WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Text Classification System Using Naïve Bayes Algorithm

Gaurav Singh[1], Mayank Upadhyay[2], Umang Sharma[3], Shahrukh Hussain[4], Ujjwal Jain[5]

[1,2,3,4]Student, Dr. Akhilesh Das Gupta Institute of Technology and Management

[5]*Assistant Professor, Department of Information Technology, Dr. Akhilesh Das Gupta Institute of Technology and Management*

Abstract –

Today Machine Learning and Artificial Intelligence has emerged out as booming technology which target at performing things that earlier or traditionally require manual or human interference. In the area of Artificial Intelligence, Machine Learning has proved itself in building robust automated system which will be when once trained then the model is made when machine learns from the data which was fed for training just like humans and on getting any new similar data the machine will automatically predict the nature, value or category of this new data. Today employing various algorithms Machine Learning has become a very important part of any human lifestyle, and its applications lies in all the areas including Ed Tech, Healthcare, Industrial, Manufacturing and many more. The day-to-day efficiency of many tasks has been rapidly increasing to a very great extent, and achieving this level of efficiency could not have been possible without using Machine Learning and Artificial Intelligence. Thus ifdiscussing one such important applications of Machine Learning is Text Classification. Text Classification is commonly also known as text tagging and text categorization. It is the process of classifying the text into several categories they belong to. As for instance, Cricket will come under the category of Sports, and stuff like that. We have used Machine Learning Classifier namely Naïve Bayes classifier. The data set which we are using for building Text Classification system is Natural Language Processing with Disaster Tweets. In this work we have used pandas, numpy libraries, for text pre-processing we have used NLTK, string libraries for model building we have used sklearn and elaborately discussed how using all these libraries and dataset how an algorithm is proposed that will classify text much better. Thus, after getting a clear view of various techniques tools available an algorithm is devised that can be implemented for building a text classification system.

Key Words: Text Classification System, Machine Learning, Naïve Bayes classifier, Natural language processing.

1 INTRODUCTION

Today, Text Classification and analysis has become a very important and one of the major application in the field of Machine Learning. Though there is one big challenge which still most of the algorithmics face is that almost all Machine Learning algorithms works only on numeric data but the input we are getting is string data, and we can not give our input data directly, so we have to convert it into numeric data. Naïve Bayes classifiers are majorly used for text analysis and classification machine learning problems.[1]

In this work we will dive deep into how Naïve Bayes works, how we an fit the data into a model after transforming them, and implement text classification system with better results.

Naïve Bayes Classifier

Naïve Bayes Classifiers are known to the a collection of various classification machine learning algorithms which are based on Bayes' Theorem. This theorem is not a single algorithm, but it is meant to the combination of various algorithms mathematically which all share a very basic and common set of rules or principles, that is every pair of feature which is taken into consideration is completely independent of each other in any aspect.[2]

The dataset is divided into two main parts that are:

1. Feature Matrix / set of inputs
2. Target Matrix / output column

Feature Matrix

This is the set of input matrix that is the features of an item or data from the dataset that consists of all the rows or vector we give to the model. In this each row or vector contains the value of dependent features.

Target Matrix

It is the output matrix or the target or the response matrix that contains the value of the output of that particular row of feature matrix. It basically consist of the actual classification for that particular row or set of features.

Naïve Bayes Assumptions

The Naïve Bayes algorithm work on this assumption that are very essential for the computation of the model and to make appropriate distinctive prediction, that is each feature or the input variable which belong to the same class makes an[3]:

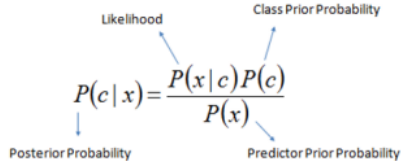
1. Independent
2. Equal

Contribution to each other in their outcome.

