An Efficient Approach For Text Classification Using Ensemble Models in Machine Learning

AXEL GORIS RAFAT KHAN
goris | rafatk @kth.se

September 13, 2022

Abstract

In machine learning, the models process given inputs and produce an outcome. The outcome is a prediction based on what pattern the models finds during the training process. Machine learning algorithms have their limitations and producing a model with high accuracy is challenging. In many cases, one model is not enough for producing a reliable prediction because of the high variance of data which might result in low accuracy as model relies heavily on too few features while making a prediction. Instead, if we build and combine multiple models, we have the chance to boost the overall accuracy. The combination of models will reduce the model error and maintain the generalization of the model by aggregating the output from each model. In ensemble learning architecture, the weak learners receive inputs as well as predictions from each weak learner. A final ensemble model can be built using the combined predictions.

Contents

List of Acronyms and Abbreviations

1 Aims, Objectives, Goals, Research questions, hypotheses

The purpose of this project is to investigate the accuracy and performance of KNN, Naive Bayes and XGBOOST on a specific dataset for text classification. Then we will ensemble the models by aggregating the output from each model with two objectives: reducing the model error and maintaining its generalization by using some pre-defined metrics.

We have planned to reach our aims by implementing models for text classification and ensembling the models by aggregating the outputs. Then conduct a comparison survey of the accuracy and overall performance. If all the experiments are designed correctly, this project will provide conclusive decision on the performance of ensemble models for text classification.

We have divided our aims to some specific goals. We will implement models for text classification using three different algorithms. Then, we will measure the accuracy and overall performance for each algorithm using experimental data. After that, we will ensemble the models by aggregating the output from each model. We will also define a metric for aggregating the output. Then we will do a comparison of the accuracy and overall performance

The research questions can be defined as, whether or not, ensembling machine learning models improves performance and accuracy for text classification and if it does, to what level?

Our hypothesis is that ensembling machine learning models will improve performance and accuracy significantly for classifying texts.

2 Background and rationale

Texts are extremely rich sources of information. But the unstructured nature of the text makes it difficult and time-consuming to extract insights from it. Text classification techniques have made the task more convenient now a days. It is a machine learning technique for identifying predefined categories in open-ended text. Machine learning algorithms have been used to classify texts for a long time [1]. Although, Support Vector Machines [2], Naive Bayes Classifier[3], XGBOOST[4] and KNN[5] algorithms can classify texts, still none of these could produce high accuracy for multi-domain text classification. To solve this problem, the concept of model ensembling was introduced. In machine learning, ensemble models combine multiple other models for prediction. By using ensemble models, we can overcome the technical challenges of building a single estimator. An ensemble model can be built by stacking, blending, boosting and/or bagging [6].

In this project, we will work on the hypothesis of Ensemble Models using model stacking using KNN, Naive Bayes and XGBOOST algorithms to build a ensemble model for text classifications to determine if this approach enhance the accuracy and performance further. We will also conduct a comparison study of the accuracy and performance.

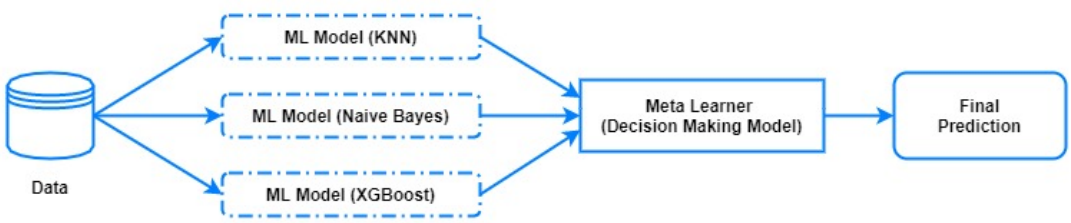

Figure 1: Architecture of proposed ensemble model.

3 Theory/literature

Accuracy and reliability can be more important over costing in different domain of text mining. To ensure maximum accuracy, we can use ensemble models. By combining multiple models instead of using just one, ensemble methods aim to improve the accuracy of results in models, combining the models increases the accuracy of the results significantly. As a result, ensemble methods have gained popularity in machine learning.

Sreemathy JayaprakashP et al. published a paper proposing text classification based on features selection and pre-processing, and thereby reducing the dimensionality of the Feature vector and increasing the classification accuracy [7]. From this paper, we got an idea to use Multiple-Model Techniques [8] to further enhance the performance. This concept of ensembling model is being used for different purposes. The ensemble forecasting approach combines different data sources, models of different types, assumptions, and/or pattern recognition methods. A more accurate ensemble model can be achieved by combining information from multiple sources, analyzing it with different techniques, and considering different sources of uncertainty in the real world [9]. Bharat Singh et al. proposed a scalable hybrid ensemble model for text classification by combining the bagging and boosting approach of the models [10]. This model gains its performance from using bagged ensemble of boosted trees. In this paper, they have used a model to build their own text classifier using the already existing or some self-modified bagging and boosting techniques. Charu Gupta et al. proposed a model for the classification of cybercrime using ensemble learning technique [11]. The proposed model is implemented with the help of model stacking, comprising Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest and gave better performance by having 96.55 % accuracy.

