OSDA project

Valentin Kopylov, 1st-year Master's student «Data Science»

19 December 2021

1 Introduction

In this paper, we are presenting the results of implementation of the algorithm of Lazy classification of objects as the binary features. We are using Tic Tac Toe data for the algorithm testing, additionally we use one-hot encoding to convert multinomial features of the dataset to binary, as it is suggested by the method. Firstly, we will present and discuss 4 versions of the classification algorithms, secondly, we will search for the optimal parameter of the third algorithm and, finally, we will compare the results using classification metrics such as accuracy, precision, recall and f1-score. Additionally, we shall compare fit of our classification algorithms to the fit of the two classical ML methods: Logistic Regression and Naive Bayes Classification.

2 Algorithm 1

Method based on the «Generators». We are interested in positive and negative train examples separately. For each test object we do 2 procedures:

1. Find intersection of the features between each positive object and test object, afterwards we are checking whether their intersection description fits into feature representation of the negative objects

2. Find intersection of the features between each negative object and test object, afterwards we are checking whether their intersection description fits into feature representation of the positive objects

If the intersections numbers are equal to each other, than we use random variable with value of 0 or 1.

Afterwards, using simple majority rule, we are deciding whether we should classify test object as positive or as negative based on the comparison between positive and negative cardinality of suitable intersections.

3 Algorithm 2

The second algorithm is based on the Algorithm 1 with the special correction. In the Algorithm 1, we were interested in comparison of the absolute values of positive and negative cardinalities. In this case, we are interested in the relative comparison, since it allows us to solve classification task with the unbalanced targets classes.

4 Algorithm 3

The third algorithm is the improvement of the Algorithm 2. Additionally, while finding the suitable intersections between the test object and train object of the particular class, we set the restriction on the minimal size of the feature map of the intersection results. Since it is the hyperparameter, in the section 6, we will provide the comparison between different values of such restriction.

5 Algorithm 4

This algorithm is based on the calculation of the mean intersection of the features between test object and train example. For the classification cause, we will be interested in the 20 largest intersection between the test object and train set of the particular class. Afterwards, we will compare 2 mean intersections and classify objects as of the largest intersection class.

6 Optimal parameter for Algorithm 3

For the third algorithms we constructed the possible restriction values from 0.6 to 0.95. On the table above, we can observe the results with the different parameter value based on the data of train1 and test1.

C	Accuracy	Precision	Recall	F1
0.6	0.8495	0.873	0.9016	0.887
0.65	0.8709	0.9622	0.0.8360	0.8947
0.7	0.8709	0.9622	0.0.8360	0.8947
0.75	0.8172	0.9782	0.7377	0.8411
0.8	0.9355	1	0.9016	0.9482
0.85	0.9355	1	0.9016	0.9482
0.9	0.4838	0.67	0.4098	0.5102
0.95	0.5268	0.6603	0.5737	0.6140

As we can see, the provided metrics are rising starting from $c = 0.6$ to $c = 0.85$, reaching its maximum value in terms of our metrics. Afterwards, the quality of the algorithms significantly reduces almost to the random part, since the accuracy is equal approximately to 0.5. Thus, it means that after this moment, there might be no intersections and we are obtaining the results through the random option of the algorithm.

7 Comparison of results

Thereafter, we used 5-fold training to obtain better classification based on the train4 data. We used StratifiedKFold, since it is better suited for the unbalanced data. Here is the table with the best results of each algorithm.

Algorithm	Accuracy	Precision	Recall	F1
1	0.7803	0.7483	1	0.8560
2	0.8843	0.8549	0.9912	0.9180
3 with $c = 0.85$	0.9884	1	0.9823	0.9910
4	0.9306	0.904	1	0.9495
Naive Bayes	0.7052	0.7672	0.7876	0.7772
Logistic Regression	0.9884	0.9826	1	0.9912

As we can see on the table, based on the accuracy and f1-score, we see that the Algorithm 3 outperforms its counter algoritms 1 and 2. Therefore, we can conclude that the suggested improvements of the original algorithms gave us significant prediction power in terms of accuracy and f1 score. The results of the algorithm 3 is similar to the Logistic Regression measures, accuracy of the both models are the same, however the f-score of the regression model is slightly higher. Also, the 4 algorithms show good results compared to the Algorithm 1 and 2, however it lacks fit in comparison to Algorithm 3. Therefore, we can conclude, that the Lazy classification algorithm with relative comparison rule and intersection size restriction achieves good fit, comparable to the Logistic Regression.