CE807 – Assignment 2 - Final Practical Text Analytics and Report

Student Id: 2211688

Abstract

In this report, we perform hate speech detection on **OLID Dataset**. The job is to identify the tweets if they are harmful, offensive or not, which target some individual or groups in some manner, such as based on their race, religion, gender and other things. Generally, this happens on internet platforms such as Instagram, Facebook and Twitter. Data from the **Legal Statistics Bureau** in the **United States Department of Law** indicates that on average, from 2004 to 2105, there were 250,000 incidents of racial crimes victimizing the US population each year [2].

In this project, we perform text classification on the given dataset. For that, we need to analyze the data, have to understand it first, what it says, shows and how we can use it to get what we want. There are various steps I performed to get the required output such as, preparing and preprocessing the data, splitting the datasets, before putting it to the models, building the machine learning models, evaluating the performances of those models and then later showing the comparison between their results.

1 Materials

Code
Google Drive Folder
Presentation:

2 Model Selection (Task 1):

Two models I have selected for this project

- Random Forest Classifier
- NuSVC

2.1 Summary of 2 selected Models

I have selected two Machine learning models for this task. One is Random Forest Classifier and the second is NuSVC. Both the models are one of the best in terms of Text Classification. Both perform best in categorical and numerical data values of the datasets. They are better when it comes to Text Classification as compared to other machine learning algorithms such as decision trees, XGBoost. Based on their ability and performance, I have selected these models.
Random Forest is the more enhanced version of Decision Tree. It performs more efficiently and quickly evaluates the results. It takes multiple trees to give the output. It takes less time to generate output such as accuracy and F1 Score[4]. It works well on both categorical and numerical data and produces best results with high accuracies as well.
NuSVC is part of SVM(Support Vector Machine). It is one of the 4 variants of Support Vector Machine. It is identical to SVC but it utilizes the parameter to control or manage the support vectors. It uses the 'nu' parameters for regularization which manages the upper bound on the percentage of training mistakes and the lower bound on the percentage of support vectors. It handles the imbalance datasets very well as compared to other SVM variants. In addition, it takes less computation time and memory usage than others[1].
Overall, both are quicker, give faster results, especially when it comes to text classification. Great choice for high imbalance datasets. Based on the requirements or the type of datasets we can select these models for our ease.

2.2 Critical discussion and justification of model selection

Random Forest and NuSVC are both suited for handling large datasets, especially when it comes to handling complex texts. Random Forest is well known for its speed and accuracy. It creates a forest of decision trees in parallel, that can be visualized and understood easily. Whereas NuSVC is well suited for high dimensional or imbalance datasets. As it uses nu parameters for regularization[3].
Random Forest is a prominent ensemble machine learning algorithm, and can be used for a broad diversity of classification such as hate speech detection. It amalgamates miscellaneous decision trees,

where each tree is trained on an arbitrary subset of the data and features and the final prognostication is made by amalgamating all the outputs of all the trees. One of the key advantages is its artistry in handling data with complex features interactions. It balances the class distribution by modifying the weights of the samples. It also grants a measure of feature importance, which can be convenient in diagnosing the most suitable features for hate speech detection. Random Forest is a development of the CART(Classification and Regression Tree) method by implementing bootstrap aggregating(bagging) and random feature selection methods[2]. Easily parallelized, permitting quicker training times and scalability to large datasets.
NuSVC, as I have mentioned earlier, is one of the variants of SVM. It has "nu" parameters that control the swapping between the support vectors and margin width. Several benefits that makes it worthy for hate speech detection:

- Treats high dimensional data.
- Manages non-linear decision boundaries, helpful for spotting intricate patterns in text.
- In comparison with other SVM variants, it is less liable to overfitting, which is important for hate speech identification since it requires encountering extraordinary and sometimes baffling words.
- In order to enhance performance, we can utilize hyperparameters tuning techniques such as grid search for randomized search to determine the ideal combination of hyperparameters.
- Permits more flexible selection of support vectors that helps in increasing the generalization performance of the classifier.
- In order to prevent overfitting, Nu parameters regulate the number and complexity of support vectors in the model.
- For text classification, input data is represented as a bag of words. NuSVC is renowned for its expertise in handling high dimensional data.

In conclusion, nuSVC is better suited for managing imbalanced data and high-dimensional data like text classification jobs, whereas Random Forest is generally faster and more interpretable, making it a solid choice for specific applications. The exact requirements and characteristics of the current challenge will determine which option is best.

Datasets	Total	NOT	OFF
Train	12313	0.66	0.33
Valid	927	0.66	0.33
Test	860	0.72	0.27

Table 1: data set details

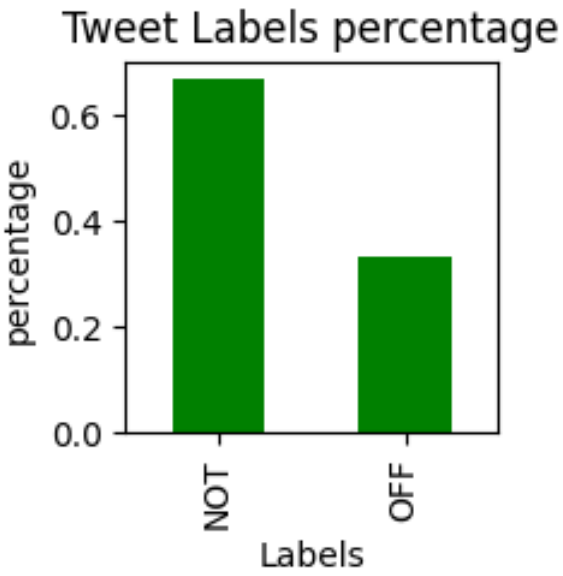

Figure 1: Tweets Labels % of Train dataset.

3 Design and implementation of Classifiers (Task 2)

Train and valid dataset have almost similar numbers of values. Test dataset has fewer values. First we need to divide the train dataset into 4 parts in such a way that first 25%, then 50%, then 75% and 100%. The splitted dataset should be merge inside of each other, that means:

25_1, 25_2, 25_3, 25_4
25% = 25_1
50% = 25_1 + 25_2
75% = 25_1 + 25_2 + 25_3
100% = 25_1 + 25_2 + 25_3 + 25_4

Before putting the models into work, I need to do some text cleaning tasks or preprocessing steps(such as removing unnecessary things from the text like special characters, punctuation, emojis, convert text into small letters) for using the datasets into the models. Also, we need to do vectorization of the tweets before using the models. I created functions for every task and then called those functions wherever needed.
I used Random Forest Classifier as my Model 1. I used my first model without any parameters and it is still giving very good accuracy, F1 Score and confusion metrics. Secondly, I used the NuSVC clas-

sifier [table(4)]. It is one of the 4 variants of SVM, best for text classification jobs. I used it with **hyperparameters('nu', 'kernel')** with some values. It has performed well, with or without hyperparameters it performs almost same results. **Note: With hyperparameters, it takes more time in execution**.
Both models performed well in all the splitted datasets [table(5)] and [table(6)]. Almost similar accuracy and F1 Score they have generated. **Note: every time we run the code, we might see minor changes in the results**.

Models	F1 score
Model1	0.71
Model2	0.70

Table 2: Model's Performance

4 Data Size Effect (Task 3)

Forest provides more better score in test dataset than validation. Whereas for NuSVC is doing vice versa.

Data split	Model 1	Model 2
25%	0.70	0.68
50%	0.72	0.70
75%	0.71	0.69
100%	0.71	0.70

Table 3: Test F1 Score of Models

Yes, I have tried some changes in hyper parameters in the NuSVC classifier. It shows the difference plus it **takes more time if we increase the value of 'nu' parameter** and for its 'kernel' parameters I kept the same values, by default it uses 'rbf'. For the random forest, I did not use any hyper tuning.
Random forest provides almost similar score in both validation [Figure(2)] and test [Figure(3)] dataset. Whereas for NuSVC is doing vice versa [Figure(4)] and [Figure(5)] . We can have a better understanding with graphical representation.
Random Forest produced more F1 Score in validation dataset than test dataset [Figure(6)]. Whereas the NuSVC produced similar F1 Scores in both test and validation dataset [Figure(7)].

5 Summary (Task 4)

5.1 Discussion of work carried out

The whole process or steps of hate speech detection was a great learning experience. I performed various steps to create different outcomes. The main part or important step for hate speech detection is the preprocessing of the datasets before putting it into the models. Then accordingly, we have to create functions for the specific tasks and call them accordingly wherever necessary.

