Article Theme Classification Based on Titles
Milestone Report I

Problem Statement

Numerous articles are being posted on the internet on a daily basis. For a platform that facilitates members sharing their ideas through writing and reading, creating an index system for information of such a high volume can be a challenging task. Although many approaches can be taken to design such systems, there are two general families, keyword-based and theme-based.

Keyword-based systems give authors the freedom to label their own work. By specifying keywords or #tags, the authors allow readers to search for information that aligns with their interest through author-generated labels. Albeit very flexible and dynamic, a keyword-based indexing system can be very costly to implement and maintain. Because keywords may be submitted by the author in a much more arbitrary way, it embeds uncertainty in the process of searching and retrieving information. For instance, a word may have various cases and derivations, and different words can also be semantically similar. In order for the readers' search to hit the target accurately and comprehensively, the index system has to be able to understand different cases and variations of the same word, as well as establish semantic similarity.

On the other hand, a theme-based system only allows articles to be labeled with predefined topics. Instead of dealing with a potentially infinite number of keywords, a theme-based indexing system can be more organized, and help readers quickly narrow down the scope of their search. Such advantages make it an ideal complementary system to keywords.

This project aims to explore different machine learning approaches and models that can automate the task of classifying articles into different themes according to their titles. The final classification algorithm should be able to suggest possible themes based on titles. Although the authors of the articles could provide their own themes, an automatic algorithm can suggest the most likely categories to the publisher and help promote a consistent framework that facilitates a more robust and efficient indexing and searching system.

Data Source

The data source comes from Kaggle (Medium Post Titles — Medium Post Titles, Subtitles, Categories, https://www.kaggle.com/nulldata/medium-post-titles). It contains the titles of 126,095 articles posted on Medium (https://medium.com). These articles belong to and are labeled with 93 different themes spanning various STEM areas, arts and social sciences.

Data Cleaning

The texts in the titles contain punctuation marks, which were not needed in the analysis. Therefore, all punctuation marks were removed. Table 1 shows the examples of the texts before and after cleaning.

CATEGORY	TITLE	TITLE (CLEANED)

WORK	"21 Conversations" - A fun (and easy) game for...	21 conversations a fun and easy game for team...
SPIRITUALITY	"Biblical Porn" at Mars Hill	biblical porn at mars hill
LGBTQIA	"CISGENDER?! Is That A Disease?!"	cisgender is that a disease
EQUALITY	"Call me Nat Love" :Black Cowboys and the Fron...	call me nat love black cowboys and the frontie...
ARTIFICIAL-INTELLIGENCE	"Can I Train my Model on Your Computer?"	can i train my model on your compute

Table 1. Comparison between texts before and after cleaning

Exploratory Analysis

In this dataset, different categories do not occur equally -- the frequency of different categories are illustrated in Figure 1. This indicates that when holding part of the data as a test set, the split should be stratified.

