**Report**

1. **Formulation of the problem:**

Develop class “NaiveBayes” with methods “predict” (determines class of input text) and “fit” (separate database to test and train bases in a relationship 20/80, using “predict” method to all docs in test base, count accuracy).

Develop methods to prepare database for “predict” and “fit” methods.

1. **Preprocessing:**

Input data:

*['0', 'The Chinese In Beijing The Chinese']*

*['0', 'Chinese Chinese In The In Shanghai']*

*['0', 'The Chinese The In In Macao']*

*['1', 'In Tokyo In The Japan The In Chinese']*

Lowercase:

*['0', 'the chinese in beijing the chinese']*

*['0', 'chinese chinese in the in shanghai']*

*['0', 'the chinese the in in macao']*

*['1', 'in tokyo in the japan the in chinese']*

Spliten DataBase:

*['0', 'the', 'chinese', 'in', 'beijing', 'the', 'chinese']*

*['0', 'chinese', 'chinese', 'in', 'the', 'in', 'shanghai']*

*['0', 'the', 'chinese', 'the', 'in', 'in', 'macao']*

*['1', 'in', 'tokyo', 'in', 'the', 'japan', 'the', 'in', 'chinese']*

Deleting stopwords with ***nltk***:

*['0', 'chinese', 'beijing', 'chinese']*

*['0', 'chinese', 'chinese', 'shanghai']*

*['0', 'chinese', 'macao']*

*['1', 'tokyo', 'japan', 'chinese']*

1. **“Predict” method:**

Test sample:

*“Chinese The Chinese In Chinese Tokyo The In The Japan”*

Test sample after preprocessing:  
*“chinese chinese chinese tokyo japan”*

Count probability of each classes:  
*0 – 0,75  
1 – 0,25*

Create vocabulary table:  
*[['chinese', 'beijing', 'shanghai', 'macao', 'tokyo', 'japan', 'ALL']]*

Count occurrence of each word:  
*[['chinese', 'beijing', 'shanghai', 'macao', 'tokyo', 'japan', 'ALL'],  
[8, 1, 1, 1, 1, 1, 0],  
[1, 0, 0, 0, 1, 1, 0]]*

Adding “+1” to each cell and count “ALL” column:  
*[['chinese', 'beijing', 'shanghai', 'macao', 'tokyo', 'japan', 'ALL'],  
[9, 2, 2, 2, 2, 2, 19],  
[2, 1, 1, 1, 2, 2, 9]]*

Count probability of each word of each class:  
*[['chinese', 'beijing', 'shanghai', 'macao', 'tokyo', 'japan', 'ALL'],  
[0.4736, 0.1052, 0.1052, 0.1052, 0.1052, 0.1052, 19],  
[0.2222, 0.1111, 0.1111, 0.1111, 0.2222, 0.2222, 9]]*

Count probability occurrence of each class:

*[0.00030121377997263036, 0.00013548070246744226]*

Log:

*[-8.10769031284391, -8.906681345001262]*

Answer of test:*['Chinese The Chinese In Chinese Tokyo The In The Japan', '0']*

1. **“Fit” method:**

**Returned** '100%' accuracy

1. **Inference:** Algorithm work correct on the test case.