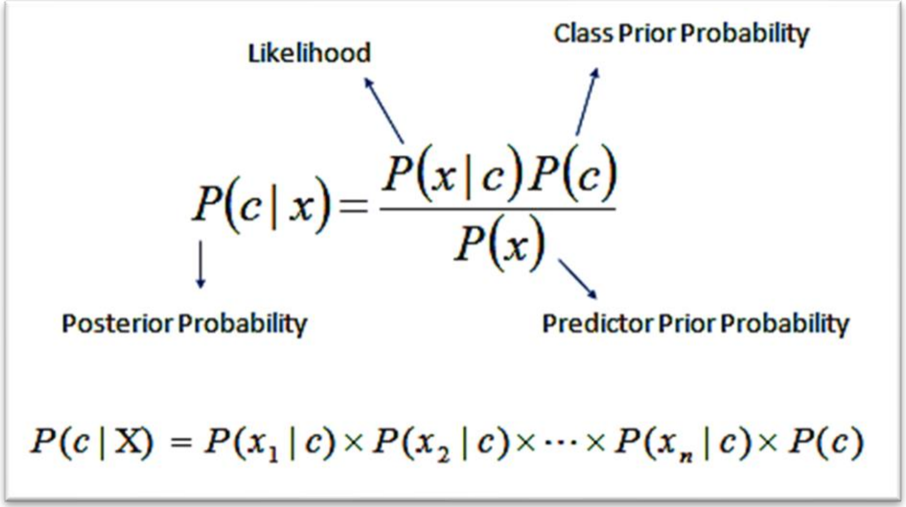
Naive Bayes Classifier with Discretization Techniques

João Soares
João Vieira

Naive Bayes is a probabilistic machine learning algorithm grounded in Bayes' Theorem.

- It is a **probabilistic** predictive model
- It is primarily used for **classification tasks** due to its simplicity and efficiency



Applications of Naive Bayes

1. **Text Classification**
2. **Medical Diagnosis**
3. **Customer Segmentation**
4. **Fraud Detection**
5. **Recommendation Systems**

Advantages and Disadvantages

Advantages

- **Efficient:** Can handle large dataset fast and with low computational cost.
- **Interpretable:** Provides probabilistic insights into predictions.
- **Resilient to noise and irrelevant attributes:** Performs well even with limited training data.
- **Works Well for Categorical Data:** Particularly effective for text classification and other categorical datasets.

Disadvantages

- **Strong Independence Assumption:** Real-world features are rarely independent
- **Sensitivity to Imbalanced Data:** May perform poorly when class distributions are skewed
- **Poor Handling of Continuous Variables:** The assumption of a specific distribution (e.g., Gaussian) often does not hold, leading to inaccuracies.

Datasets used vary in size, the number of features, the ratio of continuous to categorical features, and the number of target classes.

OpenML Datasets

Datasets	Instances	Features	Continuous Features	Classes
Diabetes	789	9	9	2
Credit_g	1000	20	7	2
Blood	748	4	4	2
Glass	214	9	9	6
ILPD	583	10	9	2
Spambase	4601	57	57	2

Preprocessing and Discretization

1. Missing Values

- Elimination of Missing Values

2. Encoding Categorical Data

- Encoding Categorical Data using One-Hot Encoding

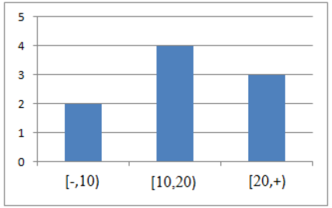
Day of Week

Tuesday

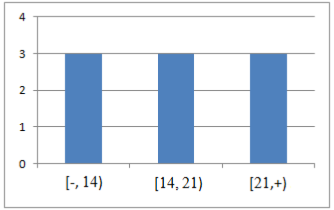
Monday	**Tuesday**	**Wednesday**
False	True	False

3. Discretization

Equal Width



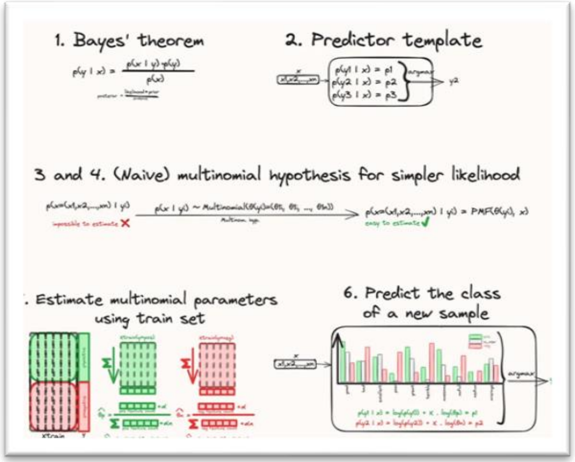
Equal Depth



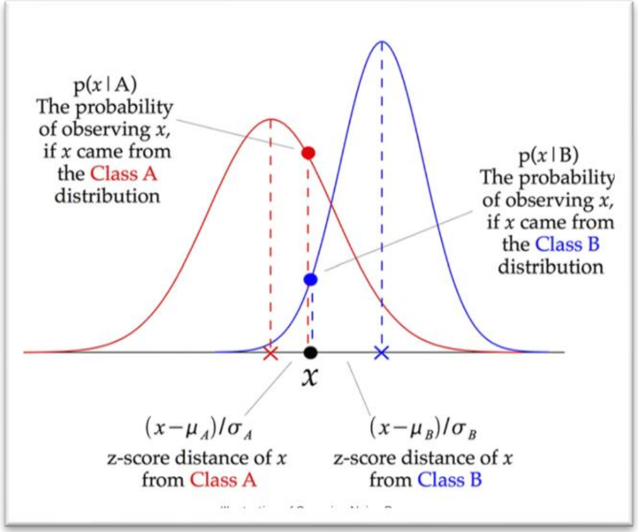
10 bins

Multinomial vs Gaussian Naive Bayes

Multinomial



Gaussian



Multinomial Naive Bayes: Discretization Impact

Key Insights

- Discretization **consistently improves** model performance
- **Equal depth outperforms** equal width in most cases. It creates **balanced bins,** addressing skewed distributions effectively.

Accuracy

Datasets	No Discretization	Equal Width	Equal Depth
Diabetes	0.6003	0.6498	0.6835
Credit_g	0.6300	0.6870	**0.7040**
Blood	0.7097	0.7620	0.7379
Glass	0.5199	0.5762	0.5251
ILPD	0.4804	0.6052	**0.6603**
Spambase	0.7903	0.8068	**0.8724**

Test: 10-Fold Cross Validation

Results: Gaussian Naïve Bayes

Key Insights

- Gaussian NB excels with **continuous data**
- Discretization often **reduces the performance slightly,** however, it **can increase** by a big margin in some cases.
- **Equal depth handles skewed data effectively.**

Accuracy

Datasets	No Discretization	Equal Width	Equal Depth
Diabetes	0.7552	0.7512	0.7460
Credit_g	0.7130	0.6800	0.6960
Blood	0.7446	0.7406	0.7366
Glass	0.4532	0.3173	0.3173
ILPD	0.5643	0.4512	**0.6756**
Spambase	0.8203	0.6805	**0.9011**

Test: 10-Fold Cross-Validation

Conclusion and Future Work

Conclusions

Multinomial Naive Bayes showed clear **benefits** from discretization

Equal depth generally **outperformed** equal width in both models when discretization was applied.

The effectiveness of each model and preprocessing technique was **highly dataset-dependent**

Testing with **Different Bin Sizes**

Exploring Additional **Discretization Methods**

Evaluating on **Diverse Datasets**

Incorporating **Hybrid Models**

AHEAD

KDE Naive Bayes

KDE Naive Bayes

- **KDE Naive Bayes** is an adaptation of the Naive Bayes classifier that **replaces the Gaussian or categorical assumptions** for feature distributions with a **KDE** approach.

- **Kernel Density Estimation (KDE)** is a non-parametric method to estimate the probability density function of data.

When to Consider KDE Naive Bayes?

- When you have continuous data that doesn't conform to common assumptions (e.g., non-normality).
- If discretization would lead to a loss of feature information.
- For datasets where traditional Naive Bayes approaches struggle due to distributional complexity.

KDE Naive Bayes

Advantages:

- Flexible with different data distributions.
- Non-parametric approximation.
- Works well with complex continuous data.
- Works better than Gaussian Naive Bayes for complex/non-normal data distributions.

Disadvantages:

- Higher computational cost due to KDE.
- Sensitive to bandwidth selection.
- Complex Parameter Tuning

KDE Naive Bayes Vs Gaussian Naive Bayes

KDE Naive Bayes:

- Outperforms in 4 out of 6 datasets
- In the Glass dataset, KDE Naive Bayes achieves 0.13 higher accuracy than GNB, demonstrating that it models multiple classes better.

The Credit_g dataset has higher accuracy in GNB because the data fits the Gaussian assumption.

