Comparison on spam email classification with different methods

Name : Chao Fu

Liu-ID : chafu696

Course Code : 732A92

1 Introduction

Remote working is prevalent during the epidemic recently. Hence, there is a significant increase in the use of email. In this situation, spam email has attracted considerable attention. Generally, the number of spam emails is significantly less than non-spam email's which can lead to a problem of imbalanced data. There are two methods in transforming imbalanced data into a balanced one, oversampling and undersampling. To process text using machine learning or Neural Network models, text data need to be encoded into vectors of numerical values. There are two typical methods for text processing, Term frequency–inverse document frequency(tf-idf) and word embedding. Overall, an optimal combination of methods and models for spam email classification is facing challenges and it is worthwhile devoting much effort to this. This project aims to compare the discrepancies among the different methods in spam email classification and find an optimal combination. With this motive, two extra data, oversampling and undersampling, are created from the original one. Then tf-idf and word embedding matrix are determined for each data. In this project, four cutting-edge models are applied which are Logistic Regression, Support Vector Machine, Random Forest and TextRNN(LSTM). Finally, the classification accuracy on the test data is measured with these models based on different tf-idf and word embedding matrix. The results exhibit that (1) Original imbalanced data has the lowest accuracy and undersampling data ranks first in all models except Support Vector Machine. (2) In word embedding method, the TextRNN(LSTM) model ranks first in all three data compared with other ones. (3) In tf-idf method, Support Vector Machine ranks first in all three data compared with Logistic Regression and Random Forest. (4) In tf-idf method, Logistic Regression is the most sensitive to imbalanced data. (5) In word embedding method, Random Forest is the most sensitive to imbalanced data. (6) Support Vector Machine with tf-idf method and undersampling data in this project has the largest accuracy which is the same result as the kaggle solution reaching 96%. (7) Logistic Regression with tf-idf method and undersampling data in this project has a slight lower result than the kaggle solution(95% to 97%). (8) TextRNN(LSTM) with word embedding method and undersampling data in this project also has the largest accuracy reaching 96%. (9) Undersampling can be used instead of the imbalanced one to obtain an optimal classification performance. (10) Although both undersampling and preprocessing by removing information on original data can have equivalent performance based on Support Vector Machine model, undersampling can save human time. (11) Many steps should be applied to transform the data into a specific form that can be used in TextRNN(LSTM). Although both TextRNN(LSTM) and Support Vector Machine can perform on the same level in small data, the simple model, support vector machine, can be an optimal selection.

2 Background

2.1 Theory

1) Oversampling and Undersampling

The performance obtained by the existing learning system can be affected by imbalanced labels in training data which means that the number of one label tremendously exceeds the other one. In this case, the learning system is facing challenges to learn the information behind the minority label. There are two non-heuristic methods to obtain balanced data by random selection from minority label's examples with replacement(oversampling) and from majority label's examples without replacement(undersampling)[1].

Without losing any information from original data is the main merit of the oversampling method. However, it has many drawbacks such as extensive time consumption, serious overfitting risk, misleading information behind minority label. Although the undersampling method can save running time, it can lose some important information in the majority label.[2]

2) Logistic Regression

The conventional logistic regression equation is as follows:

$$P(Y_i) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots\beta_p X_p}}$$

Using logit to transform the basic logistic regression into a form of multiple linear regression is given below:

$$logit(\hat{Y}) = ln(p/1-p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_p X_p = \beta^T X$$

The interpretation of this equation is given below:

1) $p = P(Y_i)$ is the estimated probability of each response variable value occurring.
2) β_0 is the intercept which is regarded as a constant value. $X_1, X_2, \ldots X_p$ are p number of explanatory variables. $\beta_1, \beta_2, \ldots \beta_p$ are their regression coefficients. β is a regression coefficient matrix($(p+1)\times 1$) and X is a sample matrix($(p+1)\times N$) in which N is the number of observations.

3) Support Vector Machine

For binary classification, Support Vector Machine(SVM) is implemented by maximizing the margin to minimize the maximum loss[3].

