EE 583 Pattern Recognition HW1

Eda Özkaynar 2375582

QUESTION 1 DECISION BOUNDARIES

A. Classifiers

*1) **Naive Bayes Classifier**:* The Naive Bayes classifier is a set of simple algorithms based on Bayes' theorem, often used for classification tasks. It assumes that features are independent of each other given the class label.

Bayes' Theorem: The foundational principle behind Naive Bayes is Bayes' theorem, which relates the conditional and marginal probabilities of random events. It is mathematically expressed as:

$$P(\omega \mid X) = \frac{P(X \mid \omega)P(\omega)}{P(X)} \tag{1}$$

Where:

- $P(\omega|X)$ is the **posterior probability** of class ω given the features X.
- $P(X|\omega)$ is the **likelihood** of features X given class ω.
- $P(\omega)$ is the **prior probability** of class ω.
- $P(X)$ is the **total probability** of the features X.

Naive Assumption

The "naive" part refers to the assumption that all features are independent of each other given the class label. This simplifies the computation of the likelihood:

$$P(X|\omega) = P(x_1|\omega) \times P(x_2|\omega) \times \ldots \times P(x_n|\omega)$$

Where $X = (x_1, x_2, \ldots, x_n)$ are the features.

Steps in Naive Bayes Classification

Training Phase:

1) Calculate the prior probabilities $P(\omega)$ for each class based on the training data.
2) For each feature x_i, compute the likelihood $P(x_i|\omega)$ for each class.

Prediction Phase:

1) For a new instance X, calculate the posterior probability for each class using Bayes' theorem:

$$P(\omega|X) \propto P(X|\omega)P(\omega)$$

2) Choose the class ω that maximizes $P(\omega|X)$:

$$\hat{\omega} = \arg\max_{\omega} P(\omega|X)$$

*2) **Discriminant Analysis**:* Discriminant analysis is a statistical technique used for classification and dimensionality reduction. It aims to find the best combination of features to separate different classes. It is particularly useful when certain assumptions are met.

Discriminant Function

This is a linear combination of features that maximizes class separation, derived from class means and variances within each class.

Linear Discriminant Analysis (LDA)

This common form of discriminant analysis assumes normally distributed predictor variables with the same covariance matrix across classes. LDA seeks to maximize the ratio of between-class variance to within-class variance.

*3) **Classification Tree**:* At each internal node, the dataset is split based on a feature that provides the best separation of the classes. Common criteria for determining the best split include:

Gini Impurity

Gini impurity measures the impurity of a node. It is calculated as:

$$Gini(X) = 1 - \sum_{i=1}^{\omega} p_i^2$$

Where p_i is the proportion of instances of class i in dataset X, and ω is the total number of classes.

Information Gain

Based on the concept of entropy, information gain measures the reduction in uncertainty about the class label after the dataset is split. The formula for entropy is:

$$Entropy(X) = -\sum_{i=1}^{\omega} p_i \log_2(p_i)$$

Information gain is then calculated as the difference between the entropy before and after the split.

K-Nearest Neighbors

K-Nearest Neighbors works by finding the closest data points (neighbors) to a given test point and making predictions based on these neighbors. The parameter k determines how many nearest neighbors should be considered when making a prediction.

B. Visualization of Data

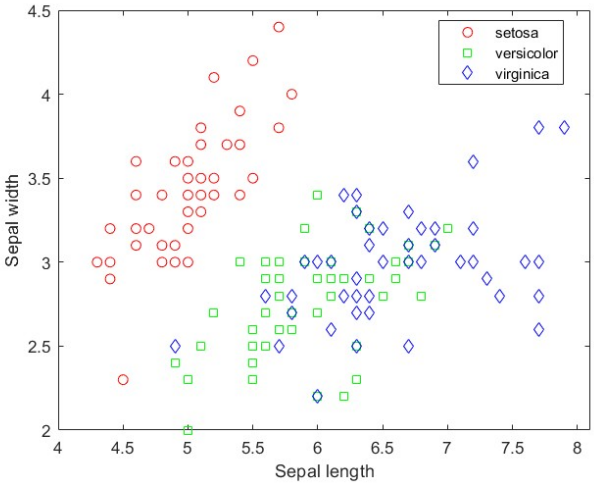


Fig. 1: Visualization of Fisher's Iris dataset.

