Here the test rate on average is 30 percent. Which means that around 30 percent of the data set belongs to clickbait dataset.

Since this classifier predicts all records as true ,

TP=0.3

FN=0

FP=0.7

TN=0

Precision=TP/(TP+FP)

=0.3/(1)

=0.3

Recall=TP/(TP+FN)

=0.3/0.3

=1

F1 score : harmonic mean of precision and recall =2\*(P\*R)/(P+R)=0.46

Here in this case the f1 score entirely depends on the distribution of the datasets(ie the one and the zero class). Based of the distribution it results a precision recall and f1 score. As the positive class increases the f1 score and the precision increases.

More than MultimodelNB(Naive base), which fails to capture the dependence between the random variable(words in this case), the better way is to collect the word features in each and every class by vectorization and use a classifier like Random forests(ensemble of decision tree), SVM, XGBoost etc to classify the text.

(a)

Rule based classifier results:

precision 1.0 recall 0.05214723926380368 f1 score 0.09912536443148688

Naive base classifier results-

precision 0.948170731707317 recall 0.9539877300613497 f1 score 0.9510703363914372

From these results it is evident that Naive base classifier performs better compared to rule based classifier.

Navie base classifier, basically computes the discrete frequency of words, and computes the probability or the likelihood that a particular word falls in the particular class. High probability (considering log probablity) words signifies the charecteristics of a particular word.

Using rule based classifier to find the exact match of high probable words would ignore certain records because of its inability to generalize data which leads to poor performance of unseen data.

Comparing the trivial based classifier and navie base, in navie base there are two parameters involved in the Likelihood, prior and the posterior probability. Where as in the trivial classifier it only depends on the distribution of the class(number of positive and negative class).

In rule based classifier it has a small degree of randomness in characterizing words according to its class with high likelihood where as in trivial classifier it does not account randomness except the positive and negative class distribution.

(b)

I would take on some preprocessing tasks like TF-IDfs to filter down the words and perform word embedding to get linguistic features. This would give scope to explore and deepen further modelling techniques like XG Boost, RNNs, Deep Neural Networks and state of the art transformers(capturing global context and dependencies)