IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 8, Issue.11, November 2020 E-ISSN: 2347-2693

Designing Shopping Cart and Determining Fake Product Comments Using Multinomial NB

N. Bhargavi

Computer science and systems engineering, Andhra University College of engineering, Andhra University, Visakhapatnam, India

Author's Mail Id: beaulah5219@gmail.com

DOI: https://doi.org/10.26438/ijcse/v8i11.4852 | Available online at: www.ijcseonline.org

Received: 02/Nov/2020, Accepted: 19/Nov/2020, Published: 30/Nov/2020

Abstract— In these days, there are a lot of shopping websites and apps that are very good for people requirements in their daily lives. But the quality of the product is known to the customer with the help of the reviews or comments of previous users. Some product producers are doing fake actions on those comments. Hence we don't know the right quality. Hence in this paper, I developed a shopping cart with simplicity and flexibility, user friendly. And it is incorporated with the safe comments. The admin can recognize the list of comments is of safe or unsafe. Here, multinomial naïve bayes technique is used for fake ones and python programming is used for shopping cart development. We can block the fake customer through customer credentials.

Keywords—flexibility, quality, admin, multinomial NB

I. INTRODUCTION

In this project, designing shopping cart and determining fake product comments using multinomial NB algorithm, I first developed an e-commerce website that is too a shopping cart. It is developed in Django framework with python programming language. Where I invoked a shirt to buy. A user can add the shirt to cart and proceed to pay and comment on the product that is shirt. All these actions are available to do only when the user is authenticated. First of all, user must login into the shopping cart. After that, many number of users commented the product, to guess the product has genuine comments or fake comments, download the comments table into csv format of document. And then I used another software to analyse the document of comments whether they are fake or genuine. The software is Jupyter notebook. Where I used Multinomial NB and predicted the result including plots also. Multinomial naïve bayes classifier is used for text classification. Here, Feature vectors represent the frequencies with which certain events have been generated by a **multinomial distribution**. This is the event model typically used for text classification. The simple design of Naive Bayes classifiers make them very attractive for such classifiers. Moreover, they have been demonstrated to be fast, reliable and accurate in a number of applications of NLP: Text classification. Hence clearly, shopping the items with good quality without fake items, we can purchase. The admin can remove the fake user from the cart and the activities can not be done. Our algorithm helps us to fond fake product comments. We are having genuine and safe shopping.

II. RELATED WORK

A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis. These classifiers are widely used for machine learning because they are simple to implement. Naive Bayes is also known as simple Bayes or independence Bayes.Naive Bayes is a **linear classifier** while K-NN is not; It tends to be faster when applied to big data. In comparison, k-nn is usually slower for large amounts of data, because of the calculations required for each new step in the process. If speed is important, choose Naive Bayes over K-NN. Although statistical learning methods have achieved success in e-commerce platform product review sentiment classification, two problems have limited its practical application: 1) The computational efficiency to process large-scale reviews; 2) the ability to continuously learn from increasing reviews and multiple domains. (fengXu, 2018).[1] The algorithm used in conducting sentiment analysis is Naïve Bayes because it has a high degree of accuracy in classifying sentiment analysis. The stages in conducting sentiment analysis in this study are preprocessing data, processing data, classification, and evaluation. The sentiment analysis obtained in this study shows that Twitter users in Indonesia give more neutral comments. In this study, an accuracy of 86.43% was obtained from testing data using Naïve Bayes Algorithm in RapidMiner tools, where the accuracy is higher than the other algorithms, Decision Tree and Random Forest which is 82.91%. (AmiliaFitr, 2018)[2]. Constructing a classifier

