NLP Modeling Task

The report is structured into 3 sub topics.

1. Data Analysis
2. Text Classification
3. Results

Data Analysis

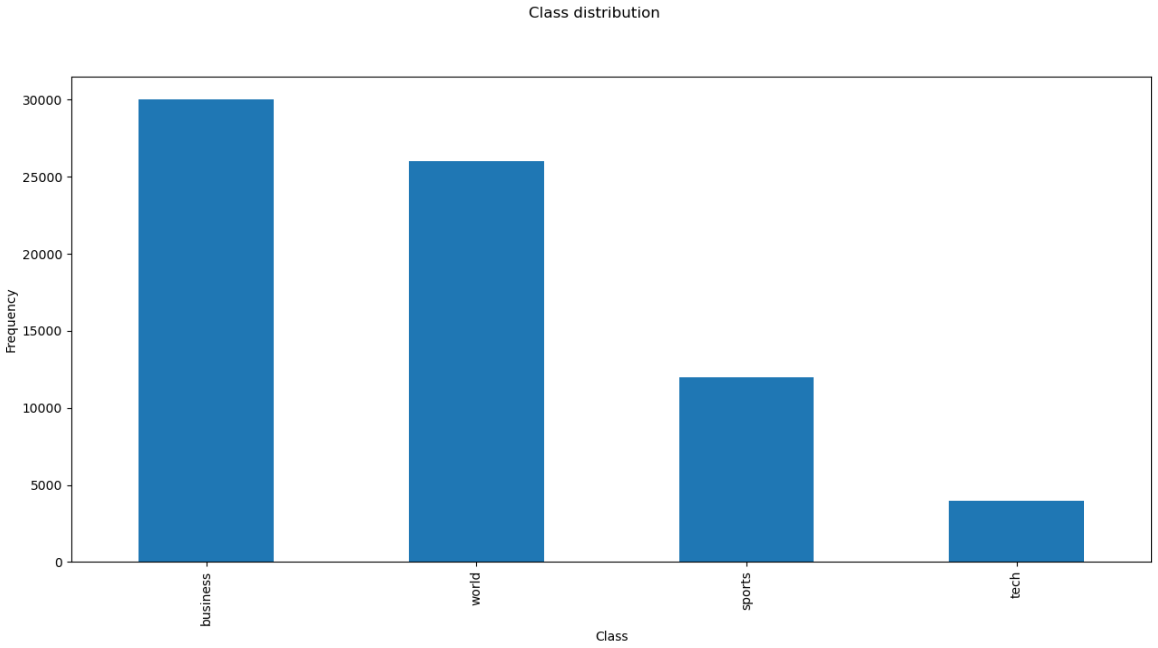

Figure 1: Class distribution in the training data

The given dataset has 4 classes as shown in the figure 1. The dataset is imbalanced with labels like business and world appearing more frequently than others. The size of the vocabulary after preprocessing is around 33897. The dataset has text written in multiple languages e.g., Spanish, English, Russian, Arabic, Japanese among others. The majority text appears in English. The train dataset has 72000 titles and the test dataset has 7600 titles.

Text Classification

Data preprocessing: Along with converting words to lowercase to remove redundancies, special characters and stopwords are removed only for the English language, as the majority of the dataset is in English. Ideally, they should be removed for other languages as well.
After the aforementioned steps, to get the correct root word, lemmatization is performed.
This step is carried out on both the train and test dataset for consistency.

Feature Extraction and Selection: Since the train dataset is large enough and one wants to avoid the high dimensionality problem associated with text, feature selection is performed.
After preprocessing, word count of the features are extracted on the training data and the test data is just represented using these features. Since this results in a vocabulary size of about 33897 words, chi-square feature selection is performed to extract 10% of the features to ensure enough coverage for all classes.

Model Selection: Three algorithms were chosen: Majority-Stratified as baseline, Support vector machine (SVM with class weighting), and Random Forest (with class weighting). Majority classifier which looks into the class distribution (stratified), is a simple baseline for text classification. Due to the nature of the dataset it is interesting to see if complex classifiers outperform baseline. SVM has proved to work well in the research area over text dataset. Since decision tree based algorithms tend to learn well over large datasets, random forest was chosen. SVM & Random forest with class weighting is used to penalize misclassification of the minority classes. Owing to time constraints, parameter tuning has not been carried out.

Results

Results are reported using accuracy and balanced accuracy. However performance measured using balanced accuracy should be considered due to the nature of the dataset.

There are 2 yardsticks of evaluation.
First yardstick: How is the model performance when the model is directly trained on the entire training data and evaluated on it? This yardstick is present because of its mention in the task “Model should correctly classify titles in the train file”.

Algorithm	Baseline	LinearSVC	RandomForest
Balanced accuracy	25.0664	80.9135	**91.7508**

