**REPORT**

Name: K. Sreesh Reddy

Phone no.: +91-9731078279

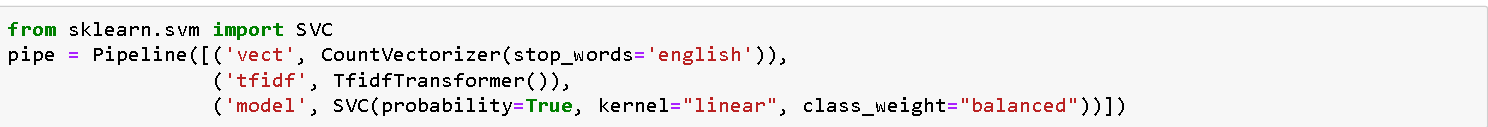
Email-ID: reddy.sreesh224@gmail.com

# SENTIMENT ANALYSIS PROBLEM STATEMENT

The objective of the task is to analyze Twitter sentiment. People use twitter every day and their tweets can have either of the three sentiments – positive, neutral, negative. A training dataset is provided containing two columns, ‘text’ and ‘sentiment’. Build a model using the training dataset and use it to predict the sentiment (Positive, Neutral, Negative) of the tweets in the validation dataset.

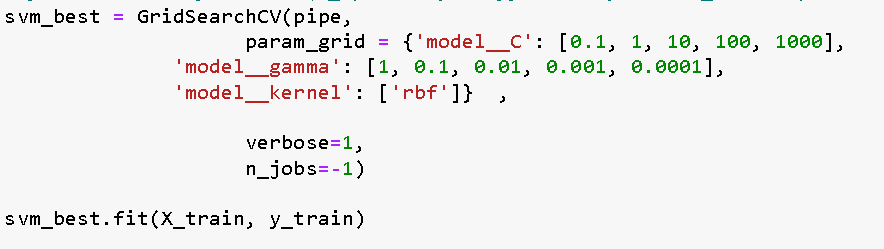
# DETAILS OF MODEL

Support vector machine (SVM) is a learning technique that performs well on sentiment classification. Which features do we have to use in order to classify texts using SVM? The most common answer is word frequencies, which is similar to what is done in Naïve Bayes. This means that we treat a text as a bag of words, and for every word that appears in that bag we have a feature. The value of that feature will be how frequent that word is in the text. The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. There is a balance between SVC maximizing the margin of the hyperplane and minimizing the misclassification. In SVC, the latter is controlled with the hyperparameter CC, the penalty imposed on errors. C is a parameter of the SVC learner and is the penalty for misclassifying a data point. When C is small, the classifier is okay with misclassified data points (high bias but low variance). When C is large, the classifier is heavily penalized for misclassified data and therefore bends over backwards avoid any misclassified data points (low bias but high variance).



HYPERPARAMTERS TUNED:

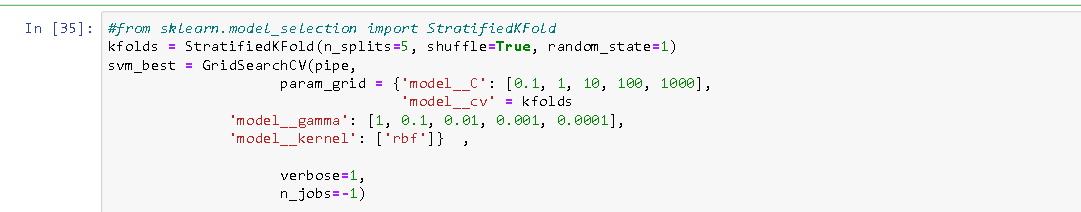
* C: Regularization parameter.
* Gamma: if gamma='scale' (default) is passed then it uses 1 / (n\_features \* X.var()) as value of gamma, if ‘auto’, uses 1 / n\_features.s
* Kernel: Specifies the kernel type to be used in the algorithm. It must be one of ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’ or a callable.



