International Journal of Advanced Studies in Computer Science and Engineering
IJASCSE Volume 5, Issue 1, 2016

Comparative study of Naïve Bayes, Support Vector Machine and Random Forest Classifiers in Sentiment Analysis of Twitter feeds

Anmol Nayak
Department of Electrical and Electronics Engineering
PES Institute of Technology
Bangalore, India

Dr. S Natarajan
Department of Information Science Engineering
PES University
Bangalore, India

***Abstract*—Sentiment analysis is one of the most active and widely investigated techniques in Machine Learning. It is used in a variety of problems like determining the viewpoint of a writer, overall emotion of the document and so on. The fundamental aim is to classify the polarity of the given document, sentence or feature. We performed Sentiment analysis on a standard Movie reviews Twitter feed dataset to investigate the three prominent supervised learning classifiers: Naïve Bayes, Support Vector Machine (SVM) and Random Forest, and hence determine the most accurate of them based on the results obtained for positive and negative polarity tweets.**

Keywords—Supervised Learning; SVM; Machine Learning; Natural Language Processing

I. INTRODUCTION

Machine learning is a field of computer science that has evolved into one of the most powerful domain that has enabled computers to make decisions without being programmed explicitly. Machine learning and AI algorithms are based on mathematical optimization to understand and make accurate predictions on data. It is widely employed in search engines, recommendation systems, driverless cars, spam filtering, etc. The tasks handled by machine learning are typically classified under supervised learning (labeled data), unsupervised learning (unlabeled data) and reinforcement learning (computer interacts with a real time environment where it has to perform a goal without being taught on how to achieve it). Sentiment analysis (also called as Opinion mining) uses computational linguistic and natural language processing [1] models for text understanding. It faces challenges in short string texts, varying contexts and a myriad of opinions of individuals, which makes it extremely hard to analyze.

II. DATA PRE-PROCESSING

In any dataset there will always be noisy, redundant and irrelevant data which needs to be cleaned up. Failing to do so makes the training phase extremely hard which results in poor performance of the classifiers. This is an important step in achieving high accuracy from any classifier, as it helps in training with the best possible data. There are several techniques involved in data pre-processing and cleaning up the dataset. We have used multiple stages to pre-process a standard movie reviews tweet dataset. This includes maintaining a uniform letter case, stemming the data using Porter stemmer, stripping of special symbols commonly found in tweets such as '@, #, \," etc., removing additional whitespaces and redundant words, and many such steps. We have also used term frequency- inverse document frequency (tf-idf) statistic as it is a powerful indicator to show the importance of a given word in the dataset. This also helps in stop words (words which do not affect the polarity of a given text) removal from the dataset. After the dataset has been pre-processed, it is used to train the classifiers.

III. DATA CLASSIFICATION

During the classification and prediction stage, different classifiers can be used. In this paper, we have used three popular supervised learning classifiers namely, Naïve Bayes, Support vector machines and Random Forests. Machine learning classifiers tend to be probabilistic, non-probabilistic or ensemble in nature.

A. Naïve Bayes Classifier:

Naïve Bayes [2] is a probabilistic classifier based on Bayes theorem, with the features being independent of each other. Each feature is considered to contribute to the probability of any given test instance to belong to a particular class. Consider ***n*** features to be represented as a vector:

$$\mathbf{x} = (x_1, \ldots, x_n) \quad (1)$$

The probabilities that the Naïve Bayes model [3] assigns to the ***k*** classes will be as follows:

$$p(C_k|x_1, \ldots, x_n) \quad (2)$$

Implementing Bayes theorem, we can determine the conditional probability of predicting the class given a feature.

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}. \quad (3)$$

The Bayes classifier then designates a particular class label to a given test instance based on which is the most probable class. Since we have represented the data in terms of tf-idf vectors, we have used Multinomial variant of Naïve Bayes classifier. Let the distributed dataset have parameter vectors as:

$$\theta_y = (\theta_{y1}, \ldots, \theta_{yn}) \quad (4)$$

, for every class ***y*** and ***n*** being the number of features (or the size of the vocabulary in our case).

In the above equation, θ_{yi} represents the probability of a feature ***i*** found in sample of class ***y*** given by P ($\mathbf{x_i}$ | **y**). The parameters of the vector is optimized by a smoothing factor alpha (=1 in our case for Laplace smoothing) in the following equation:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (5)$$

, where N_{yi} is count of occurrence of feature ***i*** found in a sample of class ***y*** and N_y is the total count of all the features occurring in a sample of class ***y***.

