UI Solution for Semantic Analysis using Machine learning: A comparative study

Arnav Sankhe
Dept. of physics and astronomy
University of Nottingham
Nottingham, England
ppxas5@nottingham.ac.uk

I. Introduction

Semantic analysis is the study to identify people's emotions in the written text like a review of a product, service or organization. It also includes finding opinions based on the comments of users on Youtube or Twitter. Sentiment analysis has become one of the most active research areas in natural language processing since the early 2000s as product reviews and tweets exploded on the internet. Identifying these emotions in product reviews or tweets and comments can lead to insights which can give a competitive edge to the business or an individual. Therefore three different Machine Learning models were trained Linear SVC, Extreme Gradient Boosting (XGBoost) and Random Forest respectively to extract semantics from the input text.

The dataset used to train these models is a Kaggle dataset consisting of Twitter tweets and Reddit comments. Before any type of operation could be done on raw data, the dataset needed pre-processing. In pre-processing the data was first cleaned and then the labels were normalized to the range 0-2, where 0 represents neutral, 1 represents positive and 2 represents negative emotions so that we could use these labels to train our models. Linear SVC, XGBoost and Random Forest algorithms were used to train models to predict the semantics of the user's input text, and gradio library was used to create a user interface to help users to interact with the trained models.

This report holds the finding of all the model performances respectively, that are trained on the same dataset for the given hyperparameters. In the methodology section, we'll take a detailed look at data pre-processing, and modelling algorithms like Linear SVC, XGBoost and Random Forest. The evaluation section consists of an evaluation metric that was used to measure the performance of the models. All the obtained models are compared in the results and discussion section, where we talk about the obtained result and the models performances. Finally, the conclusion consists of the final results and reflects on the success of this project as well as an idea for the future works.

II. Methodology

In this section, we'll have a detailed look at the specific methods chosen.

A. Data Pre-processing

The first step is data pre-processing where the dataset is checked for any missing values, and all the instances with the missing values are dropped, the reason to drop them was there were very few missing values. The next step is to normalize the label column, the values were normalized to range 0-2, where 0 represents neutral, 1 represents positive and 2 represents negative sentiment. The given figure shows the distribution of the dataset based on its labels. The last step in data pre-processing is data cleaning where data is converted to lower case, stop words and terms like mentions, hashtags and more are removed using regular expression [1].

Fig. 1. Distribution based on lables

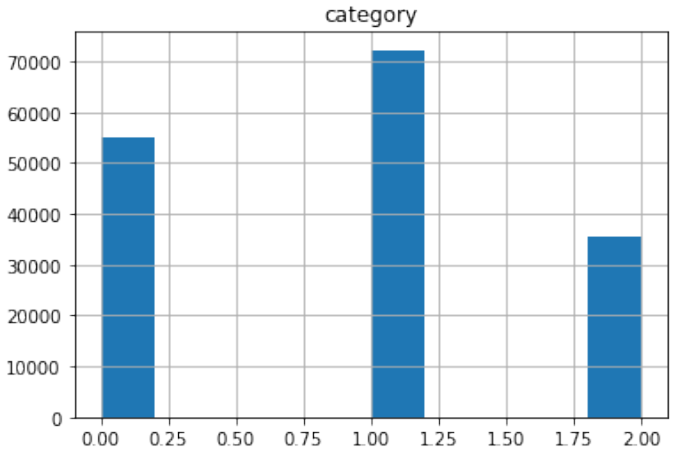


Fig. 2. Pre-processed data

	clean_text	category
0	when modi promised “minimum government maximum...	2.0
1	talk all the nonsense and continue all the dra...	0.0
2	what did just say vote for modi welcome bjp t...	1.0
3	asking his supporters prefix chowkidar their n...	1.0
4	answer who among these the most powerful world...	1.0

Next TfidfVectorizer is used to convert a set of raw documents into a TF-IDF feature matrix[2]. The TF-IDF allows us to associate each word in a document with a number that indicates how important that word is in that document. The hyperparameter max features was set to be 5000, this was decided by brute force method where few values were tried and best one was picked among them. Then the dataset was splited 80:20 for training and testing purpose using train test split from scikit-learn. Three models were trained i.e. LinearSVC, XGBoost and Random Forest using scikit-learn library.

