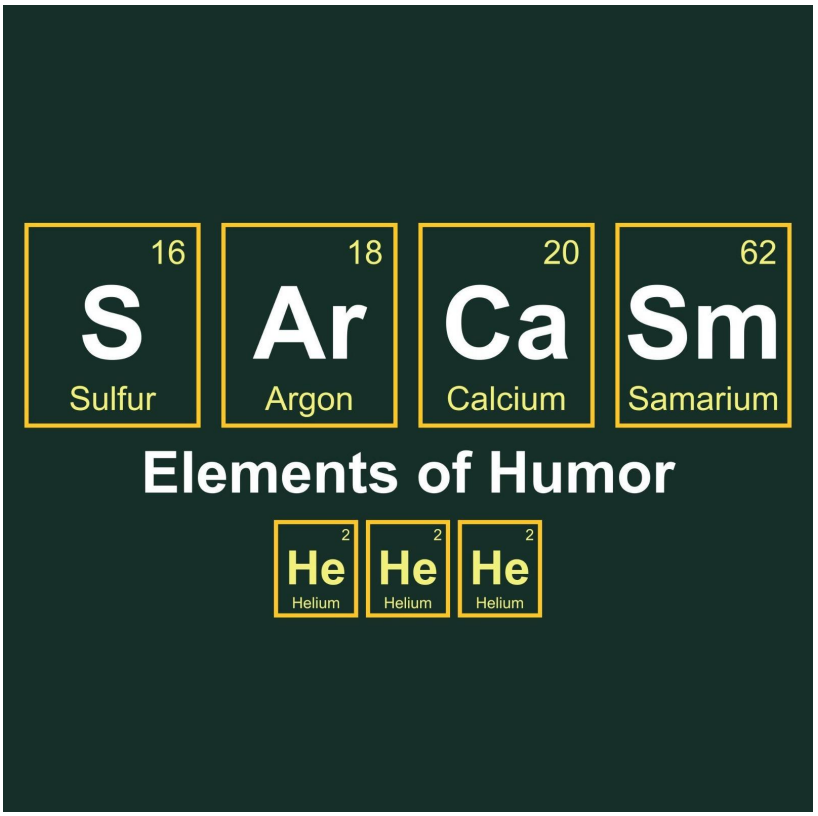
Sarcasm Detection on Reddit



Authors:
Saishree Godbole (skgodbol@iu.edu)
Tanvi Kolhatkar (tckolhat@iu.edu)

Contents

Abstract

This report addresses the approach to analyse the effect of different attributes in determining whether a comment is Sarcastic or Sincere and study the different trends in the Reddit comments. We aim to determine the label of a comment based on the comment, date of the comment, subreddit the comment was posted under etc. Since comments left on social media platforms and under products on various e-commerce websites play an important role in determining business decisions, this approach will help companies better understand their users and help make recommendations more personal.

Keywords

Keyword	Description
Sarcasm	The use of irony to mock or convey contempt
Reddit	Reddit is an American social news aggregation, web content rating, and discussion website. Registered members submit content to the site such as links, text posts, images, and videos, which are then voted up or down by other members.
Kaggle	Kaggle is an online community of data scientists and machine learning practitioners that allows users to find, publish and explore data sets and build models in a web-based data-science environment.
Text Classification	Text classification is a machine learning technique that assigns a set of predefined categories to open-ended text. Text classifiers can be used to organize, structure, and categorize pretty much any kind of text – from feedback comments, documents, medical studies and files, and all over the web.
ColumnTransformer	Applies transformers to individual columns of an array or pandas DataFrame.
TFIDFVectorizer	Term Frequency — Inverse Document Frequency (TFIDF) is a technique for text vectorization that converts a collection of raw documents to a matrix of TF-IDF features.

Introduction

We are using a balanced dataset on Kaggle, (Sarcasm on Reddit) containing ~1.3 million labelled comments . The dataset was generated by scraping comments from Reddit containing the \s (sarcasm) tag.

We aim to analyse the dataset to answer relevant questions like which topics people tend to react to sarcastically, which subreddit has the highest percentage of sarcastic comments and whether the linguistic features(capital letters etc.) have any bearing on the sarcasm score.

