INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

Sarcasm Detection on Reddit

(Minor Project Report)

Submitted By

Muskan Gupta
Shreya Bansal
Sheenu Mittal
Tanya Sharma
Sejal Banta
Neeraj Rani

Under the supervision of

Rishabh Kaushal
Asst. Prof.
IT Dept, IGDTUW

Abstract

Sarcasm is an act of commenting on a person, place or thing in an indirect manner where the other person's conscience is required to understand the meaning beneath it. On social platforms people often comment sarcastically. There should be some criteria on the basis of which we can know the nature of that comment. Our project focuses on Reddit social platform where people post a lot of comments. The aim is to classify these comments as sarcastic or non sarcastic by using various classification and regression machine learning algorithms. We will also find the accuracy of each algorithm.

Contents

1 Introduction

- Sarcasm conveys the meaning different than the literal one. Its presence changes the polarity of sentiment analysis.
- We will use the Reddit comment dataset taken from Kaggle. Reddit is an online discussion platform, where the community members can post information regarding news, technology, politics and other areas of interest. The areas of interests are categorized as subreddits (/news, /politics etc). The sarcastic comments are obtained by scraping /s from the comments.
- As the literal meaning and sarcastic meaning of the comments are different, the aim is to do sentiment analysis of the comments.

1.1 Problem Statement

To classify the comments on Reddit platform as sarcastic(1) or non sarcastic(0).

2 Related Work

hazarika2018cascade et al. [1] intoduced CASCADE (a Contextual Sarcasm detector) for detecting sarcasm in online social media discussions by using content and context-driven models. They also used user embeddings for enoding stylometric features of users alongwith content based feature extractors such as Convolutional Neural Networks (CNN).

Khodak2017 et al. [2] introduced the Self-Annotated Reddit Corpus (SARC) for sarcasm detection which has has 1.3 million sarcastic statements. In this, sarcasm is labeled by the name of the author which is provided with user, topic, and conversation context.

2.1 Comparison of Proposed Approaches and research gaps

We have used only supervised learning algorithms like Naive Bayes, Logistic Regression, Linear Support Vector Machines and Random forest classifier for classification of comments. Our work is different from above stated research papers as we have have not used CNN or any other unsupervised algorithm.

3 Data Collection And Analysis

3.1 Dataset Description

We have used data set from Kaggle website which is open source website for data sets. https://www.kaggle.com/danofer/sarcasm

Table 1: Details of the dataset.

Details	Count
Number of unique parent comments	984286
Number of unique comments	962294
Number of authors	256561

Table 2: Details of Data Attributes.

Data Attributes	Brief Explanation
Comment	Reply to the parent comment that need to be stated as sarcastic/non sarcastic
Author	User Id of the writer of comment
Subreddit	subreddit for the current comment
Score	Count of number of upvotes minus number of downvote
Ups	Count of upvotes
Downs	Count of downvotes
Date	Contains month and year of creation of comment
Created-utc	Timestamp for the comment
Parent-comment	Parent comment for reply comment
Label	Determines if the comment is sarcastic(1) or non sarcastic (0)

3.2 Data Pre-processing

The dataset is first tested to remove any null values, if existing. The sarcasm dataset taken from kaggle had null values in the comment column.

Data Attributes	Number of null values
label	0
comment	53
author	0
subreddit	0
score	0
ups	0
downs	0
date	0
created-utc	0
parent-comment	0

