DOI: http://dx.doi.org/10.26483/ijarcs.v17i1.7399

ISSN No. 0976-5697

Volume 17, No. 1, January-February 2026

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

EFFECT ON NAIVE BAYES CLASSIFIER OF FEATURE TRANSFORMATION: AN EMPIRICAL STUDY ON DIVERSE DATASETS

Dr. Deepanshu Mishra
Research Scholar, Department of Statistics,
University of Lucknow, Uttar Pradesh, India

Dr. Ashok Kumar
Assistant Professor, Department of Statistics,
University of Lucknow, Uttar Pradesh, India,

Abstract: Naïve Bayes classifiers are widely used for classification tasks because of their simplicity and for its efficiency along with strong theoretical foundation. However, their operational limitations stem from both the independence assumption concerning features and data distribution characteristics. The research investigates multiple techniques which apply feature transformations to improve both accuracy and stability of Gaussian Naïve Bayes classifiers. A comprehensive assessment of log transformation together with polynomial feature expansion on benchmark datasets consisting of Iris, Wine, Diabetes and Breast Cancer datasets occurred in this research. An evaluation of these transformations used cross-validation accuracy as well as precision, recall and F1-Score to assess their effects. Experimental outcomes show that distributions with skewed data can become more suitable for classification after log transformation while polynomial feature expansion improves feature characteristics to produce enhanced decision boundaries. In the Breast Cancer dataset Gaussian Naive Bayes with log-transformation outperformed the baseline model by producing superior recall and F1-Score results that are essential for medical diagnostics. The Wine dataset showed improved accuracy outcomes when using polynomial feature expansion because this method effectively extracts information about feature interactions which enhance class separability. Relevant preprocessing methods applied to Naïve Bayes classification yield improved predictive performance according to the results of this study. Future investigations will expand these transformation methods across different probabilistic classifiers as well as study their results within high-dimensional and imbalanced datasets.

Keywords: Naïve Bayes, Feature Transformation, Log Transformation, Polynomial Features, Classification, Machine Learning, Performance Evaluation

1. INTRODUCTION

The Naive Bayes classifier represents a vital machine learning algorithm which bases its operations on the probabilistic foundation of Bayes' theorem for updating beliefs through evidence. Naive Bayes classifier builds upon Bayes' theorem which serves as a statistical method to enrich beliefs through evidence input [1]. The foundational component of Naive Bayes receives its strength from this theorem which enables the classification algorithm to compute the posterior class probability from the observed features.

The key assumption of Naive Bayes involves feature independence because this simplification allows easier computation of conditional probabilities. The algorithm becomes more efficient because of this unrealistic assumption particularly in high-dimensional datasets. The computational efficiency of Naive Bayes stems from its independence assumption as explained by [2]. The algorithm functions effectively when processing numerous features. The algorithm achieves high efficiency when used for text classification and document categorization, where the feature space can be vast.

Naive Bayes proves to be highly effective in a wide range of classification tasks even though it relies on the strong independence assumption. The robust characteristics of this algorithm include performing well with inadequate training data and unreliable or missing input information according to [3]. The algorithm shows robustness because each feature's independent modeling allows it to reduce the effects of individual data points. Its efficient performance combined with reliable effectiveness made Naive Bayes popular across multiple domains. The application of Naive Bayes for text classification was thoroughly studied by [4] through investigation of document categorization using word frequency statistics. [5] developed Naive Bayes as a spam detection system which utilized the algorithm's strength to detect indicators of spam emails. Among medical diagnosis applications Naive Bayes delivers beneficial results through its handling of categorical data and provision of probabilistic outcomes for patient outcome prediction [6].

The Naive Bayes algorithm succeeds in managing both categorical and numerical data types because of its flexible design. The algorithm uses Gaussian distributions with continuous data but relies on multinomial distributions together with Bernoulli distributions with discrete data. Its flexible nature alongside simplicity and efficiency has made Naive Bayes a fundamental machine learning algorithm that people use to analyze abundant data but with limited computational resources [7].

