A NOVEL APPROACH TO APPLYING BAYES' THEOREM IN MACHINE LEARNING AND PROGRAMMING

Karimova U.
PhD in Mathematics
Azerbaijan Technical University, Baku, Azerbaijan
Akhundov R.
Student of the Faculty of Information Technologies and Telecommunications
Azerbaijan Technical University, Baku, Azerbaijan
Abdullayev K.
Doctor of Sciences in Economics, D.Sc. (Economics),
Institute of Economics of the Ministry of Science and Education of the Azerbaijan Republic, Baku, Azerbaijan
https://doi.org/10.5281/zenodo.21105817

Abstract

Probabilistic approaches constitute an essential component of Machine Learning, particularly in scenarios involving uncertainty and data-driven inference. Bayes' Theorem provides a principled framework for incorporating prior knowledge and updating beliefs as new data become available. This paper investigates the Naive Bayes algorithm from both theoretical and practical perspectives, emphasizing its mathematical foundation and applicability to real-world classification problems. Different variants of Naive Bayes, including Gaussian, Multinomial, Bernoulli, and Complement models, are analyzed with respect to their suitability for various data representations. The study focuses primarily on text classification tasks, where Naive Bayes is combined with TF–IDF-based feature extraction. Experimental evaluation is performed using the IMDB Movie Reviews dataset, and the results are compared with Logistic Regression and Support Vector Machine classifiers. The findings indicate that Naive Bayes delivers competitive performance with significantly lower computational cost, making it well suited for large-scale and real-time text processing applications. Despite its simplifying assumptions, the algorithm remains an effective and interpretable baseline for Machine Learning classification tasks.

Keywords: Machine Learning, Naive Bayes, Bayesian Inference, Text Classification, TF–IDF, Sentiment Analysis, IMDB Dataset

Introduction.

Machine Learning (ML) is one of the scientific fields that forms the core foundation of modern artificial intelligence systems. Over recent decades, the development of ML in both theoretical and practical domains has radically transformed the way humans work with data. Today, the application of Machine Learning models is rapidly expanding across numerous fields, including healthcare, education, finance, cybersecurity, natural language processing, robotics, and many others. The effective operation of these models largely depends on methodologies based on statistics and probability theory. Statistical analysis, modeling of uncertainty, and data-driven decision-making constitute the theoretical backbone of Machine Learning. The primary goal of Machine Learning is to extract patterns from data, learn from them, and provide accurate predictions about future events. This learning process involves not only the application of simple rules but also the correct calculation of probabilities and the discovery of hidden relationships within data. For this reason, statistical methods are regarded as an integral part of ML. In the analysis of large-scale data accumulated over time, probability distributions, statistical hypotheses, random variables, and Bayesian approaches play a crucial role. In reality, data are almost always observed under uncertainty. For example, in medical diagnostics, even if symptoms of a disease are observed in a patient, a definitive diagnosis can only be provided through probabilistic approaches. Probability-based models enable ML systems to understand the uncertain nature of data and to make mathematically justified decisions based on that data. Through these models, systems evaluate the likelihood that newly incoming data belong to particular categories by relying on previous observations and probability values.

Bayes' theorem is based on the calculation of conditional probabilities and enables the updating of prior knowledge as new information becomes available. In many modern ML methods especially probabilistic models, generative models, and tasks such as classification and segmentation—approaches based on Bayes' theorem are widely used. Bayesian-based methods allow models to be updated in accordance with constantly changing data environments in real-world scenarios. The Naive Bayes method is a simple yet extremely powerful classification algorithm based on Bayes' theorem. Although the assumption of complete independence among features is rarely satisfied in real life, this model often demonstrates very high performance in practice. Particularly in areas such as text classification, spam filtering, sentiment analysis, and medical diagnostics, the speed and efficiency of Naive Bayes have led to its widespread adoption. Today, the Naive Bayes model remains relevant in many real-world problems, including text classification, medical diagnostics, cybersecurity, social media analytics, and financial forecasting. The aim of this research is to comprehensively present the mathematical foundations of Naive Bayes

theory, explain its application principles in Machine Learning, comparatively analyze its performance in real-world problems, investigate its advantages and limitations, and, based on experimental results, determine the scenarios in which Naive Bayes is most effective.