3.3 Data Visualization

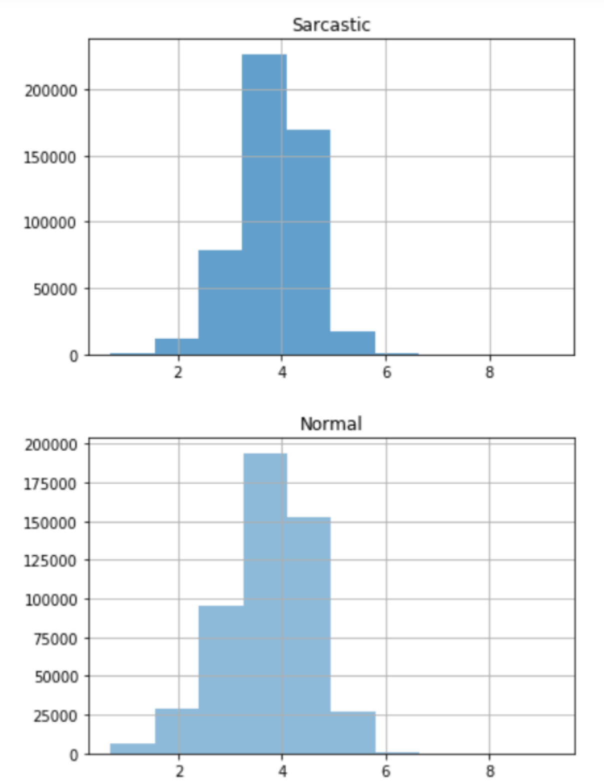
3.3.1 Step 1

First we observe the number of sarcastic and non-sarcastic comments.

Target Value	**Number of rows**
0	505405
1	505368

3.3.2 Step 2

We see the distribution of lengths for sarcastic and normal comments which is almost the same.