Mathematically, the Naive Bayes classifier can be expressed as follows:

Given a feature vector $x = (x_1, x_2, ..., x_n)$ and a set of classes $C = \{(c_1, c_2, ..., c_n\}$, the classifier identifies the class ci that maximizes the posterior probability P(ci|x). The probability equation can be written as according to Bayes' theorem [8]:

$$P(c_i \mid x) = \frac{P(x \mid c_i)P(c_i)}{P(x)}$$

where:

- $P(c_i \mid x)$ is the probability of class ci given the features x.

- $P(x \mid c_i)$ is the likelihood of features x given class c_i .
- $P(c_i)$ is the prior probability of class c_i.
- $P(x)$ is the marginal likelihood of features x.

The 'naive' assumption of feature independence simplifies the likelihood term:

$$P(x \mid c_i) = \prod_{J=1}^{N} P(x_j \mid c_i)$$

This assumption allows for the separate computation of the likelihood of each feature given the class, significantly reducing computational complexity.

GNB makes the assumption that all features in the data set will follow a Gaussian distribution pattern. Thus, the likelihood of a feature x_j given class c_i is:

$$P(x_j|c_i) = \frac{1}{\sqrt{2\pi\sigma_{c_i}^2}} exp\left(-\frac{(x_j - \mu_{c_i})^2}{2\sigma_{c_i}^2}\right)$$

where μ_{c_i} and $\sigma_{c_i}^2$ are the mean and variance of feature x_j in class c_i, respectively.

The decision rule for classifying a feature vector x into class c_i is then given by:

$$\hat{y} = arg\ max_{c_i \in C} P(c_i) \prod_{J=1}^{N} P(x_j \mid c_i)$$

where $\hat{y}$ is the predicted class.

Despite its strong assumptions, Naive Bayes classifiers, particularly the Gaussian variant, have demonstrated remarkable efficacy across a wide range of applications. Their efficiency, ease of implementation, and often surprising effectiveness make them an important tool in the machine learning.

2. METHODOLOGY

2.1 Datasets:

We utilize a variety of datasets from the scikit-learn library and OpenML repository to evaluate our approach. These include:

 Iris: A small with 150 samples and 4 features and 3 classes, well-structured, multiclass dataset. It is widely used dataset.

 Wine: A continuous, multiclass dataset with 178 samples with 13 features and 3 classes. It is more complex than Iris dataset.

 Diabetes: A continuous dataset. It has 442 samples and 10 features.

 Breast Cancer: A binary dataset, widely adapted for classification. It has 569 samples, 30 features along with 2 classes.

2.2 Feature Scaling:

In this study, *StandardScaler* was applied to standardize the features. This transforms the features to have zero mean and unit variance.

The formula for standard scaling is:

$$X_{scaled} = \frac{x - \mu}{\sigma}$$

Where μ is the mean of features and σ is standard deviation of features.

2.3 Feature Transformation Techniques:

- **Polynomial Feature Expansion:** We use *PolynomialFeatures* from scikit-learn to create polynomial features, capturing non-linear relationships between features.
- **Logarithmic Transformation:** We apply a logarithmic transformation to features with positive values to normalize skewed distributions.

Polynomial Features:

For a given dataset with n features $(x_1, x_2, \ldots, x_n)$, the polynomial features up to degree d were generated by:

$$PolynomialFeatures(X) = \{x_i^a * x_j^b * \ldots * x_k^c \mid 0 \le a + b + \ldots + c \le d\}$$

where X is the original feature set, and a,b,c are non-negative integers representing the powers of the features.

Logarithmic Transformation of Features:

For a feature x with positive values, the logarithmic transformation was applied as:

$$x' = log(x - min(x) + 1)$$

where x' is the transformed feature. Here, the addition of 1 was used to avoid taking log of zero values.

