COMP-551 Project 2 Report: Classification of Textual Data

Group 20: Diyang Zhang, Yuhang Zhang, Olivia Xu

March 2020

1. Abstract

In project 2, we implemented classification algorithms including Logistic Regression, Decision Trees, Support Vector Machines, AdaBoost, Random Forest on 2 datasets : the *20newsgroup* dataset and *IMDb* reviews. Multinomial Naïve Bayes is also implemented as an Extra, with the same analysis as previous classifiers. Different accuracies based on different training & testing data are carefully studied.

To investigate the individual performance of different hyperparameters, GridSearchCV & RandomSearchCV in *sklearn* are used on all classifiers and on both datasets. After these cross-validations, we applied the optimal model again on testing data and compare the optimal results with previous ones. Highest KCV scores are 0.76101 (SVM) & 0.89284 (SVM) for *20newsgroup* and *IMDb* respectively, whereas highest testing accuracies are 0.68104 (SVM) & 0.88316 (LR) for the two datasets.

2. Introduction

Beginning with examining our datasets, we removed headers, footers and quotes and then tokenized text files with scikit-learn package to get vocabulary frequency features. We used the built-in split of training and test. Different pipelines are made for the 2 datasets. And we use the *CountVectorizer* & *TfidfTransformer* to transform the raw textual data to vectors for later use.

Next, we built six classifiers in different files: Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), Ada Boost (AdaBoost), Random Forest (RF) and Multinomial Naïve Bayes (MNB). We fit our training data to construct the model and computed the K cross-validation (KCV) results for each model and dataset. We presented training & testing accuracies and several other observations as well.

Finally, we adjusted hyperparameters for each model to obtain higher accuracy. We searched for the best hyperparameters by tuning the important ones using *GridSearchCV* & *RandomizedSearchCV*. Logistic Regression and SVM showed the best performance on both datasets. And Random Forests also perform well, compared to other algorithms. Moreover, hyperparametes tuning successfully results in slight improvement on our KCV scores and testing accuracies.

In addition, we separated the 20 news group data into four subsets : *Comp, Rec, Sci and Talk*. Six classifiers were implemented respectively. We got interesting patterns, which was presented in detail in 6.4. We found out that all classifiers achieved relatively the highest accuracies on subset *Rec* and the lowest accuracies on subset *Talk*.

3. Related work

Some previous studies have suggested that Decision Tree, Naïve Bayes, SVM are among the most popular text classification methods. Performance depends on different datasets. A common approach is to implement and compare different classifiers on each dataset. Hyperparameters tuning could have a high computational (time) cost, but typically (sometimes slightly) improves the accuracy.

4. Datasets and Setup

First, we used *sklearn* datasets package to load the data and remove header, footers and quotes. Then we imported *CountVectorizer* which built a dictionary of features and transformed documents to feature vectors. Finally, we applied *TfidfTransformer* to get vocabulary frequency features before fitting the data into classifiers. Different pipelines are made. In addition we split the 20newsgroup dataset into 4 subsets for further study.

4.1 *20newsgroup*

The *20newsgroups* dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training and the other one for testing. The split between the train and test set is based upon a message posted before and after a specific date. Our goal was to predict the topic of news.

4.2 *IMDb* Reviews

This is a dataset for binary classification. It is a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. 50% of the reviews in this dataset are positive.

5. Proposed Approach

We used the built-in split of training and testing of both datasets. Six classifiers were then trained. Logistic Regression, Decision Trees, Support Vector Machines, Ada Boost and Random Forest could all be implemented for multi-categorical classifications. We also impletmented the multinomial Naïve Bayes classifier as an EXTRA for some interesting findings. Thess classifiers are all applied to both of *20newsgroup* and *IMDb* datasets, and we attempt to find the most accurate one in each case.