Conclusion: KDE Naive Bayes improves accuracy for complex datasets by handling complicated data distributions better.

Accuracy

Datasets	Gaussian Naive Bayes	KDE Naive Bayes
Diabetes	0.73	0.76
Credit_g	0.72	0.69
Blood	0.76	0.76
Glass	0.53	0.66
ILPD	0.67	0.70
Spambase	0.84	0.85

10-fold cross-validation with hyperparameter tuning for both models

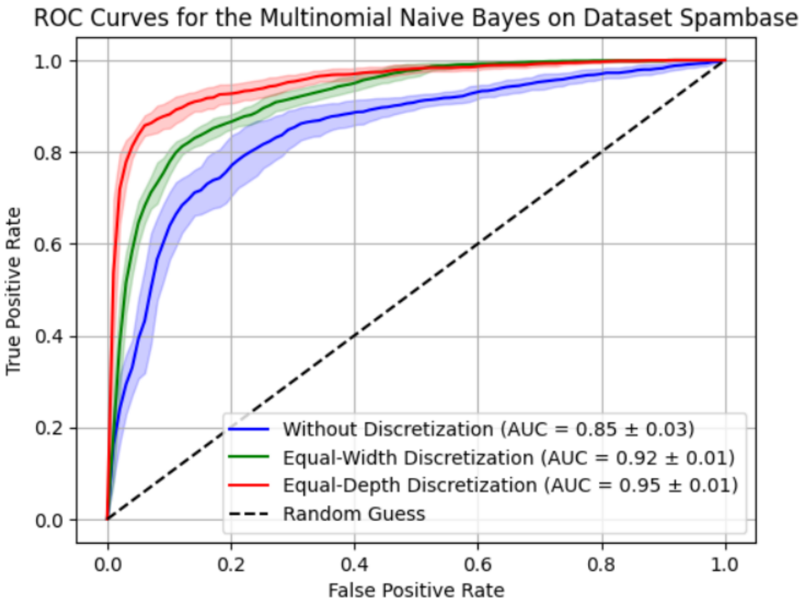
Thanks

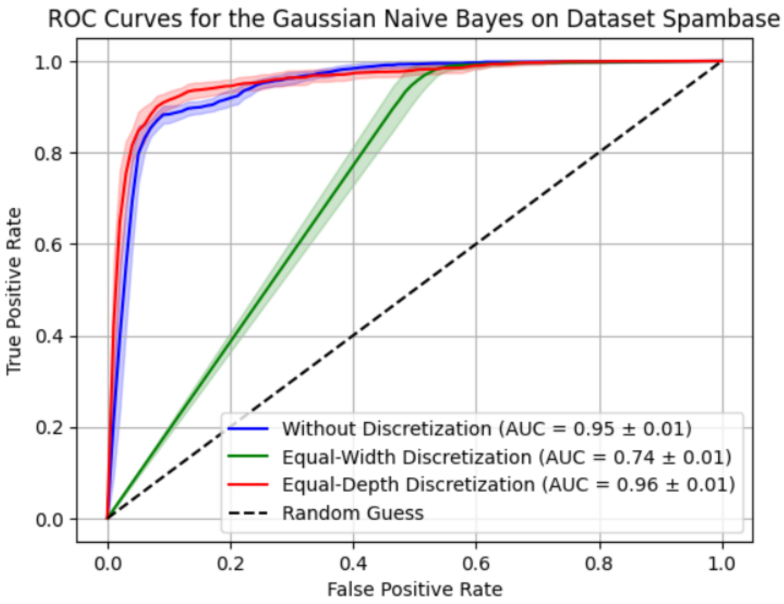
João Soares
João Viterbo

Different bin sizes applied to Multinomial Naive Bayes with Equal Depth Discretization

Datasets\Bins	5	10	15	20
Diabetes	0.6770	0.6835	0.6653	0.6717
Credit_g	0.7020	0.7040	0.6650	0.7020
Blood	0.7647	0.7379	0.7272	0.7632
Glass	0.5353	0.5251	0.5154	0.5473
ILPD	0.6808	0.6603	0.6602	0.6741
Spambase	0.8550	0.8724	0.8870	0.8555

ROC for the Multinomial Naive Bayes and Gaussian Naive Bayes and discretization techniques





Performance Metrics Comparisson between Gaussian Naive Bayes and KDE Naice Bayes for Ilpd Dataset

Models\Metrics	Accuracy	Precision	Recall
Gaussian	0.6722	0.6280	0.6430
KDE	0.7015	0.6485	0.6654