The hard SVM equation is given below[4]:

$$\begin{aligned}&\underset{w,b}{min}\,\frac{1}{2}\|w\|^2\\&subject\ to\ y_n(w^T X + b) \geq 1\ for\ all\ n = 1, 2, \ldots, N\end{aligned}$$

The interpretation of this equation is given below:

1) b is the intercept which is regarded as a constant value. w is a vector $(p \times 1)$ normal to the hyperplane and X is a sample matrix($p \times N$) in which N is the number of observations.

4) Random Forest

"Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest."[5]

5) Long Short-Term Memory (LSTM)

Hochreiter and Schmidhuber developed an innovative method called "Long Short-Term Memory"(LSTM) which can fix the limitation of recurrent neural network(RNN)[6]. Hence, the LSTM method is prevalent in text classification based on the context.

6) TextRNN(LSTM)

TextRNN is used for text classification with LSTM, which means bi-directional LSTM[7].

2.2 Method

1) Creating four data

The original data is imbalanced, in which the samples of Non-Spam heavily outnumber Spam. Random selection of the same number of Spam from Non-Spam samples without replacement to obtain undersampling data(balanced data). Then, the random selection of the same number of Non-Spam from Spam samples with replacement to obtain oversampling data(balanced data). With additional test data, four data are obtained, original(imbalanced data), oversampling(balanced data), undersampling(balanced data)finally and test data.

2) Creating pre-trained word embedding weights matrix

The number of all three data is not sufficient to train the word embedding weights in TextRNN(LSTM) learning system. Hence, the pre-trained word embedding weights matrix is applied in its word embedding layer. The original, oversampling and undersampling data generate pre-trained word embedding weights matrix respectively based on the spacy module.

3) Finding the best hyper-parameters

For both word embedding and tf-idf method, the original data is applied to find the best hyper-parameters in Logistic Regression, Support Vector Machine and Random Forest model. Then, the best parameters are fixed in each of the three models.

The pre-trained word embedding weights matrix is only used in TextRNN(LSTM). 70% of original data is used as training data, the rest is for validation data. Then, they are used to find the best epoch number based on the pre-trained word embedding weights matrix of original data. This process is also applied in oversampling and undersampling data.

4) Training models with best hyper-parameters and obtaining the accuracy on test data

After the first 3 steps, 7 learning systems are obtained with the best hyper-parameters respectively. Then, these systems are trained by original, oversampling and undersampling data respectively. The accuracy of these trained systems based on test data which is unseen in the training process is measured.

3 Numerical example

This *data* was obtained from Kaggle.

The train and test data both have 2 variables :

One is the response variable, a binary variable with two classes: "Spam" and "Non-Spam". The other is the explanatory variable with the email message.

Both train and test data don't contain missing data. However, the training data is imbalanced, "Spam" with 122, "Non-Spam" with 835.

With the aim of this project, two other balanced data are created from the original data, one is undersampling and the other is oversampling.

4 Results

Table 1: The accuracy of tf-idf

Method	Original_tf-idf	Oversampling_tf-idf	Undersampling_tfidf
Logistic Regression	79%	90%	95%
Support Vector Machine	92%	90%	96%
Random Forest	82%	82%	87%