Testing accuracies, as well as Training Accuracies as an EXTRA, are both presented, while we perform training both on training data but calculate accuracies of predictions based on the original testing data & training data respectively. Some analysis are made toward the comparisons between these two accuracies. K-fold cross validation (KCV) follows to guarantee the authenticity of our experiments, where we choose by default k=5. Eventually, we attempt to tune parameters, using GridSearchCV & RandomizedSearchCV in *sklearn* in order to find the optimal training model along with the optimized parameters for each of the datasets and overall performance.

The default values of parameters that we used for training are given below:

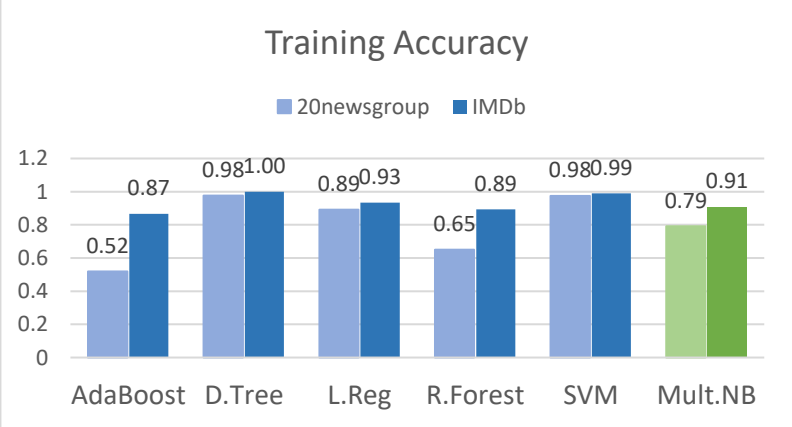
- AdaBoost: n_estimators = 200, learning_rate = 0.8.
- Decision Tree: *Sklearn* Defaults. Note that the nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- Logistic Regression: solver = “liblinear” which handles l1 penalty.
- Random Forest: n_estimators = 200, min_sample_split = 16, min_sample_leaf = 16.
- SVM: tolerance = 1e-5.
- Multinomial Naïve Bayes: *Sklearn* Defaults

Remark that Some EXTRA experiments are also implemented, and the relative results are shown below.

6. Results

6.1 Training Accuracies (Extra)

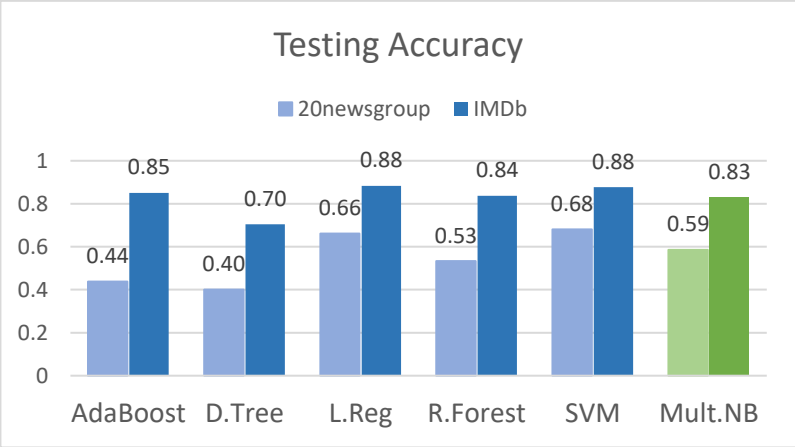
Here we show the training accuracies based on predictions on original training data.



We observe that Decision Tree & SVM results in higher accuracy score than other models. They have both an accuracy approaching 100%. This result is especially obvious for Decision Tree, as mentioned before, since we chose *Sklearn* Default parameters, for which the nodes of tree are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

6.2 Testing Accuracies

Here we show the results of testing accuracies based on predictions of testing data.