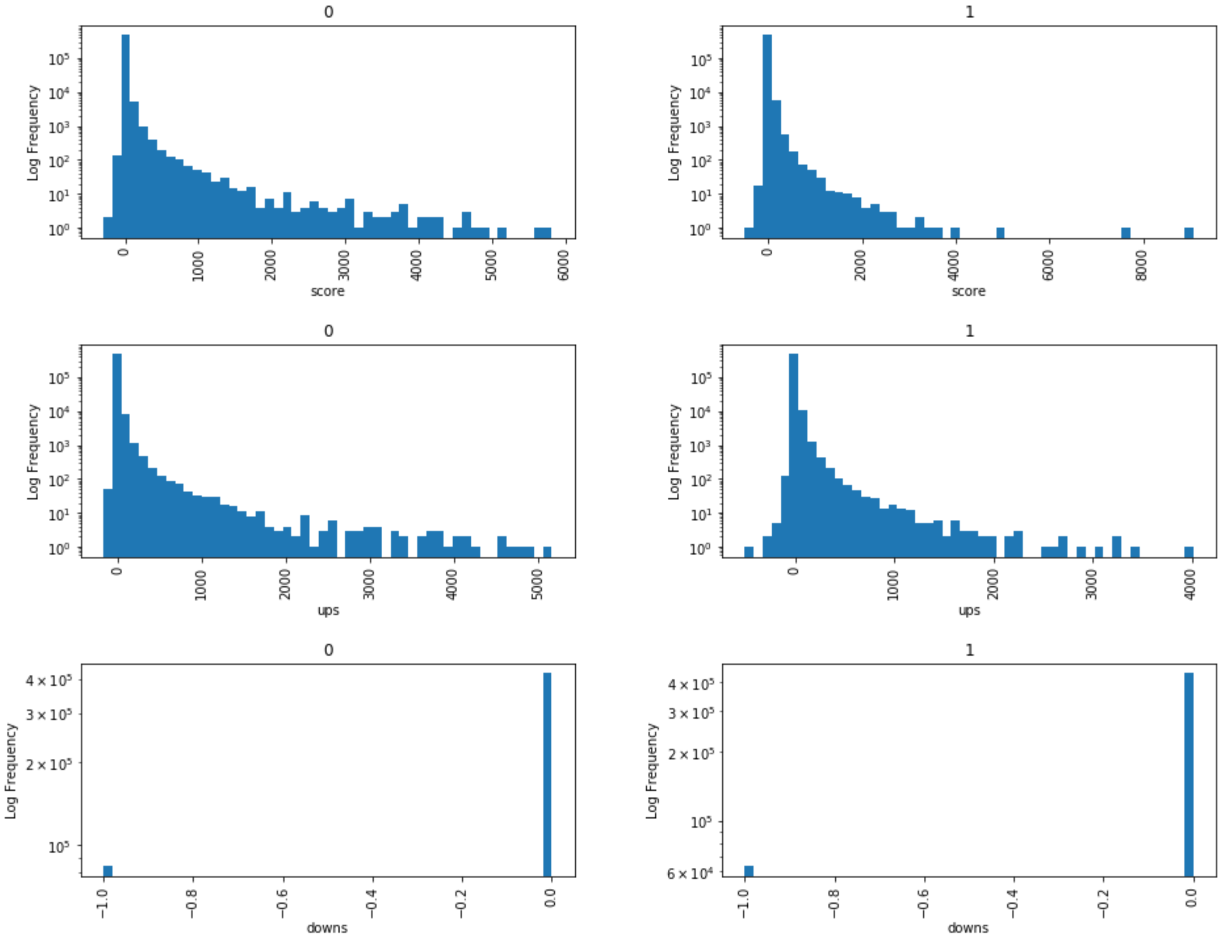

Sarcastic

Normal

Figure 1: Lengths of sarcastic and non-sarcastic comments

3.3.3 Step 3

Visualize data by plots that show frequencies of score, upvotes and downvotes for both the classes 0 and 1. These graphs are frequency vs score, upvotes and downvotes.



3.3.4 Step 4

Here, word cloud is made on the dataset for sarcastic and non-sarcastic words separately. It represents not only the frequency, but importance of each word.

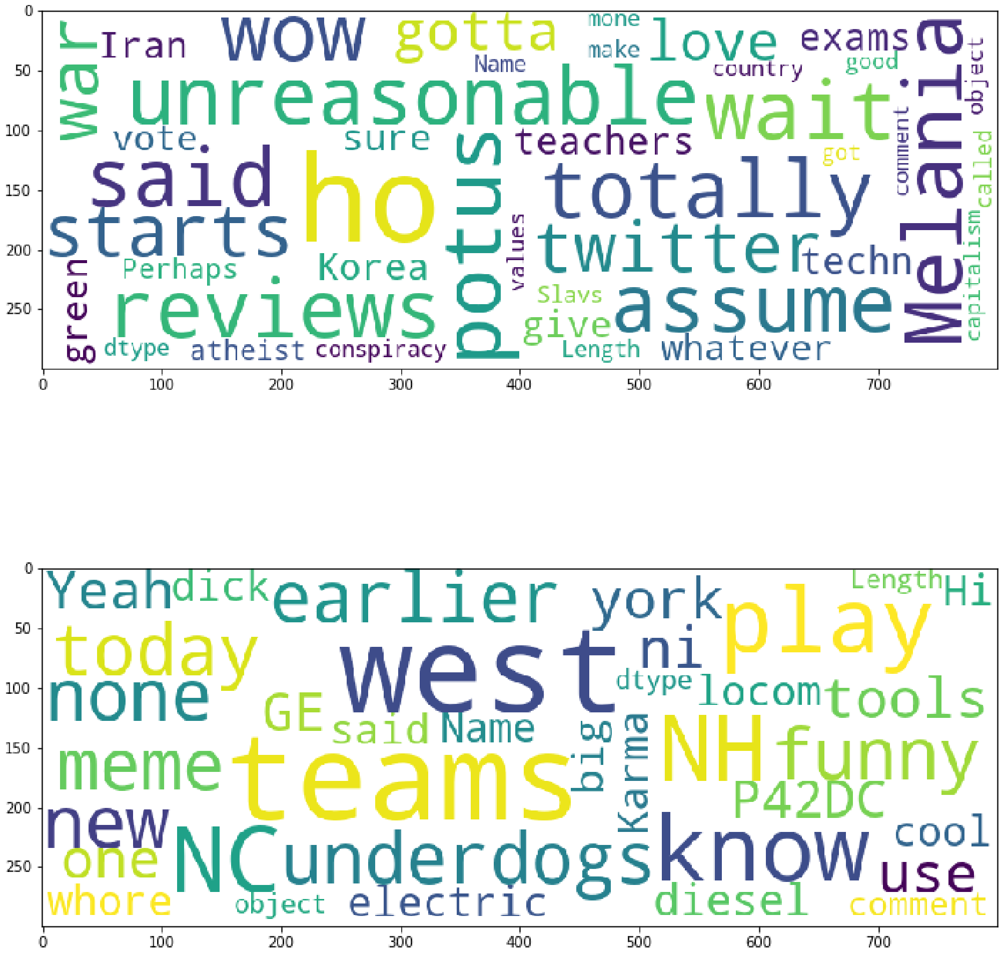


Figure 2: Stop words cloud

4 Proposed Methodology

Machine Learning Task
Sarcasm detection is a classification based Machine Learning task. We are solving this problem using supervised learning approaches.
Inputs Comments and Score Values of each comment.
Output Predicted label 1 for sarcastic and 0 for non-sarcastic comment.