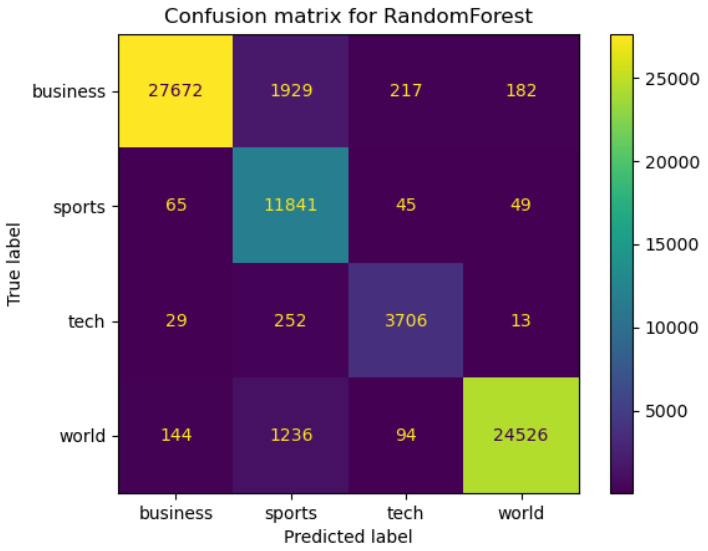

Figure 2: Confusion matrix of RandomForest on training data

On yardstick 1, both linear SVM and random forest perform better than the baseline.
The ensemble random forest seems to perform better than a linearSVM, as decision tree based algorithms tend to learn well over large datasets. The confusion matrix of random

forest when evaluated on the training data is depicted in Figure 2. From the matrix, one can see that the precision for the class business is around 92%, 99% for sports and so on.

Second yardstick: How is the model performance when stratified 5-fold CV is performed on the training data?

Algorithm	Baseline	LinearSVC	RandomForest
Balanced accuracy	25.4595±0.2755	76.4467±0.6587	**76.9993 ± 0.5503**

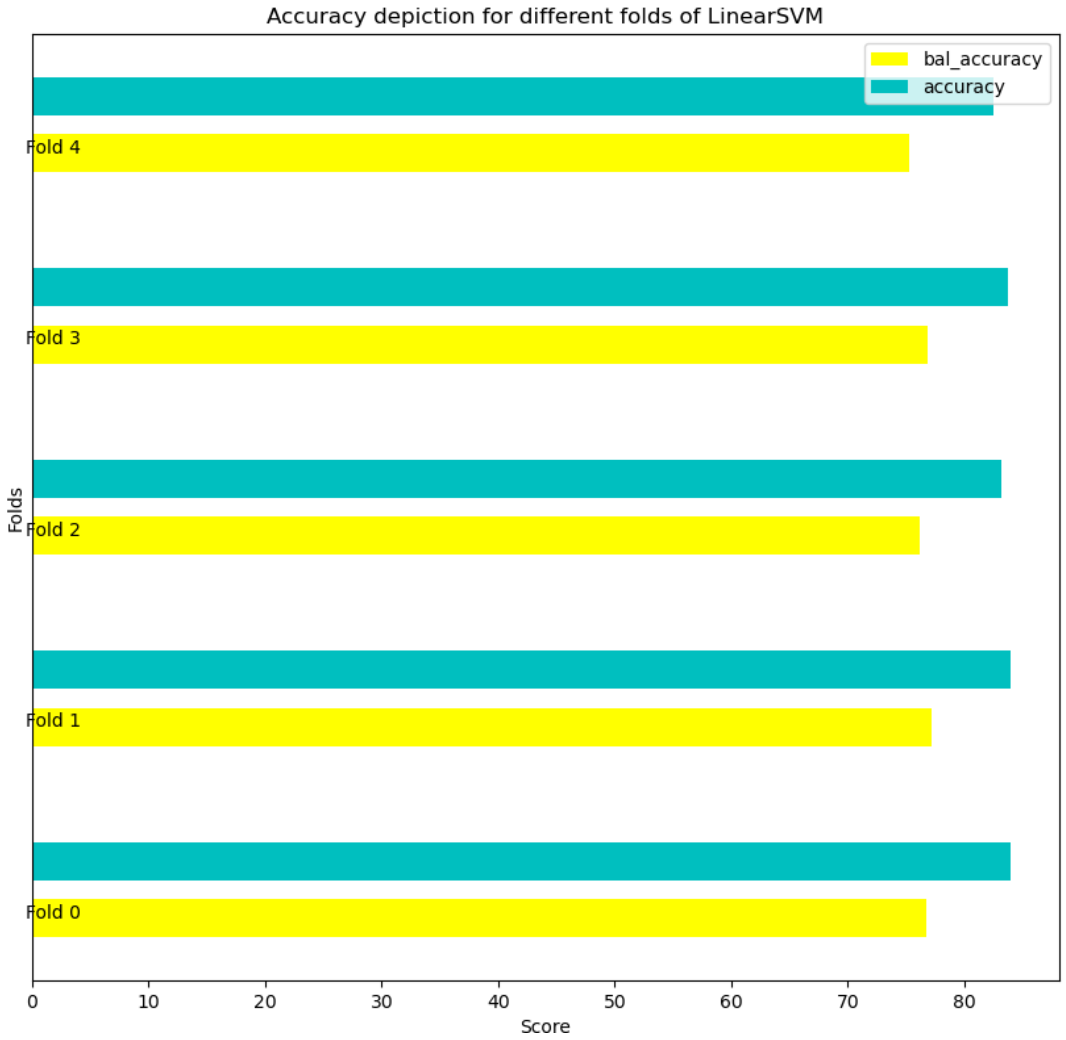

Figure 3: Accuracy score depictions for Linear SVM(stratified 5-fold CV)

The idea behind this yardsticks is to infer about the algorithm's generalization power. In Figure 3 the accuracy and balanced accuracy on each fold of 5-fold CV is depicted. Although linear SVM's mean balanced accuracy is close to Random forest on yardstick 2, overall random forest is the algorithm of choice to make predictions on unseen data as it wins on both yardstick 1 and yardstick 2. Although it was not attempted, it would be interesting to see the average performance of classifiers after parameter tuning.