Reddit sentiment analysis

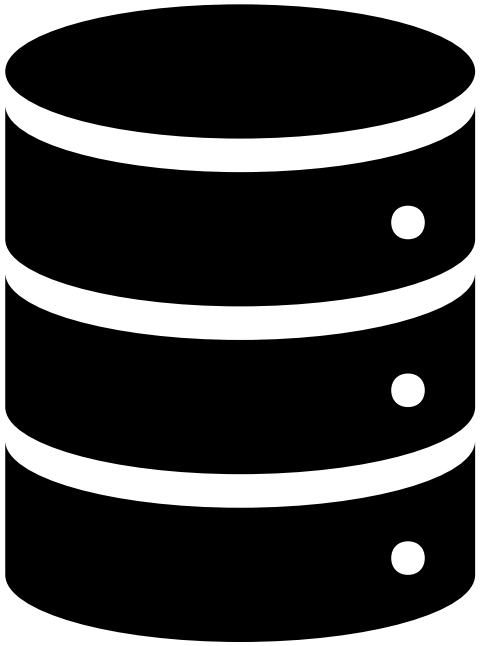
Using naïve bayes sentiment analysis algorithm

Project introduction

This is a prototype for my final project, which is a real-time sentiment analysis system. The system uses Kafka to stream textual data from Reddit, and then uses natural language processing (NLP) techniques and naïve bayes algorithms to analyze the sentiment of the data.

What is Kafka?

Kafka is a distributed streaming platform that can be used to collect and process data in real time. Kafka is built on top of ZooKeeper and uses a publish-subscribe model to communicate between producers and consumers.



What is Kafka

- Producers and Consumers
- Producers publish data to Kafka topics. Consumers subscribe to topics and receive data that is published to those topics.
- Topics
- Topics are logical channels that data is published to and consumed from. Topics can be partitioned to distribute data across multiple Kafka brokers.
- Brokers
- Kafka brokers are responsible for storing and delivering data to consumers.

What is praw?

PRAW is a Python wrapper for the Reddit API. It provides a convenient way to access and interact with the Reddit API from Python.

How to use Kafka and praw to collect data from reddit

- Create a Kafka topic to store the data.
- Write a Python script to connect to the Reddit API and stream the data to the Kafka topic.
- Start a Kafka consumer to read the data from the Kafka topic and process it.

Naïve Bayes algorithm

- The Naive Bayes algorithm is a simple yet powerful machine learning algorithm for classification. It is based on Bayes' theorem, which states that the probability of a hypothesis given the evidence is equal to the probability of the evidence given the hypothesis, times the probability of the hypothesis, divided by the probability of the evidence.
- $P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$

Naïve bayes for classification algorithm

- In binary classification, we are trying to predict whether a given data point belongs to one of two classes, such as positive or negative. To do this, we can use the Naive Bayes algorithm to calculate the probability of the data point belonging to each class.

$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}$$

Naïve Bayes inference rule

- Count the frequency of each word in a given class(positive and negative) and the no. Of words in a class which is N.
- Add their frequencies in positive class divided by the frequencies of negative class

Laplacian smoothing

$$P(\mathrm{w_i}|\mathrm{class}) = \frac{\mathrm{freq}(\mathrm{w_i}, \mathrm{class})}{\mathrm{N_{class}}}$$

class ∈ {Positive, Negative}

$$P(\mathrm{w_i}|\mathrm{class}) = \frac{\mathrm{freq}(\mathrm{w_i}, \mathrm{class}) + 1}{\mathrm{N_{class}} + \mathrm{V}}$$

N_{class} = frequency of all words in class

V = number of unique words in vocabulary

Laplacian smoothing

- Laplacian smoothing is used to avoid P(w_i | class) = 0 to avoid numerical underflow error in products of probabilities.

Naïve bayes inference rule

$$\frac{P(pos)}{P(neg)} \prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}$$

- To avoid the risk of numerical underflow in product of values we use log
- Log(a*b) = log(a) + log(b)

$$log(\frac{P(pos)}{P(neg)} \prod_{i=1}^{n} \frac{P(w_i|pos)}{P(w_i|neg)}) \Rightarrow log\frac{P(pos)}{P(neg)} + \sum_{i=1}^{n} log\frac{P(w_i|pos)}{P(w_i|neg)}$$

log prior + log likelihood

Implementation of naïve bayes model

1. Collect or annotate a dataset with positive and negative texts
2. Preprocess the text
3. Compute frequency of words freq(w, class)
4. Get P(w|pos) and P(w|neg)
5. Compute log likelihood = log(P(w|pos)/P(w|neg))
6. Compute logprior = log(P(pos)/P(neg))

Predict using naïve bayes

P = logprior

For each word in given text:

If the loglikelihood of word is given add it it to p

Test naïve bayes model

- X_value, Y_value, loglikelihood and logprior
- Score = predict(X_value, loglikelihood, logprior
- Pred = score >0

$$\begin{bmatrix} 0.5 \\ -1 \\ 1.3 \\ \vdots \\ score_m \end{bmatrix} > 0 = \begin{bmatrix} 0.5 > 0 \\ -1 > 0 \\ 1.3 > 0 \\ \vdots \\ score_m > 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ pred_m \end{bmatrix}$$

$$\frac{1}{m}\sum_{i=1}^{m}(pred_i == Y_{val_i})$$

Test naïve bayes

- X_value, Y_value, loglikelihood and logprior
- Score = predict(X_value, loglikelihood, logprior
- Pred = score >0

Calculate metrices for further improvement

- Compute confusion matrix
- Compute the following values
  precision, recall, f1_score, mean squared error, linear kappa, quadratic kappa and accuracy.

Benefits of using Kafka

- Scalability: Kafka is a highly scalable platform that can handle large volumes of data.
- Real-time processing: Kafka allows you to process data in real time, which is important for many applications.
- Fault tolerance: Kafka is a fault-tolerant platform that can continue to operate even if some of its components fail.