yashvdas@bu.edu Yashvardhan Das U63030800

# CS - 506 MIDTERM REPORT

## KAGGLE DISPLAY NAME - DasYashvardhan

---

## TOOLS & TECHNIQUES USED

I. The primary independent variable that has been used is the ***TEXT*** column present in the dataset. The strings present in this column have been preprocessed to remove noise. The preprocessing included the **removal of punctuations**, converting of all letters into to **lower case**, and the **removal of stop-words**.
II. The primary machine learning based algorithm that has been programmed into the data includes the **Linear SVC** (Support Vector Machine) in conjunction with the **TF-IDF Vectorizer**. Both these are included in the sklearn library.
III. Upon further iterations, ensemble models were fitted to increase the overall efficiency/accuracy. The other algorithms programmed as part of the ensemble model, apart from Linear SVC and TF-IDF Vectorizer, include the **Multinomial Naive Bayes Classifier** and the **Random Forest Classifier**.

## VALIDATION & IMPROVEMENT OF MODELS

I. The goal behind choosing the Linear SVC is that it is a component of Support Vector Machines (SVM), which, in turn is one of the most efficient classification algorithms owing to its features of hyperplanes and kernel tricks for linear and non-linear domains.
II. The crucial point in optimising the models involves tuning the hyper-parameters of the classifiers. I took the advantage of utilising the **GridSearchCV** and **RandomizedSearchCV** functionalities of the sklearn library.

III. Not every hyper-parameter of each classifier was iteratively tested though. The documentation of sklearn provided much needed insights regarding the selection of the relevant hyper-parameters.
IV. Based on the results obtained after processing through GridSearchCV and RandomizedSearchCV, these values were subsequently treated as the most optimised numerical quantities for improving the model efficiency.

## CREATIVE THOUGHT PROCESS & GOOD CODING DECISIONS USED

I. My initial intuition was to remove as much noise from the data as possible. Hence, preprocessing the text was a carefully-looked after process.
II. Since, the dataset was quite huge, it was necessary to use a partition of the dataset for the purpose of selecting the best hyper-parameter values with GridSearchCV and RandomizedSearchCV.
III. As far as good coding practices are concerned, I used the functionality of **Pipelining** in programming the algorithms. This assisted in avoiding different individualised stops for processing the TF-IDF Vectorizer and the respective Machine Learning algorithms (Linear SVC, Multinomial Naive Bayes and Random Forest Classifier).
IV. To avoid probable loss or overwriting of the data-frames after the preprocessing of text values, I utilised the programming concept of **DeepCopy** available in python. On creating a DeepCopy, the changes reflected in the copy of the data frame do not reflect in the original one. As a lot of subsetting and processing was required with respect to text analysis, it was quite useful to keep a backup of the crucial training and sliced data-frame so that one does not have to restart from scratch owing to the programming of unintentional erroneous commands.