Ayah Hamdan

Programing Assignment: Medical Reports

To build a system which is able to automatically classify medical reports, first the data set was explored which is the reports written by doctors, where each report describes a single case and is stored as a text file with the first line summarizing the case, and the others providing more detailed information. From observing multiple files, it can be concluded that in each class, some words are repeated and are a related to the class, for example, in the class Cardiovascular/ pulmonary, some of the common words are: heart catheterization, EKG, rapid heart rate, coronary, ventricular, CHF, and Pulmonary. Also, it was concluded that these and the other words are presented sometimes in the first line (the summary), while other time, they are not represented clearly. So, the latter gives us 2 options, either only using the first line (summary) in the classification or using the whole report with the detailed information.

Machine learning algorithms cannot work with raw text directly and the text needs to be converted into numbers, such as vectors of numbers, which is called feature extraction. And one of the methods used is the bag-of-words model of text, where it describes the occurrence of words within a document. And to achieve better results with the BOW representation, the data has to be cleaned by ignoring the case and changing all the words into lower cases, ignoring punctuations, and removing the stop words which are the frequent words that don't contain much information, such as: "a, the, of, an, etc.". Also, the stemmer function should be used to crude heuristic process that chops off the ends of words using the porter library.

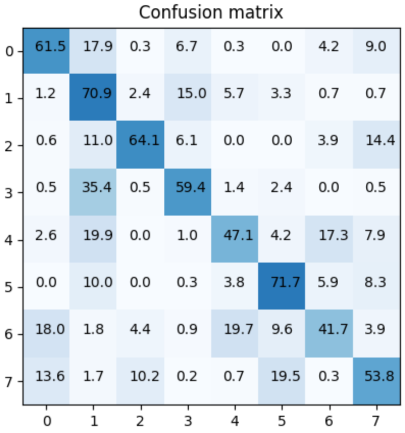
The steps to use the BOW representation is first to build a vocabulary file for each of the train, test, and evaluate files, taking in consideration the cleaning steps mentioned before. When writing the resulted vocabulary in a file, not all of the results were saved due to its big number, so only the n most frequent words were used. And for choosing n, many test cases where tried (based on the naïve bayes classification), to choose the best choice, and it resulted that choosing n=2000 gave the best result.

	Training Acc	Testing Acc	Validation Acc
N=10000	59.2	80	77.0
N=2000	58.84	83.33	84.33
N=1000	57.3	80	77.66

And when the classifications were based on the short description only (first line), the n used was equal to 1000, because the words were less.

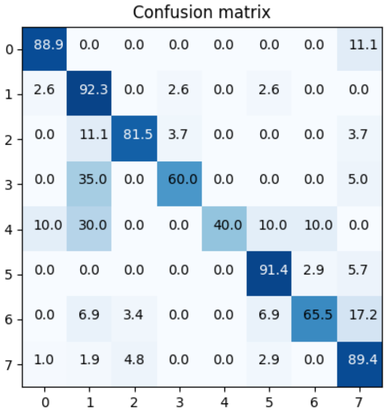
Then in the extract_features file, the Bow representation were calculated for each vocabulary file, where after reading the file, the BOW representation was calculated by first initializing the counters as zero, and for each word in the file (this operation was implemented for each file) which is also in the vocabulary file, the counter will be incremented. If the words were not in the vocabulary file, then the counter will remain 0. Thus, for each set of files, a double list was computed to represent the words. And for the labels for each file, a single list was computed with the number of classes from 0 to 7, where the classes were sorted lexicographically in an ascending order. Finally, both lists were concatenated and saved in ”.gz” extension file to reduce the size.

For the first used classifier, the multinomial Naïve Bayes was used. Where it is suitable for text classification, and despite its simplicity, it is very effective. The following figures show the confusion matrix for each the training (left figure) and testing (right figure) results. Where from the coloring of the squares, it can be concluded that the testing classification was better and more classes were classified correctly, thus darker colored squares.



Confusion matrix

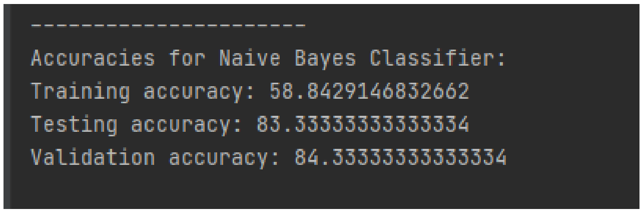
	0	1	2	3	4	5	6	7
0	61.5	17.9	0.3	6.7	0.3	0.0	4.2	9.0
1	1.2	70.9	2.4	15.0	5.7	3.3	0.7	0.7
2	0.6	11.0	64.1	6.1	0.0	0.0	3.9	14.4
3	0.5	35.4	0.5	59.4	1.4	2.4	0.0	0.5
4	2.6	19.9	0.0	1.0	47.1	4.2	17.3	7.9
5	0.0	10.0	0.0	0.3	3.8	71.7	5.9	8.3
6	18.0	1.8	4.4	0.9	19.7	9.6	41.7	3.9
7	13.6	1.7	10.2	0.2	0.7	19.5	0.3	53.8



