Naive Bayes Classifier

Presented By

Hetvi Shah

Heet Trivedi

Ramya Surati

Instructor: Dr. Yasser Alginahi

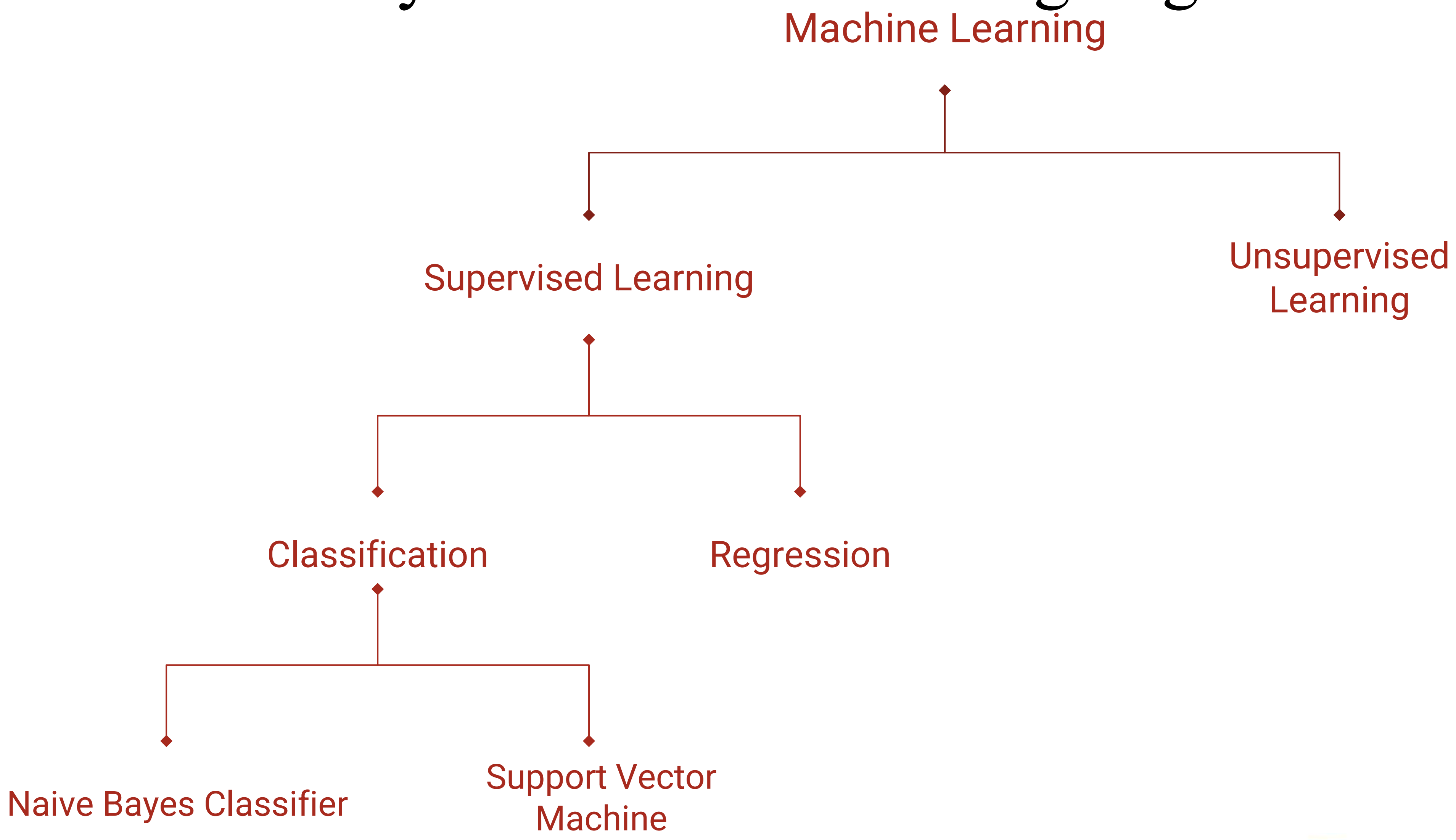
October 20th, 2023

University of Windsor

Agenda

- ❖ Introduction to Naive Bayes Classifier
- ❖ Bayes’ Theorem
- ❖ Conditional Probability
- ❖ Types of Naive Bayes Classifier
- ❖ Limitations
- ❖ Real Life Examples
- ❖ Code Implementation

Hierarchy of Machine Learning Algorithms



Introduction

- A classification algorithm in machine learning and statistics
- Used to estimate the probability of an input belonging to a particular category or class based on Bayes’ Theorem
- “Naive” assumption in Naive Bayes is about features being conditionally independent
- Calculates the probability of each class and then pick the one with the highest probability
- Based on Conditional Probability concept

Bayes' Theorem

Theorem for Conditional Probability

Likelihood of the Evidence given that the Hypothesis is True

Prior Probability of the Hypothesis

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Posterior Probability of the Hypothesis given that the Evidence is True

Prior Probability that the evidence is True

Source: https://tinyurl.com/ye5b8cw8

What is Conditional Probability ?

- Probability of an event occurring given that another event has already occurred
- Quantifies the likelihood of an event happening under a specific condition
- The probability of event A given probability of event B, which is denoted by $P(A/B)$

Example of Conditional Probability

Bag 1 (B1)

2 Red (R)
3 Black (B)

Bag 2 (B2)

4 Red (R)
3 Black (B)

$$P(B1) = \frac{1}{2}$$

$$P(B2) = \frac{1}{2}$$

$$P(R) = \frac{1}{2} \cdot \frac{2}{5} + \frac{1}{2} \cdot \frac{4}{7}$$

$$P(R/B1) = \frac{2}{5}$$

$$P(B1/R) = \frac{P(R/B1) \cdot P(B1)}{P(R)}$$

$$P(B1/R) = 0.42$$

Chain Rule

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

We can breakdown P(X|y) as:

$$P(X|y) = P(x_1, x_2, \dots, x_n|y)$$

$$= P(x_1|x_2, \dots, x_n, y) * P(x_2|x_3, \dots, x_n, y) \dots P(x_n|y)$$

So,

$$P(y|X) = \frac{P(x_1|y) * P(x_2|y) \dots P(x_n|y) * P(y)}{P(x_1) * P(x_2) \dots P(x_n)}$$

Spam Email Detection

Given: Number of *Not Spam* emails= 15
&
Number of *Spam* emails= 10

Examining various possibilities:
P(Dear|Not spam) = 8/34
P(Visit|Not spam) = 2/34
P(Dear|Spam) = 3/47
P(Visit|Spam) = 6/47

Table: Frequency of Words for Spam Detection

	Not Spam	Spam
Dear	8	3
Visit	2	6
Invitation	5	2
Link	2	7
Friend	6	1
Hello	5	4
Discount	0	8
Money	1	7
Click	2	9
Dinner	3	0
Total Words	34	47

Source: https://tinyurl.com/4c7kmwf9

Spam Email Detection

P(Not spam|Hello friend)= $\dfrac{P(Hello\,Friend Not\;Spam)\cdot P(Not\;spam)}{P(Hello\,Friend)}$

Ignoring the denominator,
P(Not spam|Hello friend)=P(Hello Friend|Not spam)* P(Not spam)

Technically, P(Not spam|Hello friend) should be 0, as this case (Hello friend does not exist in the dataset. We consider single words and the not complete sentences.

Here’s when Naive Bayes comes into picture which assumes that the features that we use to predict the target are independent.

Spam Email Detection

-Probability of Hello Friend being Not Spam:

P(Not spam|Hello friend)=P(Hello|Not spam)*P(Friend|Not spam)* P(Not spam)

P(Not spam|Hello friend)= $\frac{5}{34} \cdot \frac{6}{34} \cdot \frac{15}{25} = 0.0155$

-Probability of Hello Friend being Spam:

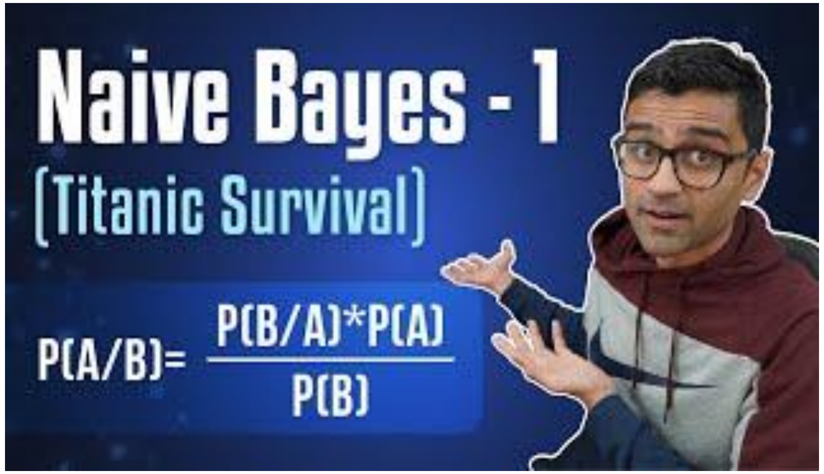
P(Spam|Hello friend)=P(Hello|Spam)*P(Friend|Spam)* P(Spam)

P(Spam|Hello friend)= $\frac{4}{47} \cdot \frac{1}{47} \cdot \frac{10}{25} = 0.00072$

Therefore, the message "Hello Friend" is NOT SPAM.