As shown in Figure 1, Setosa, Versicolor, and Virginica species are displayed with different markers and are somewhat separable based on sepal dimensions.

C. Decision Boundaries

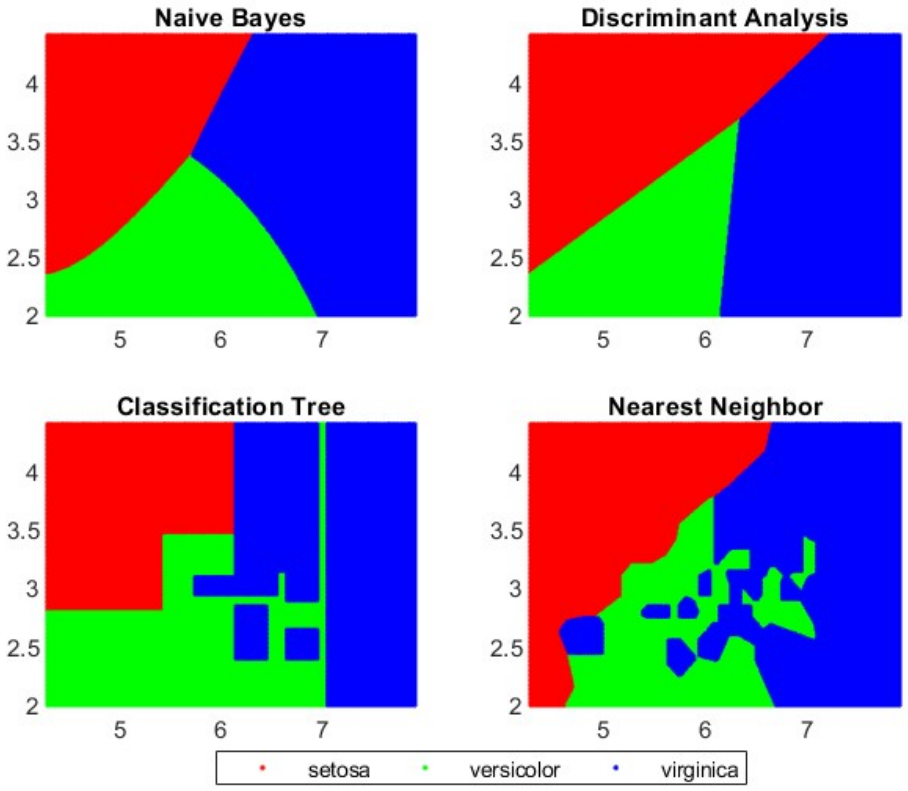
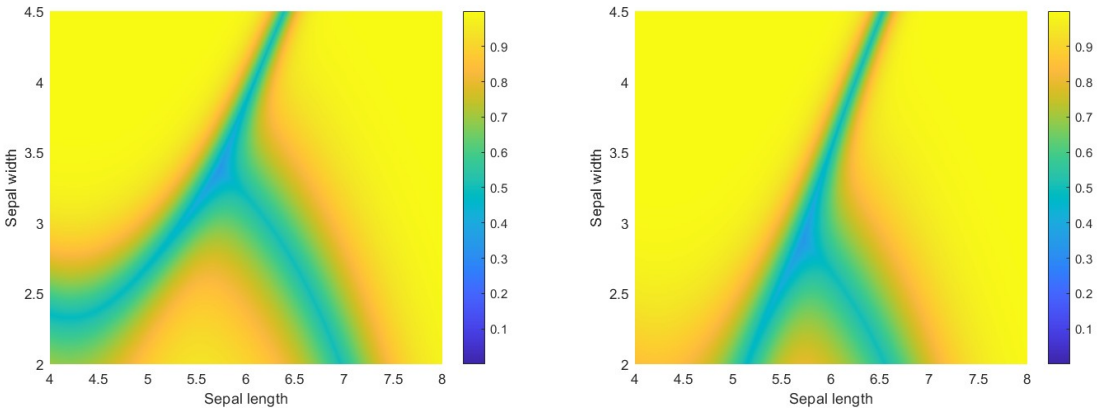


Fig. 2: Visualization of Fisher's Iris dataset.

