

Sarcasm has been part of our language for many years. It means being the opposite of what you mean, usually with a distinct tone of voice in a fun way. If you think that anyone can understand sarcasm, then you are wrong, because understanding sarcasm depends on your language skills and your knowledge of other people's minds. But what about a computer? Is it possible to train a machine learning model that can detect whether a sentence is sarcastic or not? Yes, it is!



Sarcasm means being funny by being the opposite of what you mean. It has been part of every human language for years. Today, it is also used in news headlines and various other social media platforms to gain more attention. Sarcasm detection is a natural language processing and binary classification task. We can train a machine learning model to detect whether or not a sentence is sarcastic using a dataset of sarcastic and non-sarcastic sentences that I found on Kaggle.

I hope you now have understood what sarcasm is. In the section below, I'll walk you through the task of detecting sarcasm with machine learning using the Python programming language. The dataset that I am using for this task can be downloaded from here: <https://raw.githubusercontent.com/amankharwal/Website-data/master/Sarcasm.json>

Now let's start with the task of sarcasm detection with machine learning using Python. I'll start this task by importing the necessary Python libraries and the dataset:

```
In [ ]: import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import BernoulliNB

data = pd.read_json("Sarcasm.json", lines=True)
print(data.head())
```

The "is\_sarcastic" column in this dataset contains the labels that we have to predict for the task of sarcasm detection. It contains binary values as 1 and 0, where 1 means sarcastic and 0 means not sarcastic. So for simplicity, I will transform the values of this column as "sarcastic" and "not sarcastic" instead of 1 and 0:

```
In [ ]: data["is_sarcastic"] = data["is_sarcastic"].map({0: "Not Sarcastic", 1: "Sarcastic"})
print(data.head())
```

Now let's prepare the data for training a machine learning model. This dataset has three columns, out of which we only need the "headline" column as a feature and the "is\_sarcastic" column as a label. So let's select these columns and split the data into 20% test set and 80% training set

```
In [ ]: data = data[["headline", "is_sarcastic"]]
x = np.array(data["headline"])
y = np.array(data["is_sarcastic"])

cv = CountVectorizer()
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)
```

Now I will be using the Bernoulli Naive Bayes algorithm to train a model for the task of sarcasm detection:

```
In [ ]: model = BernoulliNB()
model.fit(X_train, y_train)
print(model.score(X_test, y_test))
```

0.8448146761512542 Now let's use a sarcastic text as input to test whether our machine learning model detects sarcasm or not:

```
In [ ]: user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(output)
```

Enter a Text: Cows lose their jobs as milk prices drop ['Sarcasm']

# So this is how you can easily train a machine learning model for the task of sarcasm detection

## Summary

So this is how you can use machine learning to detect sarcasm by using the Python programming language. Sarcasm has been part of our language for many years. It means being the opposite of what you mean, usually with a distinct tone of voice in a fun way.

In [ ]:

In [ ]: