

Name: - Tushar Goel

Class: - 3MSTAT

Registration number: - 2248119

Topic: - Generative Models for Classification (Discriminant Analysis, Naïve Bayes’)

Submitted to: - Dr. Saleema JS

**Generative Model for Classification**

A generative model for classification is a type of machine learning model that learns the joint probability distribution of the features and the class labels from the training data. This means that the model is capable of generating new data points by sampling from the learned probability distribution, and it can also estimate the probability of a given data point belonging to a particular class. Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Naive Bayes are all considered generative models for classification.

**Linear Discriminant Analysis**

Linear Discriminant Analysis (LDA) is a supervised machine learning technique used for classification and dimensionality reduction. It is commonly used in pattern recognition and statistical analysis to find linear combinations of features that can best discriminate between different classes of data.

The main objective of LDA is to project the original feature space onto a lower-dimensional space while maximizing the class separability. This is achieved by finding a linear transformation that maps the original features to a new set of features, called discriminant functions, which have the property that the classes are well-separated and the within-class variance is minimized.

The key assumptions of LDA are:

Linear relationship: LDA assumes that the relationship between the features and the class labels is linear, i.e., the data points of each class can be separated by linear decision boundaries.

Multivariate normality: LDA assumes that the feature distributions within each class are multivariate normal distributions with equal covariance matrices.

Equal covariance matrices: LDA assumes that the covariance matrices of all classes are equal, i.e., the data points of all classes have similar variances and covariances.

**Performing LDA on iris dataset:**

Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique that can be applied to the Iris dataset to find linear combinations of features that can best discriminate between different classes of iris flowers (setosa, versicolor, and virginica). Here's how LDA works on the Iris dataset.

#### 

#### 

In this code, we first load the Iris dataset using load\_iris() from sklearn.datasets. We then create an instance of LinearDiscriminantAnalysis from sklearn.discriminant\_analysis, setting n\_components to 1 to specify that we want to project the data onto a 1-dimensional subspace. Next, we use the fit\_transform() method of the LDA object to perform LDA on the input data X with the corresponding target labels y, and obtain the projected data X\_lda. Finally, we visualize the projected data using a scatter plot, where each species is represented by a different color. The x-axis represents the first linear discriminant (LD1), which is the 1-dimensional subspace onto which the data is projected.[¶](http://localhost:8888/notebooks/CAC1_Component1_SML.ipynb#In-this-code,-we-first-load-the-Iris-dataset-using-load_iris()-from-sklearn.datasets.-We-then-create-an-instance-of-LinearDiscriminantAnalysis-from-sklearn.discriminant_analysis,-setting-n_components-to-1-to-specify-that-we-want-to-project-the-data-onto-a-1-dimensional-subspace.-Next,-we-use-the-fit_transform()-method-of-the-LDA-object-to-perform-LDA-on-the-input-data-X-with-the-corresponding-target-labels-y,-and-obtain-the-projected-data-X_lda.-Finally,-we-visualize-the-projected-data-using-a-scatter-plot,-where-each-species-is-represented-by-a-different-color.-The-x-axis-represents-the-first-linear-discriminant-(LD1),-which-is-the-1-dimensional-subspace-onto-which-the-data-is-projected.)

#### 

#### 

In this example, we first load the Iris dataset using the load\_iris function from scikit-learn. We then perform Linear Discriminant Analysis using the LinearDiscriminantAnalysis class from scikit-learn, specifying the number of components to be 2 to obtain a reduced-dimensional representation of the data. Finally, we plot the transformed data using matplotlib to visualize the results, with different colors and markers representing the different classes of Iris flowers.

