Medical Text Classification [50 marks]

In this assignment, we will build a classifier with medical-NLP corpus. This is a classification task with the input as a medical transcription (text) and the output as the corresponding medical transcript type. This is a clinical dataset which consists of a medical transcript from one of the 4 classes {Surgery - 1 , Medical Records - 2, Internal Medicine - 3 and Other - 4} as an input. The task is to classify the transcript (text) to the corresponding classes i.e the transcript type. The dataset consists of 4000 transcripts in the training set and 500 in each of the validation and test sets.

1. Most of the algorithms described in the class take the input as a vector. However, the reviews are natural language text of varying number of words. The first step would be to convert this varying-length movie review to a fixed-length vector representation. We will consider two different ways of vectorizing the natural language text: binary bag-of- words representation and frequency bag-of-words representation (as explained in the end of the assignment). Convert both datasets into these representations. Instructions for dataset submission are given in the end of the assignment (do not include the dataset in the report). [10 marks]

   See submission data

2. For this question, we will focus on the Medical-NLP dataset with binary bag-of-words (BBoW) representation. We will use the F1-score as the evaluation metric for the entire assignment. [19 marks]

(a) As a baseline, report the performance of the random classifier (a classifier which classifies a review into a uniformly random class) and the majority-class classifier (a classifier which computes the majority class in the training set and classifies all test instances as that majority class). [2 marks]

Classifier	F1-score (Macro)		
	Training set	Validation set	Test set
Random	0.24307372996492	0.23049138907743	0.24824586018318
Majority class	0.1209967784726	0.12424698795180	0.14183381088825

(b) Now train Naive Bayes, Decision Trees, Logistic regression and Linear SVM for this task. [Note: You should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for binary input features (also called Bernoulli naive Bayes).] [8 marks]

For this part of the assignment, the binary bag of words has been used. Four models were realized, a Naive Bayes model, a decision tree, a logistic regression model and the linear SVM model. First, let's look at the different F1-score which measures the test's accuracy for each classifier before trying to improve the models with hyper-parameters tuning

Classifier	F1-score (Macro)		
	Training set	Validation set	Test set
Naïve Bayes	0.53511119569591	0.46021778739106	0.46725082086627
Decision Trees	0.90737693142189	0.70594605117156	0.74560633465819
Logistic Regression	0.90840784427494	0.69187301569780	0.70493146106597
SVM	0.90745594581330	0.74194776383834	0.78500773426403

With these results, we can see that the naive classifier is much less efficient than the other three types. Moreover, we observe that the models overfit the training data because the scores obtained for the validation and test set are significantly smaller.

(c) Report the list of hyper-parameters you considered for each classifier, their range, as well as the best values for these hyper-parameters, chosen based on the validation set performance. [3 marks]

A k-fold set was produced from the training and validation set. Using cross validation, every record is used exactly once for validation and the hyper parameter tuning can give better results.

Using this technic:

- For Naïve Bayes, the principal hyper-parameters considered were the alpha. This parameter controls the smoothing. First, the validation sets were maximized using several powers of ten. When the best power was found, the value close to the best value was tested.

```
'alpha': 0.4
```

- For the decision trees, the different criterion (the function that measure the quality of a spill) were testes, once the best was found. The maximum depth of the tree and the minimum samples split were tested in priority. The depth limits the tree to overfit and the sample split the width of the tree.

```
'criterion': 'gini'
'min_samples_split': 25
'max_depth': None
```

- For the logistical regression, all the different algorithm that optimise the problem were tested and the best was chosen. After that, a regularization of a parameter (C) that increase the mag nitude of parameter values was tested, but it didn't give me a significant result. The next impo rtant hyper parameter tested was to add a type of penalty to the regression. To best one was th e l1 penalty that limit the coefficient value and it did improve the result in a significant way.

```
'solver': 'liblinear'
'C': 1
'penalty': 'l1'
```

- For the linear SVC (SVM), the loss function was the first hyper parameters tested and the best found was the 'hinge', the standard deviation lost. After that, the next important parameter was the maximum number of iterations to limit the chance that the model overfit. In addition, the next parameter considered was the penalty. The best penalty found was the l1 that leads the vector to be sparse. Finally, the last parameter was the regularization parameter C. For this model, no regularization was needed.

```
'loss': 'hinge'
'max_iter': 10000
'penalty': 'l2'
'C': 1
```

(d) Report the training, validation, and test F1-score for all the classifiers (with best hyper-parameter configuration). [3 marks]

Classifier	F1-score (Macro)		
	Training set	Validation set	Test set
Naïve Bayes	0.5479827560736	0.45669159541292	0.4751461438243
Decision Trees	0.8432807218136	0.79134753596383	0.80597684131314
Logistic Regression	0.9063689835553	0.76346884800857	0.78580334450991
SVM	0.9044567741939	0.74247552722874	0.7818678137257

(e) Comment on the performance of different classifiers. Why did a particular classifier perform better than the rest? What was the role of the hyper-parameters in finding the best results? [3 marks]

With the hyper parameters tuned, we can see a 5% amelioration one the decision tree and a 8% amelioration the logistical regression. The Naïve Bayes also better, but not in a significant way. For the SVM, the results are almost the same, but the cross validation removed a bit of over fitting.