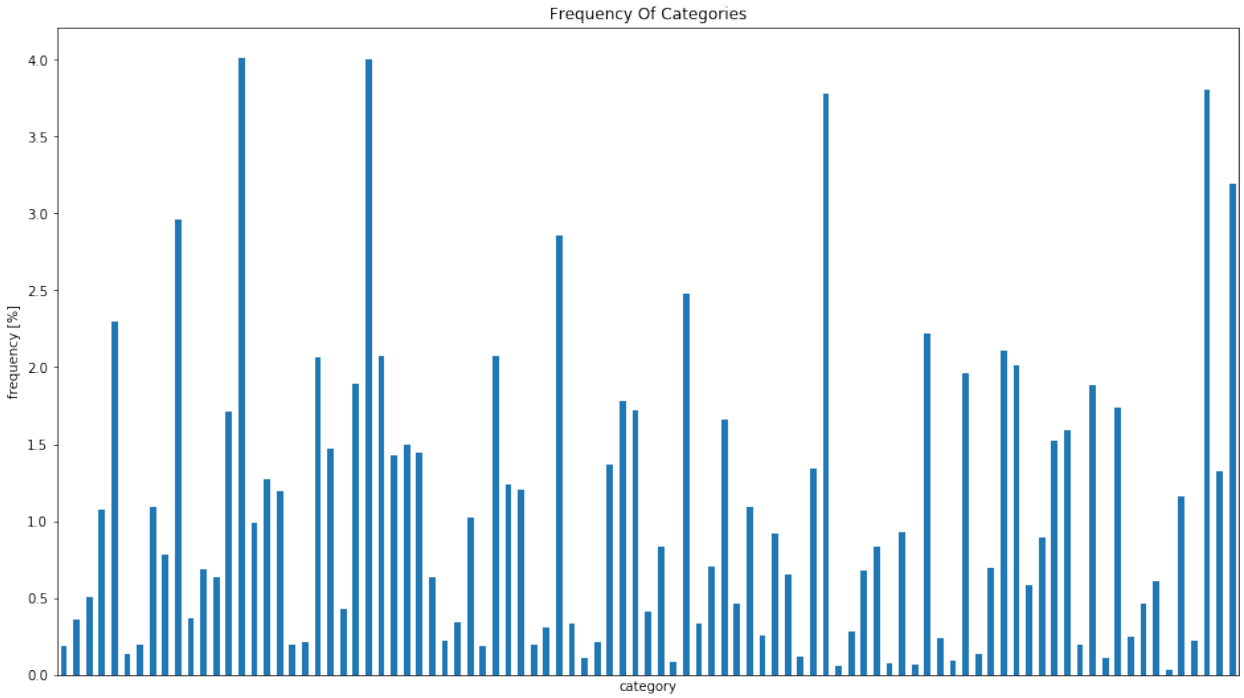

Figure 1. Occurrence of categories

I first used the bag-of-words model to extract features from the texts. Using TF-IDF, for each title, a vector representation was created. For exploratory purposes, the vocabulary was constrained to unigrams and bigrams that appear more than twice but less than 50% of times throughout the dataset. Common English stop words – (prescribed by *scikit-learn*) were also ignored.

Averaging the vectors within a category, I was able to generate a mean vector representation for each theme. Since each element in the vector indicates a categorical mean TF-IDF score for a word, the mean vectors can be visualized as wordcloud charts.

The generated wordclouds shows that the vector representations were able to capture the most prominent features shared by titles within the same category. The high-score words picked by the model all seemed reasonable given the corresponding themes (Figure 2).

Figure 2. Wordcloud visualization of some categories based on mean TF-IDF vector representation

Cosine similarities were computed between the mean vector representations of each theme (Figure 3). By ordering the cos-similarity matrix based on hierarchical linkage, I was able to group together categories with intuitive relations. For instance, *artificial-intelegence*, *data-science* and *machine-learning* are grouped together; *ux*, *design* and *visual-design* are grouped together. Thus, when I assigned each category an integer label according to their order lined up by the hierarchical clustering based on cosine similarity, conceptually related themes got neighboring labels.

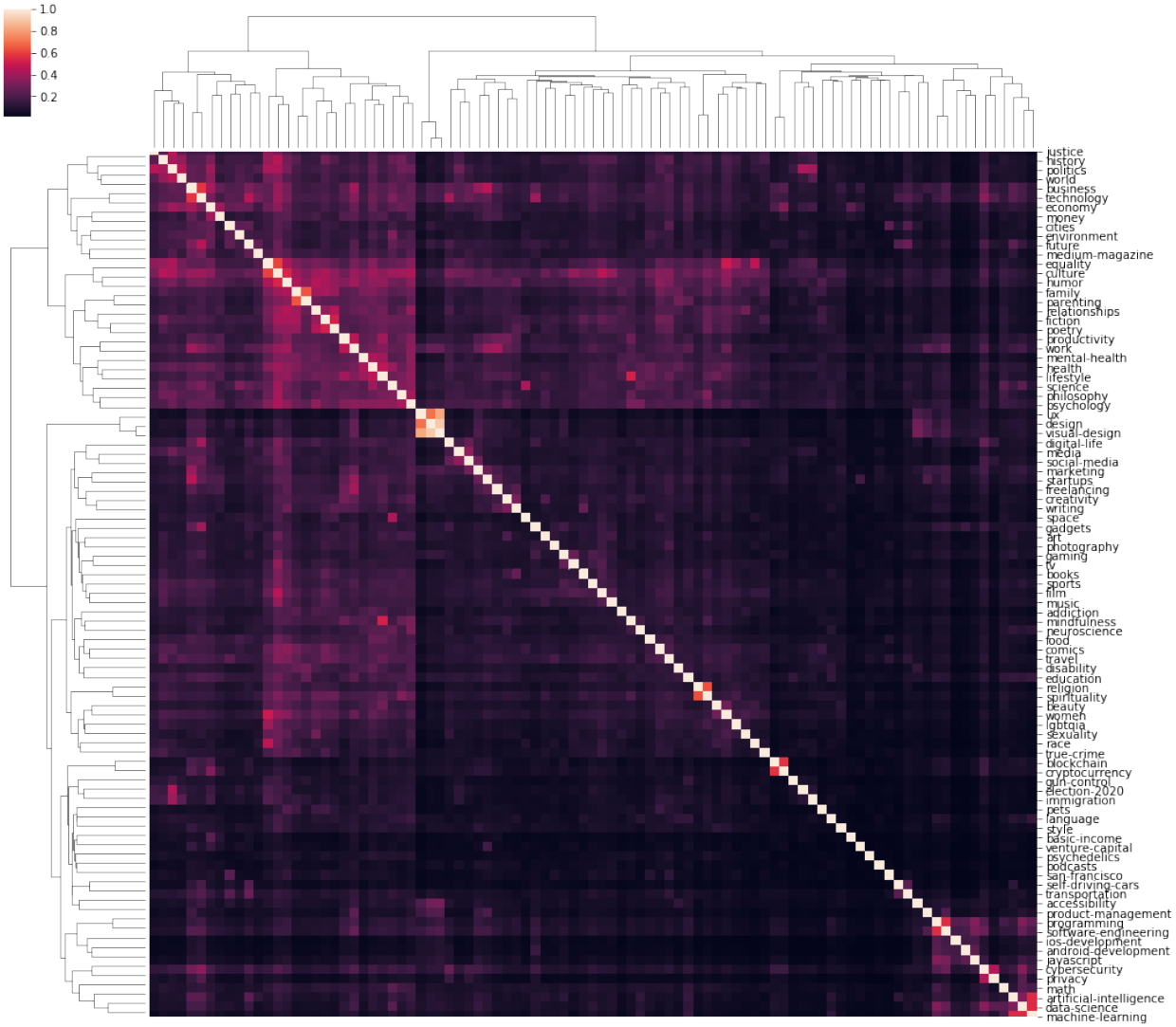

Figure 3. Cosine similarity matrix

The congregation of similar themes can also be captured by unsupervised machine learning algorithms. Figure 4 illustrates the result of PCA-TSNE analysis of the vector representations of the titles. The vector representation of every title was first projected to a 100-dimension space spanned from the first 100 principle component axis (each component accounts for at least 0.05% of the variance in the data). They then had TSNE applied to them, visualized on 2D plane. In the graph, the dots, each of which represents a document, were colored based on its category according to the above-mentioned integer labels, such that categories with higher similarities are colored chromatically closer. The graph shows that although the data were reduced to only 100 dimensions, we can still identify some dense clusters consisting of similar themes. Although, patterns on a larger scale failed to emerge.