2.4 Experimental Setup:

This methodology assesses the performance of Gaussian Naive Bayes (GNB) on the above datasets, first by splitting the data into training and testing sets with stratified k-fold cross-validation (*train_test_split, StratifiedKFold*) to ensure representative class distribution across folds. A baseline GNB model is established after standardizing the features (*StandardScaler*), followed by evaluations using *classification_report* and *cross_val_score*. Subsequently, the methodology explores feature engineering by incorporating polynomial features (*PolynomialFeatures*) and log transformations (custom function), each followed by standardization and GNB model training. Polynomial expansion was taken as 2 degrees in the study as higher degree may lead to overfitting. The performance of these models, including precision, recall, and F1-Score, is then compared to the baseline, providing insights into the impact of feature engineering on GNB classification accuracy. The research technique aligns with typical machine learning operations for assessing models alongside feature preparation.

2.5 Evaluation Metrics:

We used Accuracy, Precision and Recall together with F1-Score to judge the performance of Naïve classifier. The four metrics deliver unique evaluations of predictive ability [9]. Each performance measure receives individual explanation followed by an explanation of its role in classification assignments below.

2.5.1. Accuracy

Definition:

Accuracy measures the ratio between successfully predicted instances and all the instances present in the dataset. For classification evaluation purposes Accuracy stands as a primary metric used by researchers [10].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- **TP (True Positives)** = It is the positive cases in dataset that are correctly classified as positive
- **TN (True Negatives)** = It is the negative cases in dataset that are correctly classified as negative cases
- **FP (False Positives)** = It is negative case in dataset that are incorrectly classified as positive cases
- **FN (False Negatives)** = It is the positive cases in dataset that are incorrectly classified as negative cases

Accuracy is an effective metric when the dataset is balanced i.e., has nearly equal numbers of positive and negative samples [11]. However, in imbalanced datasets, it can be misleading. For example, in a dataset where 95% of the samples belong to the negative class, a model predicting all samples as negative would achieve 95% accuracy but fail to detect any positive instances [12].

2.5.2. Precision (Positive Predictive Value)

Definition:

Precision measures the proportion of correctly predicted positive cases out of all predicted positives. It focuses on the reliability of positive predictions [10].

$$Precision = \frac{TP}{TP + FP}$$

High precision means that when the model predicts a sample as positive, it is correct most of the time. Low precision indicates a high number of false positives, meaning the model frequently misclassifies negative instances as positive [13]. Precision is important in applications where false positives are costly, such as medical diagnosis for e.g., misdiagnosing a healthy person as sick may lead to unnecessary treatments [14].

2.5.3. Recall (Sensitivity or True Positive Rate, TPR)

Definition:

Recall measures how many of the actual positive cases were correctly identified by the model. It focuses on capturing as many true positives as possible [10].

$$Recall = \frac{TP}{TP + FN}$$

A model with high recall demonstrates its capability to identify most of the existing positive instances. A model exhibits low recall which indicates high numbers of false negatives since it fails to detect positive outcomes. Recall is important in scenarios where missing positive cases is costly. For example, scenario such as cancer detection, where failing to identify a cancerous tumor can have severe consequences [13].

2.5.4. F1-Score

Gaussian Naïve Bayes With Log Transformation	CV Accuracy	Precision	Recall	F1 Score
Iris	92.38	97.94	97.78	97.77
Wine	97.6	96.62	96.3	96.28
Diabetes	55.98	61.63	60.9	61.21
Breast Cancer	94.73	94.76	94.74	94.75

Definition:

The F1-score calculates its value by applying the harmonic mean method to precision values combined with recall values for achieving balanced measurement resolution. F score is especially useful in cases where there is an imbalance between precision and recall [15].

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1-score helps balance the trade-off between precision and recall, ensuring that neither metric is disproportionately emphasized [16]. If a model has a high F1-score, it means that the classifier maintains both high precision and recall. If a model has a low F1-Score then it suggests poor performance in at least one of these aspects. It is particularly useful when working with imbalanced datasets, as it provides a better representation of classifier performance than accuracy alone [9].