B. Model Algorithms

1) LinearSVC:: A Linear SVC (Support Vector Classifier) is intended to fit data and produce the best fit hyperplane, which separates or categorises the data. [3][4]. After you've obtained the hyperplane, you can feed some features to your classifier in our case the input is text input which can be a product review or a tween to determine the "predicted" class i.e positive, negative or neutral.

2) XGBoost:: Extreme Gradient Boosting: XGBoost is a distributed gradient boosting algorithm designed for speed, flexibility, and portability. [5]. It implements machine learning methods using the Gradient Boosting framework. XGBoost solves a number of data science challenges rapidly and accurately using parallel tree boosting (also known as GBDT or GBM). [5]. For this project four hyperparameters for XGBoost were tuned n estimators which was set to be 100 learning rate which was set to be 1.0 and max depth was set to be 1 with random state set to 0.

3) Random Forest: A random forest is made up of a huge number of individual decision trees that work together as an ensemble. Each tree in the random forest generates a class prediction, with the highest votes becoming our model's forecast. [6]. The hyperparameters for this models were set as follows n estimators was set to be 100 and max depth was set to be 30 with random sate as 0. The main reason Random Forest did't perform very well was due to hyperparameters not being properly tuned, the random forest model with max depth 100 had one of the the best accuracy as well as F1 score but its size was too big when compared to other two i.e. Linear SVC and Extreme Gradient Boosting (XGB) models, which made it infeasible to even be considered for this study.

C. Technologies and challenges

Pickle format was used to save the trained models. Gradio a python library that makes it super easy to create user interfaces that was used to develop interface to help the users interact with the trained models. The input for this web app is basically the text in the form of tweet or a product review, typed by the user and the output is the predictions of all the given three models LinearSVC, XGBoost and Random Forest respectively. The other feature that is performed by the agent but dosent have an interface for is where the agent is able to takes input from the csv file, reads it analyse all the text,reviews and coments and gives its predicted output for the respective input. Python was used as the scripting language and scikit-learn library was mostly used for majority of work and gradio library was used to build the user interface and the whole code was coded and tested on google collab.

1) Challenges: The major challenges faced while developing this project were, firstly to select the machine learning algorithms, my first plan was to use deep learning to train the model but later decided to go with machine learning algorithms and compare the results with deep learning models. Three algorithms Linear SVC, Extreme Gradient Boosting and Random Forest were selected because of availability of relevant information and correct documentation. The second challenge that was faced while developing this agent was to select correct technology to used to create an interface to interact with the agent anvil library was given a try but later decided to go with gradio which is relatively much simpler and easy to use with proper documnetation.

D. Evaluation

To evaluate the performance of the models several metrics are used. In [7] mention that the most relevant metrics for imbalanced dataset is F-measure, recall and precision. F-measure is the harmony between precision and recall. Based on the definition, the focus in this project, is to get the highest F-measure possible. Therefore, those metrics are calculated in this project. Accuracy is also one of the metric used to evaluate the performance of the model.The other metric that was used, is human experiments, where a set of 10 questions were prepared and 10 people were asked to answer those questions and the results of these experiments were aggregated and final result for the following experiment was concluded.

III. Results and discussion

To evaluate the results, several comparisons are made. Such as different models, accuracy, and human experiments.