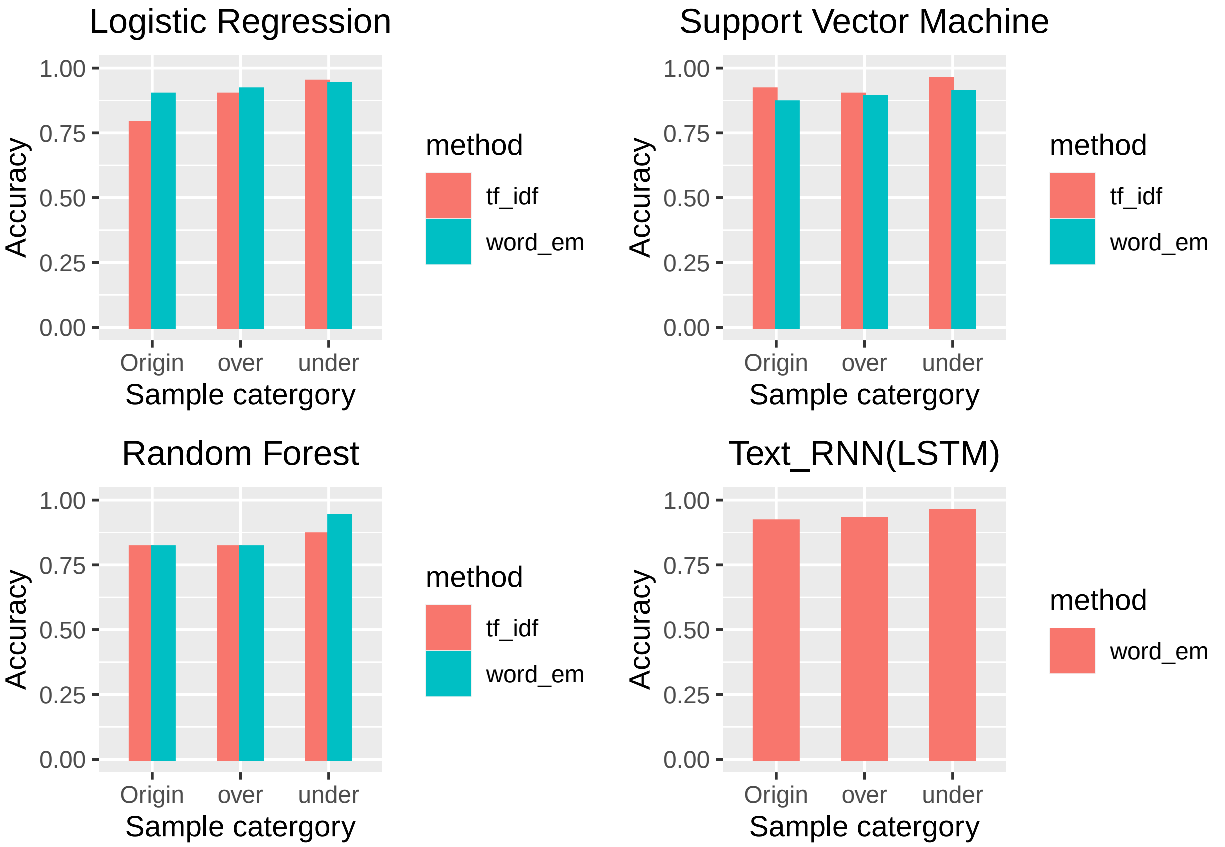
Three models apply the tf-idf method. The values in the table are classification accuracy on the test data.

Table 2: The accuracy of word-embedding

Method	original_word-embedding	oversampling_word-embedding	undersampling_word-embedding
Logistic Regression	90%	92%	94%
Support Vector Machine	87%	89%	91%
Random Forest	82%	82%	94%
TextRNN(LSTM)	92%	93%	96%

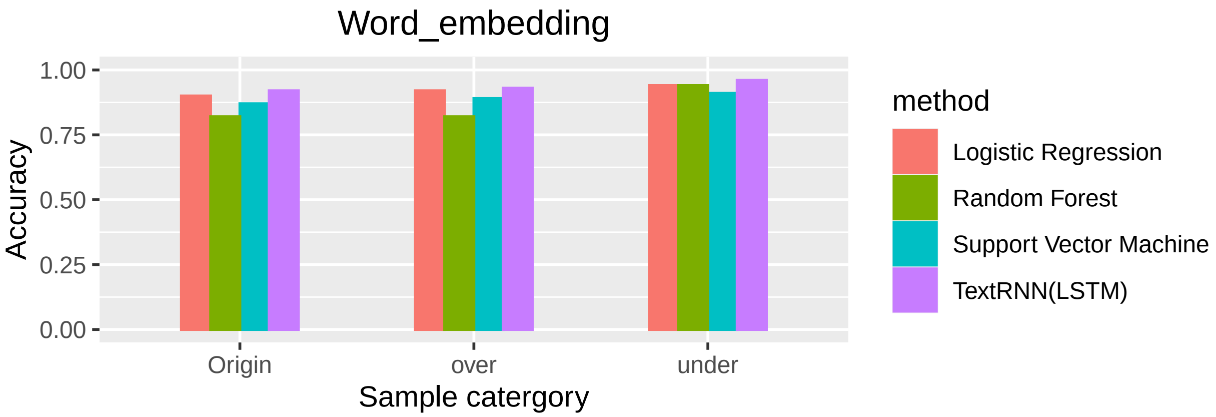
Four models apply the word embedding method. The values in the table are classification accuracy on the test data.

