

## Naive Bayes klassifikatori: ehtimollar nazariyasi asosida C# tilida modellashtirish

**Ismonjanova Sumbula Sherzod qizi**

Namangan Davlat Universiteti,

Fizika-matematika fakulteti,

Amaliy matematika yo‘nalishi,

2- bosqich talabasi

Email: [sumbula.sherzatovna@gmail.com](mailto:sumbula.sherzatovna@gmail.com)

**ANNOTATSIYA:** Ushbu maqolada ehtimollar nazariyasining amaliy yo‘nalishlaridan biri — Naive Bayes klassifikatori tahlil qilinadi. Klassifikatsiya masalalarida keng qo‘llaniluvchi ushbu usul Bayes teoremasi asosida ishlaydi va hodisalarning o‘zaro bog‘liqligini soddalashtirgan holda, ularni statistik asosda toifalash imkonini beradi. Maqola davomida Naive Bayes algoritmining nazariy asoslari bayon etildi, shuningdek, matnli ma’lumotlarni (masalan, e-mail xabarlarini) “spam” yoki “haqiqiy” deb ajratuvchi model C# dasturlash tilida ishlab chiqildi. Dasturiy modellashtirish orqali ushbu algoritmning ishlash prinsipi yoritildi va natijalari tahlil qilindi. Ilmiy yondashuv orqali ushbu usulning algoritmik strukturasi va amaliy samaradorligi ko‘rsatildi.

**Kalit so‘zlar:** Naive Bayes klassifikatori, ehtimollar nazariyasi, Bayes teoremasi, C# dasturlash tili, klassifikatsiya, ma’lumotlar tahlili, toifalash, spam filtrlash, statistik modellashtirish, sun’iy intellekt.

## Наивный байесовский классификатор: моделирование на языке программирования C# на основе теории вероятностей

Исмонжанова Сумбула Шерзод кизи

Наманганский государственный  
университет, Физико-математический

факультет, 2-й курс, направление

“Прикладная математика”

Email: [sumbula.sherzatovna@gmail.com](mailto:sumbula.sherzatovna@gmail.com)

**АННОТАЦИЯ:** В данной статье рассматривается один из прикладных методов теории вероятностей — наивный байесовский классификатор. Этот

широко используемый алгоритм классификации основан на формуле Байеса и позволяет определять принадлежность объектов к категориям, предполагая независимость признаков между собой. В рамках работы были рассмотрены теоретические основы наивного байесовского подхода, а также разработана программная модель на языке C#, классифицирующая текстовые данные (например, электронные письма) как «спам» или «нормальные». Программное моделирование позволило наглядно продемонстрировать принцип работы алгоритма и проанализировать его эффективность. Исследование подчеркивает важность алгоритмического мышления и применение вероятностных методов в современных задачах анализа данных.

**Ключевые слова:** наивный байесовский классификатор, теория вероятностей, формула Байеса, язык программирования C#, классификация, анализ данных, фильтрация спама, статистическое моделирование, искусственный интеллект, вероятностные алгоритмы.

**Naive Bayes Classifier: Modeling with C# Programming Language Based  
on  
Probability Theory**

Ismonjanova Sumbula Sherzod qizi  
Namangan State University,  
Faculty of Physics and Mathematics,  
2nd-year student, Applied  
Mathematics department

Email: [sumbula.sherzatovna@gmail.com](mailto:sumbula.sherzatovna@gmail.com)

**ABSTRACT:** This article explores one of the practical applications of probability theory — the Naive Bayes classifier. This widely used classification algorithm is based on Bayes' theorem and assumes independence among features, which allows for efficient categorization of data. The paper outlines the theoretical foundation of the Naive Bayes approach and presents a program model implemented in the C# programming language to classify text data (e.g., emails) as either “spam” or “legitimate.” The modeling process illustrates the algorithm's logic and enables evaluation of its classification accuracy. The research highlights the importance of

algorithmic thinking and the application of probabilistic methods in modern data analysis tasks.

**Keywords:** Naive Bayes classifier, probability theory, Bayes' theorem, C# programming language, classification, data analysis, spam filtering, statistical modeling, artificial intelligence, probabilistic algorithms.

Zamonaviy davrda axborot oqimining keskin ortib borishi sababli, katta hajmdagi ma'lumotlarni avtomatik tarzda tahlil qilish va ular asosida tez va aniq qarorlar qabul qilish zarurati kundan-kunga oshib bormoqda. Ayniqsa, matnli ma'lumotlarni toifalash — masalan, elektron pochta xabarlarini “spam” yoki “normal” deb ajratish — informatsion xavfsizlik, marketing va sun'iy intellekt sohalarida muhim o'rin tutadi.