from the probability model, then move to data preprocessing, training and hyperparameters optimization stages. We will write our script in Python using Jupyter Notebook. (Rubtsova, 2018)[3]. In Machine Learning approach, the classifier is built automatically by learning the properties of categories from a set of pre-defined training data. Hence, it can process complex furthermore, multi assortment information in dynamic situations. Here we propose a naïve bayes classifier which scales directly with number of indicators and data points which can be used for both binary and multiclass classification problems. (Asatani, 2018)[4]. As the name suggests, classifying texts can be referred as text classification. Usually, we classify them for ease of access and understanding. We don't need human labour to make them sit all day reading texts and labelling categories. Document classification is a classical machine learning problem. If there is a set of documents that is already categorized/labeled in existing categories, the task is to automatically categorize a new document into one of the existing categories.There are many different algorithms we can choose from when doing text classification with machine learning. One family of those algorithms is known as Naive Bayes (or NB) which can provide accurate results without much training data. In this article, we will explore the advantages of using one of the members of the bayesian family (namely, Multinomial Naive Bayes, or MNB) in text classification and will help you get started with MNB-based models.The adapted NBC is realized through programming on MBNC experimental platform and applied to the nature scene classification, the results show that it is effective. (Qin, 2012)[5]. The most important task is to maintain context data between successive user sessions. Although several methods which can be applied to implement the shopping cart system on the Web system have been proposed, none of them can attain the task of maintaining context data sufficiently. In this paper, we analyze the task and point out the following difficulties: (1) reliability, (2) safety and (3) session management. We then propose a new mechanism, called the context data storage (CDS) mechanism, to solve all of the above difficulties. (Uehara, 2002)[6]. In the modern world, shopping has become an essential day to day activity for most of the people. However, their busy life style has lessened the time to do shopping. This has made them to look for quicker and easier ways to do their shopping. Some of the difficulties that people have to go through when they do shopping include having to travel a long distance without knowing the availability of the items, difficulty in finding relevant shops inside a shopping mall, forgetting to buy some items which they intended to buy. In order to overcome the above mentioned problems a fully functional shopping mall application is proposed in this paper. (Karunarathna, 2014).[7] Although these websites can help consumers get the parity price of commodities, the search results are not so ideal. Because these websites may occur problems about the difference commodity between search results and consumers want to search, or the difference commodity price between search results and commodity web page. (ying, 2014)[8].

III. METHODOLOGY

A. Naïve Bayes Classifier:

It works on the famous **Bayes theorem** which helps us to find the conditional probabilities of occurrence of two events based on the probabilities of occurrence of each individual event.

Starting more than half a century ago, scientists became very serious about addressing the question: “Can we build a model that learns from available data and automatically makes the right decisions and predictions?” Looking back, this sounds almost like a rhetoric question, and the answer can be found in numerous applications that are emerging from the fields of pattern classification, machine learning, and artificial intelligence.

Data from various sensoring devices combined with powerful learning algorithms and domain knowledge led to many great inventions that we now take for granted in our everyday life: Internet queries via search engines like Google, text recognition at the post office, barcode scanners at the supermarket, the diagnosis of diseases, speech recognition by Siri or Google Now on our mobile phone, just to name a few.

One of the sub-fields of *predictive modeling* is *supervised pattern classification*; supervised pattern classification is the task of training a model based on labeled training data which then can be used to assign a pre-defined class label to new objects. One example that we will explore throughout this article is spam filtering via naive Bayes classifiers in order to predict whether a new text message can be categorized as spam or not-spam. Naive Bayes classifiers, a family of classifiers that are based on the popular Bayes’ probability theorem, are known for creating simple yet well performing models, especially in the fields of document classification and disease prediction.

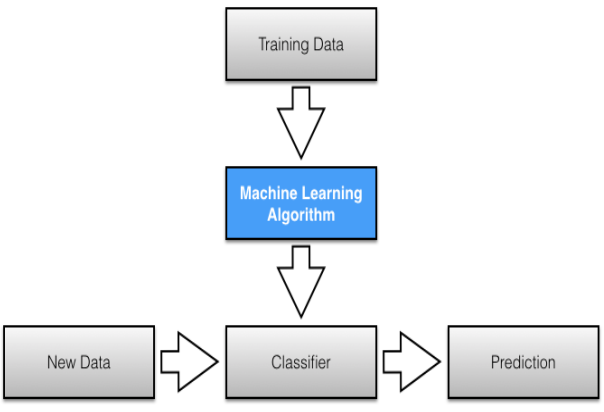

Fig. 1

B. Naive Bayes Classification

Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective *naive* comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well under this unrealistic assumption . Especially for small sample sizes, naive Bayes classifiers can outperform the more powerful alternatives.

