****

**Sarcasm detection using feature selection**

**Submitted to:** Dr. Muhammad Rafi

**Group members:**

Syed Hassaan Saleem (16k-3633)

Osama Abid (16k-3728)

# Introduction:

Sentiment Analysis is one of the most difficult tasks in Natural Language Processing and sometimes it is even difficult for humans to detect it especially sarcasm because the context of a sentence matters the most. Sarcasm is used as a hidden message to target someone critically.

There are different approaches for automatic sarcasm detection, which include:

* Rule-Based
* Statistical
* Machine Learning Algorithms
* Deep Learning

Our project is Sarcasm detection using feature scaling. This project aims to detect sarcastic sentences in a given dataset and was able to achieve an accuracy of 76.92%.

## Data Processing:

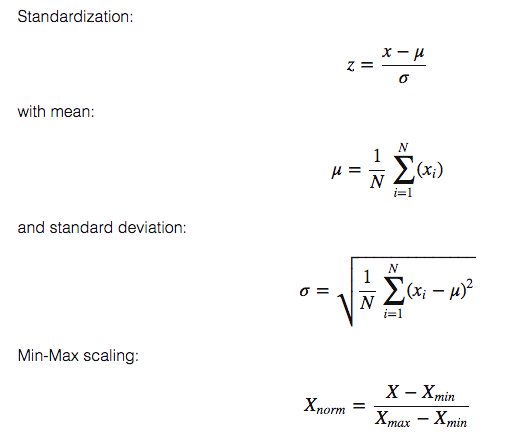
**Data Format: Sentences**

Data Preprocessing involves following steps:

1. Tokenization.
2. Convert the data to lowercase, remove stop-words and special characters.
3. Removing duplicates.
4. Removed words with frequency less than 3 in order to remove outliers.
5. Removed nouns, pronouns and foreign words from data.
6. Removed words with length less than 3 characters.

## Features Selection:

* TF-IDF vectorization is used for feature selection.
* For Feature Scaling: Standard-Scaler of Scikit-learn is used.



# Model:

PCA

Data Processing

Input

TF-idF Vectorization

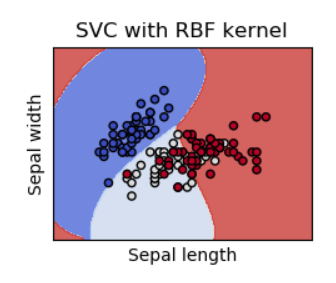
SVM

Split Data Training-Test set

Output

# Project Description:

* Data was tokenized and then the tf-idf value was used as feature because, it converted text-data into numeric form so that machine learning techniques can be applied easily.
* After creating the panda data frame, the data is sent to the standard scalar and Principal component analysis (PCA) is applied on the scaled data. Standard scalar is used to normalize the data and PCA is used to reduce the dimension space which in turn reduces the computational cost of the model and allow us to control variance.
* Data is split into training set and testing set with the ratio of 80:20
* Data is sent to the Support Vector Machine (SVM) with RBK kernel for classification. SVM is a powerful classifier and its accuracy and efficiency is better than other classifiers.



* The results are plotted and accuracy of the model is measured.

# Result:

We applied different techniques. First, we applied SVM without features scaling keeping training and testing set ratio to 70:30 and get an accuracy of 58% and then we tried with Min-Max scaler which increased accuracy by 2% and then we Tried Standard-Scaler which increased the accuracy by 5%. We then passed data through PCA with explained variance ratio 92% the accuracy was increased by 5% making it a total of 68% thus reducing the computation cost. Then we applied PCA with explained variance ratio 20% the accuracy of the model was increased to 71%. Then we finally improved the pre-data processing part and apply PCA with explained variance ratio 15% and split data into training and testing set by a ratio of 80:20. The accuracy of the applied model was **76.92%**. Further changes did not show any noticeable improvements.

## Acknowledgements

This dataset was collected from TheOnoin and HuffPost. <https://rishabhmisra.github.io/publications/>