4.1 Logistic Regression

Logistic Regression uses sigmoid function with the real value obtained after vectorization of each comment as input.

$$f(z) = \frac{1}{1 + e^{(-z)}} \tag{1}$$

Output value of the sigmoid function lies between 0 to 1 and is mapped to the output class label on the basis of the threshold which is used as a decision boundary.

$$\textbf{predict } y = \begin{cases} 1 & f(z) > 0.5 \\ 0 & f(z) < 0.5 \end{cases}$$

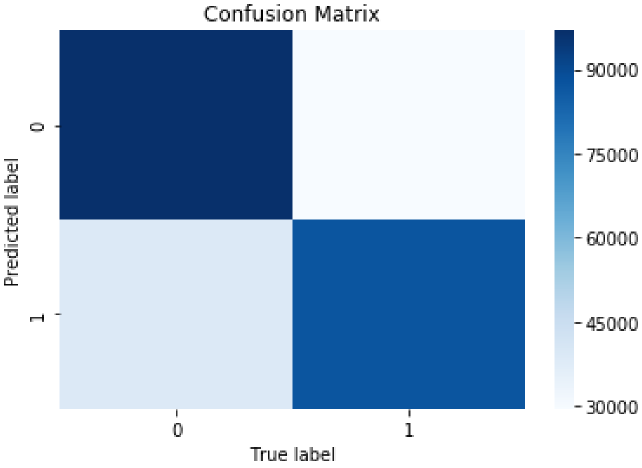


Figure 3: Confusion Matrix

We observe from the confusion matrix that the true predictions (TP + TN) are much greater than the false predictions (FP + FN).

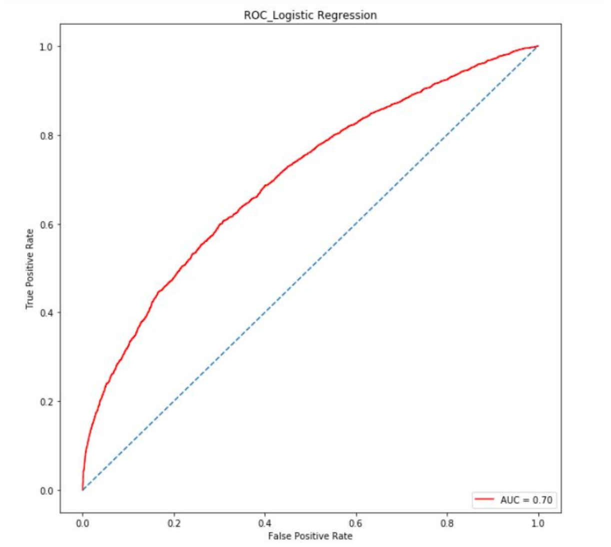


Figure 4: ROC Curve

We see that Area under the curve is 0.70 which implies that the model distinguishes between the classes correctly with 70% probability.

4.2 Gaussian Naive Bayes

Gaussian Naive Bayes uses TF-IDF Vectorization where each comment is represented as TF-IDF score.
TF-IDF score of word t in document d can be represented as :

$$W_{i,j} = \mathbf{tf}_{i,j} \times \log(\frac{N}{dfi}) \tag{2}$$

- $\mathrm{tf}_{\mathrm{i,j}}$ = number of occurences of i in j
- df_{i} = number of documents containing i
- N = total number of documents

Now after the TF-IDF score is calculated, probability is calculated using Bayes Theorem. The class with higher probability is the predicted class.

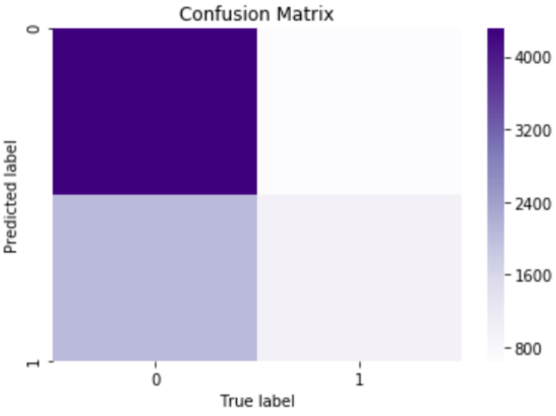


Figure 5: Confusion Matrix

We observe from the confusion matrix that the true predictions (TP + TN) are much greater than the false predictions (FP + FN), but not as accurate as in logistic regression model.