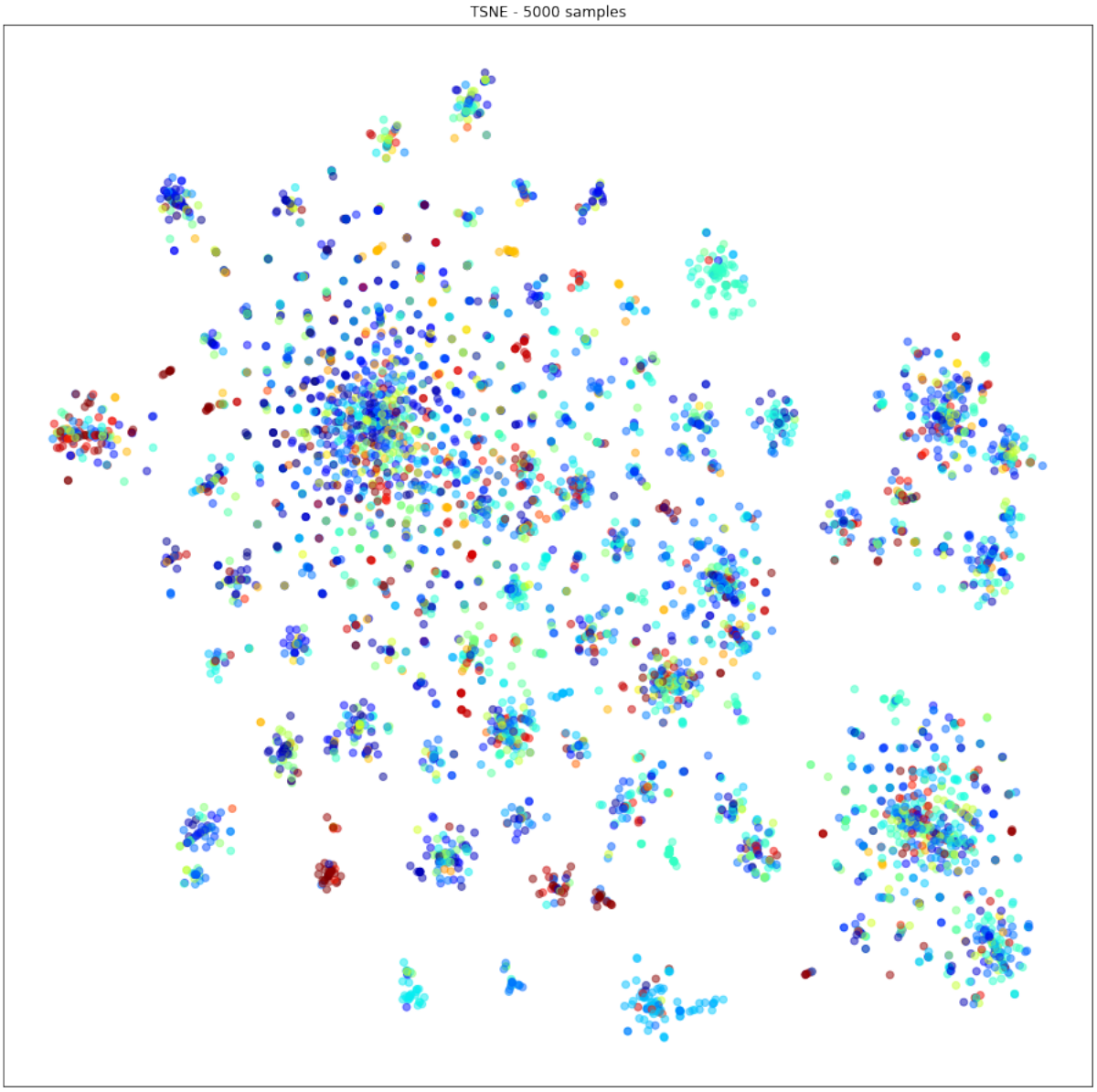

Figure 4. TSNE representation of 5000 titles

Naïve Bayes Classifier

I built a naïve Bayes classifier attempting to assign the correct themes for the titles. 20% of the data were held out as the test set while 80% of the data were used during the training of the model.