Few of these things that can be deduce from these assumptions is that these assumptions are not correct or would be appropriate when we are talking about the real-world scenarios. Thus, also these assumptions due to which the features are independent of each other are often not meet and this is the reason the classifier has been stated as "Naïve". The reason for this is that it assumes something that cannot match the real problem and might not be true always.

Bayes' Theorem

Bayes' Theorem compute the probability of an event occurring given the probability of another event when that has already been take place or occurred[8]. Mathematically, it can be written as:



$$P(c|X) = P(x_1|c)\times P(x_2|c)\times\cdots\times P(x_n|c)\times P(c)$$

Fig -1: Bayes Theorem Mathematical Formula [8]

Where,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

How Naïve Bayes Classifier works

We can re write the Bayes' theorem like this. Where we are trying to find the probability of y to occur when X event has already been done.[3]

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Fig -2: Bayes Theorem [8]

Now converting these parameters according to our need, we can call y as the class variable which will the outcome of all the feature taken into consideration and variable X is the given parameters/features, x is given by

$$X = (x_1, x_2, x_3, ..., x_n)$$

Fig -3 Feature/Row matrix

X is the vector which consists of x1,x2,x3… xn which are the feature or the input parameters for the given data point.

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

Fig -4: Removing constants and inducing proportionality.

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remains static. Therefore, the denominator can be removed, and proportionality can be injected.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Fig -5: Taking Maximum of all possible outcomes.

Using the above function, we can obtain the class, given the predictors/features.[3]

The posterior probability **P(y|X)** can be calculated by first, creating a **Frequency Table** for each attribute against the target. Then, molding the frequency tables to **Likelihood Tables** and finally, use the Naïve Bayesian equation to calculate the posterior probability for each class.[4] The class with the highest posterior probability is the outcome of the prediction. Below are the Frequency and likelihood tables for all three predictors.

2 LITERATUREREVIEW

As discussed in the approach for applying Bayes theorem for the implementation of Naïve Bayes theorem. We can apply different types of Naïve Bayes algorithm and how it can be used. Son we have few different types of Naïve Bayes algorithms variation. First is Gaussian Naïve Bayes, This type of Naïve Bayes assumes that the variables used in the data provided is from the normal distribution. In case if existing values does not have this type of values we will convert it into normal distribution.[7] Second one is Multinomial Naïve Bayes. This is used when we want to extract all the unique words and then make a frequency table of how many times a particular word is existing then we use Multinomial Naïve Bayes. Third we have Bernoulli Naïve Bayes, This is specifically used when we have binary features. So where we earlier had frequency table , in this Naïve Bayes we will have binary distinction or features that will in zeros 0s or ones 1s.[4]

Thus now it boils down to what Naïve bayes , we want to use. This all depend upon what type of data we have and how exactly we want our data to perform. Thus inputs that we have decides what kind of Naïve Bayes we should apply. Such that, the efficiency and accuracy are both maintains a good score overall.

3 PROPOSEDWORK

3.1 Data

The Dataset which we have used for this work is Natural Language Processing wit Disaster Tweets Dataset. This is a very popular and widely used Dataset which is being taken from Kaggle. This dataset comes with various invariants which are of different sizes that is10k dataset, 100k dataset, 500k dataset, 800k dataset.[11] For your work we have used 10k dataset. In this data. The output is labelled as either 0 or 1 , which is "1" when the disaster is real so target is 1 and target is "0" when the tweet is about fake disaster. [5]

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

Fig -6: Distance Functions[5]

Importing the libraries/Dataset

Importing the essential libraries such as Numpy, Pandas, Matplotlib, Seaborn. The Natural Language Processing with disaster tweets dataset is loaded and read using pandas library. For text preprocessing, various libraries like string and nltk, which is natural language tool kit library is being used. For model building, most popular library sklearn library is being used.

Data carpentary/Cleaning

There are several columns that can have NULL values or some of rows have some fields as null values. So, we have to process the data, for that we can use inbuilt methods for this.[6]

Analysing Missing Data:

```
id              0
keyword        61
location     2533
text            0
target          0
dtype: int64
```

Number of words in a tweet*:*

The number of tweets about a real disaster turn out to be more than non disaster tweets.

