CS 584: Data Mining (HW3)

Md. Ridwan Hossain Talukder
miner2 ID: Luffy
Rank on miner2: 1
F1 Score: 0.83

I. Problem Statement

In this homework, we are trying to develop an ensemble classifier with boosting (AdaBoostClassifier) that can determine given a particular compound whether it is active (1) or not (0).

II. Methodology

A. Balancing Imbalanced Dataset

The training data set has an imbalanced distribution of 78 actives (1) and 722 inactives (0). As the number of active records is deficient compared to inactive records, I have used RandomUnderSampler, an undersampling technique to undersample the minority records.

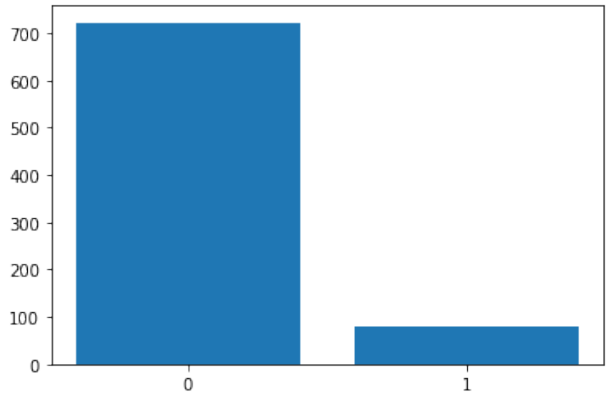

Fig. 1. Imbalanced distribution of dataset

B. Feature Selection

SelectKBestFeatures selects the feature according to the k highest scores, for that I used the chi_square score function to select the k best features. This function "weeds out" the features that are most likely to be independent of class and, therefore, unimportant for classification because the chi-square test analyzes dependence between stochastic variables. I have also stratified my training data for validation, as this is an imbalanced classification problem we need to allocate the samples evenly based on sample classes so that the training set and validation set have a similar ratio of classes. So I used StratifiedKFold for that purpose. I used 5 fold to cross-validate the f1_score and took the average. In Figure 2 as we can see that the f1 score is highest for the k value of 255 (in a range of 190 to 300), so I decided to use the 255 best features selected by the SelectKBestFeatures method. I used DecisionTree (with max depth 1) as Base Classifier in my AdaBoostClassifier while doing this experiment (selecting the K value). And it is also supported by the precision-recall curve in the same figure, where the recall and precision(close to highest) are highest for the k value of 255.

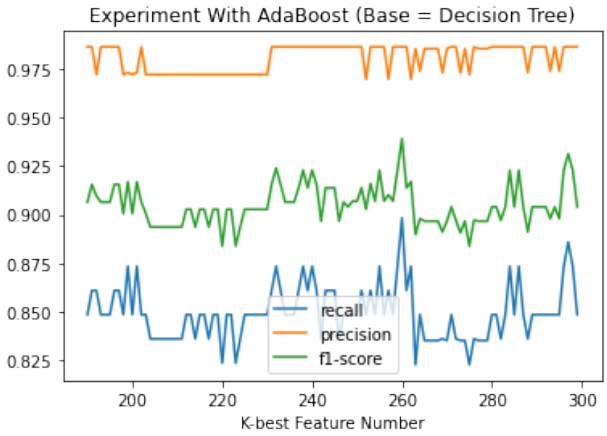

Fig. 2. F1, Precision, Recall Score for different K value (Range 190-300)

C. AdaBoostClassifier Implementation

In short, my approach to building the AdaBoost classifier is as follows:

- Initially set the same weights to all the samples (as we have yet to see which samples are not well classified by our model)
- We then choose a base classifier either Decision Tree (with max depth 1) or BernouliNB based on a boolean value.
- Classify using the weak classifier and determine the misclassified labels by comparing the true labels with the predicted labels. And using that misclassified labels we then compute the misclassification rate which is computed by dividing the total weight of the misclassified label by the total weights of the label. Total weight of the misclassified labels is simply the sum of all weights of the respective labels.
- Using the misclassification rate (error) we calculate the alpha value for each weak classifier which is calculated by the equation $log\frac{1-error}{error}$
- We update each label's weight with the new alpha value and normalize the weights. As more weights are put on

labels that have been wrongly classified now the decision tree will use the samples that have misclassified earlier and now has more weight. Each weight is updated using this equation $w_i = w_{i-1} * e^{\pm\alpha}$

- Once the weight is updated for the samples we repeat the process for the next classifiers and it goes on till the number of weak classifiers reaches the given limit (n_estimators). During this process, we store the fitted weak classifiers and the alpha value for each classifier for future prediction.
- Now for prediction we use each weak classifier to predict the labels of the test instances. Then we take the weighted sum of the predictions for each label calculated by this equation $\sum \alpha * predition$ predicted by the classifiers. If the weighted sum is above a threshold we predict the label as 1, otherwise 0. For this classification problem, I converted all the 0 label's as -1 so that we can set the threshold at 0.