Youtube Video

This video provides an explanation of the fundamental principles behind Naive Bayes' Algorithm.



Types of Classifier in Naives

- Multinomial Naive Bayes

Uses Multinomial distribution (“n” values possible) of each feature i.e. discrete / count data.

- Bernoulli Naive Bayes

Features having binary attributes.

Types of Classifier in Naives

- Gaussian Naive Bayes

Features follow Gaussian Distribution i.e. features are continuous and assumed that they are distributed normally.

The conditional probability is given by :

Formula for Normal Distribution

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Source: https://tinyurl.com/3fbrww4m

Here,

μ - mean of the feature

σ - standard deviation of the feature

Limitations

- Independence Assumption
- Imbalanced Dataset
- Zero Frequency Problem

Zero Frequency Problem

Suppose some event is not there in our example then the model will assign zero probability for that event and also, any set of unseen events [5].
And when all the possibilities are multiplied, we will have a zero.

Example : Spam Detection

Solution to Zero Frequency Problem

Laplace Smoothing for Feature (w’)

$$P(w'|positive) = \frac{\text{number of reviews with w' and y = positive} + \alpha}{\text{N} + \alpha * K}$$

Here,

alpha represents the smoothing parameter,

K represents the number of dimensions (features) in the data, and

N represents the number of reviews with y=positive

Source: https://tinyurl.com/h56rtzcx

Real Life Applications

- Email Spam Detection
- Weather Forecast
- Sentiment Analysis
- News Article Classification
- Credit Scoring

Dataset and Code Implementation

Training Dataset: [Spam.csv]

Here the data set contains 2 columns which are “category” and “message”.

The column “category” contains values such as “Ham” and “Spam” to classify the email type whereas the column “message” contains the email body.

Jupyter Notebook: [Implementation of Multinomial naive Bayes Classifier for Email Spam Detection]

Code

```
import pandas as pd
df = pd.read_csv("spam.csv")
df.groupby('Category').describe()
df['spam']=df['Category'].apply(lambda x: 1 if x=='spam' else 0)
df.head()
```

Here, first we will load the dataset and explore the dataset through describe function to understand spam and ham emails with the help of Pandas library.

Then by using the lambda function we will add a column called spam to classify the email category.

Code

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.Message,df.spam ,testsize=0.25)
from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer()
X_train_count = v.fit_transform(X_train.values)
X_train_count.toarray()[:2]
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train_count,y_train)
```

Here, we are splitting the data into training set (75%) and test set (25%) and by the help of Count Vectorizer we are turning the text body into the frequency table.

Later, we are using the Multinomial Naïve Bayes classifier to train our model.

Code

```
emails = [    'Hey mohan, can we get together to watch footbal game tomorrow?',
'Upto 20% discount on parking, exclusive offer just for you. Dont miss this reward!’]
emails_count = v.transform(emails)
model.predict(emails_count)
X_test_count = v.transform(X_test)
model.score(X_test_count, y_test)
```

Here, we are taking the sample test email to understand the working of our model.

First, we need to the transform the email body into a count using the transform() of Count Vectorizer and then we will that transformed data into our model to predict the category of the email.

Moreover, we can use the pipeline to reduce the steps of transformation.

References

[1] B. Gamal, “Naïve Bayes Algorithm,” *Analytics Vidhya*, Apr. 11, 2021. https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2

[2] A. A. Awan and A. Navlani, “Naive Bayes classifier tutorial: With Python Scikit-Learn,” DataCamp, https://www.datacamp.com/tutorial/naive-bayes-scikit-learn

[3] L. S, “Gaussian naive Bayes algorithm for credit risk modelling,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2022/03/gaussian-naive-bayes-algorithm-for-credit-risk-modelling/

[4] R. Gandhi, “Naive Bayes classifier,” Medium, https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[5] “Naïve Bayes Algorithm: Everything you need to know,” KDnuggets, https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html

Thank you