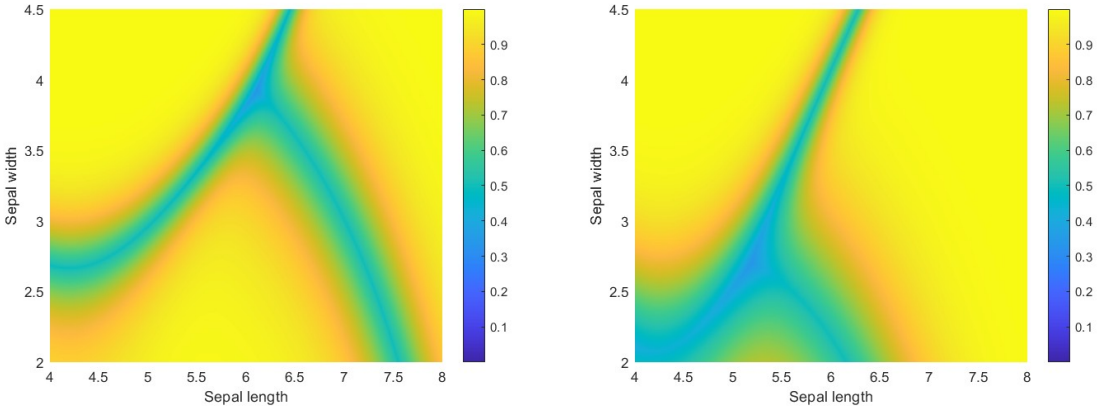
Figure 2 shows the decision boundaries for four different classifiers. The data is memorized in the "Classification Tree" and "K-NN" classifiers. Overfitting occurs, and the models fail to capture the underlying distribution. Complex decision boundaries can be seen for these classifiers. The boundaries of "Naive Bayes" and "Discriminant Analysis" classifiers can give more correct results for unseen data. That is, they are better at generalization. I would choose "Naive Bayes" since it is good at generalization and is not a linear classifier like "Discriminant Analysis." Linear classification would be insufficient for data that is not linearly separable.

QUESTION 2 BAYESIAN CLASSIFIER



(a) Prior probabilities 0.33 0.33 0.33

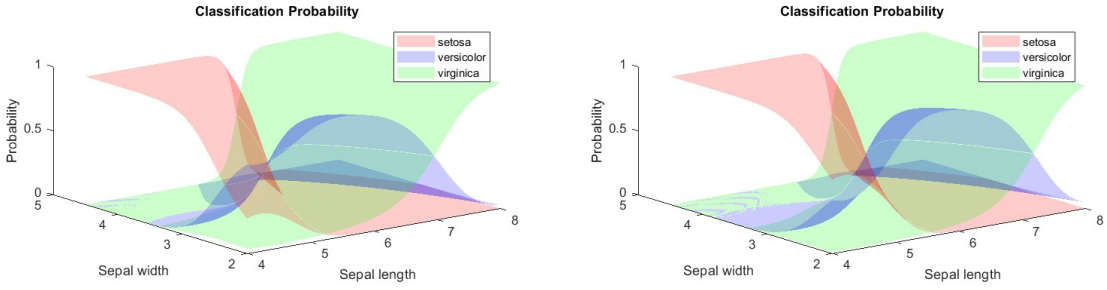
(b) Prior probabilities 0.7 0.2 0.1



(c) Prior probabilities 0.2 0.7 0.1

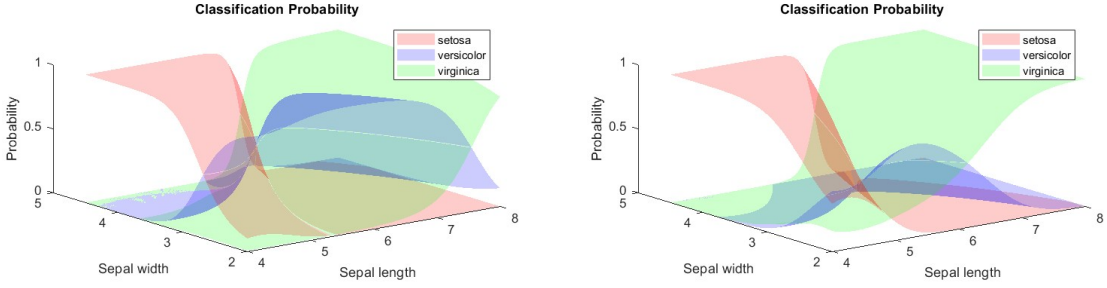
(d) Prior probabilities 0.2 0.1 0.7

Fig. 3: The posterior probability distribution for each species (2D)