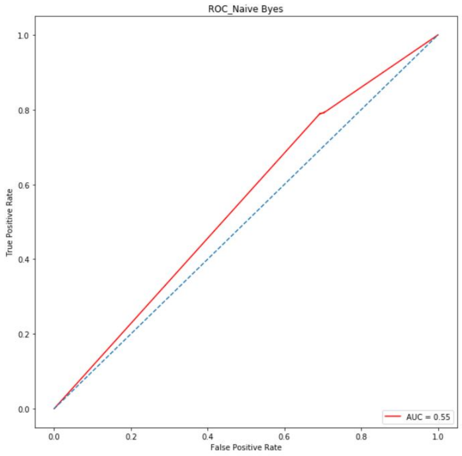


Figure 6: ROC Curve

We see that Area under the curve is 0.55 which implies that the model distinguishes between the classes correctly with 55% probability. This model is not suitable for this problem due to bad accuracy.

4.3 Linear Support Vector

Linear Support Vector uses hyperplanes to divide classes. Since we have two input features, we get a linear function.

Output value of the linear function lies between -1 and 1 where 1 is mapped to one of the output class label and -1 is mapped to the other class.

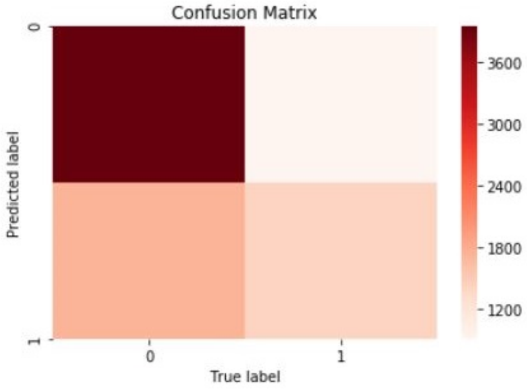


Figure 7: Confusion Matrix

We observe from the confusion matrix that the true predictions (TP + TN) are much greater than the false predictions (FP + FN), which is better than what we observed in Naive Bayes Classifier.

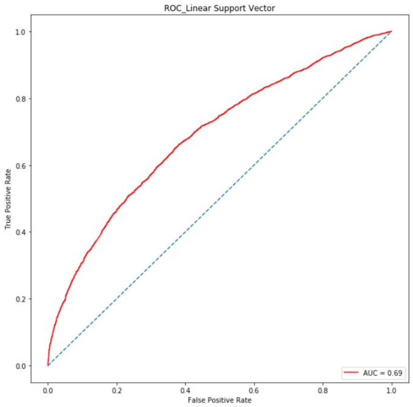


Figure 8: ROC Curve

We see that Area under the curve is 0.69 which implies that the model distinguishes between the classes correctly with 69% probability.

4.4 Random Forest

Random Forest classifier constructs a decision tree against each sample of the training set and predicts the vote for each decision tree. Majority vote is our predicted class.

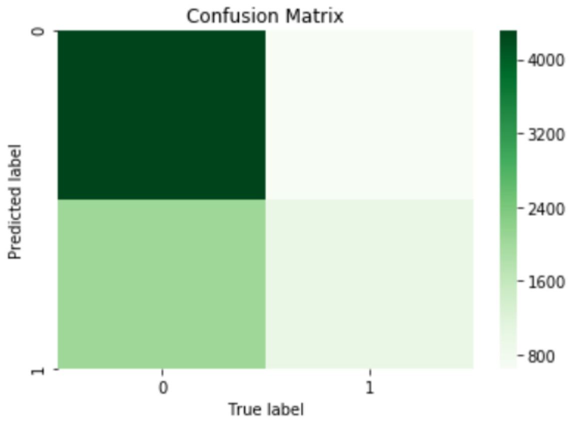


Figure 9: Confusion Matrix

We observe from the confusion matrix that the true predictions (TP + TN) are much greater than the false predictions (FP + FN).

