Lab Experiment-1

Naïve Bayes Classifier: Iris and Diabetes Dataset

CS3103: Machine Learning
Department of Computer Science and Engineering

July 30, 2025

Experiment Name

Implementation of Naïve Bayes Classifier on Iris and Diabetes Datasets

Aim

To implement the Naive Bayes classification algorithm from scratch using Python, train the model on Iris and Diabetes Datasets, and evaluate its performance using accuracy, precision, recall, and the confusion matrix.

Platform / Tools Used

- Python 3.x
- Jupyter Notebook / Google Colab / VS Code
- Libraries: `numpy`, `pandas`, `matplotlib`, `scikit-learn`, `seaborn`

Introduction

Naïve Bayes is a **supervised learning** algorithm based on Bayes' Theorem, which calculates the probability of a hypothesis based on prior knowledge. Despite the assumption of independence between features (hence "naïve"), it performs well in real-world applications, especially in high-dimensional data.

Bayes Theorem

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

Where:

- $P(A|B)$: Posterior probability
- $P(B|A)$: Likelihood
- $P(A)$: Prior probability
- $P(B)$: Marginal probability

Tasks to be Performed

Task 1: Iris Dataset

- Load Iris dataset(from CSV or sklearn).
- Preprocess the data (show the preprocessing).

Table 1: Dataset Preprocessing Steps and Their Purpose

Step	**Purpose**
Convert object → int	For model compatibility
Handle missing values	To ensure complete data
Scaling (optional)	Improves performance in other models
Encode categorical	For class labels or nominal features
Split train/test	Prevents overfitting, allows evaluation

- Create and apply your own class of Naive Bayes classifier to classify the species.
- Evaluate using Confusion Matrix, Accuracy, Precision, and Recall.
- Visualize the confusion matrix.
- Compare the result of your own class of Naive Bayes classifier and sklearn

Task 2: Diabetes Dataset

- Load Diabetes dataset (from CSV or sklearn).
- Preprocess the dataset as per table-1.
- Create and apply your own class of Naive Bayes classifier to classify diabetic outcome.
- Evaluate model performance with Confusion Matrix, Accuracy, Precision, and Recall.
- Visualize the confusion matrix.
- Compare the result of your own class of Naive Bayes classifier and sklearn

Expected Output

- Trained Naïve Bayes models for Iris and Diabetes datasets.
- Model accuracy score.
- Confusion matrix and classification report.
- Precision and recall values.
- Visualization plots of confusion matrices.

Conclusion

The Naïve Bayes classifier demonstrates fast and accurate classification for structured datasets like Iris and Pima Diabetes. It is highly effective in high-dimensional problems and can be applied to real-time classification tasks. The performance evaluation shows good accuracy and reliability in predictions.

References

- `https://scikit-learn.org/stable/modules/naive_bayes.html`
- `https://archive.ics.uci.edu/ml/datasets/diabetes`
- `https://www.kaggle.com/datasets/uciml/iris`
- Python Documentation - `https://docs.python.org/3/`