(a) Prior probabilities 0.33 0.33 0.33

(b) Prior probabilities 0.7 0.2 0.1



(c) Prior probabilities 0.2 0.7 0.1

(d) Prior probabilities 0.2 0.1 0.7

Fig. 4: The posterior probability distribution for each species (3D)

In Figure 3 and 4, it can be seen that posterior probabilities increase when the prior probabilities of classes increase. As seen in Equation 1, the prior probability $P(\omega)$ is a multiplicative factor in the numerator of Bayes' theorem. If you increase $P(\omega)$ for a class, it directly increases the value of the numerator. Increasing the prior probability of a class makes it more likely that the Bayesian classifier will assign new data points to that class, as the posterior probability for that class will increase. If the class-conditioned likelihood for each class is similar, the effect of prior probability can be more observable.

QUESTION 3 MAHALANOBIS DISTANCE

The Mahalanobis distance measures the distance between a point and a distribution.

$$d_M = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{2}$$

$$d_E = \sqrt{(x-\mu)^T I^{-1} (x-\mu)} \tag{3}$$

As can be seen in Equations 2 and 3, in contrast to the Euclidean distance, which only considers the direct distance between two points, the Mahalanobis distance considers correlations in the data by using the covariance matrix.

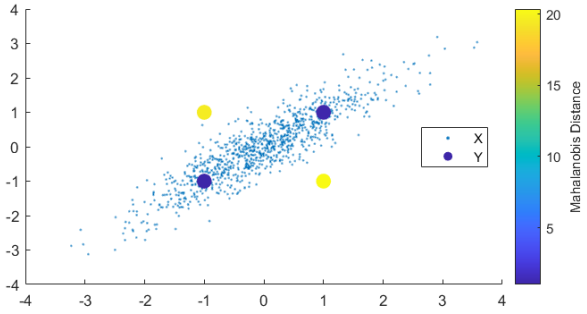
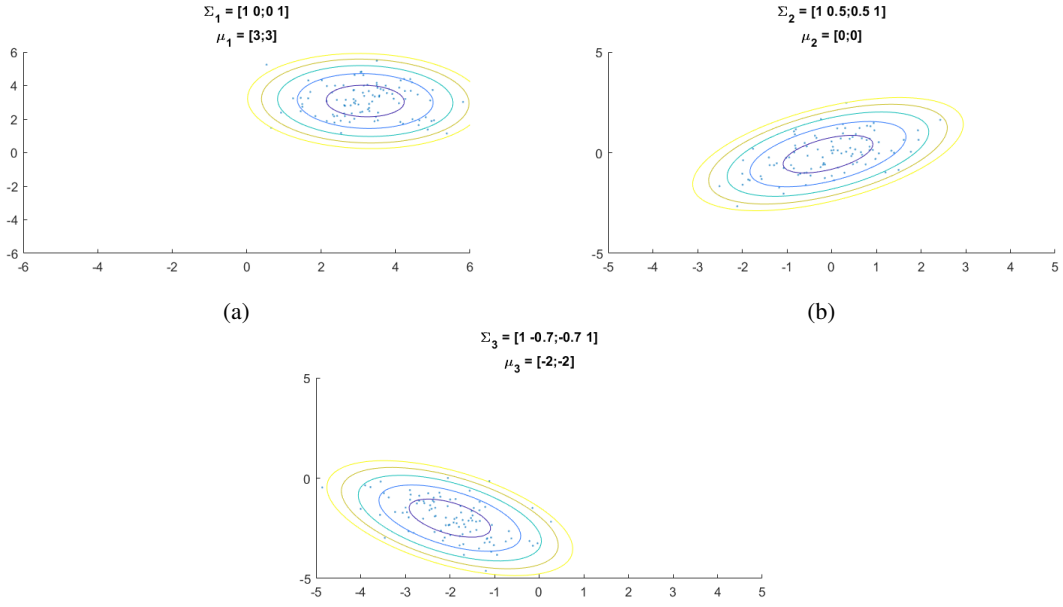


Fig. 5: Mahalanobis Distance of 4 equidistant points in Euclidean distance

In Figure 5, one can show that the equidistant point in Euclidean distance is not the same as the Mahalanobis distance because the Mahalanobis distance takes into account the data's covariance and the scales of the different variables.



