……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Name:** Kishan Kumar

**Email address:** kishankumar09ei@gmail.com

**Contact number:** 9066712922

**Anydesk address: NA**

**Years of Work Experience: 7**

**Date: 09-03-2022**

……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Self Case Study -2: Sarcastic comment detection in Reddit comments**

“After you have completed the document, please submit it in the classroom in the pdf format.”

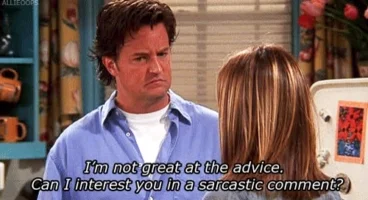
Please check this video before you get started: <https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg>

# Overview

\*\*\* Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** \*\*\*

* **Introduction**:

Sarcasm is "a sharp, bitter, or cutting expression or remark; a bitter gibe or taunt". Sarcasm may employ ambivalence, although sarcasm is not necessarily ironic. Most noticeable in spoken word, sarcasm is mainly distinguished by the inflection with which it is spoken and is largely context-dependent. <https://en.wikipedia.org/wiki/Sarcasm>.



In sarcasm, ridicule or mockery is used harshly, often crudely and contemptuously, for destructive purposes. It may be used in an indirect manner, and have the form of irony, as in "What a fine musician you turned out to be!" "It's like you're a whole different person now...," and "Oh... Well then thanks for all the first aid over the years!" or it may be used in the form of a direct statement, "You couldn't play one piece correctly if you had two assistants." The distinctive quality of sarcasm is present in the spoken word and manifested chiefly by vocal inflection.

* **Problem Statement:**

Sarcasm is the part of daily life and conversations now. For every business entity there is a social media platform where the people share their experience with the product and services a company offered. This is actually feedback which is actually helps the business to expand and retrieve more customers. This is equally important to the customers as well, because it’s a general trend now ...before using a product or service, people tend to search for the reviews and comment by other people. Sometimes these comment/reviews are SARCASTIC in some ways. This mean that the sentiment of underlying meaning of the sentence is just opposite to what they pretend. But getting sarcasm is little non trivial. It’s like you crack a tricky joke and some people laughed instantly (they know sarcasm...), some laughed after 2-3 minutes (it seems they are using a single core CPU rather than GPU in their mind. yeah, this is SARCASM !!) and some don’t even understand (couldn’t even processed this data... SARCASM again !!!).

Sarcasm detection is an important component in many natural language processing (NLP) systems, directly relevant to natural language understanding, dialogue systems, and text mining. However, detecting sarcasm is difficult because it occurs infrequently and is difficult for even humans to discern. and platforms want to build a automated system that can understand the sentence being Sarcastic. So they can process the information and can take further action towards improvement.

* **ML Problem Formulation**
* **Objective:**  
   The objective of this project is to use labeled data for a comments for being sarcastic or not and create a ML based mechanism to predict the future sentence’s orientations. This is a standard supervised classification model

**Supervised**: The labels are included in the training data and the goal is to train a model to learn to predict the labels from the features.  
**Classification**: Here we are trying to create a Binary classifier which can predict with given features (comment and other features) and predict whether acomment is

* Comment is not sarcastic: **1**
* Comment is sarcastic: **0**
* **Real world constraints**
* No strict low-latency requirement. It may take some seconds to predict a comment’s being sarcastic of not.
* Probability of prediction is useful.
* We need to be susceptible about
* False positives are instances where a comment is incorrectly labeled as sarcastic.
* False negatives are instances where a comment is sarcastic but labeled as not sarcastic.
* Interpretability is somewhat important.
* **Data description and column analysis**

This dataset contains 1.3 million Sarcastic comments from the Internet commentary website Reddit. The dataset was generated by scraping comments from Reddit containing the \s ( sarcasm) tag. This tag is often used by Redditors to indicate that their comment is in jest and not meant to be taken seriously, and is generally a reliable indicator of sarcastic comment content.

The data was gathered by: **Mikhail Khodak ,Nikunj Saunshi and Kiran Vodrahalli for their research paper article “**[A Large Self-Annotated Corpus for Sarcasm](https://arxiv.org/pdf/1704.05579.pdf)**”**.

**Data-Source**: <https://www.kaggle.com/sherinclaudia/sarcastic-comments-on-reddit>.

There are actually 10 columns in our dataset.

|  |  |
| --- | --- |
| Column name | Description |
| Label | Sarcastic (1) or not (0) |
| comment | Reply to Parent Reddit comment |
| author | Person who commented |
| subreddit | Commented under which subreddit |
| score | Number of upvotes – No. of downvotes |
| ups | No of upvotes |
| downs | No of downvotes |
| Date | Commented date |
| created**\_**utc | Commented time in the UTC Time zone |
| parent\_comment | The Parent Reddit comment to which sarcastic replies are made |

* **Performance metric**

Performance on this task can be measured by various metrics.

1. <https://recentscientific.com/sites/default/files/10765-A-2018.pdf> research paper used **AUC\_Score**
2. Original research paper of Mikhail and team used Normal **classification score**

I think **AUC\_Score** will be better option because it considers FP and FN both because I suppose False Positive and False Negative is equally important to us. It’s a matter to further discussion.

# Research-Papers/Solutions/Architectures/Kernels

\*\*\* Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. \*\*\*

1. Mikhail Khodak and Nikunj Saunshi and Kiran Vodrahalli for their article "A Large Self-Annotated Corpus for Sarcasm". .

<https://arxiv.org/abs/1704.05579>

1. DEEP AND DENSE SARCASM DETECTION <https://arxiv.org/pdf/1911.07474v2.pdf>
2. <https://towardsdatascience.com/sarcasm-detection-step-towards-sentiment-analysis-84cb013bb6db>
3. <https://towardsdatascience.com/sarcasm-detection-with-nlp-cbff1723f69a>
4. <https://journals.sagepub.com/doi/full/10.1177/1470785320921779>
5. <https://www.kaggle.com/mpwolke/sarcastic-comments>
6. <https://paperswithcode.com/task/sarcasm-detection>
7. <https://ieeexplore.ieee.org/abstract/document/9420094>
8. <https://recentscientific.com/sites/default/files/10765-A-2018.pdf>

# 

# First Cut Approach

\*\*\* Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** \*\*\*

My Approach will be

1. EDA of the data.
2. Data cleaning
3. Data preprocessing(using NLTK and regular expressions)
4. Handling missing values
5. Using different Word2Vec model
   1. Word2vec
   2. AvgWord2vec
   3. TFIDF vectorizers
   4. Glove vectors
   5. Creating BERT based vectors
6. Applying Classical Machine learning approaches for modeling
7. Trying some deep learning models.
8. Evaluating the results and select the best model based on the cross-validation performance metrices.

Notes when you build your final notebook:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument “X” (a single data points i.e 1\*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
   1. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
   2. so in your final notebook, you need to pass only those two values
   3. def final(X):

preprocess data i.e data cleaning, filling missing values etc

compute features based on this X

use pre trained model

return predicted outputs

final([time, location])

* 1. in the instructions, we have mentioned two functions one with original values and one without it
  2. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
  3. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y\_predict)

1. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
2. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
3. Check this live session: <https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models>