Literature Review

The development of probability-based models in the field of Machine Learning and the formation of their theoretical foundations particularly the application of the Bayesian approach have emerged as the result of many years of research. The importance of the Bayesian approach in Machine Learning is especially related to theoretical interpretability, model evaluation, and the ability to make decisions under conditions of uncertainty. Kevin Murphy, in his book "Machine Learning: A Probabilistic Perspective", states that nearly all ML models can be explained through a Bayesian interpretation, which enhances the transparency of their prediction and decision-making mechanisms [7[. The paper investigates machine learning models for predicting early student dropout, trained using student data on personal characteristics and academic performance [6]. The novelty lies in applying explainable AI (XAI) methods to interpret these models. In author provides a systematic literature review (SLR) on how Artificial Intelligence (AI) and Machine Learning (ML) technologies are being applied to enhance software project management processes [1]. It consolidates research findings from existing literature to highlight major trends, opportunities, challenges, and research gaps in the integration of AI/ML within project management in software engineering. Research article [3] elucidates the key role of Mathematics that plays in development of artificial intelligence (AI) [3]. This research paper's contribution is to pointing out the mathematical rules that must be at their peak in order to construct an ideal artificial intelligence model [3]. The paper proposes a novel Explainable Artificial Intelligence (XAI) framework combining UNet-based semantic segmentation with Bayesian machine learning to classify brain tumors (BTs) using MRI images [4]. This integrated approach aims to support automated, interpretable, and accurate diagnosis from medical scans. In article discussed how Artificial Intelligence (AI) and Machine Learning (ML) are reshaping materials research and outlines a visionary roadmap for India's role in this transformation, emphasizing strategic directions for future research and innovation in the materials domain [9]. It highlights the growing importance of data-driven science in accelerating discovery, optimizing processes, and enabling new materials development paradigms. Paper [10] reviews and systematically analyzes the educational applications of deep learning in the following six aspects: learning effect prediction, educational game, learning recommendation, automatic scoring, assisted teaching and medical education [10]. In research author focused on addressing the problem of class imbalance in Naive Bayes classification, which is a common issue in real-world datasets where one class (or a few classes) significantly outnumbers others [5]. Class imbalance can degrade classifier performance, particularly for minority classes that are often most important in applications like fraud detection, medical diagnosis, and rare-event prediction. The study investigates how various factors influence risk assessment (RA) outcomes in scientific research projects (SRPs) by applying the Naive Bayesian algorithm, with an enhanced version called Tree-Augmented Naive Bayes (TANB) to better model dependencies among features [2]. The book is a comprehensive guide to machine learning using Python, aimed at students, researchers, and practitioners [8]. It focuses on building practical ML systems using widely adopted Python libraries such as NumPy, pandas, scikit-learn, TensorFlow, and Keras.

Mathematical Statement of the Problem.

The application of probability based models in Machine Learning requires a strong mathematical foundation. The effectiveness and practical efficiency of models such as Naive Bayes largely depend on the correct calculation of probabilities, a clear understanding of Bayes' theorem, and the assumption of feature independence. In this section, the concept of probability, Bayes' theorem, and the Naive independence assumption are discussed. The concept of probability forms the basis of statistical analysis and Machine Learning. Probability provides a mathematical framework for measuring the likelihood of events occurring. This concept is applied in both classical and frequentist approaches. In classical probability, it is assumed that all possible outcomes have equal likelihood. This approach is based on ideal conditions and is used to determine theoretical probabilities. In contrast, frequentist probability is based on the relative frequency of observed outcomes.

Bayes' theorem enables the updating of prior probabilities based on new data. The theorem is expressed as follows:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \quad (1)$$

The concept of Bayesian inference enables a model to learn dynamically as new data become available. The fundamental assumption of the Naive Bayes method is that features are conditionally independent of one another. Although this assumption is rarely fully satisfied in real-world scenarios, it often yields very strong performance in practical applications.

Simplified probability model:

$$P(X_1, X_2, \dots, X_n \mid C) = \prod_{i=1}^{n} P(X_i \mid C) \quad (2)$$

This assumption allows the conditional probability of each feature to be calculated independently, significantly increasing the computational efficiency of the model. The primary objective of Naive Bayes classification models is to determine, based on probability theory, which class a given observation (feature vector $x = (x_1, x_2, \dots, x_n)$) belongs to. The probabilities are computed for all possible classes, and the class with the highest probability is selected.