The training set was first used to build a TF-IDF vectorization model. The complete data set were then transformed into the TF-IDF space. I used the multinomial naïve Bayes model for the classification task. The choosing of hyperparameters were completed through a 5-fold cross validation search, which determined the range of n-grams, the threshold and cut-off frequency for a word to be included in the vocabulary and the pseudo-count for additive smoothing.

The best choice of hyperparameters generated an accuracy of about 35% on the test set, while the model achieved a multiclass precision of 46% on average. Figure 5 shows the macro-average of multiclass precision and recall. To achieve an average precision more than 90%, the decision threshold should be no less than 0.975. At that level, the average recall is 0.08%.

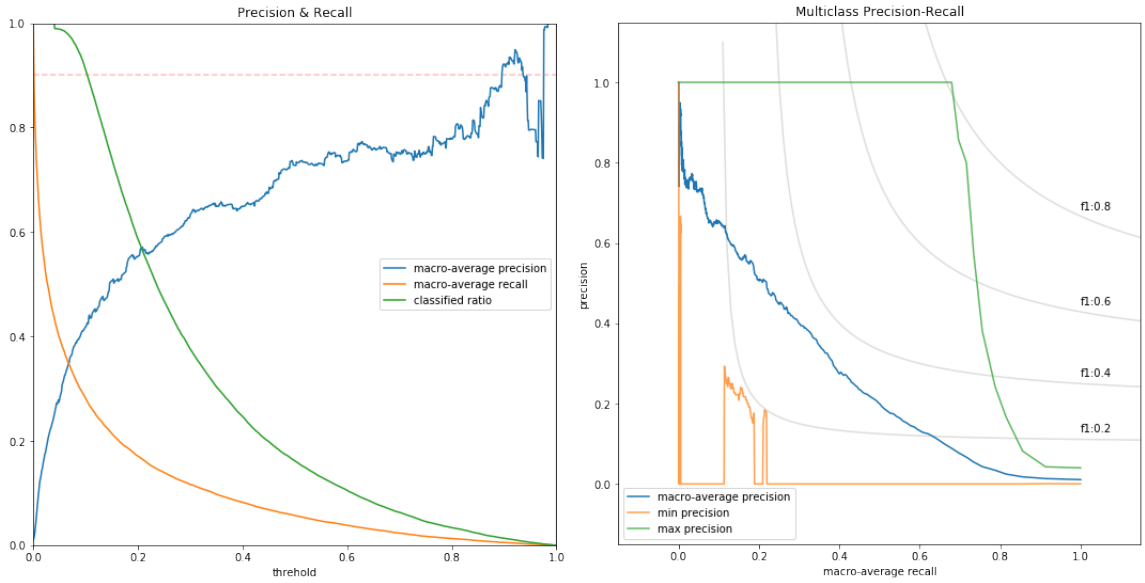

Figure 5. Precision-recall of naïve Bayes model

The confusion matrix generated from the test data (Figure 6.) points out some major pairs of misclassification, partly because those involved categories are tightly related -- *venture-capital* misclassified as *business*, *basic-income* misclassified as *economy*, *election-2020* misclassified as *politics*, *lgbtqia, race* and *women* misclassified as *equality*…

