<https://www.kaggle.com/c/nlp-getting-started/data?select=test.csv>

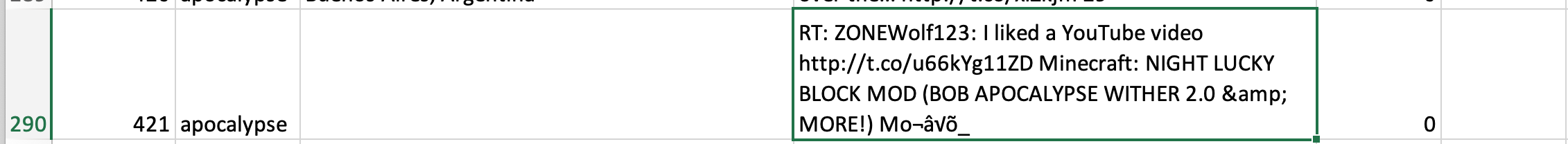
Tweet pre-processing

<https://medium.com/analytics-vidhya/pre-processing-tweets-for-sentiment-analysis-a74deda9993e>

NLTK doc

<https://realpython.com/nltk-nlp-python/#filtering-stop-words>

Retweets



Word2Vec in python <https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/>

TF-IDF in python <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>

Tensorflow <https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding>

NOTE: use tensorflow.keras and not keras

Workflow: <https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/>

Word2Vec – LSA <https://www.sciencedirect.com/science/article/pii/S0957417420302256>

Fasttext using genism - <https://radimrehurek.com/gensim/models/fasttext.html>

Word2Vec using genism - <https://radimrehurek.com/gensim/models/word2vec.html>

GLoVE: <https://nlp.stanford.edu/projects/glove/>

Graphical user interface, text, application, email

Description automatically generated

<https://github.com/ketanvaidya25/IMDb-Movie-Sentiment-Analysis/blob/main/IMDb_Movie_Sentiment_Analysis.ipynb> ,

<https://towardsdatascience.com/sentiment-analysis-using-lstm-and-glove-embeddings-99223a87fe8e>

(code for GLoVE)

SEE for glove: <https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010>

SVM: <https://scikit-learn.org/stable/modules/svm.html>

XGBoost - <https://suatatan.com/posts/sklearn_xgboost_tc/> ,

<https://towardsdatascience.com/finding-similar-quora-questions-with-word2vec-and-xgboost-1a19ad272c0d> ,

<https://github.com/susanli2016/NLP-with-Python/blob/master/Word2vec_xgboost.ipynb>

XGBClassifier parameters - <https://xgboost.readthedocs.io/en/latest/python/python_api.html>

XGBoost vs Tensorflow: <https://blog.doit-intl.com/xgboost-or-tensorflow-63f4c92d4377>

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text, letter

Description automatically generated

Eg: What is the Precision for our model? Yes, it is 0.843 or, when it predicts that a patient has heart disease, it is correct around 84% of the time

A picture containing text

Description automatically generated

The recall is the measure of our model correctly identifying True Positives.

Text

Description automatically generated with medium confidence

Table

Description automatically generated with low confidence

F1-score is the Harmonic mean of the Precision and Recall

This is easier to work with since now, instead of balancing precision and recall, we can just aim for a good F1-score and that would be indicative of a good Precision and a good Recall value as well.

<https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/>

<https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05>

LDA2VEC –

<https://radimrehurek.com/gensim/models/ldamodel.html>

<https://github.com/Arikskigin/Lda2vec_Implementation/blob/master/Code.ipynb>

Validation split, callback, early stopping - <https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd>

EXTRA:

<https://www.tensorflow.org/text/guide/word_embeddings>

ROUGH:

# removing punctuations

# PUNCTUATION\_LIST = list(string.punctuation)

# print(PUNCTUATION\_LIST)

# df\_clean.tokens = df\_clean.tokens.apply(lambda word\_list: [w for w in word\_list if w not in PUNCTUATION\_LIST])

# print(df\_clean)

# print(df\_clean.loc[268, 'tokens'])

# print(df\_clean.loc[145, 'text'])