The average number of words in a disaster tweet is 15.17 as compared to an average of 14.7 words in a non-disaster tweet.[9]

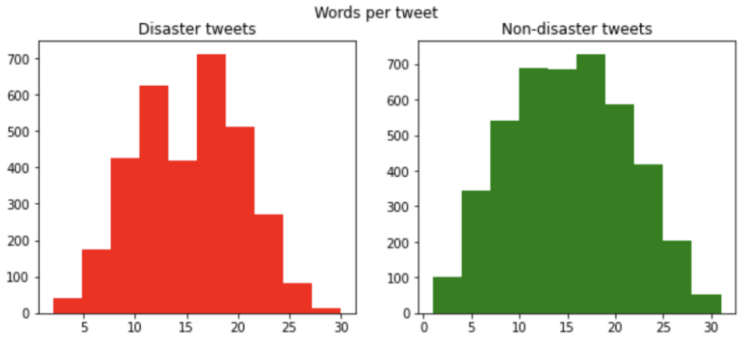


Fig -7: Number of words per tweet

3.1.4 Importing the libraries/Dataset

Before we start out building our model, we have to convert our dataset by removing , cleaning, applying lemmatization.

Clean Text

Remove unnecessary characters, URLs, punctuations, abbreviations in the data.

Stop-word Removal

Words like “He”, “She”, “it”, “I”, “am”, ”you”, ”a ”,…etc are unnecessary words and should be removed completely.

Stemming/Lemmatization

In his process we will , break down the words by slicing and removing any grammatical prefixes or suffixes.

Final Processing

After removing all unwanted data , we will get is this data which is shown as below.[10]

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

Fig -8: Original Textual Data (Part-1)

target	clean_text
1	deed reason earthquake may allah forgive u
1	forest fire near la ronge sask canada
1	resident ask shelter place notify officer evac...
1	people receive wildfire evacuation order calif...
1	get sent photo ruby alaska smoke wildfires pou...

Fig -9: Cleaned Text form Original Textual Data (Part-2)

```
              precision    recall  f1-score   support

           0       0.78      0.90      0.83       871
           1       0.82      0.66      0.73       652

    accuracy                           0.79      1523
   macro avg       0.80      0.78      0.78      1523
weighted avg       0.80      0.79      0.79      1523

Confusion Matrix: [[780  91]
 [224 428]]
AUC: 0.8445135694815211
```

Fig -10: predicted data

3.1.5 Running Naïve Bayes ML Algorithm

Naïve bayes classifier is known to be a classifier which is probabilistic In nature that uses Bayes Theorem, theorem that computes probability of an event based in some event already been occurred. We have applied Laplace Correction as the data set may have zero frequency issue. We can now remove correlated feature as the highly correlated features as it can be considered twice. Thus, we have built Naïve Bayes Classifier successfully[8]

4 RESULT ANDDISCUSSION

Here (fig-11) it can be seen that the text which we have cleaned is now have categorized into two labels either 0 or 1. Initially we had raw data which we have first converted into cleaned data that we can feed to the model. This model is further trained with the cleaned data applying Bayes Theorem. We have also seen it simple to understand and very much easier to build. It is one of the most scalable and most used Machine Learning algorithms for making Text Classification System. It has also proven to be very much faster when we compare to various other algorithm available to us in the field of Machine Learning. So today it has become a popular choice for further study and also for implementing Text classification System in real world scenarios.

	clean_text	target
0	happen terrible car crash	1
1	heard earthquake different city stay safe ever...	1
2	forest fire spot pond geese flee across street...	1
3	apocalypse light spokane wildfire	1
4	typhoon soudelor kill china taiwan	1
5	shake earthquake	1
6	probably still show life arsenal yesterday eh eh	0