Optimal class selection (Decision Rule):

$$\hat{y} = arg \max_{C_k} P(C_K) \prod_{i=1}^{n} P(x_i \mid C_k) \quad (3)$$

where $P(C_K)$ – the prior probability of class C_k, $P(x_i \mid C_k)$ -the likelihood, i.e., the probability of feature $x_i$given class C_k, $\prod$ the product of feature probabilities based on the conditional independence assumption.

Using this approach, it is determined which class both occurs more frequently a priori and better matches the observed features of the given instance. Naive Bayes models are formulated in different variants depending on the type of data being used.

Gaussian Naive Bayes (for normally distributed features)

This model assumes that each feature follows a normal (Gaussian) distribution within each class. It is commonly applied in medical diagnostics, biometric measurements, the Iris dataset (flower measurements), sensor data, and technical measurement applications. The probability density function of the distribution is given by:

$$P(x_i \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{\left(-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}\right)} \quad \textbf{(4)}$$

Multinomial Naive Bayes (count-based – particularly for text data)

This model is especially used in text classification tasks, as it is based on word counts (frequencies).

$$P(x_i \mid C_k) = \frac{(x_i)!}{x_{i1}! \ldots x_{in}!} \prod_j \theta_{kj}^{x_{ij}} \quad (5)$$

where x_{ij} - the number of times a word appears in a given document; θ_{kj} - the relative frequency (proportion) of the word within a class. It is applied in spam detection, sentiment analysis, and news categorization.

Bernoulli Naive Bayes (for binary features)

In this model, each feature is represented as a binary value, taking either 0 or 1.

$$P(x_i \mid C_k) = \theta_k^{x_i}(1 - \theta_k)^{1 - x_i} \quad (6)$$

It is applied for determining whether a word appears in a text or not, for binary features, and in simple NLP models.

Complement Naive Bayes

In text classification tasks with class-imbalanced datasets, Complement Naive Bayes often yields better results than the Multinomial variant. It is particularly recommended for datasets containing rare classes.

The training of a Naive Bayes model consists of three main stages.

a. Calculation of the prior probability : the prior is simply the proportion of each class

b. Calculation of likelihood probabilities: this step varies depending on the type of Naive Bayes model used:f or Gaussian Naive Bayes, the mean and variance $(\mu_k, \sigma_k^2)$are estimated for each feature within each class; for Multinomial Naive Bayes, word counts (frequencies) are computed; for Bernoulli Naive Bayes, the frequencies of binary values (0/1) are determined.

c. Laplace Smoothing: to prevent unseen words in the text from being assigned zero probability:

$$P(x_i \mid C_k) = \frac{n_{i,k} + \alpha}{N_k + \alpha V} \quad (7)$$

where

$$n_{i,k} = n(x_i, C_k), N_k = \sum_j n_{j,k.}$$

V - the size of the vocabulary, α — the smoothing parameter (for Laplace smoothing, $\alpha = 1$), $n_{i,k}$ — the number of occurrences of feature $x_i$in class C_k, N_k − the total number of observations in class C_k. This approach makes Naive Bayes more stable and robust.

d. Testing phase (Inference Stage): during testing, the model computes the probability for a given observation.

e. Probability computation

$$P(C_k) \prod_i P(x_i \mid C_k)$$

f. Class selection: The class with the highest probability is selected.

g. Computations in log-space: Since the product of very small probabilities may approach zero, the logarithm of the probabilities is taken:

$$\log P(C_k) + \sum_i \log P(x_i \mid C_k)$$

This approach resolves the underflow problem, improves computational stability, and plays a significant role in handling high-dimensional text data.

Naive Bayes in Text Classification and Information Retrieval Systems

Text classification is one of the most widely used application areas of the Naive Bayes algorithm. Since the model is based on the independence assumption, it treats each word or attribute in textual data as independent of others. This assumption simplifies the process and significantly accelerates computation.The Bag-of-Words (BoW**)** model is one of the most classical approaches for structuring text data. In this method: text is tokenized into individual words, word frequencies are calculated, each document is represented in a vectorized form. When Naive Bayes is combined with the BoW model, probability distributions are constructed based on word frequencies, and posterior probabilities are computed for each class.