Our end goal is to build an efficient classification model that can detect sarcasm in comments on the Internet commentary website Reddit

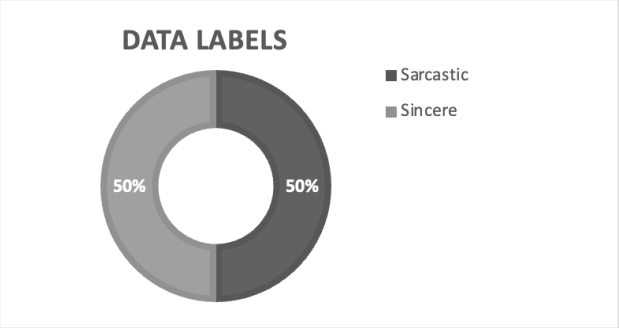
Methods

1. Dataset Selection

We started the project by parsing through Kaggle for ideas. We came across Sarcasm Detection which proved interesting. We collected and read over some research papers which outlined some of the techniques used for sarcasm detection on other platforms like Twitter. Those papers and some Kaggle notebooks helped us plan out a strategy for our project.

2. Exploratory Data Analysis

The dataset we have used is a balanced dataset from Kaggle, named Sarcasm on Reddit, containing ~1.3 million labelled comments.



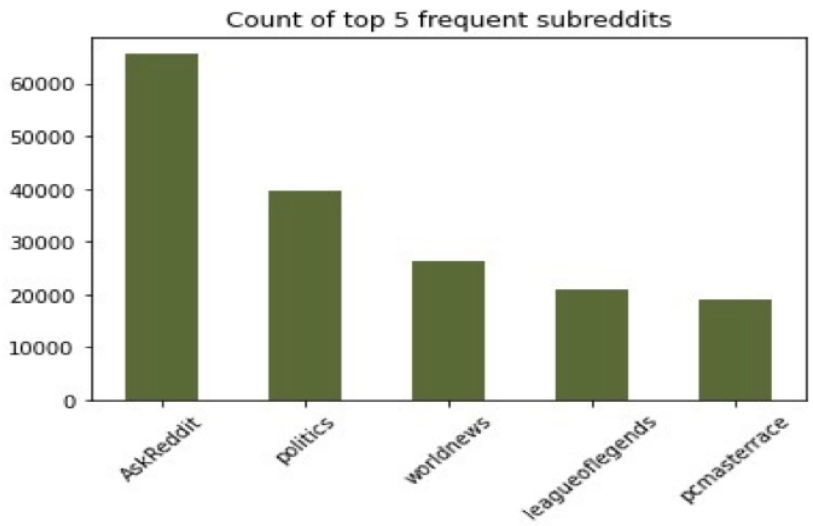
	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	NC and NH.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	Yeah, I get that argument. At this point, I'd ...
1	0	You do know west teams play against west teams...	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	The blazers and Mavericks (The wests 5 and 6 s...
2	0	They were underdogs earlier today, but since G...	Creepeth	nfl	3	3	0	2016-09	2016-09-22 21:45:37	They're favored to win.
3	0	This meme isn't funny none of the "new york ni...	icebrotha	BlackPeopleTwitter	-8	-1	-1	2016-10	2016-10-18 21:03:47	deadass don't kill my buzz
4	0	I could use one of those tools.	cush2push	MaddenUltimateTeam	6	-1	-1	2016-12	2016-12-30 17:00:13	Yep can confirm I saw the tool they use for th...

Please find below the different data features and their meaning:

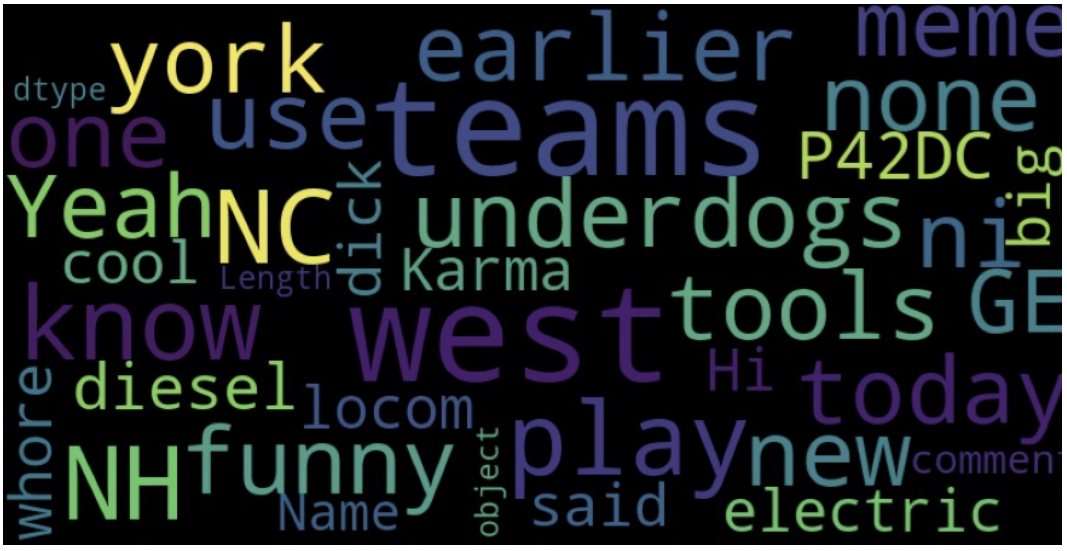
Sr. No.	Feature Name	Feature Description
1	Label	0/1 indicator where 1 denotes a sarcastic comment and 0 denotes a sincere comment
2	Comment	Reddit comment
3	Author	User who posted the Reddit comment
4	Subreddit	Topic specific forums on Reddit
5	Score	Score voted on by Reddit users
6	Ups	Upvotes on the comment
7	Downs	Downvotes on the comment
8	Date	Date the comment was posted (YYYY-MM)
9	Created UTC	Date the comment was posted in UTC format (YYYY-MM-DD HH:MM:SS)
10	Parent Comment	Parent comment of the Reddit comment under feature 'Comment'

We also performed a few EDA techniques to help us better understand the data. Please find below our observations:

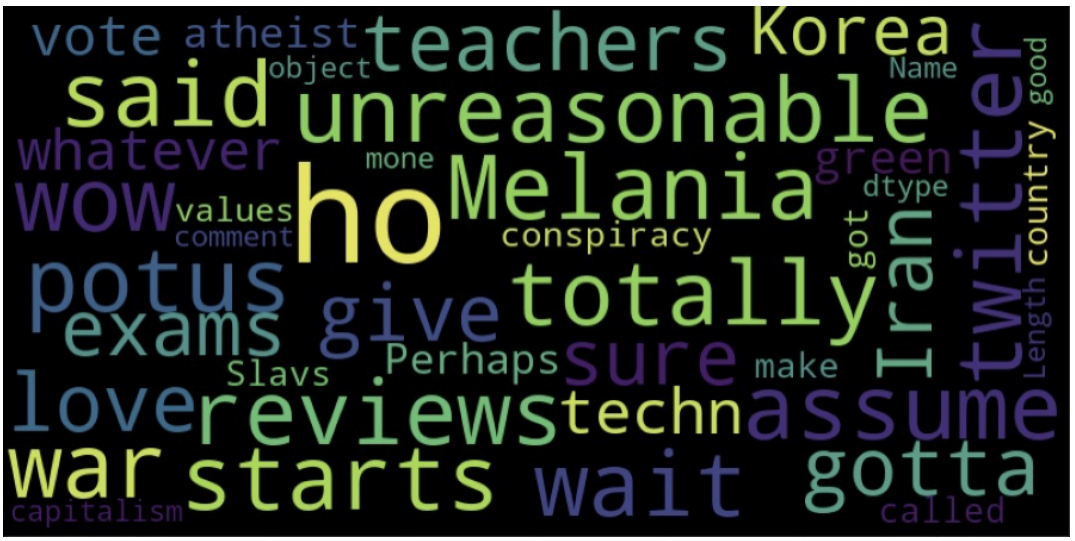
- The comment feature had 53 missing values. The other features did not have any missing data
- The top 5 subreddits in the dataset are:



- Additionally, the highest percentage of sarcastic comments were left under the subreddit topic of Politics, Worldnews and League of Legends
- The features with the highest correlation with the target label were : Created UTC, Downs, Comment and Subreddit
- We created a column to count the number of capital letters in the comment but observed that it did not have much correlation with the target label
- Word Cloud of most frequent words in Sincere comments:



- Word Cloud of most frequent words in Sarcastic comments:



3. Feature Engineering and Selection

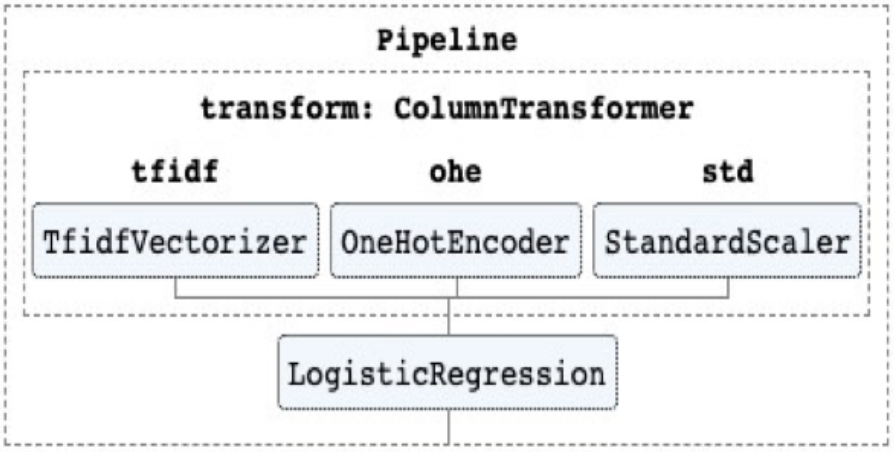
- Based on the results of the EDA, we performed some basic data cleaning techniques like dropping rows with missing comments.
- We converted the feature 'created_utc' to a datetime object and extracted 'year', 'month', 'day', 'hour', 'minute', 'seconds' attributes as separate features from it
- By using the Pearson's correlation method, we have retained only six features of all the existing and newly created features, :
    1. Three numerical: Downs, Year, Month
    2. Two categorical: Subreddit, Created UTC
    3. One text feature: Comment

4. Modelling Pipelines

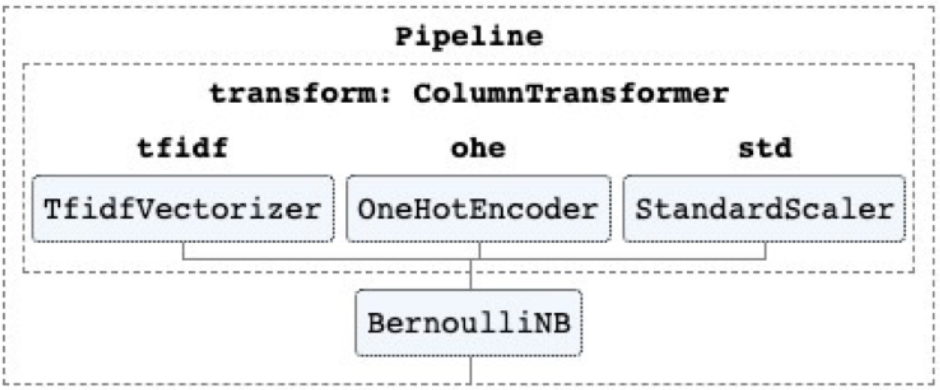
Once we finalized the features, we built a Logistic Regression model as a baseline pipeline with the following steps:

1. Data preprocessing:
    - Tokenizing and learning vocabulary of the comments column using the TfidfVectorizer
    - Converting the categorical feature values to binary using the OneHotEncoder
    - Scaling the numerical features using StandardScalar
2. Split the dataset into train and test
   Training: 80% and Testing: 20%
3. Train and evaluate the model

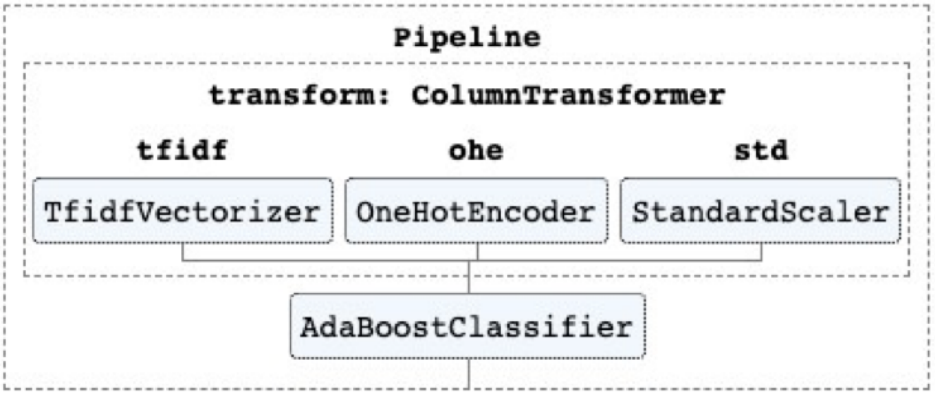
4. Evaluating the model using accuracy_score metric