3. RESULTS AND DISCUSSION

Baseline Gaussian Naive Bayes	CV Accuracy	Precision	Recall	F1 Score
Iris	91.43	97.94	97.78	97.77
Wine	95.97	100	100	100
Diabetes	55.01	61.95	62.41	62.16
Breast Cancer	92.97	93.55	93.57	93.56

Gaussian Naive Bayes with Polynomial Features	CV Accuracy	Precision	Recall	F1 Score
Iris	92.38	97.94	97.78	97.77
Wine	95.13	98.24	98.15	98.15
Diabetes	49.5	61.36	60.9	61.21

Breast Cancer	92.72	92.38	92.4	92.38

The below tables compare the performance of Baseline Gaussian Naive Bayes (GNB) with two feature transformation techniques:

Log Transformation and Polynomial Features. We compared the performance across five widely used datasets: Iris, Wine, Diabetes, and Breast Cancer. The performance of the models was evaluated using Cross-Validation Accuracy (CV Accuracy), Precision and Recall along with F1-Score.

Iris Dataset

Gaussian Naive Bayes demonstrates consistent performance when used to classify the Iris dataset across its three evaluation scenarios. For the Iris dataset the baseline model reaches 91.43% CV Accuracy yet both log transformation and polynomial features enhance it to 92.38%. Throughout every configuration of models Precision, recall and F1-Score values stabilized at 97.94% and 97.78% and 97.77% respectfully. Iris dataset performance exhibits little change between raw and transformed features because its current state performs well without further transformation.

Wine Dataset

The Wine dataset sees notable improvements with log transformation, as CV accuracy increases from 95.97% (baseline) to 97.6%, though polynomial transformation slightly reduces it to 95.13%. The baseline model achieves perfect precision and recall (100%), but log transformation balances performance more effectively, yielding 96.62% precision, 96.3% recall, and 96.28% F1-Score. Polynomial features show a minor improvement over the baseline but do not outperform log transformation. This indicates that logarithmic scaling helps better distribute the features for classification, whereas polynomial features do not provide significant benefits.

Diabetes Dataset

The Diabetes dataset remains the most challenging across all models, with relatively low classification performance. The baseline model achieves 55.01% accuracy, which only marginally improves to 55.98% with log transformation, while polynomial features cause a decline to 49.5%. Precision, recall, and F1-Score also see minimal improvement with log transformation but drop with polynomial features. This suggests that the data distribution of the Diabetes dataset does not align well with Gaussian assumptions, and complex transformations do not enhance separability. More advanced preprocessing or alternative classifiers may be needed for better performance.

Breast Cancer Dataset

Gaussian Naive Bayes performs well on the Breast Cancer dataset, with the baseline achieving 92.97% accuracy. Log transformation further enhances performance, increasing accuracy to 94.73%, while polynomial features slightly reduce it to 92.72%. Precision, recall, and F1-Score follow a similar pattern, with log transformation improving all metrics. This indicates that logarithmic scaling helps in this dataset by refining feature representation, while polynomial transformations may introduce unnecessary complexity, leading to minor performance degradation.

Overall Trends

- Log transformation generally improves performance, particularly for Wine and Breast Cancer datasets, while its impact on Diabetes is marginal.
- Polynomial Features show mixed results, sometimes leading to a decline in performance (e.g., Diabetes dataset).
- Baseline Gaussian Naive Bayes performs well, but feature transformations help refine results.

Log Transformation generally enhances performance across datasets, especially for Wine, and Breast Cancer datasets. Polynomial Feature transformation does not consistently improve results and can degrade performance as seen in Diabetes. For simpler datasets like Iris, transformations do not make a significant difference. For the challenging Diabetes dataset, all models struggle, but Log Transformation slightly helps. Wine dataset benefits the most from transformations, with Log Transformation achieving the best performance (97.6% accuracy).

4. DISCUSSION

The results of this study highlight the significant impact of feature transformation techniques on the performance of Gaussian Naïve Bayes classifiers. By applying log transformation and polynomial feature expansion, we observed notable variations in classification accuracy, precision, recall, and F1-Score across different datasets. This section explores the analyzed findings with detailed evaluations of their importance and review of their advantages and constraints.

