

Jaypee Institute of Information Technology, Sector-62, Noida

Project members: Archita Mittal, Paluck Deep, Easha Pandey
Under the Supervision of: Ms. Sakshi Agarwal

Sarcasm Predictor - User Manual

January-May, 2020

Overview

Sarcasm is the use of words that mean the contrary to what an individual really wants to state, particularly as to affront someone, or to show aggravation, or just to be funny. It activates a person's abstract thinking. To understand sarcasm people rely on the speaker's facial expression, tone of voice, writing style and how the speaker might feel about it. It is twice as difficult to identify sarcasm in writing than in verbal communication - particularly as we are living in a digital age. Most of the cues necessary for detection of sarcasm is lost in writing.

This project aims to develop a web application to predict sarcasm from a given text.

Goals

1. **Binary Classification:**

To categorise the tweets into sarcastic and non-sarcastic comments.

2. **Multi-class Classification:**

To categorise the sarcastic tweets into positive, negative, self-deprecating and miscellaneous categories.

Hardware Specifications

- Processor: Intel core i5-7300U CPU @ 2.60GHz 2.71 GHz
- RAM: 8 GB

Software Specifications

- OS: 64-bit
- Tool: Anaconda -> Jupyter Notebook

Install the following python libraries: (some libraries needs to be installed via the command "pip install" in anaconda prompt)

- pandas
- numpy
- nltk
- bs4
- re
- textblob
- pyspellchecker
- import_ipynb
- sklearn
- seaborn
- matplotlib
- pickle
- flask
- nbimporter
- waitress

Instructions

The file “result_web.ipynb” consists of the code for *binary classification*. Open it using Jupyter Notebook. Give the path to the files “sarcasm_tweets.txt” and “nonsarcasm_tweets.txt” in the variables “data” and “data_non”. Now, run the whole file.

The file “Multi.ipynb” consists of the code for *multi-class classification*. Open it using Jupyter Notebook. In the line of variable “Pragmatic_features_Sar”, give the path to the file “a_features_sarcastic.csv”, for the variable “Lexical_features_Sar”, give the path to “b_features_sarcastic.csv”, for the variable “Linguistic_features_Sar”, give the path to “c_features_sarcastic.csv”, and for the variable “data”, give the path to “Multiclass_sarcasm.csv”. And then, run the whole file.

The file named “app.ipynb” contains the code for the *web application*. Open the file and run it. In the last cell, a link will appear for the website. If the link doesn’t appear, then open a new tab and paste “<http://localhost:8080/>”. On pressing enter, the sarcasm predictor form will be displayed. Enter a sentence and click on the “Predict” button. The next page will show the entered sentence and the predictions by Logistic Regression, Naive Bayes, Random Forest and Support Vector Machine.

