• How did you decide to handle this amount of data

The amount of data was quite small, there was no need to create pipelines with batches.

• How did you decide to do feature engineering

I tokenized and stemmed sentences, cleaned them from stopping words and unwanted signs (like comas etc.) at the end I normalized the data. Also I removed some of the words manually.

• How did you decide which models to try (if you decide to train any models)

Since the given problem is classification I tried few popular models, including logistic regression, KNN, XGB and SVM. All of them are assign observation to one of three existing classes.

• How did you perform validation of your model

I used cross-validation with main metric set to f1\_score (the classed are imbalanced), I also took 20% of the sample simulate testing set (I am aware that with sample size = 100 I should stick to just CV – it was done just for the sake of exercise) and measure sensitivity, specificity, accuracy and provide confusion matrix.

• What metrics did you measure

As above.

• How do you expect your model to perform on test data (in terms of your metrics)

With Logistic Regression expected value on test set should be from 60%-70% (f1\_score).

• How do you think you would be able to improve your algorithm if you would have more data

Web scraping can be improved for avoiding possible errors while data preparation part doesn’t fully clean the data from possibly unwanted values. In modeling part with better data quality & wider range of grid searched models surely can achieve better results. In my opinion with more data, It will be reasonable to try using recurrent neural networks.

• What potential issues do you see with your algorithm

Data extraction and preparation parts are not adjusted for all possible scenarios as well as data preparation part surely can be coded more optimally. Also, the data engineering part is done superficially.