**Sarcasm Detection with GloVe/Word2Vec**

**Introduction**

Sarcasm is a form of language that involves saying one thing but meaning the opposite, often in a humorous or ironic way. Recent advancements in natural language processing (NLP) have made possible to automatically detect sarcasm in text using machine learning models. GloVe and Word2Vec are two popular algorithms used for training word embeddings in NLP. Word embeddings are numerical representations of words that capture their meaning and semantic relationships with other words in a language. These embeddings can be used to train machine learning models for various NLP tasks, including sarcasm detection.

Sarcasm detection with GloVe or Word2Vec typically involves training a supervised machine learning model on a large dataset of labelled examples of sarcastic and non-sarcastic text. The word embeddings are used to represent the words in the text, and these representations are fed into the model along with other features, such as punctuation, capitalization, and sentence structure. The model then learns to classify text as sarcastic or non-sarcastic based on these features. Overall, sarcasm detection with GloVe or Word2Vec is an exciting area of research in NLP, with many practical applications in social media analysis, customer service, and more.

# OVERVIEW OF THE DATASET

**Past studies in sarcasm detection mostly make use of Twitter datasets collected using hashtag-based supervision, but such datasets are noisy in terms of labels and language. Furthermore, many tweets are replies to other tweets, and detecting sarcasm in these requires the availability of contextual tweets.**

**To overcome the limitations related to noise in Twitter datasets, this news headline dataset for sarcasm detection is collected from two news websites. The Onion aims at producing sarcastic versions of current events, and we collected all the headlines from the News in Brief and News in Photos categories (which are sarcastic). We collect real (and non-sarcastic) news headlines from HuffPost.**

**The dataset consists of about 28000 text data points, where each data category belongs to two categories: sarcastic or not sarcastic.**

**Implementation**

Step 1: Word embeddings

* Word embeddings are numerical representations of words that capture their meaning and semantic relationships with other words in a language.
* Two popular algorithms for training word embeddings in NLP are GloVe and Word2Vec.
* These embeddings can be used to train machine learning models for various NLP tasks, including sarcasm detection.

Step 2: Pre-trained GloVe embeddings

* **GloVe method is built on an important idea, You can derive semantic relationships between words from the co-occurrence matrix. Given a corpus having V words, the co-occurrence matrix X will be a V x V matrix, where the i th row and j th column of X, X\_ij denotes how many times word i has co-occurred with word j.**

Step 3: Converting text to GloVe embeddings

* To convert a sentence into a matrix of GloVe embeddings, we can define a function that takes a sentence, the embeddings index, and some optional parameters such as the maximum number of words to consider and the embedding size.
* This function splits the sentence into a list of words, initializes a matrix of zeros, and fills it with the corresponding GloVe embeddings for each word in the sentence.
* We can then use this matrix as input to a machine learning model for sarcasm detection.

Step 4: Training a machine learning model

* To train a machine learning model for sarcasm detection, we need a dataset of labeled examples of sarcastic and non-sarcastic text.
* We can use the GloVe embeddings matrix along with other features, such as punctuation, capitalization, and sentence structure, as input to the model.
* The model learns to classify text as sarcastic or non-sarcastic based on these features.
* We can evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score.