Being relatively robust, easy to implement, fast, and accurate, naive Bayes classifiers are used in many different fields. Some examples include the diagnosis of diseases and making decisions about treatment processes , the classification of RNA sequences in taxonomic studies , and spam filtering in fake product reviews in shopping cart. However, strong violations of the independence assumptions and non-linear classification problems can lead to very poor performances of naive Bayes classifiers. We have to keep in mind that the type of data and the type problem to be solved dictate which classification model we want to choose. In practice, it is always recommended to compare different classification models on the particular dataset and consider the prediction performances as well as computational efficiency.

In the following sections, we will take a closer look at the probability model of the naive Bayes classifier and apply the concept to a simple toy problem. Later, we will use a publicly available SMS (text message) collection to train a naive Bayes classifier in Python that allows us to classify unseen messages as spam or ham.

1) Posterior Probabilities

In order to understand how naive Bayes classifiers work, we have to briefly recapitulate the concept of Bayes' rule. The probability model that was formulated by Thomas Bayes (1701-1761) is quite simple yet powerful; it can be written down in simple words as follows:
posterior probability=conditional probability·prior probabilityevidenceposterior probability=conditional probability·prior probabilityevidence

Bayes' theorem forms the core of the whole concept of naive Bayes classification. The *posterior probability*, in the context of a classification problem, can be interpreted as: "What is the probability that a particular object belongs to class ii given its observed feature values?" A more concrete example would be: "What is the probability that a person has diabetes given a certain value for a pre-breakfast blood glucose measurement and a certain value for a post-breakfast blood glucose measurement?"
P(diabetes|xi),xi=[90mg/dl,145mg/dl]P(diabetes|xi),xi=[90 mg/dl,145mg/dl]
Let

- xixi be the feature vector of sample i,i∈{1,2,...,n}i,i∈{1,2,...,n},
- ωjωj be the notation of class j,j∈{1,2,...,m}j,j∈{1,2,...,m},
- and P(xi|ωj)P(xi|ωj) be the probability of observing sample xixi given that is belongs to class ωjωj.

The general notation of the posterior probability can be written as
P(ωj|xi)=P(xi|ωj)·P(ωj)P(xi)P(ωj|xi)=P(xi|ωj)·P(ωj)P(xi)
The objective function in the naive Bayes probability is to maximize the posterior probability given the training data in order to formulate the decision rule.

2) Class-conditional Probabilities

One assumption that Bayes classifiers make is that the samples are *i.i.d.*
The abbreviation *i.i.d.* stands for "independent and identically distributed" and describes random variables that are independent from one another and are drawn from a similar probability distribution. Independence means that the probability of one observation does not affect the probability of another observation (e.g., time series and network graphs are not independent). One popular example of *i.i.d.* variables is the classic coin tossing: The first coin flip does not affect the outcome of a second coin flip and so forth. Given a fair coin, the probability of the coin landing on "heads" is always 0.5 no matter of how often the coin if flipped.

An additional assumption of naive Bayes classifiers is the *conditional independence* of features. Under this *naive* assumption, the *class-conditional probabilities* or (*likelihoods*) of the samples can be directly estimated from the training data instead of evaluating all possibilities of xx. Thus, given a dd-dimensional feature vector x the class conditional probability can be calculated as follows:

P(x|ωj)=P(x1|ωj)·P(x2|ωj)·…·P(xd|ωj)=∏k=1dP(xk|ωj)P(x|ωj)=P(x1|ωj)·P(x2|ωj)·…·P(xd|ωj)=∏k=1dP(xk|ωj)
Here, P(x|ωj)P(x|ωj) simply means: "How likely is it to observe this particular pattern xx given that it belongs to class ωjωj?" The "individual" likelihoods for every feature in the feature vector can be estimated via the maximum-likelihood estimate, which is simply a frequency in the case of categorical data:
P^(xi|ωj)=Nxi,ωjNωj(i=(1,...,d))P^(xi|ωj)=Nxi,ωjNωj(i=(1 ,...,d))

- Nxi,ωjNxi,ωj: Number of times feature xixi appears in samples from class ωjωj.
- NωjNωj: Total count of all features in class ωjωj.