Ehtimollar nazariyasining amaliy natijalaridan biri bo'lgan Naive Bayes klassifikatori — shunday masalalarni yechishda eng samarali va sodda algoritmlardan biri hisoblanadi. Ushbu usul Bayes teoremasi asosida tuziladi va har bir xususiyatning tasnif natijasiga ta'sirini mustaqil deb hisoblashga asoslangan. Garchi bu “soddalashtirilgan” taxmin bo'lsa-da, u ko'plab real vaziyatlarda yuqori aniqlikdagi natijalarni beradi.

Naive Bayes klassifikatori — bu Bayes teoremasiga asoslangan va xususiyatlarning (features) o'zaro mustaqilligiga tayanuvchi ehtimollik asosidagi klassifikatsiya algoritmidir. U “Naive” deb ataladi, chunki barcha kiruvchi belgilarning (atributlarning) mustaqil ekanligi taxmin qilinadi, bu esa real hayotda kamdan-kam hollarda to'liq haqiqat bo'ladi. Shunga qaramasdan, bu algoritim ko'plab klassifikatsion muammolarda juda samarali ishlaydi.

Bayes teoremasi quyidagicha yoziladi:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Bu yerda:

$P(A|B)$  — B hodisasi sodir bo'lgandan keyin A hodisasining ehtimoli

$P(B|A)$  — A hodisasi sodir bo'lganda B hodisasining ehtimoli



$P(A)$  — A hodisasining boshlang‘ich ehtimoli

$P(B)$  — B hodisasining umumiy ehtimoli

Agar bizda quyidagicha tasniflanadigan xususiyatlar bo‘lsa:

$$x = (x_1, x_2, \dots, x_n)$$

va biz bu  $x$  to‘plam qaysi sinfga tegishli —  $C_k$  — ekanini aniqlamoqchi bo‘lsak, Naive Bayes formulasi:

$$P(C_k | x_1, x_2, \dots, x_n) = \frac{P(C_k) \cdot \prod_{i=1}^n P(x_i | C_k)}{P(x)}$$

Ammo  $P(x)$  barcha sinflar uchun bir xil bo‘lgani uchun, biz faqat yuqori qiymatni topamiz:

$$\text{Classify}(x) = \arg \max [P(C_k) \cdot \prod_{i=1}^n P(x_i | C_k)]$$

Bu yerda:

- $P(C_k)$  — har bir sinfnig prior ehtimoli
- $P(x_i | C_k)$  — har bir atributning sinfga bog‘liq ehtimoli

Naive Bayes klassifikatori ko‘pincha matnni toifalash (text classification) kabi masalalarda qo‘llaniladi. Quyida biz  $C\#$  dasturlash tilida matnli ma‘lumotlar asosida xabarni “Spam” yoki “Normal” turiga ajratuvchi soddalashtirilgan model yaratamiz. Biz quyidagi bosqichlarni bajarishimiz kerak:

1. O‘quv ma‘lumotlari — “spam” va “normal” xabarlardan iborat matnlar to‘plami
2. Har bir sinf (spam/normal) uchun so‘zlar soni va ehtimollarini hisoblash
3. Kiruvchi yangi xabar uchun har bir sinfga Naive Bayes formulasi asosida ehtimol hisoblash
4. Ehtimollari eng katta sinfni natija sifatida chiqarish

using System.Collections.Generic;

using System.Linq;

class NaiveBayesClassifier

{

private Dictionary<string, int> spamWords = new Dictionary<string, int>();

private Dictionary<string, int> normalWords = new Dictionary<string,

int>();