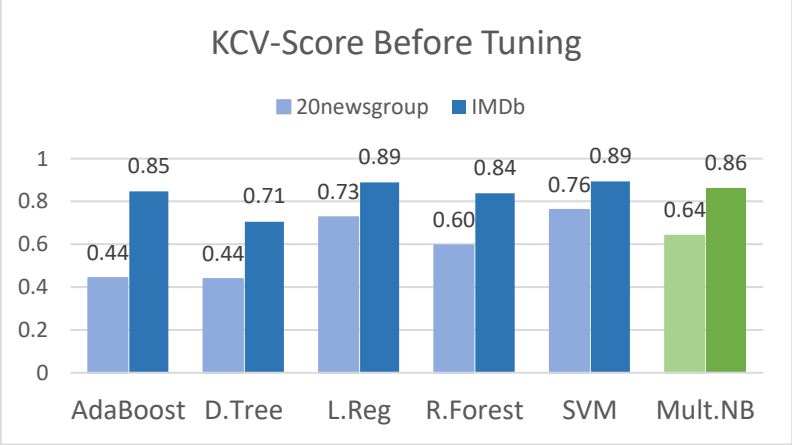
First, note that the test accuracies using Decistion Tree models for both of the two datasets are significantly lower than the training accuracies shown above. This should be a direct result of overfitting, as with the *Sklearn* Default parameters the tree always tries to be expanded until all leaves are pure. Therefore although the model performs well on the amount of original training data, it has relatively bad performance when predicting new (or unknown) instances.

Generally speaking, the accuracies of binary predicitions for *IMDb* are much higher than the accuracies of multi-class predictions for *20newsgroup*. For *20newsgroup*, highest accuracies of above 65% are obtained using LR or SVM models, acceptable accuracies from 50% to 60% are obtained using RF or Mult.NB, and relatively lowest accuracies less than 45% are obtained by AdaBoost or Decision Tree. For *IMDb* reviews, except Decision Tree, all of the remaining models result in a high accuracy greater than 80%.

We also remark that AdaBoost yields the greatest difference of accuracies when prediciting the two different datasets. And its accuracies change extremely few, compared to other models, when perform predictions on training data or testing data. This is a consequence of the sensitivity of Adaboost to noisy data and outliers. It can be less susceptible to the overfitting problem than other learning algorithms in this case. Moroever, we know that Adaboost is a process of accumulatinig individual weaker learners, which corresponds perfectly to our observation that running Adaboost requires more time for predictions than other algorithms.

6.3 KCV Score before parameters tuning

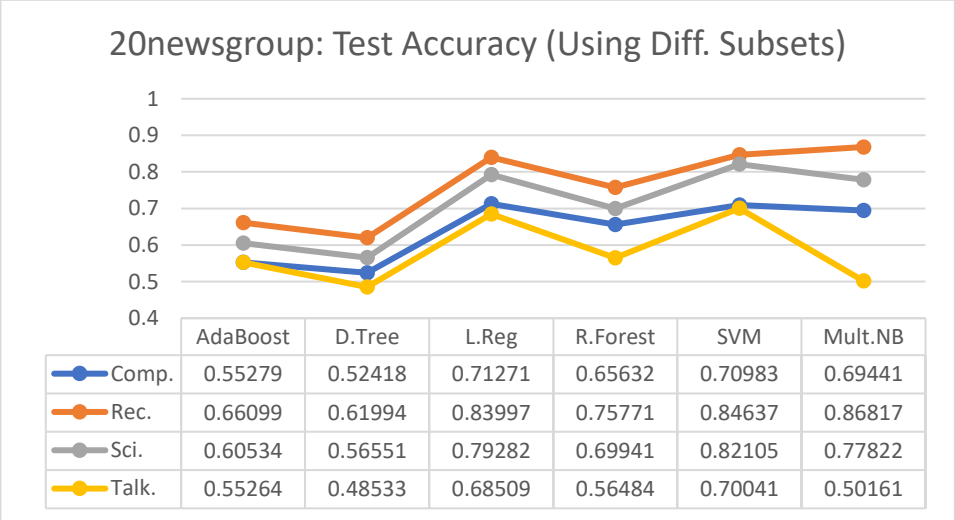
Here we show the KCV (k=5) score before tuning hyperparameters.