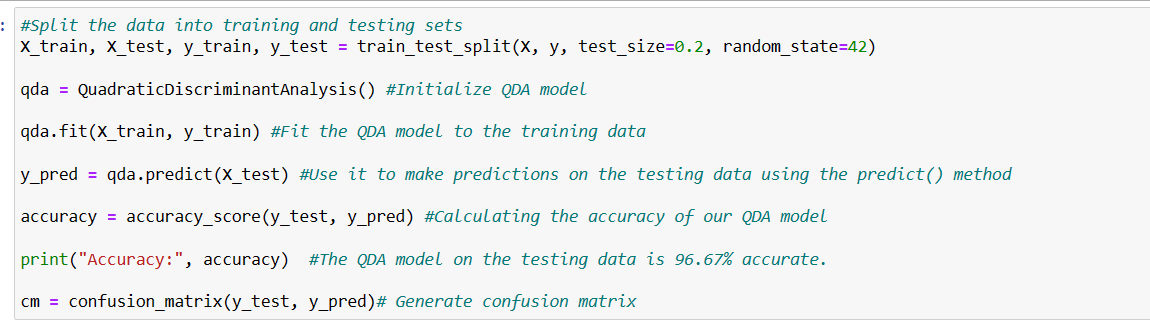
**Quadratic Discriminant Analysis**

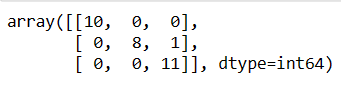
Quadratic Discriminant Analysis (QDA) is a statistical technique used for classification, similar to Linear Discriminant Analysis (LDA). However, unlike LDA which assumes that the covariance matrices of the features are equal across all classes, QDA relaxes this assumption and allows for different covariance matrices for each class. QDA can be seen as a generalization of LDA, where each class has its own covariance matrix.

The main idea behind QDA is to model the probability distribution of the features within each class using multivariate Gaussian distributions, and then use these models to estimate the probability of a new data point belonging to each class. QDA uses the estimated probability distributions to calculate the discriminant functions, which are used to determine the decision boundaries between different classes.

QDA has some advantages over LDA in cases where the assumption of equal covariance matrices across classes is not valid. However, it also has some limitations, such as increased computational complexity due to the need to estimate a separate covariance matrix for each class, and potential risk of overfitting with limited training data. It is important to carefully evaluate the data and model assumptions before choosing between LDA and QDA for a particular classification problem.

**Performing QDA on iris dataset:**

****

****

The confusion matrix shows the true positive, true negative, false positive, and false negative counts for each class. It provides a detailed view of the model's performance in terms of classification accuracy and misclassifications.

## **Naïve Bayes**

Naive Bayes is a popular and simple probabilistic machine learning algorithm used for classification tasks. It is based on the Bayesian probability theorem and makes the assumption that the features used for classification are conditionally independent given the class label. This assumption simplifies the model, making it computationally efficient and easy to implement, hence the name "naive".

The basic idea behind Naive Bayes is to calculate the probability of a data point belonging to a particular class, given its feature values, and then select the class with the highest probability as the predicted class label.

Here are the main steps in Naive Bayes algorithm:

Data Preprocessing

Model Training

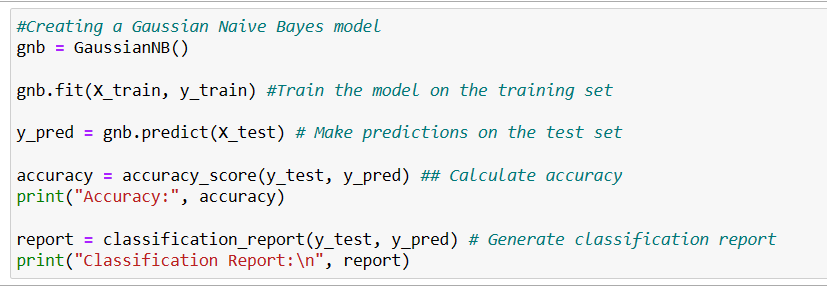
Feature Assumption

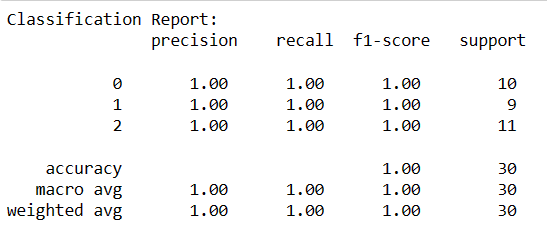
Probability Calculation

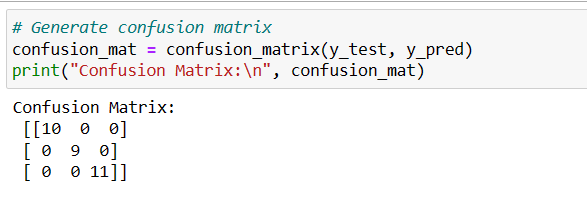
Model Evaluation

Naive Bayes is known for its simplicity and speed, and it performs well on many real-world classification tasks, such as spam detection, sentiment analysis, and document classification. However, the assumption of feature independence may not always hold true in real-world datasets, and the algorithm may not perform well in such cases. Nevertheless, Naive Bayes is a useful algorithm to consider as a baseline or as a simple and efficient solution for certain classification tasks.

Naïve Bayes on iris dataset:



****

****

The accuracy represents the percentage of correctly predicted samples in the test set. The classification report provides detailed metrics such as precision, recall, and F1-score for each class. The confusion matrix displays the number of correct and incorrect predictions for each class, which can help assess the performance of the model in terms of false positives, false negatives, true positives, and true negatives