CS 464

HW 1

Ahmet Hakan Yılmaz

21803399

SECTION 01

1.1

Probability of a student is motivated = (Probability of a student gets H and motivated) + (Probability of a student gets L and motivated) +(Probability of a student gets F and motivated)

(64/100)*(87/100) + (24/100)*(21/100) + (12/100)*(4/100) = 0,6120

1.2

Probability of a getting H when student is motivated = (Probability of a student gets H and motivated) / (Probability of student is motivated)

We already found probability of student is motivated in 1.1

((64/100) * (87/100)) / ((64/100)*(87/100) + (24/100)*(21/100) + (12/100)*(4/100)) = 0,9098

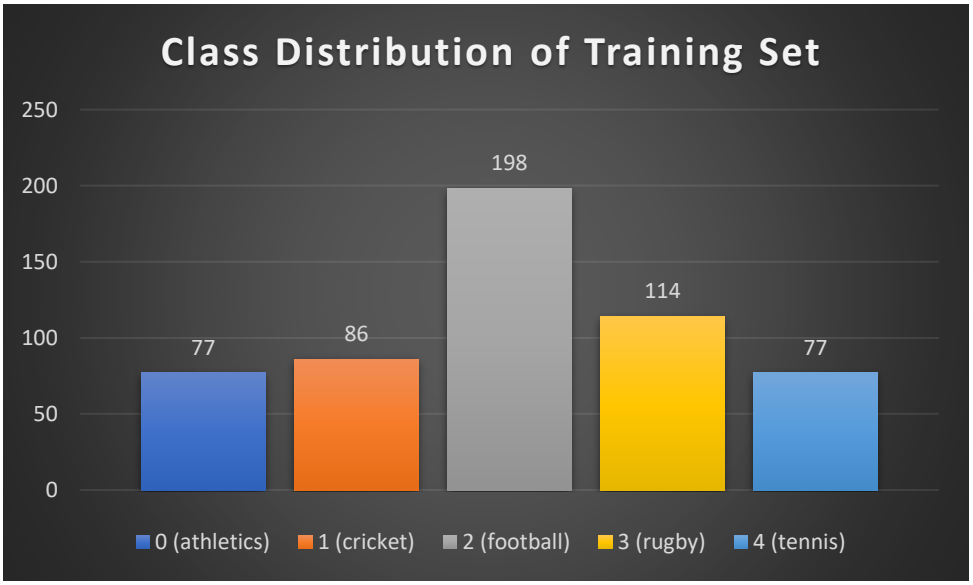
1.3

Probability of a getting H when student is unmotivated = (Probability of a student gets H and unmotivated) / (Probability of student is unmotivated)

((64/100) * (13/100)) / ((64/100)*(13/100) + (24/100)*(79/100) + (12/100)*(96/100)) = 0,2144

2.1.1

In training set there are total 552 articles. 77 articles belongs to athletics class, 86 articles belongs to cricket class, 198 articles belongs football class, 114 article belongs to rugby class and 77 articles belongs to tennis class.



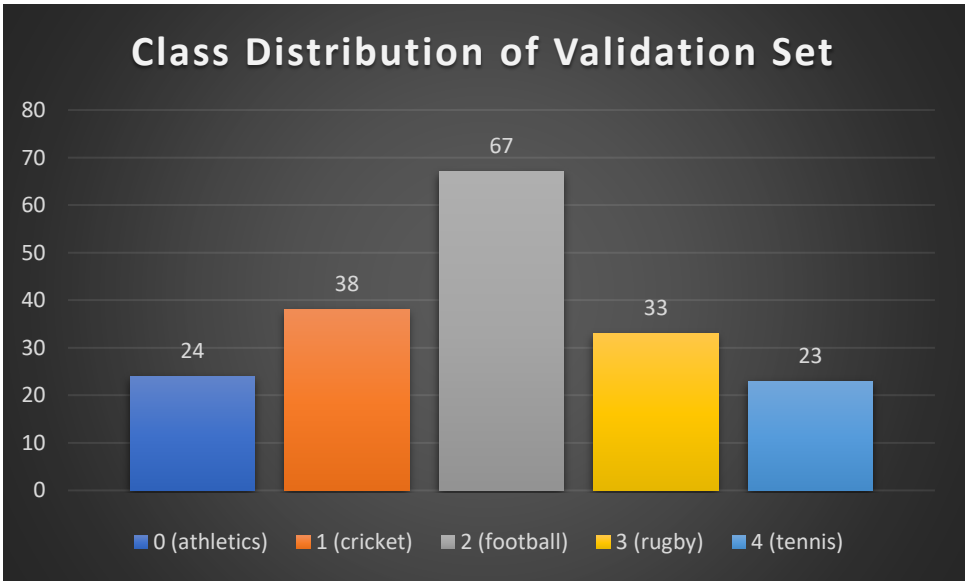
2.1.2

We can say this dataset is not balanced. It is skewed towards football class because although there are 5 classes the number of articles that have football classes more than 1/3 of the dataset.