Snapshots

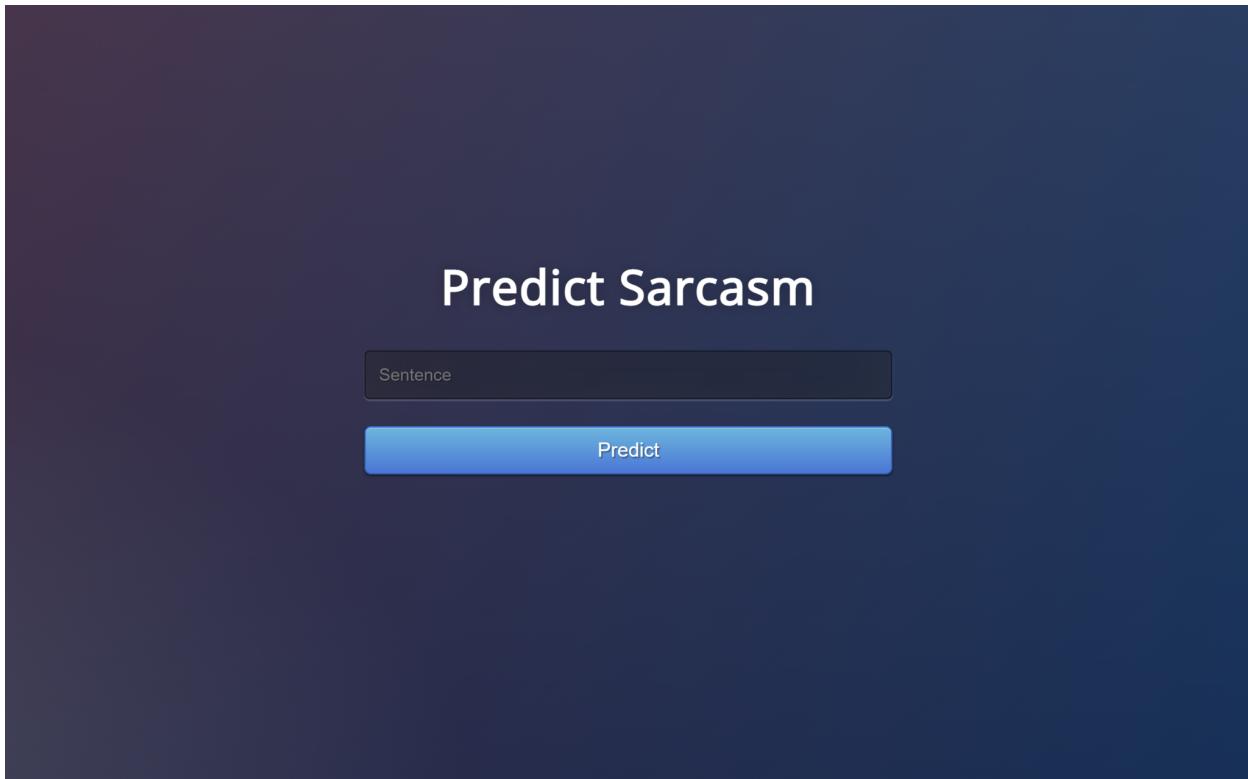


Fig 1: The Main Page of the Web App

A screenshot of the web application showing a prediction result. On the left, the "Predict Sarcasm" interface is shown with the sentence "I am fine 😊😊" entered into the "Sentence" field and the "Predict" button highlighted. On the right, the results are displayed. The text "The given sentence:" is followed by the sentence "I am fine 😊😊". Below this, the text "Results predicted by various classifiers:" is followed by a table. The table has four columns: "Logistic Regression", "Naive Bayes", "Random Forest", and "Support Vector Machine". The rows show the predictions: Logistic Regression and Random Forest predict "Sarcastic", while Naive Bayes predicts "Not sarcastic".

Logistic Regression	Naive Bayes	Random Forest	Support Vector Machine
Sarcastic	Not sarcastic	Sarcastic	Sarcastic

Fig 2: A sarcastic comment correctly predicted by Logistic Regression, Random Forest and Support Vector Machine

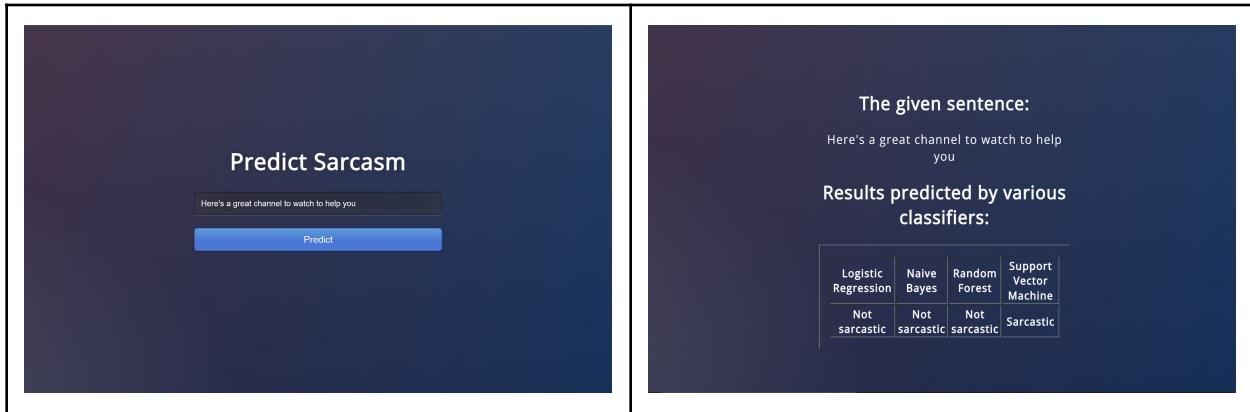


Fig 3: A non sarcastic comment correctly predicted by Logistic Regression, Naive Bayes and Random Forest

Milestones

1. **Binary Classification:**

The highest accuracy of 88.4%, precision of 89%, and recall of 88% was achieved from Random Forest.

2. **Multi-class Classification:**

The highest accuracy of 38.7%, precision of 39%, and recall of 39% was achieved from Logistic Regression.