INF8245E: Machine Learning | Assignment #3

Louis Plessis (1933334)

14 November 2021

1. Data Pre-Processing

The vocabulary can be found in “medical_text-vocab.txt”. The data can be found in “medical_text-train.txt”, “medical_text-test.txt” and “medical_text-valid.txt”.

2. Binary bag-of-words (BBoW)

(a) Random classifier performance (F1-score)

Training: 0.2486939620429286

Validation: 0.2724920606510813

Testing: 0.2552326904504254

Majority-class classifier performance (F1-score)

Training: 0.120996778472617

Validation: 0.12424698795180723

Testing: 0.14183381088825217

(b) *Please see Jupyter Notebook*

(c) Hyper-parameters

- ***Naïve Bayes***

  Values of **alpha** considered: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 (best value = 0.6)

- ***Decision Trees***

  Values of **max_depth** considered: 1, 2, 5, 10 (best value = 10)

- ***Logistic regression***

  Values of **C** considered: 0.01, 0.1, 1, 10, 100 (best value = 1)

- ***Linear SVM***

  Values of **C** considered: 0.01, 0.1, 1, 10, 100 (best value = 1)

(d) F1-score

Classifier	Training	Validation	Testing
Naïve Bayes	0.5243230447997342	0.4453180264457025	0.4672239541944292
Decision Trees	0.6538501176678703	0.5394842997924081	0.5913772738010867
Logistic Regression	0.8226849082758394	0.44471145768048514	0.4933905041682139
Linear SVM	0.8223413846361506	0.5216290554386827	0.5382102261041218

(e) Performance of classifiers

When looking at validation and testing performances, we can see that Decision Trees and Linear SVM performed best. We can see that the F1-score stays around 0.45-0.55 for these 4 classifiers, which is significantly higher than the random classifier and the majority-class classifier. One explanation of the relatively bad Naïve Bayes performance could be the very high number of features (10000). We can also see that the training F1-score for Logistic Regression and Linear SVM is higher than Decision Trees and Naïve Bayes (>0.80), which means they could probably perform better on the validation and training dataset with better hyperparameter tuning.

3. Frequency bag-of-words (FBoW)

(a) *Please see Jupyter Notebook*

(b) Hyper-parameters

- ***Naïve Bayes***

  Values of **alpha** considered: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 (best value = 0.6)

  Decision Trees

  Values of **max_depth** considered: 1, 2, 5, 10 (best value = 10)

- ***Logistic regression***

  Values of **C** considered: 0.01, 0.1, 1, 10, 100 (best value = 10)

- ***Linear SVM***

  Values of **C** considered: 0.01, 0.1, 1, 10, 100 (best value = 1)

  Values of max_inter considered: 100, 200, 300, 400, 500, 600, 700, 800, 900 (best value = 100)

(c) F1-score

Classifier	Training	Validation	Testing
Naïve Bayes	0.5243230447997342	0.4453180264457025	0.4672239541944292
Decision Trees	0.6649919768970892	0.5753923197816475	0.5863910797466454
Logistic Regression	0.45926741730734605	0.4179133177475002	0.4205989741981914
Linear SVM	0.39952614300221356	0.39011900287444246	0.3831947217586665

(d) Performance of classifiers

Decision Trees seems to have performed best on Validation and Testing dataset.

(e) FBoW vs BBoW performance

The performance of the 4 classifiers seems close to the BBoW performance. However, we can notice that Linear SVM performed worse on FBoW than BBoW, and that the training F1-score for Logistic Regression is significantly lower for FBoW than for BBoW.

(f) Best representation

The best representation is probably FBoW since it indicates the frequency of word instead of only providing information on its presence or not. This additional information should probably lead to a better prediction.