D. Experiments

1) Decision Stump as base classifier: We run our experiments using Decision Stump (Decision Tree with max depth 1) as base classifier for choosing the value for a number of classifiers needed to correctly classify the instances. For this, we do StratifiedKFold cross validation in training data using the AdaBoostClassifer(Base = DecisionTree(max_depth=1)) to predict the training labels using a value of n = 1 to 50 for the number of weak classifier while RandomUnderSampling the data and selecting 255 best features. And in training data, we got our best result (based on the f1 score) for n = 34, but it was overfitted as it gave a poor result in the test dataset. So in the training set, so we then tried the last spike (n=28) we saw in Figure 3 where we got the best value for f1, recall and precision and trying values near to that point (n=27) gave us a good result in test data (0.80).

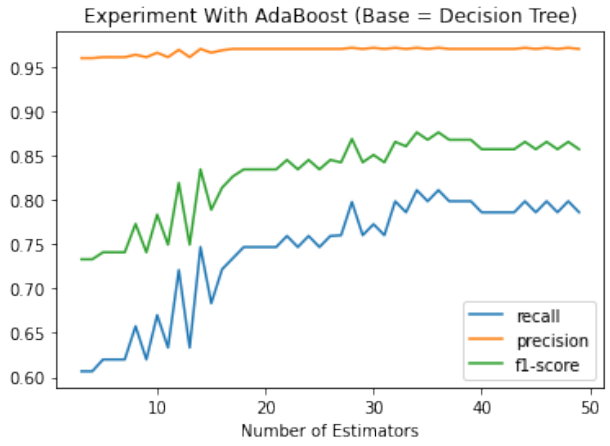

Fig. 3. F1, Precision, Recall Score for different K value (Range 1-50)

2) Bernoulli Naive Bayes as base classifier: We run our experiments using Bernoulli Naive Bayes as base classifier for choosing the value for a number of classifiers needed to correctly classify the instances. For this, we do StratifiedKFold cross validation in training data using the AdaBoostClassifer(Base = BernoulliNB) to predict the training labels using a value of n = 1 to 30 for the number of weak classifier while RandomUnderSampling the data and selecting 255 best features. And in training data, we got our best result (based on the f1 score) for n = 28, but it was overfitted as it gave a poor result in the test dataset. In the training set, values after 14 was similar and was overfitting, we can see this in Figure 4. So we used n = 14 and it gave us the best result in test data (0.83)

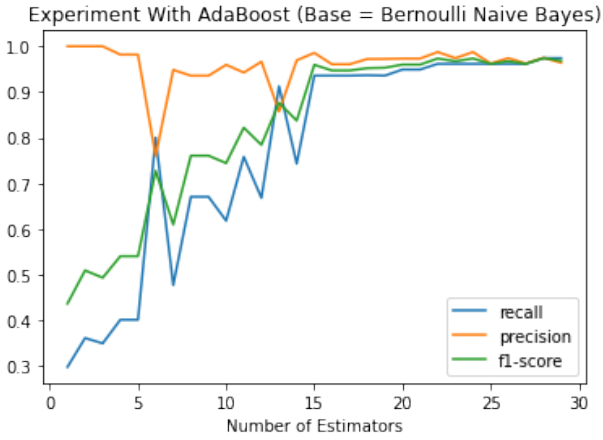

Fig. 4. F1, Precision, Recall Score for different K value (Range 1-30)

III. Results

So we choose BernoulliNB as our base classifier for the AdaBoostClassifer we designed. We do Random under sampling for balancing the data, and select (K=255) for KBest Feature Selection, and our model gave an f1 score of 0.83 on training data when we cross validated with StratifiedKFold with k-fold(k=5). It also gave an equal performance on miner (0.83 on miner2) on the test data (approximately as the percentage on the miner is approximate at the time of writing this report). To compare with previous result (HW2) we get a better performance for using AdaBooster Classifier than using a classifier alone.

TABLE I
AVERAGE F1-SCORE FOR DIFFERENT BASE CLASSIFIER FOR ADABOOSTCLASSIFER

Classifier	f1_Score	HW2 Score (without boosting)
AdaBoost(BernoulliNB)	0.83	0.77
AdaBoost(Decision Tree)	0.80	0.67