TF–IDF (Term Frequency–Inverse Document Frequency) feature extraction allows rare but informative words to be represented with higher weights in the model. The combination of TF–IDF with Naive Bayes demonstrates strong performance particularly in news article categorization, classification of academic papers by research fields, topic analysis of social media posts.

Spam and Malicious Content Detection One of the most successful classical applications of Naive Bayes is spam filtering systems. This model relies on the higher occurrence of certain words (e.g., *free, win, limited offer, prize*) in spam emails, is sufficiently fast for real-time filtering in large-scale email streams, provides high accuracy with minimal computational requirements. For these reasons, this algorithm has been widely used for many years in systems such as Gmail, Yahoo Mail, and Outlook.

Sentiment Analysis Sentiment analysis is an important area of Machine Learning aimed at identifying emotional polarity in texts positive, negative, or neutral. Naive Bayes performs particularly well in this domain because word distributions differ across emotional categories, and probabilistic approaches yield high-quality classification even for short and noisy texts. It is applied in customer review analysis (Amazon, TripAdvisor, AliExpress), public opinion measurement on social media platforms (Twitter/X sentiment analysis), brand reputation monitoring, political analysis (e.g., identifying public opinion trends during election periods).

Applications of Naive Bayes in Medicine and Bioinformatics Probability-based decision-making methods have been used in medicine since the mid-20th century. Naive Bayes is especially effective in medical modeling due to the following characteristics:stable results even with small sample sizes, the independence assumption aligns well with many diagnostic systems, provides interpretable outcomes for physicians through posterior probabilities. Naive Bayes is applied in: disease probability estimation (e.g., calculating the likelihood of diagnoses such as diabetes, heart disease, or early-stage cancer based on symptoms), genetic marker analysis (in bioinformatics, gene expression data are high-dimensional, and Naive Bayes excels due to its simplicity and speed), classification of laboratory test results (determining disease probabilities based on blood test parameters). The interpretability of Naive Bayes is a major advantage for medical experts, as the reasoning behind the model's decisions can be clearly explained.

Practical Experiments and Results

4.1. Dataset Selection (IMDB Movie Reviews)

In this experiment, the IMDB Movie Reviews dataset was used. The dataset contains positive (1) and negative (0) reviews (table 1).

Dataset characteristics:

Training set: 25,000 samples

Test set: 25,000 samples

The texts consist of real movie reviews

Words are indexed numerically

Table 1.

Data Acquisition and Preprocessing Using the IMDB Dataset

```
# ========================================
# 6.1. DATASET SELECTION (IMDB REVIEWS)
# Technology: TensorFlow / Keras
# Library: tensorflow.keras.datasets.imdb
# Purpose: Loading the IMDB Movie Review Dataset
# ========================================

from tensorflow.keras.datasets import imdb

# Keep only the 20,000 most frequently used words
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=20000)

# Word-to-index mapping
word_index = imdb.get_word_index()

# Reverse index (index-to-word mapping)
reverse_index = {v: k for k, v in word_index.items()}

# Function to convert numerical sequences back to text
def decode(review):
    return " ".join([reverse_index.get(i - 3, "?") for i in review])

# Decode training reviews into readable text
decoded_train = [decode(x) for x in X_train]

# Decode test reviews into readable text
decoded_test = [decode(x) for x in X_test]
```

Note: The decode function converts numerical indices into words, enabling the data to be processed in text format.

Data Processing

Text preprocessing and vectorization: TfidfVectorizer was used, english stopwords were removed, bigrams (two-word combinations) were taken into account

Table 2.

Preprocessing and Feature Engineering of Review Texts

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(
    max_features=20000,
    stop_words="english",
    ngram_range=(1,2)
)

X_train_vec = vectorizer.fit_transform(decoded_train)
X_test_vec = vectorizer.transform(decoded_test)
```

Source: Prepared by the author using the IMDB Movie Reviews Dataset and the TF-IDF vectorization technique implemented in Scikit-Learn.

Construction of Naive Bayes Models

The Multinomial NB and Bernoulli NB models were implemented:

Table 3.

Sentiment Classification Using Multinomial and Bernoulli Naive Bayes