B. *Support Vector Machine Classifier:*

Support vector machines [4] are associated with learning algorithms which learn from data to decipher patterns in classification and regression analysis. SVM models aim to find a hyperplane that separates the data points lying in the different classes as wide as possible so that when a new sample comes in, it is classified based on which side of the gap they fall in.

The hyperplane equation for every class ***y*** and points ***x*** has the following constraints:

$(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1$, if $y_i = 1$ (6)

$(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \leq -1$, if $y_i = -1$ (7)

, where b is a constant, ***w*** is called the weight vector, and $\|\boldsymbol{w}\|$ is minimized to maximize the separation between the classes. While implementing SVM models, one must supply parameters such as C, gamma, etc. to obtain the highest accuracy keeping in mind the bias-variance tradeoff.

Bias-variance [6] dilemma is frequently dealt with in supervised learning algorithms as we tend to generalize beyond the training set. Bias error leads to underfitting of the data as it misses important interaction between the features and classes. On the other hand, variance error leads to overfitting as it is highly sensitive to noise and fluctuations that may be present in the training set. The accuracy of a model is largely dependent on these parameters, and hence the optimum values are found using grid-search technique.

C. *Random Forest Classifier:*

Random Forests [5] is a powerful ensemble learning algorithm often used in classification tasks. It classifies based on the results obtained from the myriad of decision trees it generates while training, where the mode of the targeted outputs from each decision tree is the output of the forest. Since trees are known to overfit data as they have low bias and high variance, Random Forests tend to average out the multitude of decision trees.

Random Forests generate decision trees on random samples of the training data, thereby reducing the variance in the overall model improving its performance and controlling overfitting. In classification trees, features are represented in nodes of the trees where in the most important features are high up in the tree and leaves represent the class labels. The importance of a feature is determined by Gini impurity, where in the lesser the decrease in accuracy by randomly permuting the values of the feature, the lesser is the importance of the feature.

IV. Experimental Evaluation

The experiment has employed our Python source code that uses Machine learning and Natural language Processing libraries like nltk, scikit-learn etc. It is conducted by dividing the dataset (of 2000 positive and negative polarity tweets) into training and test samples.

Once the preprocessing is performed, each classifier is trained after which the classifiers are individually evaluated on the test samples. We have used three metrics (precision, recall and f1-score) to evaluate the performance for both positive and negative polarity tweets.

Precision is the ratio of true positive/ (true positive + false positive) items. It is a measure of the quality of the algorithm to return relevant results. An algorithm with high precision returns more pertinent results as compared to irrelevant results.

Recall is the ratio of true positive/ (true positive + false negative) items. It is different from precision in the fact that it is a quantity measure to check how many relevant results are being returned by the algorithm while precision checks from the selected items, how many of them were relevant.

F1-score [6] is the harmonic mean of the precision and recall often used as a weighted average for balancing quality vs. quantity of true positives selection of an algorithm given by:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

Table 1, 2 and 3 show the results obtained for the Naïve Bayes, Support vector machine and Random Forest classifiers respectively when deployed on the testing set. As we can see, for positive polarity tweets Naïve Bayes classifier obtains the highest precision of 85% as compared to 82% and 81% for SVM and Random Forest classifiers, while in the case of negative polarity tweets we see that both Naïve Bayes as well as SVM attain precision >90% although SVM is slightly better at 92% precision with Random Forests achieving 89% precision. A similar trend is observed for both polarity tweets when measured for recall and f1-score, with Naïve Bayes producing highly accurate classification and comparatively better performing as compared to SVM and Random Forest classifiers in this dataset.

Figures 1 and 2 sum up the results obtained for the three classifiers with respect to positive and negative polarity tweets.

A. Figures and Tables

TABLE I.

Polarity of tweets	**Results obtained on Naïve Bayes classifier**		
	Precision (%)	***Recall (%)***	***F1- Score (%)***
Positive	85	87	86
Negative	91	90	90

TABLE II.

Polarity of tweets	**Results obtained on Support Vector Machine classifier**		
	Precision (%)	***Recall (%)***	***F1- Score (%)***
Positive	82	87	85
Negative	92	88	89

TABLE III.

