Naïve Bayes and Unsupervised Learning

Mr.M. A. Jadhav
Department of Computer Studies (MCA)
Vivekanand College, Kolhapur

14 July 2025

Agenda

Introduction to Naïve Bayes Classifiers

Concept: Probabilistic classifiers based on Bayes' Theorem, assuming feature independence.

Applications:

- Text classification (e.g., spam detection)
- Sentiment analysis

Bayes' Theorem

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

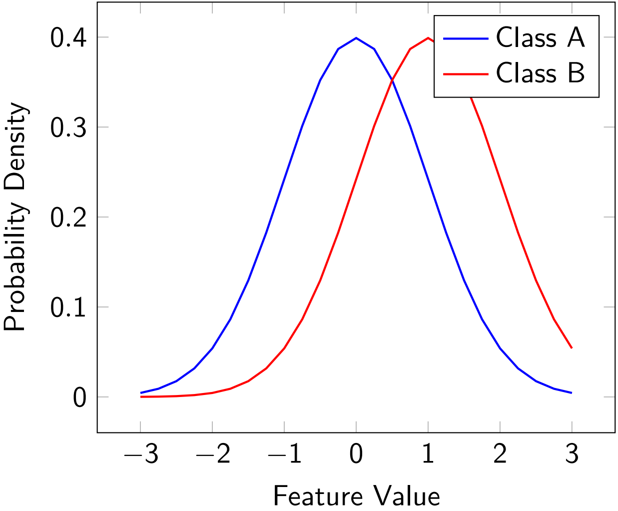
Conditional Independence Assumption: Features are independent given the class label.

$$P(C|X_1, X_2, \ldots, X_n) \propto P(C) \cdot \prod_{i=1}^{n} P(X_i|C)$$

Types of Naïve Bayes

- **Gaussian Naïve Bayes:** Assumes continuous features follow a Gaussian distribution.
- **Multinomial Naïve Bayes:** Suitable for discrete data (e.g., word counts in text).
- **Bernoulli Naïve Bayes:** Designed for binary/boolean data (e.g., presence/absence of words).

Gaussian Naïve Bayes



Continuous data modeled using Gaussian distributions.

Text Pre-processing

Steps for preparing text data:

- **Tokenization:** Splitting text into words or tokens.
- **Stop Words:** Removing common words (e.g., "the", "is").
- **TF-IDF:** Term Frequency-Inverse Document Frequency for feature extraction.

TF-IDF

Concept: Weights words based on their frequency in a document and rarity across documents.

Formula:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \log\left(\frac{N}{\text{DF}(t)}\right)$$

Text Data → Tokenization → TF-IDF → Features

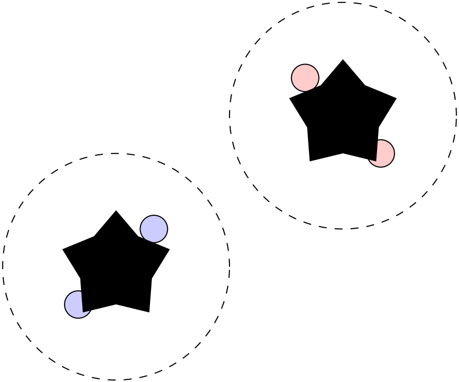
K-Means Clustering

Algorithm Steps:

1. Initialize k centroids randomly.
2. Assign data points to the nearest centroid.
3. Update centroids as the mean of assigned points.
4. Repeat until convergence.

Applications: Customer segmentation, image compression.

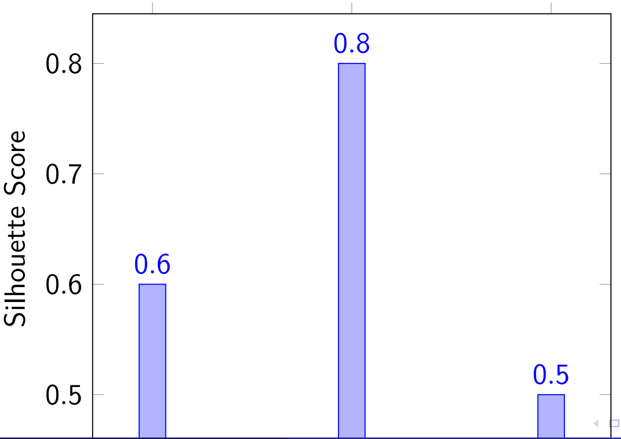
K-Means Visualization



Points clustered around centroids (C1, C2).

Clustering Evaluation Metrics

- **Inertia:** Sum of squared distances from points to their assigned centroid.
- **Silhouette Score:** Measures how similar points are to their own cluster vs. others ($[-1, 1]$).



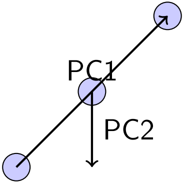
Principal Component Analysis (PCA)

Concept: Projects high-dimensional data onto lower-dimensional space while preserving variance.

Steps:

- Standardize data.
- Compute covariance matrix.
- Find eigenvectors (principal components).
- Project data onto top components.

PCA Visualization



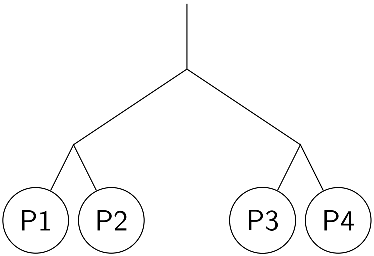
Data projected onto principal components (PC1, PC2).

Hierarchical Clustering

Concept: Builds a hierarchy of clusters, either bottom-up (agglomerative) or top-down (divisive).

Applications: Gene expression analysis, social network analysis.

Hierarchical Clustering Dendrogram



Dendrogram showing hierarchical clustering.

Naïve Bayes Implementation

Implementation (Python):

- Gaussian: sklearn.naive$_{b}ayes.GaussianNBMultinomial:$

Use Cases: Spam filtering, document classification.

K-Means Implementation

Implementation (Python): `sklearn.cluster.KMeans`

Key Parameters:

- `n_clusters`: Number of clusters (k).
- `max_iter`: Maximum iterations for convergence.

Use Cases: Market segmentation, image clustering.

PCA Implementation

Implementation (Python): `sklearn.decomposition.PCA`

Key Parameters:

- `n_components`: Number of principal components.

Use Cases: Data visualization, feature reduction.

Hierarchical Clustering Implementation

Implementation (Python):
`sklearn.cluster.AgglomerativeClustering`

Key Parameters:

- `linkage`: Method for merging clusters (e.g., ward, average).

Use Cases: Taxonomy creation, clustering documents.

Conclusion

Naïve Bayes and unsupervised learning techniques like K-Means, PCA, and hierarchical clustering are essential for classification and data analysis.

Text pre-processing and evaluation metrics enhance their effectiveness.

References

- Scikit-learn Documentation: Naïve Bayes and Clustering
- GeeksforGeeks: Unsupervised Learning
- Towards Data Science: PCA and Hierarchical Clustering