Yes, having unbalanced set affects our model. In Naive Bayes we try to calculate posterior probability by using likelihood and prior probabilities, having a unbalanced set may not affect likelihood probability so much but it affect prior probability badly and therefore it affects posterior badly. So we can have a biased model as a result of unbalanced set. As the solution we can select same number of samples from each classes for training set. For example, in our example we can select 77 samples from each classes and make a balanced training set to inhibit a biased model.

2.1.3

In validation set and training have similar distribution. there are total 185 articles. 24 articles belongs to athletics class, 38 articles belongs to cricket class, 67 articles belongs football class, 33 article belongs to rugby class and 23 articles belongs to tennis class. When we see the graphs of class distributions of validation and training set we can see they have similar distributions. If there is a bad split that means mostly **prior** possibility will affect the posterior badly and the probability we get will be biased. **Prior** would be the misleading term.



2.1.4

If our dataset is skewed to one of the classes than still our model can get high accuracy rates since our validation set is also from same dataset (if there is not bad split) . However, in unbalanced sets accuracy rate is not a good metric. For example if we have an unbalanced dataset which have 2 classes with rates %1 and %99. Than our model probably predicts always the class with majority and it will get %99 accuracy rate (if validation set is similar) however this does not mean it is a good model. In this situation reported accuracy becomes misleading.

2.2

In confusion matrixes in 2.2 and 2.3 columns shows actual classes and rows shows the predicted classes of our models.

		ACTUAL VALUES				
		0th class	1st class	2nd class	3rd class	4th class
	0th class	24	38	67	33	23
PREDICTED	1st class	0	0	0	0	0
VALUES	2nd class	0	0	0	0	0
	3rd class	0	0	0	0	0
	4th class	0	0	0	0	0

In validation set there are 185 articles. As we can see from the confusion matrix all values predicted as 0th class(athletics) and only 24 of them is true. Wrong predictions number is 38 + 67 + 33 + 23 = 161. Accuracy rate is 24/ 185 which is 12,97 %.

2.3

		ACTUAL VALUES				
		0th class	1st class	2nd class	3rd class	4th class
	0th class	24	0	0	0	1
PREDICTED	1st class	0	35	0	0	0
VALUES	2nd class	0	1	66	0	0
	3rd class	0	2	1	33	0
	4th class	0	0	0	0	22

In validation set there are 185 articles. Number of correct predictions are 24 + 35 + 66 + 33 + 22 = 180. Accuracy rate is 180 / 185 which is 97,3 %.

2.4

We can clearly see that after smoothing with Dirichlet prior $\alpha = 1$ our models accuracy rate increased from 12,97% to 97,3%. Our model without smoothing was not an ideal model it alwasy predicted the first class as we can see in part 2.2. The reason behind that is our dataset. Our dictionary consist of 4613 words and while we look for the posterior possibility of each article with each category we multiply likelihood and prior possibilities. While we calculating the likelihood possibility we multiply $\mathbf{P}(\mathbf{X_j} \mid \mathbf{Y} = \mathbf{y_k})$ for every j and as our dataset consist of 4613 (too much) words, for an article for every category in validation dataset there are some words that is not included in those categories. Therefore, since for every article in validation set there are some j that make $\mathbf{P}(\mathbf{X_j} \mid \mathbf{Y} = \mathbf{y_k}) = \mathbf{0}$ for every k. As a result of this likelihood become 0 and posterior possibility becomes 0. When we calculate through logarithm and then compare then logarithms values become -inf. As a result of this our program in 2.2 always predicts first option. To inhibit this type of situation we should do smoothing like in 2.3. When we assume every word is counted 1 more then we save probabilities from becoming 0 so it enable us to predict more accurately. The big difference between 2.2 and 2.3 caused from probability of zeros and smoothing them in 2.3.