5.2 Lessons Learned

Both models performed well in all splitted datasets. For a random forest, it takes very less time to process the model and generate the results. It gives similar accuracy and F1 score in every execution attempt. For NuSVC, it takes very less time in execution when we run it without hyperparameters. But if we **run it with some hyperparameters, it takes too much time for 75% and 100% datasets**. Then I adjust the code for selecting the best parameters, after that it runs little bit faster and it does not reduce the accuracy.

6 Conclusion

In the end, what I observed is that if we split the dataset properly, models will perform accurately. Both models performed well in all the splitted datasets. They have produced both similarities and differences in their results. The role of hyperparameters in the second model was very impressive. It really makes the difference of using it in the model. The main thing was that these models can perform well both categorical and numerical data. So, if we select a proper model based on the datasets everything will work fine and accurately.

References

[1] A. Aggarwal, T. Sahay, A. Bansal, and M. Chandra. Grid search analysis of nu-svc for text-dependent speaker-identification. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–5, 2015.

[2] K. Nugroho, E. Noersasongko, Purwanto, Muljono, A. Z. Fanani, Affandy, and R. S. Basuki. Improving random forest method to detect hatespeech and offensive word. In *2019 International Conference on Information and Communications Technology (ICOIACT)*, pages 514–518, 2019.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] J. Singh and P. Tripathi. Sentiment analysis of twitter data by making use of svm, random forest and decision tree algorithm. In *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, pages 193–198, 2021.

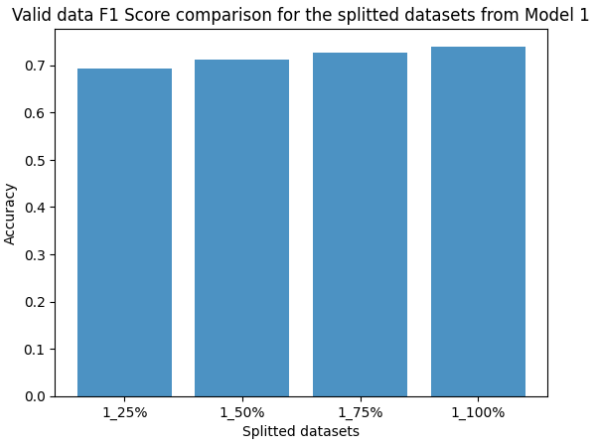

Figure 2: Valid F1 Score of Model 1.

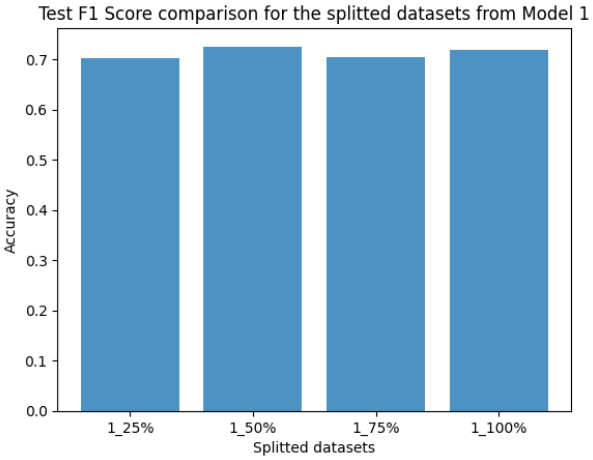

Figure 3: Test F1 Score of Model 1.

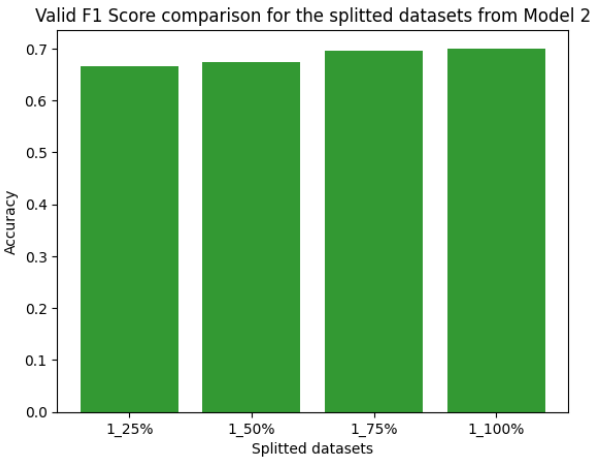

Figure 4: Valid F1 Score of Model 2.

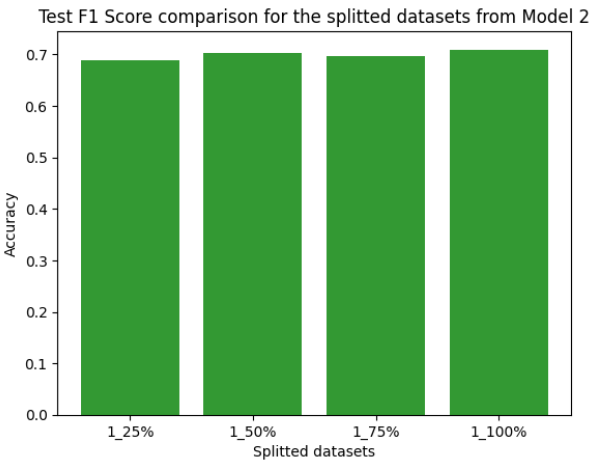

Figure 5: Test F1 Score of Model 2.

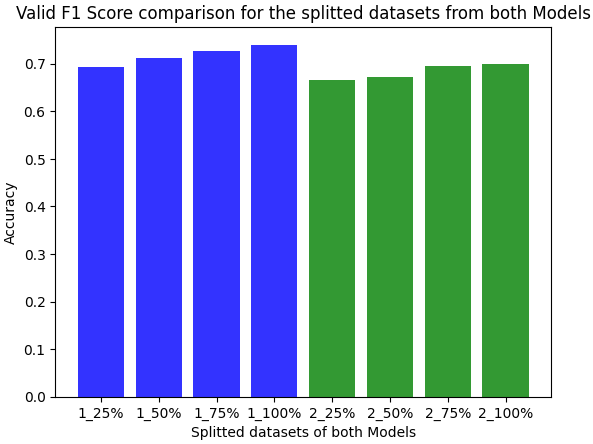

Figure 6: Valid F1 Score of Models.

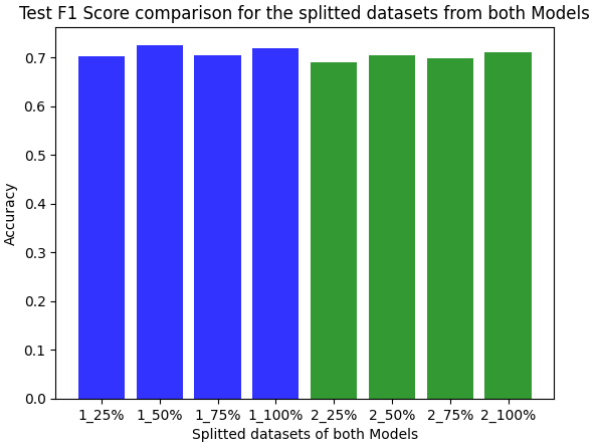

Figure 7: Test F1 Score of Models.

Tweet ID %	**GT**	**M1(100%)**	**M2(100%)**
60133	OFF	NOT	NOT
96874	NOT	OFF	NOT
95457	NOT	OFF	NOT
30899	OFF	NOT	OFF
96457	OFF	OFF	NOT

Table 4: Comparing two Model's output using 100%.

Tweet ID %	**GT**	**M1(25%)**	**M1(50%)**	**M1(75%)**	**M1(100%)**
68123	NOT	NOT	NOT	NOT	NOT
24930	NOT	OFF	OFF	OFF	NOT
83155	NOT	NOT	NOT	NOT	OFF
99680	NOT	NOT	OFF	NOT	NOT
72405	OFF	NOT	NOT	OFF	OFF

Table 5: Model output using Model 1 with different Data Size

Tweet ID %	**GT**	**M2(25%)**	**M2(50%)**	**M2(75%)**	**M2(100%)**
45712	NOT	NOT	NOT	NOT	NOT
83155	NOT	NOT	OFF	OFF	OFF
44546	OFF	OFF	OFF	OFF	NOT
33452	NOT	NOT	NOT	NOT	NOT
96661	NOT	OFF	NOT	NOT	NOT

Table 6: Model output using Model 2 with different Data Size