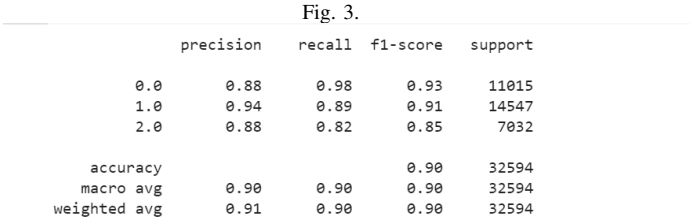
Models	F1	Precision	Recall
Linear SVC	89.66	90	89.66
XGB	82	83.33	81.66
Random Forest	55.66	74.33	59
Average	77.1	82.56	76.78

TABLE I

From the table I we can see that Linear SVC performed the best when compared to all other models based on the F1 score, the second best model was the extreme Gradient Boosting(XGB) and the last was Random Forest, the main reason for such poor performance of Random Forest model was because its hyperparameters were not properly tuned and the best hyperparameter for max depth came out to be 100 which made the model too large to even consider using even when its accuracy came out to be the best.

1) LinearSVC: From the figure 3 we can see that the accuracy of Linear SVC is 90 percent which is quite good , it has highest accuracy as well highest F1 score. Just by looking at the stats we can say that Linear SVC is the best model out of the three trained models if we are comparing the

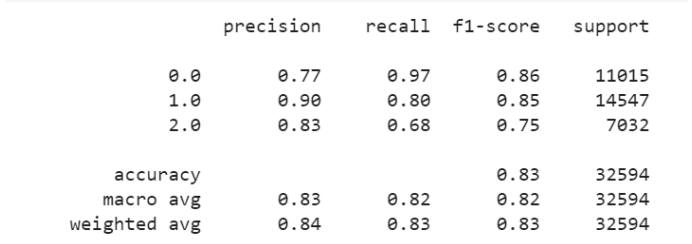
Fig. 3.



	precision	recall	f1-score	support
0.0	0.88	0.98	0.93	11015
1.0	0.94	0.89	0.91	14547
2.0	0.88	0.82	0.85	7032
accuracy			0.90	32594
macro avg	0.90	0.90	0.90	32594
weighted avg	0.91	0.90	0.90	32594

models purely based on this stats from the following figures 3,4,5.

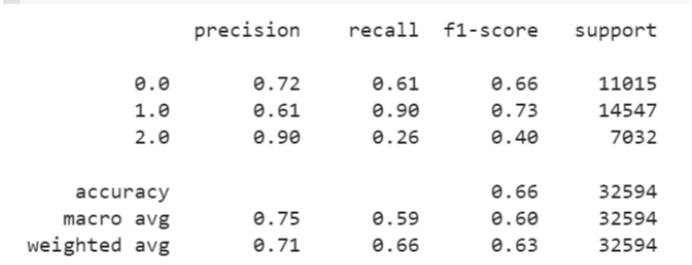
Fig. 4.



	precision	recall	f1-score	support
0.0	0.77	0.97	0.86	11015
1.0	0.90	0.80	0.85	14547
2.0	0.83	0.68	0.75	7032
accuracy			0.83	32594
macro avg	0.83	0.82	0.82	32594
weighted avg	0.84	0.83	0.83	32594

2) Extreme Gradient Boosting: From the figure 4 we can see that the accuracy of the extreme Gradient Boosting (XGB) trained on this dataset is 83 percent and its average F1 score is 82, where as the the accuracy for Random Forest is 66 percent and its average F1 score is 59.66 from figure 5. The accuracy and the F1 score of the Random Forest is the lowest between the three trained models, reason being hyperparameters not being trained currently.

Fig. 5.



	precision	recall	f1-score	support
0.0	0.72	0.61	0.66	11015
1.0	0.61	0.90	0.73	14547
2.0	0.90	0.26	0.40	7032
accuracy			0.66	32594
macro avg	0.75	0.59	0.60	32594
weighted avg	0.71	0.66	0.63	32594

3) Human Trials: Human trials were performed where 10 people were asked to answer 10 questions, a sample questionary was prepared consisting of 10 questions, 10 people were asked to answer those questions and the results were calculated by taking average of all the correct prediction for the given model. As we can see from the table II Linear SVC algorithm had the best results out of the three models for the following experiment and extreme Gradient Boosting (XGB) came second and the Random Forest performed worst which was expected.The results that we got by this human experiment was expected as we can back them by the stats we have i.e. Linear SVC had the highest accuracy as well as the highest F1 score hence we could say that theoretically Linear SVC model out of three trained models should preform better. Which was proven by the data in the following table II.

