**Multinomial Bayes**

**Feature selection**: tf-idf method to all words with lemmatization(should talk about why using this method).

**Data processing**: label coding: convert 11 text classifications into numbers. The following table is as follows:

|  |  |
| --- | --- |
| Category name | Category code |
| ARTS CULTURE ENTERTAINMENT | 0 |
| BIOGRAPHIES PERSONALITIES PEOPLE | 1 |
| DEFENCE | 2 |
| DOMESTIC MARKETS | 3 |
| FOREX MARKETS | 4 |
| HEALTH | 5 |
| MONEY MARKETS | 6 |
| SCIENCE AND TECHNOLOGY | 7 |
| SHARE LISTINGS | 8 |
| SPORTS | 9 |
| IRRELEVANT | 10 |

**Method and reasons:**

All the words are counted as features using td-idf method with lemmatization, which are trained with the whole training set. with cross validation.

Td-idf can be implemented by TfidfVectorizer() of sklearn.feature\_extraction.text, and lemmatization can be achieved by lemma() from pattern.en.Actually I tried with and without lemmatization and the results of the following parts are exactly same. But the feature number reduces from 35822 to 35290.

To prevent overfitting, we should apply cross validation method. Multinomial Bayes function of sklearn has the parameter alpha as a smoothing hyperparameter, we should use cross validation to find optimal alpha. Here we use GridSearchCV() method of sklearn to efficiently find out the best alpha in range of 0.0-10.0 with step of 0.01.We set the cross-validation method as 10-fold cross-validation and use the scoring method of f1-macro. The result of finding best alpha for this training set is 0.01.

The specific code implementation uses the multinomialnb() in the sklearn package to fit the training set, and then forecasts the test set. The prediction accuracy after cross validation for training set is 89.3% and for test set is 76.6%.

Since 10 articles are recommended for each reader for each topic, the method used here is to calculate the probability of each article in the test set for each topic by using the predict\_proba() function, and then select the 10 articles with the highest probability for each topic, and recommend them to the corresponding readers. We find that for some topics which has quite small number of articles in the test set and the recommended 10 articles method can have poor effect. For example, for the types of arts culture enterprise, the whole test set has only 3 articles, and only 5 articles are recommended for training test with Bayesian model. In this case, if you choose other unrelated articles in order to gather 10 recommended articles, the reader may feel upset and stop using the app; at the same time, when you test the test set with the model, you find that for some topics, the trained model does not have recommended articles at all, such as SCIENCE AND TECHNOLOGY, there are three pieces of labeled articles in this topic in the actual test set (9604, 9722, 9929 respectively),but the model is not able to pick any one of them, so that if there is no recommended article, it can not meet the requirements of the readers.

So for the user's good experience, the method selected here is: if the number of recommended articles of the topic predicted by the model is less than 5, then the model recommends that the top 5 most probable articles related to the topic to the user.; if the number of recommended types is more than or equal to 5 and less than or equal to 7, then all these number of articles are recommended, and no any other article are recommended, so as to prevent too many irrelevant articles.

If the number of articles of this type recommended by the model training is 8 or more, then the 10 most probable articles in the user's preference topic can be selected. (for 8 or 9 cases, one or two articles that are not classified can be mixed in. At the same time, the articles predicted by the model basically appear in the top ten of the probability of the specific topic, so the articles calculated by using the top ten of the probability calculated by the model can be used to approximate the articles predicted by the model.). The output results are in the order of probability from large to small.

**Performance Measurement**

After applying such method, we get the following result.

Performance measurement of the whole test set:

|  |  |  |  |
| --- | --- | --- | --- |
| Topic name | Precision | Recall | F1 |
| ARTS CULTURE ENTERTAINMENT | 0.20 | 0.33 | 0.25 |
| BIOGRAPHIES PERSONALITIES PEOPLE | 0.71 | 0.33 | 0.45 |
| DEFENCE | 0.62 | 0.62 | 0.62 |
| DOMESTIC MARKETS | 1.00 | 0.50 | 0.67 |
| FOREX MARKETS | 0.43 | 0.42 | 0.43 |
| HEALTH | 0.75 | 0.86 | 0.80 |
| MONEY MARKETS | 0.55 | 0.74 | 0.63 |
| SCIENCE AND TECHNOLOGY | 0.00 | 0.00 | 0.00 |
| SHARE LISTINGS | 1.00 | 0.14 | 0.25 |
| SPORTS | 0.95 | 1.00 | 0.98 |

Performance measurement of recommendation set:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Topic name | Suggested articles | Precision | Recall | F1 |
| ARTS CULTURE ENTERTAINMENT | 9604,9789,9952,9830,9526 | 0.20 | 0.33 | 0.25 |
| BIOGRAPHIES PERSONALITIES PEOPLE | 9878,9940,9758,9724,9988, 9933, 9797. | 0.71 | 0.33 | 0.45 |
| DEFENCE | 9576,9559,9607,9773,9616,  9842,9770,9783,9987,9670. | 0.80 | 0.62 | 0.70 |
| DOMESTIC MARKETS | 9796,9989,9640,9762,9767. | 0.20 | 0.50 | 0.29 |
| FOREX MARKETS | 9584,9599,9823,9743,9625, 9659,9704,9693,9902,9572. | 0.50 | 0.10 | 0.17 |
| HEALTH | 9661,9807,9873,9621,9947, 9929,9735,9911,9982,9833 | 0.90 | 0.64 | 0.75 |
| MONEY MARKETS | 9916,9586,9939,9853,9766, 9737,9863,9769,9765,9835. | 0.90 | 0.13 | 0.23 |
| SCIENCE AND TECHNOLOGY | 9617,9982,9722,9929,9621. | 0.40 | 0.67 | 0.50 |
| SHARE LISTINGS | 9601,9972,9999,9563,9732. | 0.60 | 0.43 | 0.50 |
| SPORTS | 9922,9857,9568,9800,9754, 9848,9574,9760,9858,9787. | 0.80 | 0.13 | 0.23 |

**Evaluation of results**

We focus on precision because it represents the ratio of articles recommended to users to meet the expected topic.

The result of the test set shows the model is less effective for the topic with fewer articles. For example, For SCIENCE AND TECHNOLOGY, there are only three labeled articles (9604, 9722, 9929) in the test set, The result of model prediction shows the prediction accuracy rate is 0,meaning that the model is unable to pick any of the three, after implementing the recommendation method I talked before ,the top five predictions of model test are 9617, 9982, 9722, 9929, 9621, and two of them are correct prediction, which shows performance improvement. And the same case of small test set for DOMESTIC MARKETS. This time the model prediction of precision is 1 and the recall is 0.5, indicating that 1 of 2 articles are picked out, but when recommendation number is promoted to 5,the precision accuracy is only 0.2,meaning 1 of 5 is correct prediction and the rest 4 is irrelevant to DOMESTIC MARKETS. Under this case, readers may find that the search results are far from the expectations and don't like to use the app. A larger set of tests is needed to properly pick out articles of this type. For the topics with many articles in the test set, the prediction effect is relatively good. For example, HEALTH, the accuracy of the predicted 10 articles reaches 90 percent.and some topics with general effect are used in the whole test set, but the precision is improved after using the recommended algorithm, such as MONEY MAKERS, from 0.55 to 0.90. For SHARE LISTINGS with a significant decline in precision (1.0 - > 0.6), only 9601 are obtained in model prediction. Although it is a correct prediction, only one prediction obviously fails to meet the readers' expectations. When it is increased to 5 recommendations, the precision is 0.6, indicating that two more articles of SHARE LISTINGS are picked out, and the recommendation is relatively in line with the readers' requirements.

Generally speaking, there is no mathematical basis for this kind of recommendation, but it is more in line with human nature (considering the readers' feelings and the number of articles recommended in a balanced way in the maximum of 10 cases), and the training effect is relatively good. The promotion requires a larger test set, so the recommended articles will be more and more in line with the expectations of the readers.

Reference:

https://towardsdatascience.com/text-classification-in-python-dd95d264c802

<https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624>

<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>

Introduction to Machine Learning with Python: A Guide for Data Scientists, chapter 7