# VALIDATION STRATEGIES AND ACCURACY

# Train/Test split:  The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model’s prediction on this subset.

1. Holdout Set: Creating another set other than the train test data so as to check if the model might have been overfitted.
2. K-fold Split: This is where k-fold cross-validation comes in. It splits the data into k folds, then trains the data on k-1 folds and test on the one-fold that was left out. It does this for all combinations and averages the result on each instance.



Accuracy on the validation dataset: 69.56%

# CHALLENGES

Challenges I faced during model building were:

1. Time consumption: The model sometimes took way too much time to fit the model and to predict the sentiment. This was due to the constraint of the local machines.

2. Overfitting: The model built worked well with the testing data but when a completely new (unseen) data was used for the model to predict, the accuracy was pretty low.

# HOW WOULD YOU LIKE TO IMPROVE MODEL/ACCURACY?

1. Feed more training data: We can collect more tweets and use the model to predict the sentiments. If the model provides sufficient accuracy then the model works fine. If not, the data can be used to train the model more
2. Improved Feature Engineering: For the model to perform optimally, it must have optimal inputs. Therefore, all the **unnecessary** data must be removed and all the **important** data must be present in the right form. For example, the links in a tweet need not be significant enough to determine its sentiment. Hence, it can be removed. And similarly, the most commonly or least commonly occurring words do not contribute, thus they can be excluded.
3. Using a different algorithm: It may be the case that the problem trying to be solved and the way the data is structured does not fit well into the current algorithm. In such case, A different algorithm can be used to see if it learns better from the data.
4. Ensemble Methods: Ensemble method can improve the performance of prediction of more than any single model. And random forest is the technique used many times for ensembling the machine learning model.

# RESEARCH ON STATE-OF-THE-ART SENTIMENT ANALYSIS – A WRITE UP

In the last few years, Twitter becomes the most popular platform for individuals to share their experiences and viewpoints towards different products and services. Therefore, it attracts a lot of researchers to use it as a body for sentiment analysis and opinion mining research studies. Most of the previous research studies in this area have been using the traditional machine learning-based and lexicon-based approaches more compared to the deep learning approach to classify the emotional states of English tweets. Recently, deep learning approach has achieved remarkable results over the traditional machine learning algorithms in analysing a massive amount of data as the case with social networks data. It is seen that Bi-LSTMs perform well across datasets and that both LSTMs and Bi-LSTMs are particularly good at fine-grained sentiment tasks (i.e. with more than two classes). Incorporating sentiment information into word embeddings during training gives good results for datasets that are lexically similar to the training data.

Along with traditional models such as Naïve Bayes, Logistic Regression, Random Forest Classifier, Support Vector Machine, and Gradient Boosted Trees, if CNN (Convoluted Neural Network) and RNN (Recurrent Neural Network) are used, it was noticed that CNN and RNN gave the best recall and F1-score for the analysis.

Below are the models which were predominantly used for the iMDb sentiment analysis:



**Previous Approaches**

The first important step to begin with while handling text data is pre-processing. After the tokenization of a sentence, all word types having an overall occurrence frequency of three or less in the corpus is removed. Because the word types having small occurrence frequency convey more noise than pertinent information. This pruning heuristic also allows to eliminate specific slags that could be hard to classify without any additional lexical vocabulary. To represent a text, bag-of-words assumption is adopted as one of the techniques in which each word is stemmed according to Porter’s method. Such a sequence of words corresponds to one unigram model. POS tagging information is not used in order to reduce the language specific pre-processing of the corpus. The second step is to determine the terms belonging clearly to one category. Z scores calculation was considered for this.

To take into account local proximity, extract the neighbours of each confident feature (two terms before and two terms after). The bigram indexing scheme does not always capture these expressions since a lot of times they may contain modifiers or other words in-between. Sum of the confident features can be checked and classify the sentence accordingly.

To represent textual unit, different strategies are used. First, the unigram model is based on a sequence of stems obtained with the Porter’s stemmer. Second, bigram indexing scheme is used. As a third approach, it is noticed that some of the prepositions combined with the previous term in the sentence can change its meaning, and sometimes even its polarity. Therefore, together with the experiments with unigram and bigram representations, a new indexing scheme called WiseTokenizer is implemented. In this new procedure, the terms in the sentence are indexed separately, except terms that precede the prepositions that could change the meaning of the verb.