Polarity of tweets	**Results obtained on Random Forest classifier**		
	Precision (%)	***Recall (%)***	***F1- Score (%)***
Positive	81	85	81
Negative	89	86	87

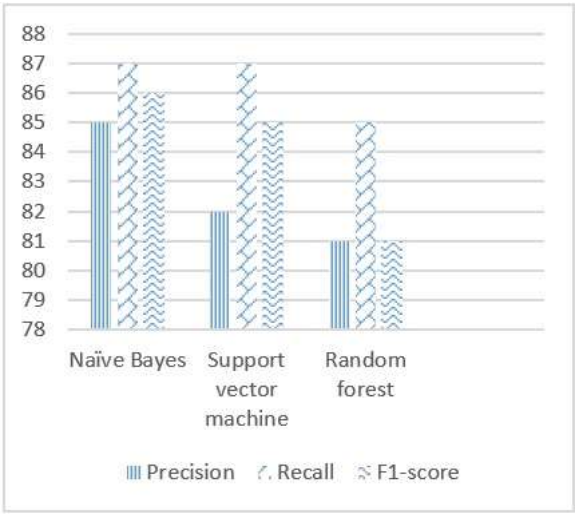

Fig. 1. Graphical results of classifiers for positive polarity tweets

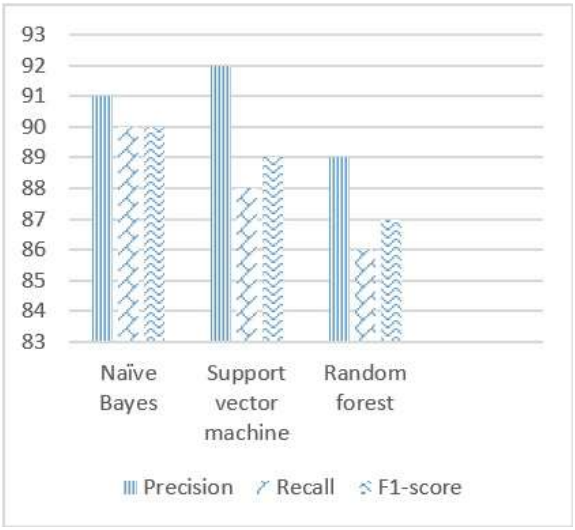

Fig. 2. Graphical results of classifiers for negative polarity tweets

V. Conclusion and Future Work

In this paper, we provided a thorough comparison and performance analysis of the salient classification algorithms used in supervised learning namely Naïve Bayes, Support vector machine and Random Forests. For our standard movie reviews tweet dataset, we can say that Naïve Bayes was the most accurate with a score of 89% followed by SVM (88%) and Random Forests (85%). As there is not definitive ranking on which classifier works best in all cases, it is important to choose algorithms based on the nature and characteristics of the database such as size, variance, reliability and many such vital factors. Compared to previous works which have shown 80% accuracy in classification for Naïve Bayes and SVM and 80.4% for Random Forests, we have shown an improvement of ~8% and ~5% for the given classifiers respectively.

As we know that machine learning, and more specifically Natural language processing are quintessential fields in Artificial intelligence, we can positively comment that there will be significant advances in algorithms, models and expansion to other niche areas of technology.

In the area of Sentiment analysis, short texts often contain emoticons [7] which are difficult to decipher by classification algorithms. These present an area for work to be carried on as emoticons when placed in texts can severely alter the polarity of the text.

Another area in sentiment analysis can be made using the feature or aspect based model. In this model, sentiment or polarity of all features of a given entity is found individually and hence the overall sentiment of the given entity can be determined. Feature based modelling gives a holistic insight to the entity as opposed to performing sentiment extraction on just individual features.

For future works, we can even carry out the experiment using Latent sentiment analysis [8] techniques which employs Singular value decomposition (SVD) in word count matrix.

REFERENCES

[1] Ralph Wieschdel, Jaime Carbonell, Barbara Grosz, Wendy Lehnert, Mitchell Marcus, Raymond Perrault, Robert Wilensky. "White Paper on Natural Language Processing" Proc. October 1989 DARPA Speech and Natural Language Workshop, pp. 481-493, October 1989.

[2] I. Rish, J. Hellerstein, and T. Jayram. An analysis of data characteristics that affect naive Bayes performance. Technical Report RC21993, IBM T.J. Watson Research Center, 2001.

[3] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In Proceedings of the Tenth National Conference on Artificial Intelligence, pages 399–406, San Jose, CA, 1992. AAAI Press.

[4] Christopher J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121-167, 1998

[5] Gerard Biau. Analysis of a Random Forests Model. Journal of Machine Learning Research 13 (2012) 1063-1095

[6] Marina Sokolova, Nathalie Japkowicz, Stan Szpakowics. Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation. Association for the advancement of artificial intelligence (2006)

[7] Davidov, D.; Tsur, O.; and Rappoport, A. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In COLING, 241–249.

[8] Lan Wang, Yuan Wan. Sentiment Classification of Documents Based on Latent Semantic Analysis. S.Lin and X. Huanh (Eds.): CESM 2011, Part II, CCIS 176, pp.356-361, 2011.