private int spamCount = 0, normalCount = 0;



```
private int totalSpamWords = 0, totalNormalWords = 0;
```

```
public void Train(string text, string label)
```

```
{
```

```
    var words = text.ToLower().Split(' ');
```

```
    if (label == "spam")
```

```
    {
```

```
        spamCount++;
```

```
        foreach (var word in words)
```

```
        {
```

```
            if (!spamWords.ContainsKey(word))
```

```
                spamWords[word] = 0;
```

```
                spamWords[word]++;
```

```
                totalSpamWords++;
```

```
            }
```

```
        }
```

```
    else
```

```
    {
```

```
        normalCount++;
```

```
        foreach (var word in words)
```

```
        {
```

```
            if (!normalWords.ContainsKey(word))
```

```
                normalWords[word] = 0;
```

```
                normalWords[word]++;
```

```
                totalNormalWords++;
```

```
            }
```

```
        }
```

```
    }
```

```
public string Predict(string text)
```

```
{
```

```
    var words = text.ToLower().Split(' ');
```

```
    double pSpam = Math.Log((double)spamCount / (spamCount +  
normalCount));
```



```
double pNormal = Math.Log((double)normalCount / (spamCount +
normalCount));

foreach (var word in words)
{
    // Laplace smoothing
    double spamWordProb = Math.Log((spamWords.ContainsKey(word) ?
spamWords[word] + 1 : 1) / (double)(totalSpamWords + spamWords.Count));
    double normalWordProb =
Math.Log((normalWords.ContainsKey(word) ? normalWords[word] + 1 : 1) /
(double)(totalNormalWords + normalWords.Count));

    pSpam += spamWordProb;
    pNormal += normalWordProb;
}

return pSpam > pNormal ? "Spam" : "Normal";
}
}
```

**Misoldan foydalanish:**

```
class Program
{
    static void Main()
    {
        var classifier = new NaiveBayesClassifier();

        // Trening ma'lumotlari
        classifier.Train("pulingiz yutib oldingiz", "spam");
        classifier.Train("sizni bank kutmoqda", "spam");
        classifier.Train("dars jadvali o'zgardi", "normal");
        classifier.Train("bugun yig'ilish soat 9 da", "normal");

        // Sinov
```



```
string testText = "siz pulingizni yutib oldingiz";  
string result = classifier.Predict(testText);  
Console.WriteLine($"Xabar: \"{testText}\" → Natija: {result}");  
}  
}
```

Ushbu model real spam filtrlash tizimlarining asosiy g‘oyasini tushuntiradi va Naive Bayes formulasi orqali qaror qabul qilishni dasturiy tarzda ifodalaydi.

Yuqoridagi dasturiy modellashtirish orqali Naive Bayes klassifikatori qanday ishlashi aniq ko‘rsatildi. Dastlabki o‘quv ma‘lumotlari asosida har bir so‘zning har bir sinfga tegishli ehtimoli hisoblab chiqildi. Kiritilgan test xabari asosida algoritm ushbu ehtimollarni jamlab, qaysi sinfga ko‘proq mos tushishini aniqladi.

#### Muhim jihatlar:

**Soddalashtirish:** Naive Bayes barcha atributlarni mustaqil deb hisoblaydi. Bu real hayotda unchalik to‘g‘ri bo‘lmasa-da, natijalar ko‘pincha juda aniqlik bilan chiqadi.

**Laplace smoothing:** Ba‘zi so‘zlar mashq to‘plamda uchramagan bo‘lishi mumkin. Bu holatda nolga teng ehtimollarni oldini olish uchun Laplace usuli ishlatildi.

**Model tezligi:** Naive Bayes algoritmi oson va tez ishlaydi. Katta hajmdagi matnlar bilan ham samarali natijalar beradi.

Ushbu maqolada ehtimollar nazariyasining amaliy ko‘rinishi bo‘lgan Naive Bayes klassifikatori nazariy va amaliy jihatdan tahlil qilindi. Bayes teoremasi asosida ishlovchi ushbu usul C# dasturlash tilida modellashtirildi va matnli xabarlarini “spam” yoki “normal” sinflarga ajratish uchun soddalashtirilgan dastur yaratildi.

Tadqiqot natijalari shuni ko‘rsatdiki, Naive Bayes klassifikatori:  
samarali,  
tez ishlovchi,  
va haqiqiy ilovalarda qo‘llash uchun qulay vositadir.



Mazkur yondashuv ehtimollar nazariyasini chuqurroq o‘zlashtirish, dasturiy modellashtirish ko‘nikmalarini rivojlantirish va real hayotdagi axborot oqimlarini avtomatlashtirish imkonini beradi.

#### Foydalanilgan adabiyotlar

1. Ширяев А.Н. Теория вероятностей. — Москва: Физматлит, 2017.
2. Mitchell T. Machine Learning. — McGraw-Hill, 1997.
3. Sebastian Raschka, Vahid Mirjalili. Python Machine Learning. — Packt Publishing, 2019.
4. Jurafsky D., Martin J. Speech and Language Processing. — Pearson, 2020.
5. MSDN Documentation. C# Programming Guide.
6. Brownlee J. Machine Learning Algorithms. — Machine Learning Mastery, 2016.
7. [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)



---

# Research Science and Innovation House