4 Research Methodology

The project will try to establish a positive relation between ensembling models and the performance, accuracy of the model. In order to achieve this, we'll need quantitative secondary data, available from Kaggle or Github. We won't be collecting data, instead we'll use available datasets while still making sure that they are not biased and contain sufficient datas to train, test and validate our models. By using highly used datas, we make sure that we don't introduce bias into the dataset and that the dataset has been tested through millions of use before. Moreover, it allows for others to quickly replicate our process.

Before starting the analysis in Python or R, we will check for missing datas, remove potential outliers and considers the independence of observations. The analysis will be conducted using F-measure [12] to verify the validity of our tests as was done before. [5].

5 Participants, Procedures, Data collection and analysis

The project will be organized as a two-person project. For conducting this research, we will need participation from our supervisors and peer reviewers.

We have found some relevent prior works and there are plenty of test data available in Kaggle [13], from where we can select suitable dataset and divide it into training data, validation data and test data. In parallel, we will start developing the experiments and implement the models.

We have selected a specific dataset to perform text classification on Kaggle. The dataset contains tweets relevent to coronavirus [14]. This dataset is publicly available, has been updated 2 years ago and is suitable for experimentation, because it has been manually labelled so that we can work on it.

6 Expected outcomes

A conclusive proof if ensembling KNN, Naive Bayes and XGBOOST can be a more efficient way for text classification in terms of efficiency, accuracy and performance.

A comparison survey on the performance of text classification between different ML algorithms.

7 Milestones/schedule, budget

The project will start on 6 September and end on October 10. There will be the following milestones and deliverable:

09 September - Find relevant prior works

12 September - Select experiment data

14 September - Start building the models

20 September - Start running the experiments

25 September - Define aggregation metrics for ensembling models

30 September - Evaluation of model for additional data points to be check by new experimental runs to check that the model accurately produces the expected outcomes.

Before 10 October - Submit final report (the report will have been written in parallel with each of the above steps)

8 Risks

For the risks involved, multiple ones are present:

- **Timeline** : considering the tight timeline and the time to build, train, validate analyse the data, if more than one of the steps involved appears to cause troubles, it might be difficult to hand the report in time.
- **Ressources** : considering the cost of training a model, even though we made sure to use a dataset that is not too big, the computer performance and the cost of upgrading it might prove to be a challenge. Furthermore, there might be a bias in the dataset that we did not clearly identify yet.
- **Results** : in addition to the previous point, we are at risk of failing to provide results significant enough to illustrate what we think. And considering the cost associated with ensembling models, even when needing robust methods, our method might be too expensive to work with.

9 Outline

Our report will follow the traditional format.

1. Abstract - Summary of the entire project.
2. Introduction - Aims, goals, importance, hypotheses, expectations, research questions.
3. Theory - What's available in the literature, what are we building on.
4. Methods - How are we going to do what we want, so that everyone can reproduce it.

5. Results and Analysis - What are our factual results, is the hypothese validated.
6. Discussion - Restate the problem, if the hypothese is validated and why, providing an explanation by using previous research paper. Where can our research be used ? How will it be used ?
7. Conclusion - What's the future for our research.

References

[1] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, vol. 34, pp. 1–47, Apr. 2001. doi: 10.1145/505282.505283

[2] C. Shan, "Research of Support Vector Machine in Text Classification," *Advances in Intelligent and Soft Computing*, vol. 119, pp. 567–573, Jan. 2011. doi: 10.1007/978-3-642-25538-0$_7$9

[3] D. Li-guo, D. Peng, and L. Ai-ping, "A New Naive Bayes Text Classification Algorithm," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, Sep. 2014. doi: 10.11591/telkomnika.v12i2.4180

[4] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, Aug. 2016, pages: 794.

[5] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Using kNN model for automatic text categorization," *Soft Computing*, vol. 10, Mar. 2006. doi: 10.1007/s00500-005-0503-y

[6] A. Kumar and M. Jain, *Ensemble Learning for AI Developers: Learn Bagging, Stacking, and Boosting Methods with Use Cases*, 01 2020. ISBN 978-1-4842-5939-9

[7] S. Jayaprakash and P. Balamurugan, "An efficient text classification using knn and naive bayesian," *International Journal on Computer Science and Engineering*, vol. 4, pp. 392–396, 03 2022.

[8] J. Brownlee, "A Gentle Introduction to Multiple-Model Machine Learning," https://machinelearningmastery.com/multiple-model-machine-learning/, 2022.

[9] H. Wu and D. Levinson, "Ensemble models of for-hire vehicle trips," *Frontiers in Future Transportation*, vol. 3, p. 876880, 04 2022. doi: 10.3389/ffutr.2022.876880

[10] B. Singh, N. Kushwaha, and O. Vyas, "A scalable hybrid ensemble model for text classification," 11 2016. doi: 10.1109/TENCON.2016.7848630

[11] C. Gupta and D. D. Virmani, "Ensem$_sldr$: $Classificationofcybercrimeusingensemblelearningtechnique,''$ $Internatio$ $-90,022022.doi$: $10.5815/ijcnis.2022.01.07$

[12] Y. Sasaki, "The truth of the F-measure," *Teach Tutor Mater*, Jan. 2007.

[13] "Text Classification." [Online]. Available: https://kaggle.com/code/matleonard/text-classification

[14] J. Brownlee, "A Gentle Introduction to Multiple-Model Machine Learning," https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification?resource=download, 2022.