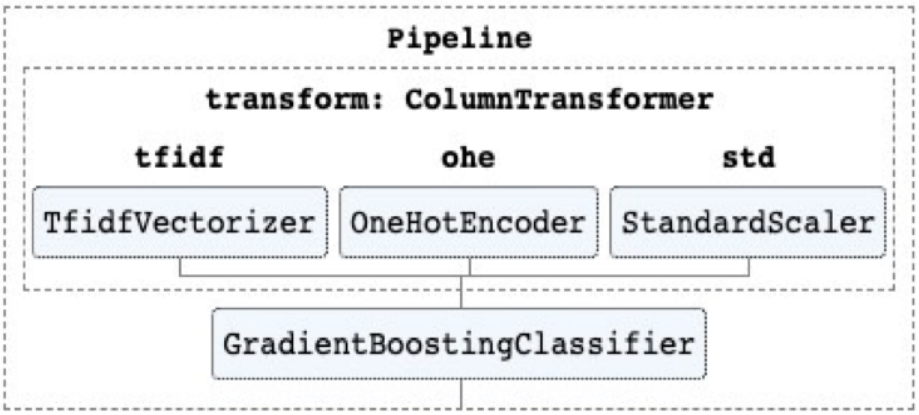
5. We also trained and evaluated additional models like
    - Bernoulli Naive Bayes Classifier



    - Adaptive Boosting Classifier



    - Gradient Boosting Classifier



To improve accuracy, we then performed hyperparameter tuning on the top two models : Logistic Regression and Bernoulli Naive Bayes. We considered the following parameters:

- For Logistic Regression
  penalty: 'l1' or 'l2'
  C: 0.1, 1 or 10
  Out of the above parameters, the model chose the parameters of penalty: 'l2', C:1, max_iter: 10,000 and solver: saga to give an accuracy of 72.8%
- For Naive Bayes,
  alpha: 0 or 1
  binarize: 0, 0.5 or 1
  fit_prior: True or False
  Out of the above parameters, the model retained the default parameters of alpha: 1, binarize: 0 and fit_prior: True to give an accuracy of 71.16%

Results

- In conclusion, over the course of this project, we have explored the dataset, generated new features, trained and tested various models to detect sarcasm.
- In data exploration, we plotted word clouds to identify the most frequent words in sarcastic and sincere comments.
- We then generated additional features by splitting 'created_utc' datetime object to its attributes and counting capital letters in comments. We then used Pearson's correlation method to reduce down the number of features to only 6 which were then used while training the models.
- We trained and evaluated a baseline model of Logistic Regression (best accuracy of 72.8%) as well as Bernoulli Naive Bayes (accuracy 71.16%), Adaboost (accuracy 64%) and Gradient Boost (accuracy 64.4%) models.
- Our best model is Logistic Regression with an accuracy of 72.8%

Discussion

As per our observation, although Logistic Regression yielded better accuracy, Bernoulli Naive Bayes had a much less processing time than the other models.

One of the main challenges we faced was while performing hyperparameter tuning. We could not execute GridSearchCV on Logistic Regression as the processing time for the same exceeded 15 hours. We used it on the second best model of Bernoulli Naive Bayes but manually checked for the best parameters for Logistic Regression to get accuracy of 72.8%.

References

- The below sklearn links have been used as a reference while building our models:
    1. Pipeline
    2. Column Transformer
    3. GridSearchCV
    4. TfidfVectorizer
    5. Logistic Regression
    6. Naïve Bayes
    7. AdaBoost
    8. GradientBoost
- The below research papers were used to understand the research techniques that have previously been used on a similar dataset in the field of sarcasm detection and text analysis :
    1. "Large Self-Annotated Corpus for Sarcasm" by authors Mikhail Khodak, Nikunj Saunshi, Kiran Vodrahalli
    2. "A Pattern-Based Approach for Sarcasm Detection on Twitter" by authors Mondher Bouazizi and Tomoaki Otsuki
    3. "Bag of Tricks for Efficient Text Classification" by authors Armand Joulin, Edouard Grave, Piotr Bojanowski and Tomas Mikolov
    4. "Modelling Context with User Embeddings for Sarcasm Detection in Social Media" by authors Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho and Mario J. Silva
    5. "Text Classification of Reddit Posts" by authors Jacqueline Gutman and Richard Nam
    6. "Representing Social Media Users for Sarcasm Detection" by authors Y. Alex Kolchinski, Christopher Potts