```
from sklearn.naive_bayes import MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score

mnb = MultinomialNB()
bnb = BernoulliNB()

mnb.fit(X_train_vec, y_train)
bnb.fit(X_train_vec, y_train)

mnb_pred = mnb.predict(X_test_vec)
bnb_pred = bnb.predict(X_test_vec)
```

Source: prepared by the author using the IMDB Movie Reviews Dataset and implemented with Python programming language and the Scikit-Learn machine learning library.

Accuracy scores:

Table 4.

Classification Accuracy of Naive Bayes Models on the IMDB Dataset

```
print("MultinomialNB Accuracy:", accuracy_score(y_test, mnb_pred))
print("BernoulliNB Accuracy:", accuracy_score(y_test, bnb_pred))
```

```
MultinomialNB Accuracy: 0.85316
BernoulliNB Accuracy: 0.8536
```

Source: Prepared by the author using the IMDB Movie Reviews Dataset and implemented with Python programming language and the Scikit-Learn machine learning library.

Evaluation of Results

Metrics: Accuracy, Precision, Recall, F1-score

Table 5.

Performance Evaluation of Naive Bayes Models Using Accuracy, Precision, Recall, and F1-Score

```
from sklearn.metrics import classification_report

print("\nMultinomialNB Report:")
print(classification_report(y_test, mnb_pred, digits=4))

print("\nBernoulliNB Report:")
print(classification_report(y_test, bnb_pred, digits=4))
```

```
MultinomialNB Report:
              precision    recall  f1-score   support

           0     0.8411    0.8709    0.8557     12500
           1     0.8661    0.8354    0.8505     12500

    accuracy                         0.8532     25000
   macro avg     0.8536    0.8532    0.8531     25000
weighted avg     0.8536    0.8532    0.8531     25000


BernoulliNB Report:
              precision    recall  f1-score   support

           0     0.8410    0.8721    0.8563     12500
           1     0.8672    0.8351    0.8508     12500

