

# **Reddit Comment Sarcasm Detection**

## Table of Contents

Reddit Comment Sarcasm Detection.....	1
Project Overview.....	1
Goals and Objectives.....	1
Scope of the Project.....	1
Proposed Methodology:.....	2
Results and Conclusion:.....	3

## **Project Overview**

Reddit Comments Sarcasm Detection is a challenging problem, which lies in the domain of Natural Language Understanding. The project involves using the publicly available “Sarcasm on Reddit” dataset to build a model that is able to classify a sarcastic comment on social media platform Reddit. This problem is unique in its nature as it involves differentiating between sentiment and sarcasm based on not just the comments but other features like subreddit(topic), parent comment and score against each comment.

To solve this problem a Machine Learning pipeline is built and the data is passed through it. It includes the following modules:

1. Data Analysis
2. Data Preprocessing
3. Feature Engineering
4. Model Training

The end product will be a trained model which would be able to generalize what it has learned from training data and distinguish between sarcastic and non-sarcastic comments on Reddit with good accuracy.

## **Goals and Objectives**

The main goals of this project are:

- To build a model that would differentiate between sarcastic vs. non-sarcastic comments on Reddit.
- Achieve high recall score.
- Conduct an analysis and provide a brief comparison of different approaches used.

## **Scope of the Project**

Build a sarcasm detection model that would detect sarcasm in commentary on Reddit utilizing several features that are available with each comment.

The following steps will be taken to accomplish this task:

- Build and test the aforementioned ML pipeline using a subset of the actual 1.3 million Reddit Comments.
- Train models on training set taken from entire dataset and establish baseline accuracy.
- Obtain good performance scores in terms of precision, recall and accuracy on validation/test data.
- Comparing results using performance metrics like precision, recall and accuracy of different models trained with different feature sets.

## Proposed Methodology

The structure of Machine Learning pipeline mentioned above will remain the same however different approaches will be taken for feature engineering and model training.

### 1. Data Analysis:

Data analysis is done to see the distribution of sarcastic vs. non-sarcastic comments. It also involves looking at some statistics about the data and checking out some information about the data like data types, missing values, etc. It includes other visualization like log-frequency distribution of numeric features as well.

### 2. Data Preprocessing:

It's a very important step which includes dealing with missing values and applying basic text preprocessing techniques like removing special characters, replacing multiple spaces with a single space, converting strings to lowercase, tokenizing and stemming text.

### 3. Feature Engineering:

There are two feature engineering approaches used in this project.

- **TF-IDF Vectorization:** Text data i.e. Comment, subreddit and parent\_comment features were represented as TF-IDF(term frequency – inverse document frequency) vectors. These features were then stacked with normalized numeric features (score, ups and downs) and passed onto the model.
- **Word Embedding Features using Word2Vec Model:** This involves generating feature vector of each document by averaging out word embedding vectors of all terms in a document generated by Word2Vec model. Document here refers to text data features i.e. comment, subreddit and parent\_comment. Like TF-IDF vectors, these features were also stacked with numeric features and passed onto the model.

### 4. Model Training:

It's a binary classification problem so simple but effective binary classification models were chosen. 4 models were trained in total:

- Logistic Regression with TF-IDF Vectorization
- Linear SVM with TF-IDF Vectorization
- Logistic Regression with Word Embedding Features
- Linear SVM with Word Embedding Features

## Results and Conclusion

Model	Feature Set	Data set	Precision	Recall	Accuracy
Logistic Regression	TF-IDF Vectorization	Training set	0.89	0.74	0.854
		Test Set	0.63	0.54	0.670
SVM		Training Set	0.97	0.71	0.846
		Test Set	0.94	0.66	0.844
Logistic Regression	Word2Vec Features	Training Set	0.64	0.37	0.661
		Test Set	0.60	0.30	0.590
SVM		Training Set	0.60	0.15	0.592
		Test Set	0.60	0.10	0.590

From these experiments it looks like SVM with TF-IDF Vectorization is working well and has achieved a test accuracy of 84% which is far better than other models. This model is capable of detecting sarcasm in any Reddit Comment with 84% confidence which is fairly good.

# Project Architecture