Confusion matrix

	0	1	2	3	4	5	6	7
0	88.9	0.0	0.0	0.0	0.0	0.0	0.0	11.1
1	2.6	92.3	0.0	2.6	0.0	2.6	0.0	0.0
2	0.0	11.1	81.5	3.7	0.0	0.0	0.0	3.7
3	0.0	35.0	0.0	60.0	0.0	0.0	0.0	5.0
4	10.0	30.0	0.0	0.0	40.0	10.0	10.0	0.0
5	0.0	0.0	0.0	0.0	0.0	91.4	2.9	5.7
6	0.0	6.9	3.4	0.0	0.0	6.9	65.5	17.2
7	1.0	1.9	4.8	0.0	0.0	2.9	0.0	89.4

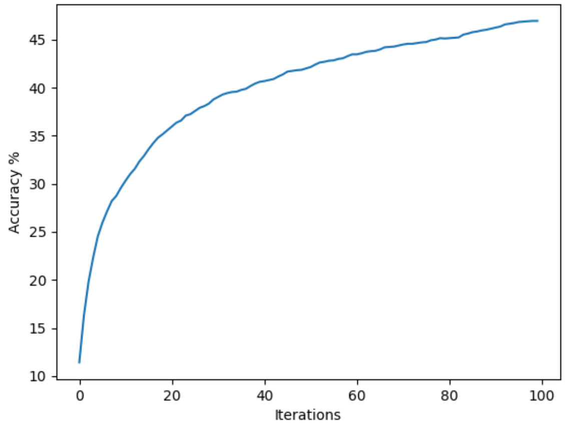
While the following figure show the accuracies for the training, testing, and validating classifications, and it can be concluded the accuracies are very good for the testing and validating, while there is a big difference in the training, and this is called “underfitting”, where it happens when the model is too simple for the classes.

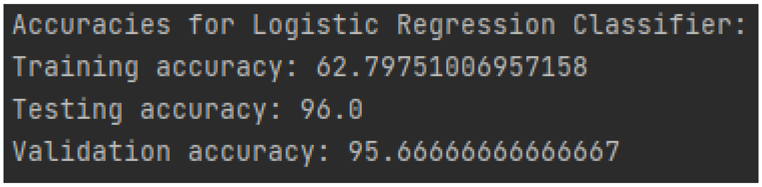


```
----------------------
Accuracies for Naive Bayes Classifier:
Training accuracy: 58.8429146832662
Testing accuracy: 83.33333333333334
Validation accuracy: 84.33333333333334
```

The second classifier used is the Logistic Regression Classifier, where to limit the complexity of a classifier, constraining the values of its parameters can be executed using the generalization, which is achieved by favoring the distribution of the weights of the classifier over a large number of features, and it consists in minimizing the cross entropy (the loss function of logistic regression) combined with L2 regularization.

The following plot shows the graph for the training accuracies. And the other figure shows the accuracies for the training, testing, and validation, where the accuracies is better than the Naïve Bayes classifier, but the underfitting problem remains.





```
Accuracies for Logistic Regression Classifier:
Training accuracy: 62.79751006957158
Testing accuracy: 96.0
Validation accuracy: 95.66666666666667
```

Other classifiers were also implemented such as the Gaussian Naïve Bayes, multi-class SVMs: one versus rest and one versus one, and the multi-layer perceptron. But they all resulted in vey low accuracies, and maybe because they don't suit this type of problems.

Also, before using the classifiers, some normalizations were introduced, such as the Min-Max, Mean-Var, Max-Abs, and L2 normalization, but all of them decreased the accuracies, and made it worse, and this is because they don't work with the BOW representation and this type of the problem.

At the end, the first 2 classifier: Naïve Bayes, and Logistic Regression were used in classification of the short description (only the first line), and the accuracies are shown in the table below. And it can be concluded that using the detailed description is better, where the accuracy increased by 2.6-9.63%.

	Training	Testing	Validation
Naïve Bayes	56.17	74	76
Logistic Regression	58.62	91.67	92

In conclusion, some modifications can be made which might improve the results, such as removing the numbers from the vocabulary file, where they probably don't give a meaning and a good indication to the class but instead decreasing the accuracy. Also, maybe a feature selection can be used with the BoW representation where it can classify exam, history, technique, findings, and impression, each into single feature, and then we can test classifying using some of them and it might resolve the underfitting problem. Finally, other classifiers can be used which might give better result, and even it can be classified using some neural network.

I affirm that this report is the result of my own work and that I did not share any part of it with anyone else except the teacher.