    accuracy                         0.8536     25000
   macro avg     0.8541    0.8536    0.8535     25000
weighted avg     0.8541    0.8536    0.8535     25000
```

Source: Prepared by the author based on the IMDB Movie Reviews Dataset. Performance metrics (Accuracy, Precision, Recall, and F1-Score) were calculated using the Classification Report function provided by the Scikit-Learn library in Python.

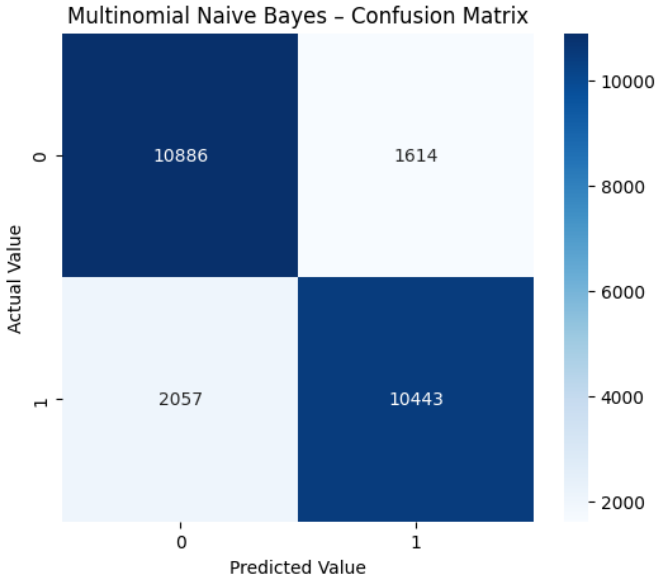

Figure 1. Confusion Matrix of the Multinomial Naive Bayes Classifier

Source: *Prepared by the author using the IMDB Movie Reviews Dataset and implemented with Python programming language, TensorFlow/Keras, Scikit-Learn, Matplotlib, and Seaborn libraries.*

The confusion matrix demonstrates that the Multinomial Naive Bayes classifier achieved strong performance on the IMDB Movie Reviews Dataset (Figure 1). The model correctly classified 10,886 negative reviews and 10,443 positive reviews. However, 1,614 negative reviews were incorrectly classified as positive, while 2,057 positive reviews were misclassified as negative. The relatively high number of correctly classified instances compared to misclassifications indicates that the model is effective in distinguishing between positive and negative sentiments. Overall, the confusion matrix confirms the reliability of the Multinomial Naive Bayes classifier for sentiment analysis tasks.

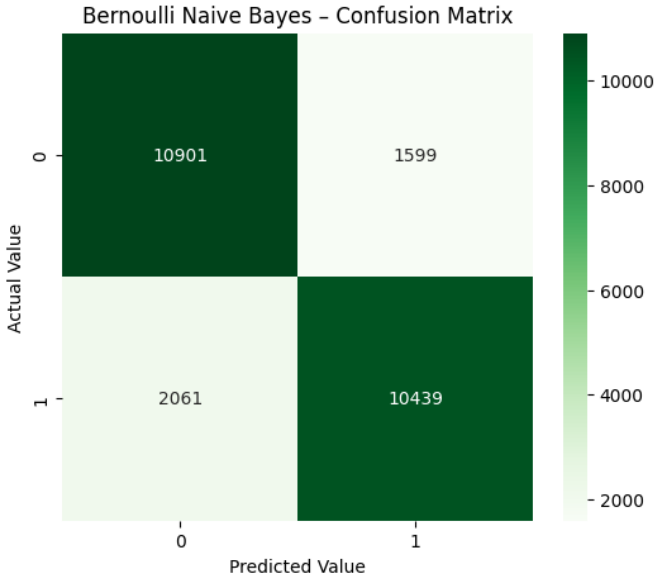

Figure 2. Confusion Matrix of the Bernoulli Naive Bayes Classifier

Source: *Prepared by the author using the IMDB Movie Reviews Dataset and implemented with Python programming language, Tensor Flow/Keras, Scikit-Learn, Matplotlib, and Seaborn libraries.*

The confusion matrix of the Bernoulli Naive Bayes classifier also shows a high level of classification accuracy (Figure 2). The model correctly identified 10,901 negative reviews and 10,439 positive reviews. Misclassification occurred in 1,599 negative reviews predicted as positive and 2,061 positive reviews predicted as negative. Compared with the Multinomial Naive Bayes classifier, the Bernoulli model achieved slightly better recognition of negative reviews while maintaining a similar performance on positive reviews. The results indicate that the Bernoulli Naive Bayes classifier is also a suitable approach for text-based sentiment classification.

Comparison with Other Models. Comparison with Logistic Regression and LinearSVC (SVM) models

Table 6.

Performance Assessment of Logistic Regression and Support Vector Machine Models on the IMDB Dataset

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC

logreg = LogisticRegression(max_iter=300)
svm = LinearSVC()

logreg.fit(X_train_vec, y_train)
svm.fit(X_train_vec, y_train)

logreg_pred = logreg.predict(X_test_vec)
svm_pred = svm.predict(X_test_vec)

print("Logistic Regression Accuracy:", accuracy_score(y_test, logreg_pred))
print("SVM Accuracy:", accuracy_score(y_test, svm_pred))
```

```
Logistic Regression Accuracy: 0.883
SVM Accuracy: 0.87072
```

Source: Dataset: IMDB Movie Review Dataset (Maas et al., 2011), loaded via tensorflow.keras.datasets.imdb.

Presentation of Results in Tabular Form

Table 7.

Performance Comparison of Machine Learning Models Based on Accuracy

Model	Accuracy
MultinomialNB	0.85316
BernoulliNB	0.8536
Logistic Regression	0.883
SVM	0.87072

Source: Prepared by the author based on the IMDB Movie Reviews Dataset , loaded via tensorflow.keras.datasets.imdb. Accuracy values were computed using the Scikit-Learn library in Python.

Interpretation

The results presented in Table 7 show a comparative evaluation of four machine learning models applied to the IMDB Movie Reviews sentiment classification task. The Logistic Regression model achieved the highest accuracy (0.88300), indicating its superior ability to generalize patterns in text data compared to the other models.

The Support Vector Machine (SVM) also demonstrated strong performance with an accuracy of 0.87072, making it the second-best performing model in this experiment. Both Naive Bayes variants, MultinomialNB (0.85316) and BernoulliNB (0.85360), showed slightly lower but still competitive results, reflecting their effectiveness as baseline probabilistic classifiers for text classification tasks.

Overall, the findings suggest that discriminative models (Logistic Regression and SVM) outperform generative models (Naive Bayes) in sentiment classification on the IMDB dataset, particularly in terms of accuracy.