**Analysis**

****

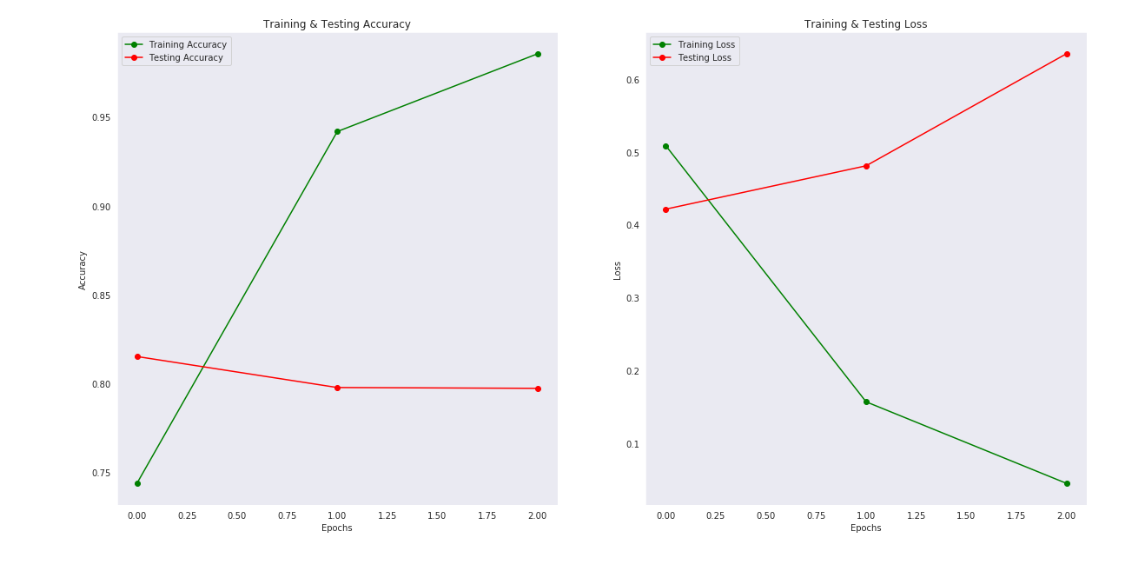


# TRAINING WORD2VEC MODEL

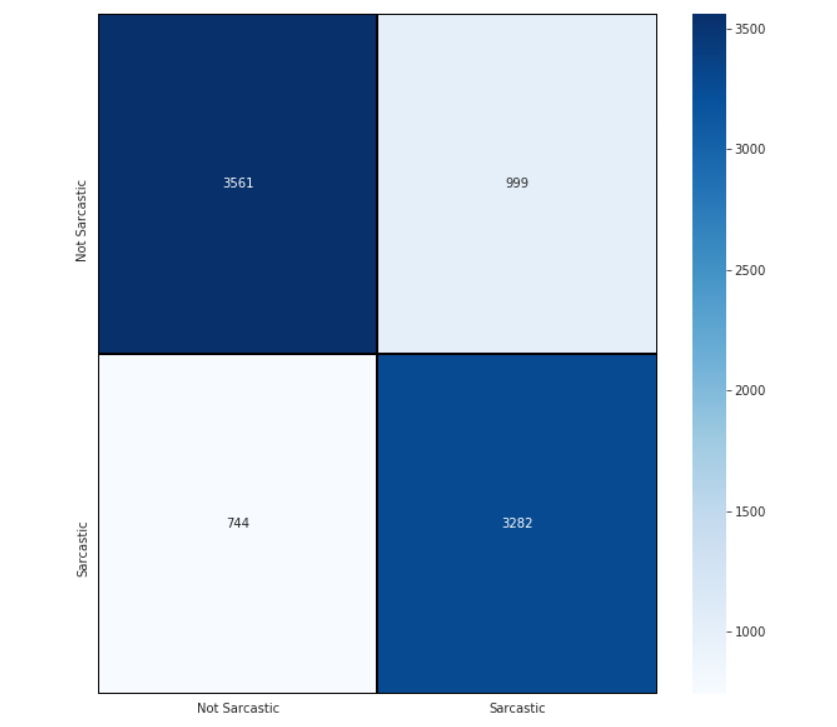


# ANALYSIS AFTER TRAINING OF WORD2VEC MODEL



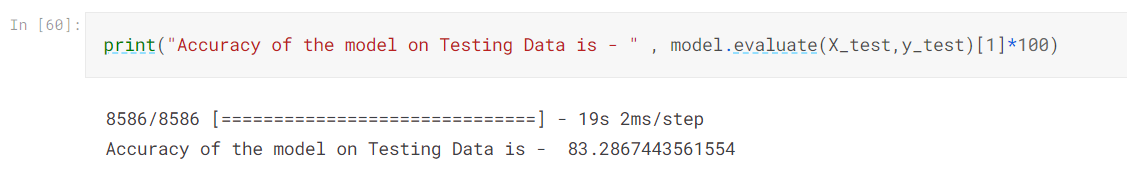


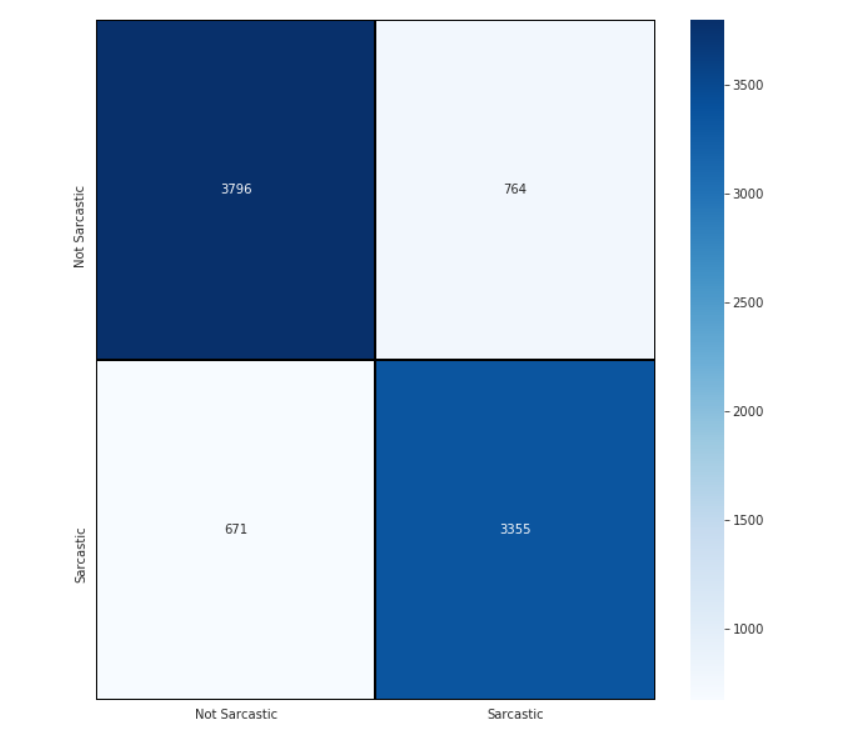
The findings indicate that the model appears to be overfitting and underperforming on the test data. Below confusion matrix observation shows the above model



# TRAINING GLOVE EMBEDDINGS MODEL



THE ACCURACY INCREASED TO 83% FROM 79%



# The confusion matrix observation shows the GLOVE EMBEDDINGS MODEL.