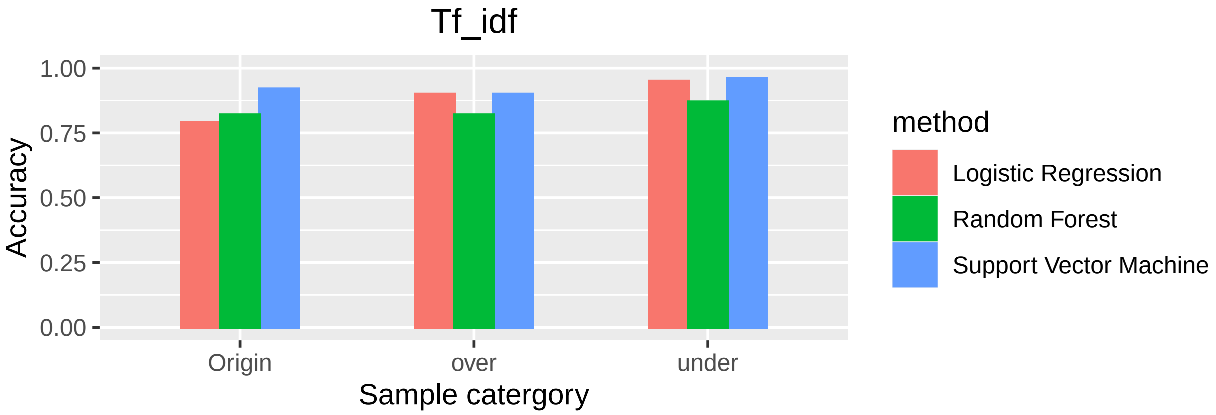
Figure 1: Comparison on each model in three data



Each model has a graph to show the comparison on the tf-idf and word embedding methods in original, oversampling and undersampling data.

Figure 2: Comparison among models in three data





Each method has a graph to show the comparison on the four different models in original, oversampling and undersampling data.

5 Discussion

(1) Imbalanced data has the lowest accuracy and undersampling data ranks first in all models except Support Vector Machine.

(2) In word embedding method, TextRNN(LSTM) model ranks first in all three data compared with other ones.

(3) In tf-idf method, Support Vector Machine model ranks first in all three data compared with Logistic Regression and Random Forest.

(4) In tf-idf method, Logistic Regression is the most sensitive to imbalanced data.

(5) In word embedding method, Random Forest is the most sensitive to imbalanced data.

(6) The *solution* based on the same data is published in kaggle. This solution used Natural Language ToolKit(NLTK) module. The spacy module is used in this project.

(7) The kaggle solution processes the original data by removing some information. However, this process can lose some potentially important information. Hence, this project does not remove any information from the original data. Moreover, the kaggle solution only used original data(imbalanced data) and tf-idf method but three different data and two different methods are used in this project.

(8) Support Vector Machine with tf-idf method and undersampling data in this project has the largest accuracy which is the same result as the kaggle solution reaching 96%.

(9) Logistic Regression with tf-idf method and undersampling data in this project has slightly lower results than the kaggle solution(95% to 97%).

(10) TextRNN(LSTM) with word embedding method and undersampling data in this project also has the largest accuracy reaching 96%.

6 Conclusion

(1) Undersampling can be used instead of the imbalanced one to obtain an optimal classification performance.

(2) Although both undersampling and preprocess by removing information on original data can have equivalent performance based on the Support Vector Machine model , undersampling can save human time.

(3) Many steps should be applied to transform the data into a specific form that can be used in TextRNN(LSTM). Although both TextRNN(LSTM) and Support Vector Machine can perform on the same level in small data, the simple model, support vector machine should be an optimal selection.

7 References

[1] G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[2] P. Kaur and A. Gosain, “Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise,” in *ICT based innovations*, Springer, 2018, pp. 23–30.

[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on computational learning theory*, 1992, pp. 144–152.

[4] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning.* Cambridge University Press, 2020, pp. 370–379.

[5] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] J. Cai, J. Li, W. Li, and J. Wang, “Deeplearning model used in text classification,” in *2018 15th international computer conference on wavelet active media technology and information processing (ICCWAMTIP)*, 2018, pp. 123–126.