Person	SVC	XGB	Random Forest
1	70	70	50
2	70	70	60
3	70	80	50
4	80	80	60
5	80	60	60
6	80	60	50
7	60	50	50
8	60	60	60
9	70	70	70
10	80	80	70
Average	72	68	58

TABLE II

IV. Conclusion, Limitations and Future Works

1) Limitations: One of the limitations of this project is that the models were not trained to detect cursed words, hence performance of the models is dropped when curse words are involved in the input text. The other limitation of this project is that there is no interface to add data from csv file directly and get the predicted output, It has to be done manually by making changes in code. Lastly in this project, ensemble method was not implemented, as wanted this to be a comparative study of the machine learning algorithms.

2) Conclusion: Three Machine Learning algorithms Linear SVC, extreme Gradient Boosting(XGB) and Random Forest were successfully implemented for the project, where Linear SVC performed best in all the tests that were considered for evaluation out of the three algorithms. Gradio library was used to create interface for the user to interact with the agent. Hence in this project semantics analysis was performed using trained models on the input provided by the user as well as input from a csv file, with clear models comparison, which clearly addressed the question at hand.

3) Future Works: Firstly if there is anything that could be improved in this project by investing more time that would be perfectly tuning hyperparameters, training more ml algorithms to have a detailed comparison and see which works best for the given dataset. Creating a GUI could get csv file filled with reviews and tweets as input and generate output for it respectively. Another thing we can implement is to have ensemble method prediction approach which will greatly improve the collective accuracy of the output which could maybe beat the deep learning approach [8].

References

[1] ”Cleaning Text Data with Python”, Medium, 2022. [Online]. Available: https://towardsdatascience.com/cleaning-text-data-with-python-b69b47b97b76. [Accessed: 09- May- 2022].

[2] Dey, A., Jenamani, M., Thakkar, J.J. (2017). Lexical TF-IDF: An n-gram Feature Space for Cross-Domain Classification of Sentiment Reviews. In: Shankar, B., Ghosh, K., Mandal, D., Ray, S., Zhang, D., Pal, S. (eds) Pattern Recognition and Machine Intelligence. PReMI 2017. Lecture Notes in Computer Science(), vol 10597. Springer, Cham. https://doi.org/10.1007/978-3-319-69900-448

[3] : Noor, T.H.; Almars, A.; Gad, I.; Atlam, E.-S.; Elmezain, M. Spatial Impressions Monitoring during COVID-19 Pandemic Using Machine Learning Techniques. Computers 2022, 11, 52. https://doi.org/10.3390/computers11040052

[4] J. Lilleberg, Y. Zhu and Y. Zhang, "Support vector machines and Word2vec for text classification with semantic features," 2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC), 2015, pp. 136-140, doi: 10.1109/ICCI-CC.2015.7259377.

[5] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[6] Qingyao Wu, Yunming Ye, Haijun Zhang, Michael K. Ng, Shen-Shyang Ho, ForesTexter: An efficient random forest algorithm for imbalanced text categorization, Knowledge-Based Systems, Volume 67, 2014, Pages 105-116, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2014.06.004.

[7] C. Weng and J. Poon, "A New Evaluation Measure for Imbalanced Datasets", Proceedings of the 7th Australasian Data Mining Conference, vol. 87, pp. 27-32, 2008

[8] Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey. WIREs Data Mining Knowl Discov. 2018;8:e1253. https://doi.org/10.1002/widm.1253