Naive Bayes models provide fast and efficient results, making them particularly suitable for the classification of large-scale text datasets. Due to their simple probability-based approach, they require relatively low computational resources and are highly effective in real-time applications. More complex linear models, such as Logistic Regression and Support Vector Machines (SVM), may outperform Naive Bayes in terms of accuracy in certain scenarios. However, these models typically demand greater computational resources and involve longer training times compared to Naive Bayes. Different variants of the Naive Bayes algorithm should be selected based on the characteristics of the dataset. Multinomial Naive Bayes is well suited for count-based or TF–IDF features, while Bernoulli Naive Bayes performs more stably on binary features and may achieve marginally better performance in such cases. For simple and time-efficient applications, Naive Bayes represents an ideal choice. Nevertheless, in situations where higher accuracy is required, it is recommended to conduct comparative evaluations with models such as Logistic Regression or SVM.

Conclusion.

This work provided a detailed examination of the Naive Bayes classifier from a probabilistic and computational perspective. By formally analyzing its reliance on conditional independence assumptions and Bayesian inference principles, the study clarified how likelihood estimation, prior modeling, and posterior maximization contribute to efficient decision-making in high-dimensional feature spaces. The use of log-probability computations and Laplace smoothing further ensured numerical stability and robustness during inference. Experimental results obtained from the IMDB Movie Reviews dataset demonstrated that Multinomial and Bernoulli Naive Bayes models achieve competitive classification performance when combined with TF–

IDF-based feature representations. While Logistic Regression and Support Vector Machines exhibited marginally higher accuracy, they required substantially greater computational resources and longer training times. In contrast, Naive Bayes maintained linear computational complexity with respect to the number of features and samples, highlighting its scalability advantages. From a practical standpoint, the findings confirm that Naive Bayes is well suited for large-scale text classification tasks where efficiency, interpretability, and rapid model deployment are critical. Future research directions may include relaxing the independence assumption through structured Bayesian models or integrating Naive Bayes with ensemble-based approaches to improve predictive performance without significantly increasing computational overhead.

References:

1. Ali, U., Naseer, M. Artificial intelligence and machine learning in enhancing software project management processes: A systematic literature review. *Autom Softw Eng*, 2026. Volume 33, (31) . https://doi.org/10.1007/s10515-025-00578-6

2. Dong, X., Qiu, W. A case study on the relationship between risk assessment of scientific research projects and related factors under the Naive Bayesian algorithm. *Sci Rep. 2024.* 14. https://doi.org/10.1038/s41598-024-58341-y

3. Kaur, R., Sharma, D. Application and Significance of Mathematics in Artificial Intelligence. In: Reddy, V.S., Wang, J., Chetti, P., Reddy, K.T.V. (eds) Soft Computing and Signal Processing. ICSCSP 2024. Lecture Notes in Networks and Systems, vol 1221. 2025. Springer, Singapore. https://doi.org/10.1007/978-981-96-0924-6_53

4. Lakshmi, K., Amaran, S., Subbulakshmi, G. *et al*. Explainable artificial intelligence with UNet based segmentation and Bayesian machine learning for classification of brain tumors using MRI images. *Sci Rep,* 2025. 15. https://doi.org/10.1038/s41598-024-84692-7

5. Luo, Y. Improvement and Research of Naive Bayes Classification Based on Unbalanced Data Sets. In: S. Shmaliy, Y. (eds) 8th International Conference on Computing, Control and Industrial Engineering (CCIE 2024). CCIE 2024. Lecture Notes in Electrical Engineering, 2024. vol 1253. Springer, Singapore. https://doi.org/10.1007/978-981-97-6937-7_3

6. Minullin, D.A., Gafarov, F.M. Analyzing Machine Learning Models Based on Explainable Artificial Intelligence Methods in Educational Analytics. *Autom. Doc. Math. Linguist.* 2024. 58 (Suppl 3), p.115. . https://doi.org/10.3103/S0005105525700189

7. Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. 2012. MIT Press.

8. Raschka, S., & Mirjalili, V. *Python Machine Learning*. 2017. Packt Publishing

9. Sen, P. Artificial intelligence transforming materials research: possible road ahead for India. *Proc.Indian Natl. Sci. Acad.* 2025. https://doi.org/10.1007/s43538-025-00641-6

10. Zhang, F., Wang, X. & Zhang, X. Applications of deep learning method of artificial intelligence in education. *Educ Inf Technol.* 2025. 30, p.1563–1587. https://doi.org/10.1007/s10639-024-12883-w