3) Prior Probabilities

In contrast to a frequentist's approach, an additional *prior probability* (or just *prior*) is introduced that can be interpreted as the *prior belief* or *a priori* knowledge.

posterior probability=conditional probability·prior probabilityevidenceposterior probability=conditional probability·prior probabilityevidence

In the context of pattern classification, the prior probabilities are also called *class priors*, which describe "the general probability of encountering a particular class."

C. Multinomial Naive Bayes

1) Term Frequency

A alternative approach to characterize text documents — rather than binary values — is the *term frequency (tf(t, d))*. The term frequency is typically defined as the number of times a given term *t* (i.e., word or token) appears in a document *d* (this approach is sometimes also called *raw frequency*). In practice, the term frequency is often normalized by dividing the raw term frequency by the document length.

normalized term frequency=tf(t,d)ndnormalized term frequency=tf(t,d)nd

where

- tf(t,d)tf(t,d): Raw term frequency (the count of term tt in document dd).
- ndnd: The total number of terms in document dd.

The term frequencies can then be used to compute the maximum-likelihood estimate based on the training data to estimate the class-conditional probabilities in the multinomial model:

P^(xi|ωj)=∑tf(xi,d∈ωj)+α∑Nd∈ωj+α·VP^(xi|ωj)=∑tf(xi,d∈ωj)+α∑Nd∈ωj+α·V

where

- xixi: A word from the feature vector xx of a particular sample.
- ∑tf(xi,d∈ωj)∑tf(xi,d∈ωj): The sum of raw term frequencies of word xixi from all documents in the training sample that belong to class ωjωj.
- ∑Nd∈ωj∑Nd∈ωj: The sum of all term frequencies in the training dataset for class ωjωj.
- αα: An additive smoothing parameter (α=1α=1 for Laplace smoothing).
- VV: The size of the vocabulary (number of different words in the training set).

The class-conditional probability of encountering the text xx can be calculated as the product from the likelihoods of the individual words (under the *naive* assumption of conditional independence).

P(x|ωj)=P(x1|ωj)·P(x2|ωj)·…·P(xn|ωj)=∏i=1mP(xi|ωj)P(x|ωj)=P(x1|ωj)·P(x2|ωj)·…·P(xn|ωj)=∏i=1mP(xi|ωj)

2) Term Frequency - Inverse Document Frequency (Tf-idf)

The *term frequency - inverse document frequency (Tf-idf)* is another alternative for characterizing text documents. It can be understood as a weighted *term frequency*, which is especially useful if stop words have not been removed from the text corpus. The Tf-idf approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents. Although Tf-idf is most commonly used to rank documents by relevance in different text mining tasks, such as page ranking by search engines, it can also be applied to text classification via naive Bayes.

Tf-idf=tfn(t,d)·idf(t)Tf-idf=tfn(t,d)·idf(t)

Let tfn(d,f)tfn(d,f) be the normalized term frequency, and idfidf, the inverse document frequency, which can be calculated as follows

idf(t)=log(ndnd(t)),idf(t)=log⁡(ndnd(t)),

where

- ndnd: The total number of documents.
- nd(t)nd(t): The number of documents that contain the term tt.