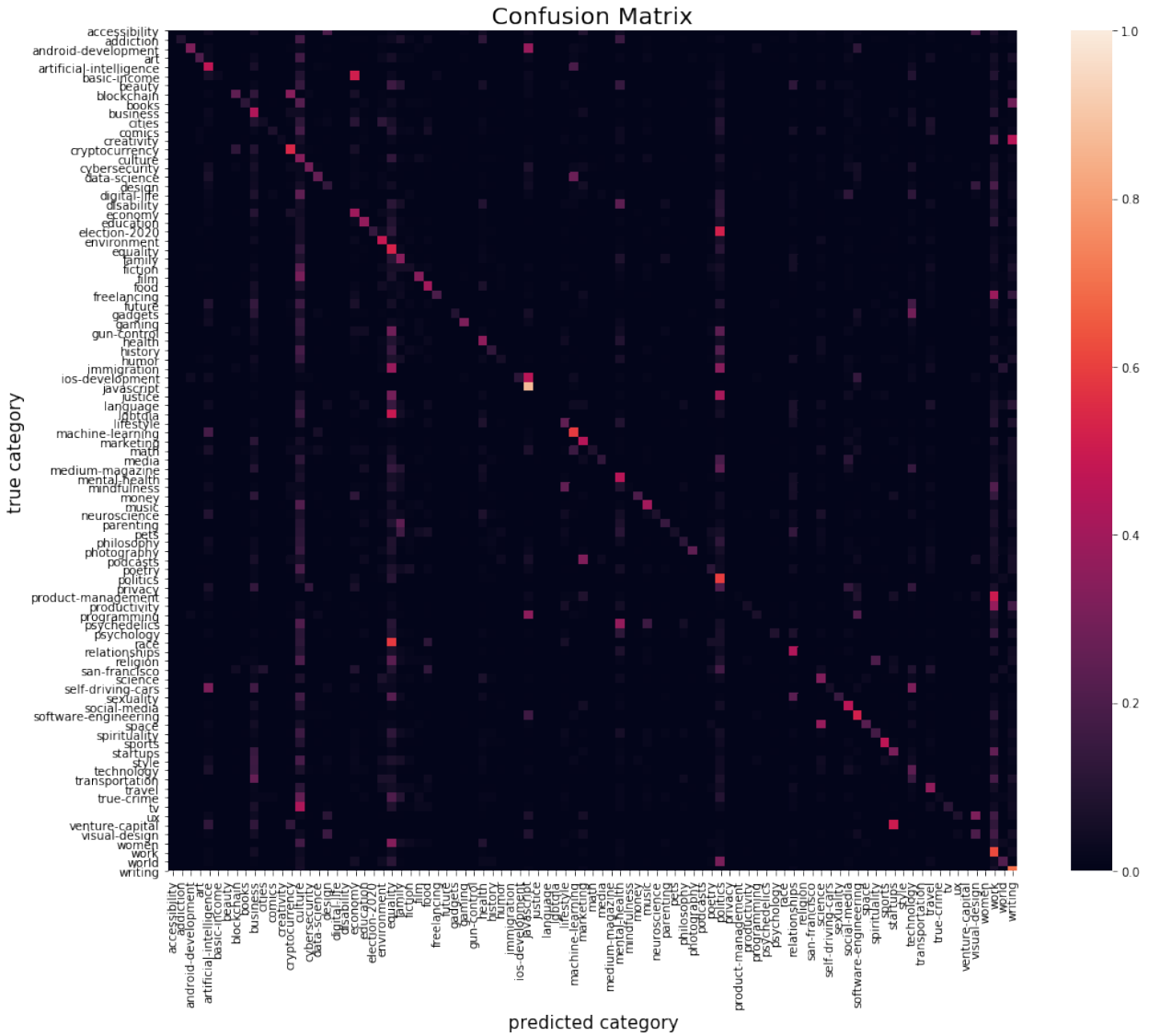

Figure 6. Confusion matrix of naïve Bayes model

Titles in many categories were misclassified into *culture*. Among them, *tv* and *film* caused major confusions (Figure 6). When I took a closer look at words with high TF-IDF scores in the real and predicted *culture* class (Figure 7), as well as the original *film* and *tv* (Figure 8), I saw that both words 'movie' and 'tv' are included in the wordcloud of the true *culture* class. A lot of other major features are also shared by either *film* or *tv* with *culture*, and made major contribution in the misclassified *culture* class, for instance, 'hollywood', 'game', 'thrones', 'black', 'netflix' and so on.

Figure 7, Wordcloud of true *culture* (left) and misclassified *culture* class by naïve Bayes classifier

Figure 8, Wordcloud of true *tv* (left) and *film* (right) class

Multilayer Perceptron Network

I built multilayer perceptron networks for the classification task with various architectures. The networks took input of TF-IDF vector representations as well. However, due to limited computing resources, only unigrams were used because n-grams of higher orders would significantly increase the size of vocabulary, i.e. the dimension of the vectors. Words that appeared less than twice or occurred in more than 10% of the documents in the training set were rejected; this is consistent with the best parameters for the naïve Bayes classifier.

Given the available computing power, the structure of 2 hidden layers of 256 units in each layer was the most cost-effective choice without compromising the performance. Such network was able to achieve an accuracy of 41% on the test set, and a multiclass average precision of 45%. Compared to the naïve Bayes model, the multilayer perceptron network had better F1 scores (Figure 9). However, for identical precision, a higher decision threshold is required compared to the naïve Bayes classifier (Figure 9). To achieve an average precision of 90%, the decision threshold should be more than 0.991. The average recall is 0.5% at that threshold. The increase of accuracy of the model can be mainly attributed to better recall compared to the naïve Bayes classifier. However, there is a trade-off observed between recall and precision.

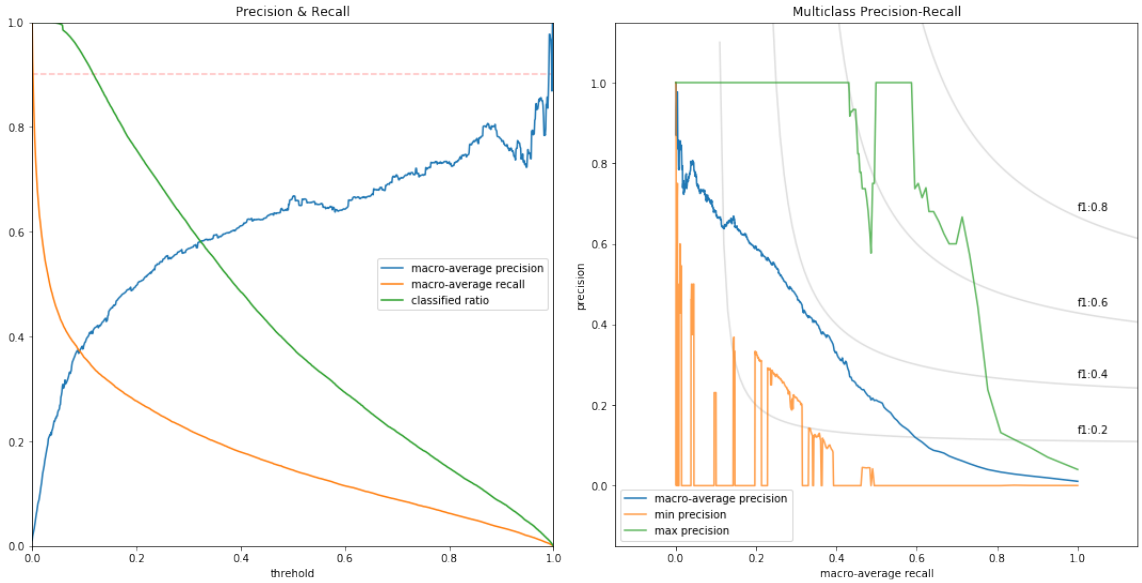

Figure 9. Precision-recall of multilayer perceptron network

The confusion matrix (Figure 10) presents some reasonable misclassification pairs, *privacy* misclassified as *technology*, *basic-income* misclassified as *economy*, *election-2020* and *immigration* misclassified as *politics*, *lgbtqia* and *race* misclassified as *equality*, *venture-capital* misclassified as *startup*, *space* misclassified as *science*...