Fig -11: Result of making the Text Classification System

```
#FITTING THE CLASSIFICATION MODEL using Naive Bayes(tf-idf)

nb_tfidf = MultinomialNB()
nb_tfidf.fit(X_train_vectors_tfidf, y_train)

#Predict y value for test dataset
y_predict = nb_tfidf.predict(X_test_vectors_tfidf)
y_prob = nb_tfidf.predict_proba(X_test_vectors_tfidf)[:,1]

print(classification_report(y_test,y_predict))
print('Confusion Matrix:',confusion_matrix(y_test, y_predict))

fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)
print('AUC:', roc_auc)
```

```
#Pre-processing the new dataset
df_test['clean_text'] = df_test['text'].apply(lambda x:
finalpreprocess(x)) #preprocess the data
X_test=df_test['clean_text']

#converting words to numerical data using tf-idf
X_vector=tfidf_vectorizer.transform(X_test)

#use the best model to predict 'target' value for the new dataset
y_predict = lr_tfidf.predict(X_vector)
y_prob = lr_tfidf.predict_proba(X_vector)[:,1]
df_test['predict_prob']= y_prob
df_test['target']= y_predict
final=df_test[['clean_text','target']].reset_index(drop=True)
print(final.head())
```

Fig -12: Code wise flow of making the actual prediction

5 CONCLUSION

Machine Learning till now has all in all proven itself that if exact calculated inputs given to it its model then it has the capability to automate various programming system that has been initially difficult for humans to do. These Machine Learning algorithms are known for learning and getting trained form the data and try to develop most appropriate mathematical graphs or clusters that in future if we give a new data to let ML model to predict, it can clearly predict or can compute what is the category and predictive value that this new data belong to. The similar is done in all Machine Learning algorithms just like in our case, we have used Naïve Bayes Classifier to classify the various texts we have got in the process of data collection and then we have successfully classified using the algorithm proposed in our work. The algorithm which is being proposed in the work can be improved by using more and more data sets and improving the conversion from text to numeric, applying Laplace smoothing factor to the test data, applying multiple combination technique like ensembling , bagging and boosting. In future, using the combination of these techniques much better text classifier can be deduced which will classify text much efficiently.[12]

REFERENCES

[1] HTTPS://TOWARDSDATASCIENCE.COM/TEXT-CLASSIFICATION-USING-NAIVE-BAYES-THEORY-A-WORKING-EXAMPLE-2EF4B7EB7D5A

[2] HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2017/09/NAIVE-BAYES-EXPLAINED/

[3] HTTPS://EN.WIKIPEDIA.ORG/WIKI/NAIVE_BAYES_CLASSIFIER

[4] HTTPS://MEDIUM.COM/ANALYTICS-VIDHYA/NLP-TUTORIAL-FOR-TEXT-CLASSIFICATION-IN-PYTHON-8F19CD17B49E

[5] HTTPS://WWW.KDNUGGETS.COM/2020/06/NAIVE-BAYES-ALGORITHM-EVERYTHING.HTML

[6] HTTPS://EN.WIKIPEDIA.ORG/WIKI/BAYES%27_THEOREM

[7] HTTP://OPENCLASSROOM.STANFORD.EDU/MAINFOLDER/DOCUMENTPAGE.PHP?COURSE=MACHINELEARNING&DOC=EXERCISES/EX6/EX6.HTML

[8] HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2021/03/INTRODUCTION-TO-NAIVE-BAYES-ALGORITHM/

[9] HTTP://CS229.STANFORD.EDU/NOTES-SPRING2019/CS229-NOTES2.PDF

[10] HTTPS://SCIKIT-LEARN.ORG/STABLE/MODULES/NAIVE_BAYES.HTML

[11] HTTPS://WWW.KAGGLE.COM/C/NLP-GETTING-STARTED/DATA

[12] HTTPS://INSIGHTIMI.WORDPRESS.COM/2020/04/04/NAIVE-BAYES-CLASSIFIER-FROM-SCRATCH-WITH-HANDS-ON-EXAMPLES-IN-R/