We observe that the KCV scores for all the xperiments do not vary a lot with respect to testing accuracies. This ensures the authenticities of our experiments. For example, the low accuracy for Decision Tree on *20newsgroup* successfully give us an insight of overfitting when using this generalized model to predict new data.

6.4 Testing Accuracies using different training data subsets (Extra)

Here, we divide the 20 categories for *20newsgroup* dataset into four "classes" *i.e.* the categories starting with *Comp.* , *Rec.* , *Sci.* and *Talk.* respectively. We then compare the test results based on test sets, which perform training only on each of these "classes" (subsets) of the original 20 categories training data individualy. The folliwing graph demonstrates our results.



	AdaBoost	D.Tree	L.Reg	R.Forest	SVM	Mult.NB
Comp.	0.55279	0.52418	0.71271	0.65632	0.70983	0.69441
Rec.	0.66099	0.61994	0.83997	0.75771	0.84637	0.86817
Sci.	0.60534	0.56551	0.79282	0.69941	0.82105	0.77822
Talk.	0.55264	0.48533	0.68509	0.56484	0.70041	0.50161

We observe that the using uniquely *Rec.* results in the highest testing accuracies, even higher than our previous testing accuracies based on training perfomed on the whole *20newsgroup* training data. For the other three subsets of training data, the testing accuracies obtained are likewise closed to our previous results, and sometimes higher. We notice that high-bias models will not benefit from more training examples, and also considering overfitting, this observation might be presumably a consequence of this.

On the other hand, another interesting found is that while using different classification models, we always get the same order of acurracies, from *Rec.* (the highest) to *Talk.* (the lowest), for using the 4 subsets of training data. Moreover, comparing to our previous accuracies, we find out that the order of accuracies by applying different models (i.e. training subsets fixed, comparisons between accuracies on different models, LR, SVM and Mult.NB higher than the others) are also almost the same.

6.5 Optimization of hyperparameters

Here we used tables to show the hyperparameters that we tried for training. The opimal ones are highlighted.

Note that for Multinomial Naïve Bayes (Mult.NB) that we implemented as an Extra, there is no actual needs to tune hyperparameters, as given by defaults, alpha equals to 1.0 for smoothing and both fit_prior & class_prior should better be set as default values for regular learning & adjusting.

For *20newsgroup* datasets:

AdaBoost

n_estimators	100	200	300
learning_rate	0.4	0.8	1.2
optimal KCV	0.46041		
optimal Test Acc	0.46044		

Decision Tree (DT)

max_depth	200	400	600
min_samples_split	8	12	16
min_samples_leaf	8	12	16
optimal KCV	0.42602		
optimal Test Acc	0.41009		

Logistic Regression (LR)

solver	newton-cg (L2)
	lbfgs (L2)
	liblinear (L1)
	sag (L2)
	saga (L1 & L2)
optimal KCV	0.72538
optimal Test Acc	0.66086

Random Forest (RF)

bootstrap	TRUE		FALSE
n_estimators	100	200	400
min_samples_split	8	12	16
min_samples_fit	8	12	16
optimal KCV	0.65034		
optimal Test Acc	0.59063		

SVM

C (Regularization)	1		2
max_iter	500	1000	1500
tol (tolerance)	1e-3	1e-4	1e-5
optimal KCV	0.76101		
optimal Test Acc	0.68104		

For *IMDb* reviews:

AdaBoost			
n_estimators	100	200	300
learning_rate	0.4	0.8	1.2
optimal KCV	0.85376		
optimal Test Acc	0.85605		

Decision Tree (DT)			
max_depth	200	400	600
min_samples_split	8	12	16
min_samples_leaf	8	12	16
optimal KCV	0.71724		
optimal Test Acc	0.71219		

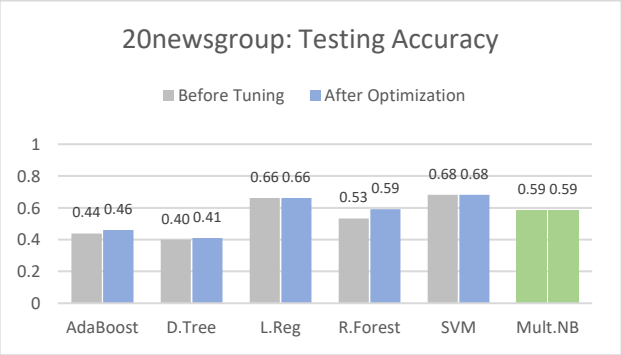
Logistic Regression (LR)	
solver	newton-cg (L2)
	lbfgs (L2)
	liblinear (L1)
	sag (L2)
	saga (L1 & L2)
optimal KCV	0.88836
optimal Test Acc	0.88316

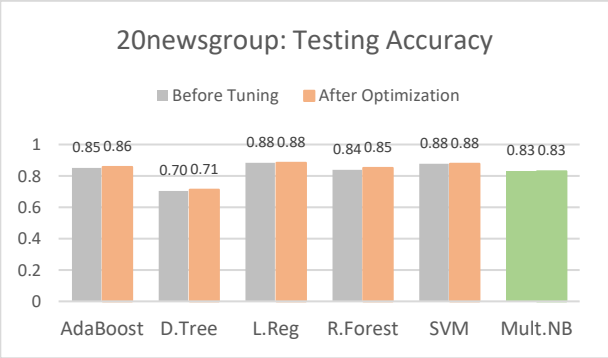
Random Forest (RF)			
bootstrap	TRUE		FALSE
n_estimators	100	200	400
min_samples_split	8	12	16
min_samples_fit	8	12	16
optimal KCV	0.8482		
optimal Test Acc	0.84996		

SVM			
C (Regularization)	1		2
max_iter	500	1000	1500
tol (tolerance)	1e-3	1e-4	1e-5
optimal KCV	0.89284		
optimal Test Acc	0.87718		

Note that the optimal parameters are in general different for the two different datasets.
The highest K Cross-Validation (KCV) scores (k=5) can be obtain by:
20newsgroup: 0.76101 (SVM with L1-regularization, 500 max iterations & 1e-3 tolerance).
IMDb: 0.89284 (SVM with L1-regularization, 500 max iterations & 1e-4 tolerance).

Then we compare these optimized testing accuracies with the previous results:





The accuracies after tuning parameters are always greater or equal to the accuracies we obtained before. This unsure the correctness of our algorithms.

The optimal parameters are shown on the above tables. And we conclude that our highest testing accuracies are:
20newsgroup: 0.68104 (SVM with L1-regularization, 500 max iterations & 1e-3 tolerance).
IMDb: 0.88316 (Logistic Regression with solver *lbfgs* which handles L2 penalties).

7. Discussion and Conclusion

Predictions on *IMDb* datasets have in general higher accuracy than on *20newsgroup* dataset. While some models (Decision Tree) have a extremely high accuracy on training data, it doesn't imply the high score on new testing data for the reason of overfitting. Logistic Regression & SVM have a better overall performance than other models through our implementation, while AdaBoost, Random Forests and Multinomial NB also perform well on *IMDb* dataset. Tuning hyperparameters is time-consuming. Different datasets have different optimized parameters. But this can slightly improve the accuracy of our predictions.

8. Statement of Contribution

All members made adequate contributions to this project.

9. References

[1] Rafael G., Andre L., Joaquin V., Bernd B. and Andre C., "To tune or not to tune: recommending when to adjust SVM hyperparameters via Meta-learning", International Joint Conference on Neural Network, 2015
[2] Le Luo, Li Li, "Defining and Evaluating Classification Algorithm for High-Dimensional Data Based on Latent Topics", PLOS One, vol9(1), 2014