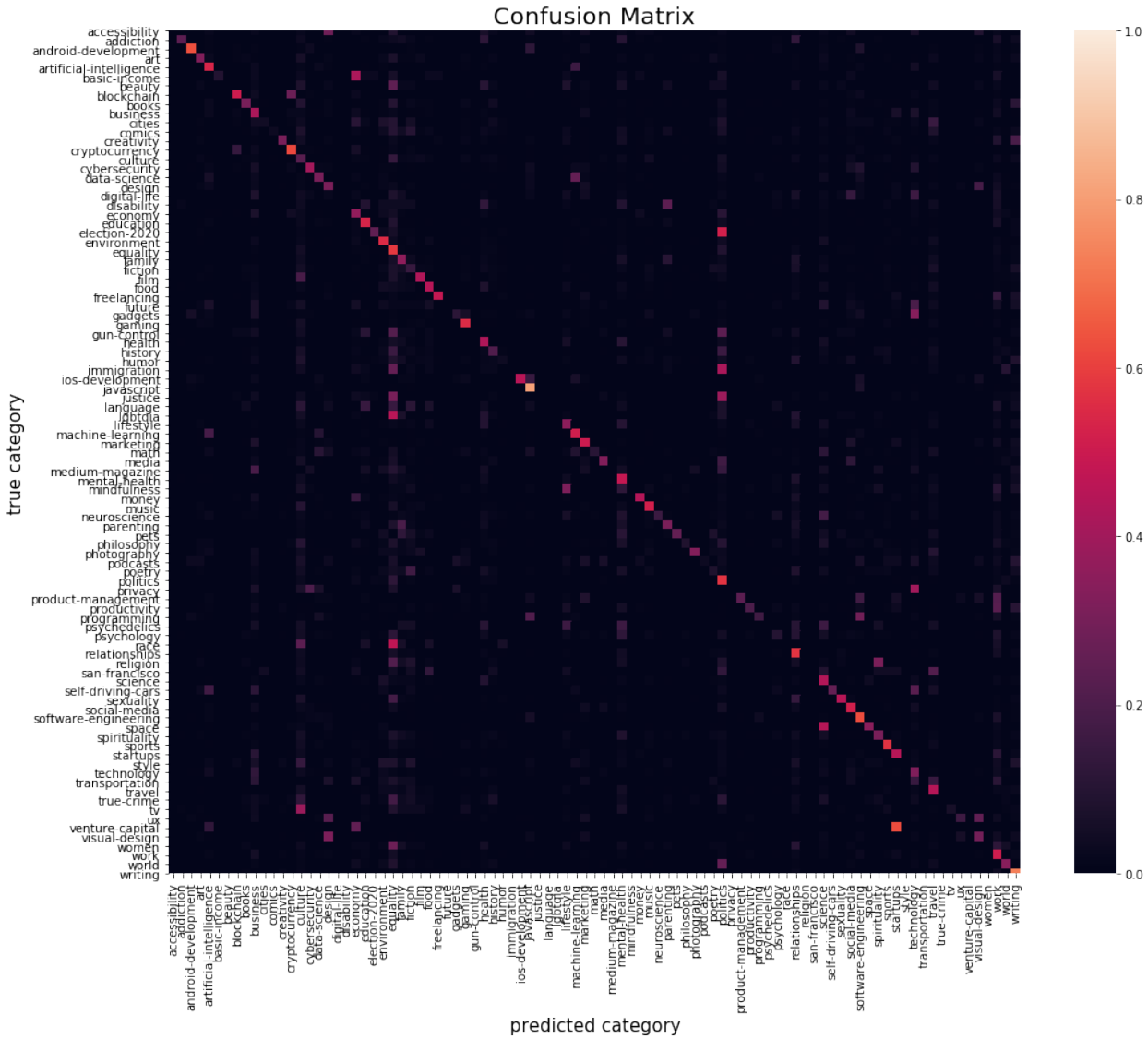

Figure 10. Confusion matrix of multilayer perceptron network