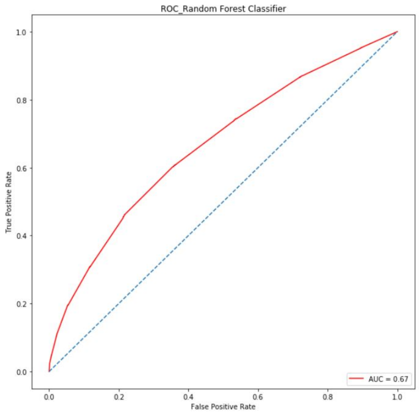


Figure 10: ROC Curve

We see that Area under the curve is 0.67 which implies that the model distinguishes between the classes correctly with 67% probability.

5 Evaluation & Results

After all the models have been applied, we check the accuracy of each model in the table below.

Table 3: Accuracy as an evaluation metric

Model	**Accuracy**
Logistic Regression	67.85
Gaussian Naive Bayes	49
Linear Support Vector	66.91
Random Forest	65.63

We compare all the models with respect to their accuracy, precision and recall.

5.1 Accuracy Comparison Graph

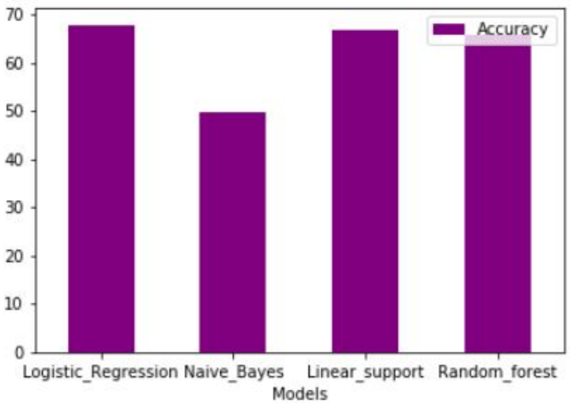


Figure 11: Accuracy Graph

5.2 Precision Comparison Graph

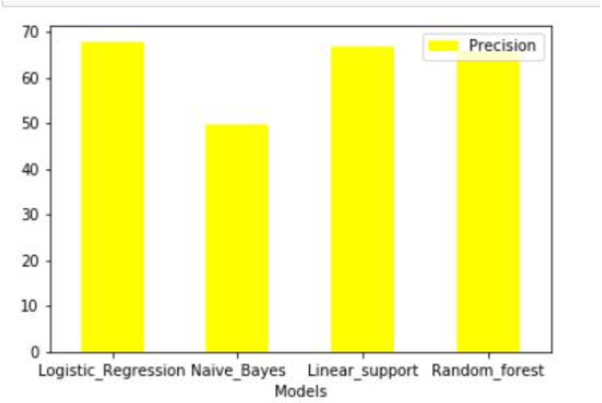


Figure 12: Precision Graph

5.3 Recall Comparison Graph

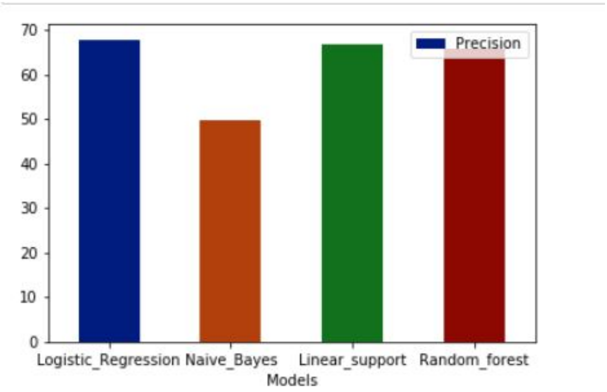


Figure 13: Recall Graph

6 Conclusion and Future Work

On comparing all the four models, **Logistic Regression** has the best performance in terms of accuracy followed by linear support vector, random forest and naive bayes.

Future work related to the project

- Detecting trending subreddits
- Evaluating author activeness
- Predicting trending hashtags

References

[1] Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. Cascade: Contextual sarcasm detection in online discussion forums. *arXiv preprint arXiv:1805.06413*, 2018.

[2] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*, 2017.