D. *Overview of Scikit* ***Learn***

Scikit learn is a library used to perform machine learning in Python. Scikit learn is an open source library which is licensed under BSD and is reusable in various contexts, encouraging academic and commercial use. It provides a range of supervised and unsupervised learning algorithms in Python. Scikit learn consists popular algorithms and libraries. Apart from that, it also contains the following packages:

- NumPy
- Matplotlib
- SciPy (Scientific Python)

To implement Scikit learn, we first need to import the above packages. If you are not familiar with these libraries, you can have a look at my previous blogs on Numpy and Matplotlib. You can download these two packages using the command line or if you are using PyCharm, you can directly install it by going to your setting in the same way you do it for other packages.

Next, in a similar manner, you have to import Sklearn. Scikit learn is built upon the SciPy (Scientific Python) that must be installed before you can use Scikit-learn. You can refer to this website to download the same. Also, install Scipy and wheel package if it's not present, you can type in the below command:

pip install scipy

Pandas is used for data manipulation, analysis and cleaning. Python pandas is well suited for different kinds of data, such as:

- Tabular data with heterogeneously-typed columns
- Ordered and unordered time series data
- Arbitrary matrix data with row & column labels
- Unlabelled data
- Any other form of observational or statistical data sets

E. Count Vectorizer:

CountVectorizer uses to collect simple text documents to build a vocabulary of known words.

IV. RESULTS AND DISCUSSION

We obtain a confusion matrix and draw a plot named heatmap()

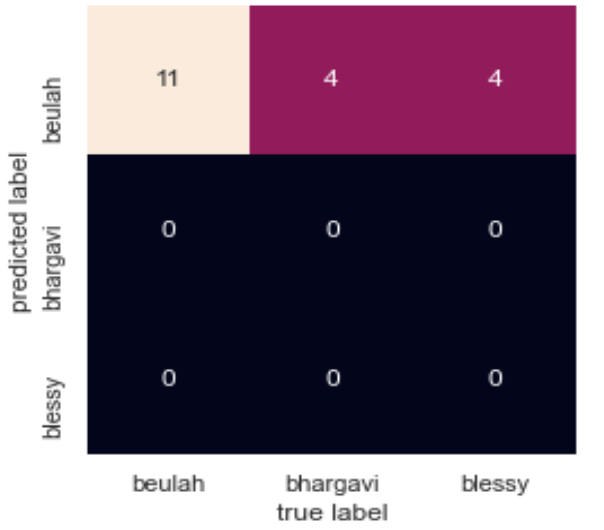

Fig. 2

V. CONCLUSION AND FUTURE SCOPE

I build a shopping cart website using python and django. Where i used sqlite database and some well formed packages. The methodology used is multinomial naive bayes algorithm for finding fake product comments. I took df.user as my testing data set and df.body as my training dataset. It checks same user how many times posted the comments. By observing the confusion matrix, we observed that list of comments has fake comments. I found the repeated user commenting on same data item. We can further find the semantically fake comments like abused words etc.

REFERENCES

[1] fengXu. Mixture of latent multinomial naive Bayes classifier., **p. 4, 2018**

[2] AmiliaFitr, V. Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm. *Information systems*. **2018.**

[3] Rubtsova, Y. Constructing a Corpus for Sentiment Classification Training. *Software & Systems*. **2018.**

[4] Asatani, P. K. (2018). Text classification. *World Symposium on Communication Engineering, WSCE* **2018**.

[5] Qin, F. Application and research of multi_label Naïve Bayes Classifier. *https://ieeexplore.ieee.org/xpl/conhome/6338367/proceeding*. IEEE. **2012.**

[6] Uehara. An implementation of electronic shopping cart on the Web system using component-object technology. *Proceedings Sixth International Workshop on Object-Oriented Real-Time Dependable Systems*. IEEE. **2002.**

[7] Karunarathna, K. . A Fully Functional Shopping Mall Application -- SHOPPING EYE. *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*. IEEE. **2014.**

[8] ying, m. . A commodity search system for online shopping based on ontology and web mining. *International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things 2014*. IEEE. **2014.**

AUTHORS PROFILE

N. Bhargavi pursed Bachelor of Technology from Acharya nagarjuna university, guntur in 2018 and doing Master of Technology from andhra university. Her main research work focuses on , Big Data Analytics, Data Mining.