In the proposed procedure, the extraction and weighting of confident features are based on Z score model, able to determine term specificity according to two or more categories. Based on the information gain measure, neighbourhood of confident terms are suggested to be considered. The model that was able to achieve comparable results here was classical methods such as SVM and Naïve Bayes.

**ML approach**

In sentiment analysis mostly the supervised learning type is employed to classify the content. It mainly involves 4 main steps- collecting data, preparing data for processing, training and testing data, then finally the classification step. The sentiment classification model is built based on the training data which is then used later with new data inputs. The tweets were pre-processed firstly using tokenization, normalization, and POS tagging. Then, naïve bayes classifier has been used to categorize tweet content as positive or negative by comparing each word with the labelled words in the dictionary.

**Lexicon traditional approach**

The idea behind lexicon-based approach is used to build a dictionary which holds all the positive and negative words and then use it to measure the sentiments of given text based on the appearance of positive and negative words on the input text. The process starts by converting the input text into tokens then scan all tokens, and if the encountered token matches the lexicon in the dictionary, the sentiment score is changed on the type of word. Negation handling has been found to be challenging task in sentiment analysis. Therefore, fuzzy logic is proposed as solution to this.

The main differences between machine learning technique and lexicon-based technique was based on text pre-processing, employed method, dictionary, datasets, and soft-computing approaches.

One of the researches has provided a lexicon-based method to measure and analyse the sentiment of English customer reviews in twitter data. Their method is based on the following rule: the collective polarity of a document or sentence is the polarities sum of the singular words or phrases. They found this approach resulted in a high accuracy sentiment analysis, and they recommended the use of hybrid approaches in future to improve the performance.

**Deep learning approach**

Deep learning has received a lot of attention recently and for a good reason; it achieved classification and analysing results that were not possible before. As deep learning achieved surprising performance on tasks such as language understanding and image analysis, it’s believed it might bring the same level of accuracy and performance in the field of sentiment analysis.

One of the researchers applied their deep learning model at the message using Semeval2015 dataset to analyse twitter morale. They proposed a solution that combines, unsupervised learning of text and learning about poorly supervised data. Their system gave high results across all other test sets.

Another researcher, reported three main findings after performing an experiment. First, that deep learning models are not always better than traditional machine learning methods. Second, the classification accuracies in the real-world setting are much lower than in the experimental setting, which means the current reported performance of twitter sentiment analysis in previous studies may be over-claimed. Third, they reported that support vector machine models have rather good fitness to NLP tasks, especially when the size of the dataset is limited.

In one of the experiments, tweets were labelled into three categories: positive, negative and neutral based on the Euclidian distance and the cosmopolitan similarity. Correlation based feature selection was used as well to determine the best classification features. KNN and similarity-based cosine outperformed all other models.

Another system proposed from a different experiment was a hybrid design of both convolutional neural networks and recurrent neural networks to implement the deep learning model. They performed the following main steps to create the model: pre-processing, word embedding and then feeding input text into a deep neural network architecture. They stated that a hybrid architecture of both convolutional neural networks and recurrent neural networks could achieve an optimal accuracy result.

**Conclusion**

The traditional sentiment analysis approach uses a bag-of-words approach which has failed to take into consideration the order of the words and hence leads to a less accurate model. This approach also often fails to classify negative sentences, i.e., sentences with negation and often give it a positive label. On the other hand, deep learning fixes these issues by generating new feature representations, instead of using term frequency-inverse document frequency vector, the model creates its representations of the word. Given that word2vec identify the semantic context of words in a given dataset as numeric vectors, which facilitates find words that have similar semantic meanings. However, it should be noted that the practicality of the neural networks of deep learning is having both large-scale data for training and super-fast computers for processing.

Deep learning is still in early stages for sentiment analysis research field. Even though in most of the cases, the sentiment analysis results for the machine learning based approach outperform deep learning, it’s still believed that deep learning approach needs more investigation to prove their effectiveness, capability and limitations. Taking in consideration that deep learning automatically finds out the best text features needed for sentiment classification, but this is not the case in machine learning-based approach.