4.1. Impact of Log Transformation on Naïve Bayes Performance

Log transformation proves helpful for skewed distribution datasets because it normalizes distribution of features while reducing the impact of outlier samples. The implementation of log transformation enhanced Naïve Bayes classifier performance throughout most datasets while achieving its most significant benefits with higher recall and F1-Score in the Breast Cancer dataset. This is a crucial finding because high recall is very important in various cases such as in medical diagnostics, ensuring that positive cases are identified with minimal false negatives. The analysis on data showed that log transformation applied to the data led to better model performance because it effectively works when features present wide range distribution. The implementation of feature transformation achieved minimal improvement when used for analysis on the Diabetes dataset therefore demonstrating that feature manipulation alone cannot solve all class imbalance and complex attribute interaction problems.

4.2. Effectiveness of Polynomial Feature Expansion

The addition of polynomial feature expansion generates new interaction terms between features so that the classifier becomes able to detect previously undetectable non-linear associations between features. Our research demonstrates that polynomial transformations provided notable improvement in results in both Wine data sets because they produced enhanced accuracy alongside precision improvements. This suggests that these datasets contain strong feature interactions, which the Naïve Bayes model can leverage when provided with transformed features. The findings indicate that these datasets have robust feature interactions which enhance the Naive Bayes model when it receives transformed features. The polynomial transformation showed no substantial benefits in predicting the Diabetes dataset probably because of its natural class distribution and minimal feature correlations. Additionally, while polynomial expansion increased accuracy in some datasets, it also increased computational complexity, which may not always be desirable for large-scale applications.

4.3. Comparison Between Different Feature Transformations

Both log transformation and polynomial feature expansion demonstrated dataset-dependent effects. While log transformation helped mitigate skewness and improved performance in cases where numerical features varied widely, polynomial feature expansion was more effective in scenarios where feature interactions played a critical role in classification. However, neither transformation universally outperformed the baseline across all datasets, suggesting that the choice of transformation should be data-specific.

4.4. Limitations and Challenges

Despite the improvements observed, certain challenges remain. First, Naïve Bayes classifiers rely on the assumption of feature independence, which is often unrealistic in real-world datasets. While feature transformations help improve performance, they do not fully address this limitation. Second, class imbalance remains a challenge, particularly in datasets like Diabetes, where neither transformation significantly enhanced performance. Future work should explore oversampling techniques, cost-sensitive learning, or hybrid models to address this issue. Lastly, computational cost is a concern, particularly for polynomial feature expansion, which increases the number of features and, consequently, the processing time and memory usage.

Overall, the results confirm that feature transformations can significantly enhance Naïve Bayes classification performance by improving feature representation and mitigating some of the limitations of the feature independence assumption. Future research will focus on extending this work to high-dimensional datasets, investigating the impact of additional transformation techniques (such as kernel-based methods), and exploring their integration with ensemble learning approaches to enhance classification robustness further.

5. FUTURE WORK

The research showed that feature transformation methods produce better performance with Gaussian Naïve Bayes classifiers yet multiple investigation fields remain available. Future research can extend this work in the following directions:

- Investigate other feature transformation techniques, such as wavelet transformations or kernel transformations.
- Explore the integration of feature selection methods to reduce dimensionality and improve generalization.
- Conduct a more in-depth analysis of the impact of hyperparameter tuning on the performance of transformed Naive Bayes models.
- Test on a wider range of datasets, including real-world datasets with complex feature dependencies.
- Compare the performance of the transformed Naive Bayes models with other classification algorithms.

By addressing these areas, future research can further refine Naïve Bayes classification, making it more adaptable to diverse data types and real-world scenarios.

REFERENCES

[1] Jaynes, E. T. (2003). Probability theory: The logic of science. Cambridge university press.

[2] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

[3] Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

[4] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.

[5] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98-105).

[6] Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in medicine, 23(1), 89-109.

[7] Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29, 103-130.

[8] Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. arXiv preprint arXiv:1410.5329.

[9] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

[10] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437.

[11] Kelleher, J. D., Mac Namee, B., & D'arcy, A. (2020). Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT press.

[12] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

[13] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).

[14] Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets. J Inf Eng Appl, 3(10).

[15] Rijsbergen, V. (1979). Information retrieval; Butterworth, 1978. J. librariansh., 11, 237.

[16] Sasaki, Y. (2007). The truth of the F-measure. Teach tutor mater, 1(5), 1-5.