used with real streaming of data. It is an ensemble of Machine Learning algorithms.

* Learning Outcomes of the Study in respect of Data Science

I have run model on GPU, with 2 cores in parallel, for faster computation, as earlier on CPU it was taking a huge amount of time, during ‘model training’ and ‘Data Cleaning’ steps. Along with that, it took a huge amount of time while doing LSTM hyperparameter tuning. But, there was not much improvement in model, by increasing ‘vocabulary\_size’ or ‘sentence\_length’ and other attributes related to ‘embedding\_layer’ features.

* Limitations of this work and Scope for Future Work

There is still a huge amount of data cleaning is to be done in the model, so that. The word cloud can infer better information. But, from the above models, Passive Aggressive works best, as it has only one hyperparameter need or not to be tuned, but still giving better results in lesser amount of time, compared to other models. Though, XGBoost is also giving same accuracy, but it required hyperparameters tuning. Also, can infer information from dataset, while utilizing ‘headline’ feature, instead of ‘news’, and adding new features, performing topic modelling (as discussed earlier).