The best classifier obtained is the tree-based one. The tree classifier, compared to the others, can search for the words having the most impact on the type of conversation, but can also consider a high degree of combination. The Naïve Bayes is the worst because this type of model considers all the features to be independent. Finally, the two other models have interesting predictive capabilities, and it would be interesting to combine them with the decision tree to improve the results.

3. Now we will repeat question 2 but with frequency bag-of-words (FBoW) representation. [21 marks]

(a) Train Naive Bayes, Decision Tree, Logistic regression and Linear SVM for this task. [Note: Again, you should do a thorough hyper-parameter tuning by using the given validation set. Also, note that you should use the appropriate naive Bayes classifier for real valued input features (also called Gaussian naive Bayes).] [8 marks]

For the second part of this assignment, we are using the frequency bag of words instead of the binary. This mean that we can not use the Bernoulli class because it has more than two value. To remedy to this problem, we have used a gaussian Naïve Bayes. As before, the models a decision tree, a logistic regression model and the linear SVM model we tested. First, let's look at the different F1- before trying to improve the models with hyper-parameters tuning.

Classifier	F1-score (Macro)		
	Training set	Validation set	Test set
Naïve Bayes	0.5171964299159	0.10999999999999	0.11841285063533
Decision Trees	1.0	0.21554428491806	0.17420969070172
Logistic Regression	0.64883897538310	0.1757513807786	0.07911457265833
SVM	0.66694238202743	0.09031198686371	0.07911457265833

Where we can see that using the frequency bag of word give us much worst results than the binary bag of words. All the models are overfitting the training data and the decision tree also give us an 83% different between the training and the test f1-score.

(b) Report the list of hyper-parameters you considered for each classifier, their range, as well as the best values for these hyper-parameters, chosen based on the validation set performance. [3 marks]

- For the Naïve bayes, the only hyper parameter is the smoothing of the variance. By augmenti ng this parameter, the model can give us better results and increase the stability of calculation at the same time.

```
'var_smoothing': 0.01
```

- To reduce the overfitting, the hyper paramter tested in priority the function to measure the q uality of the slip, the splitter, the maximum depth of the tree and the minimum sample split. Combining those parameters can reduce overfitting greatly.

```
'criterion': 'gini',
'max_depth': 1000,
'min_samples_split': 2,
'splitter': 'random'
```

- For the logistical regression the same hyper parameters were considered. All the different algo rithm that optimises the problem were tested and the best was chosen. After that, a regulariza tion of a parameter (C) that increase the magnitude of parameter values was tested, but it did n't give me a significant result. The next important hyper parameter tested was to add a type o f penalty to the regression. The best one was the l1 penalty that limit the coefficient value and it did improve the result in a significant way.

```
'C': 100,
'penalty': 'l1',
'solver': 'liblinear'
```

- SVM: For the Linear SVC, all the parameters were the same except for the regularization para meter C. In this case, the model needed a strong parameter to increase its performance.
-

```
'C': 4200
'loss': 'hinge',
'max_iter': 10000,
'penalty': 'l2'}
```

(c) Report the training, validation, and test F1-score for all the classifiers (with best hyper-parameter configuration). [3 marks]

Classifier	F1-score (Macro)		
	Training set	Validation set	Test set
Naïve Bayes	0.40110946721224	0.37264129511422	0.38376685481751
Decision Trees	0.65101605467084	0.61994518476075	0.66563414645069
Logistic Regression	0.64299342018717	0.65792744515080	0.62310290760143
SVM	0.57115169585776	0.50129083740845	0.47341201133379

(d) Comment on the performance of different classifiers. Why did a particular classifier perform better than the rest? What was the role of the hyper-parameters in finding the best results? [3 mark]

Using the different hyper-parameters, we clearly see that we removed the overfitting of each model. Here all the f1-score are almost the same for each model and the decision tree also give us 1.5% more accuracy in the test set than the training set. The Naïve Bayes model gives us the worst results of the four models. This is since this model considers that the features are independent, and they are not in reality. The decision tree and the logistic regression give the best results of the four models. the use of hyper-parameters has greatly increased the results of the test set by reducing the overfit, but the results are not exceptionally good. And finally, the linear SVC gives us a very bad results on the test set even if a strong regularization parameter was use. Even if the results are not as good as with the other type of data, we see that the use of hyper parameters have helped a lot. Like a said earlier, all the overfitting is now gone and the f1-score of the decision tree is now better in the test set, the data that the tree never saw before.

(e) Compare the performance with the binary bag-of-words based classifiers. Why is there difference in the performance? Give a brief explanation comparing BBoW Naive Bayes and FBoW Naive Bayes and similarly for other models. [2 mark]

Comparing the naive models, we see that the binary model gives better results. This may be since some words are important in the decision, no matter how often they are said. The same logic can be used to explain the difference in performance between the other models, even if they can adapt somewhat to the word frequency.

Which representation is better? Why? [2 mark]

Here the binary model is the best. This could be explained by the fact that some words weigh a lot in the decision, even if they are not used at the same frequency in all conversations. To improve the frequency bag, it would be interesting to change the method of traintement of the words by removing the words like the stop words and to use the steaming to group the words family.