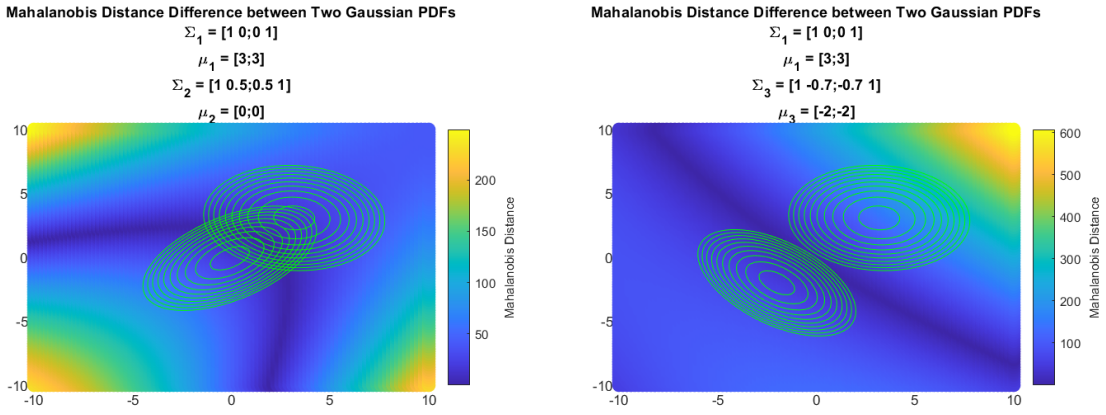
(a) (b)

(c)

Fig. 6: Three different Gaussian Distributions

In Figure 6, we can observe the effect of different covariance matrices. If the covariance matrix is an identity matrix, Mahalanobis distance is the same as the Euclidean distance in equation 3. In 7b, data points have a positive correlation with

each other; therefore, we observe a tilted ellipsoid. Similarly, in 6c, they are negatively correlated, and we observe again a tilted ellipsoid.

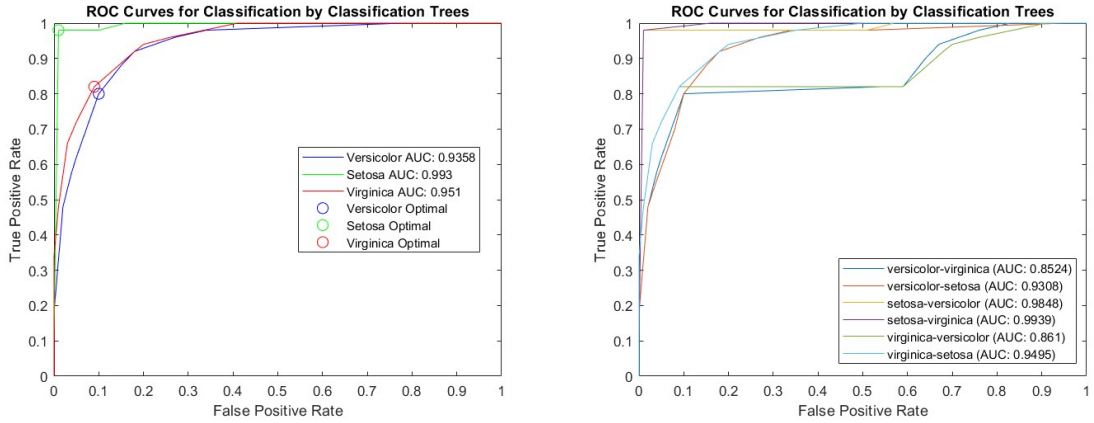


(a) (b)

Fig. 7: Absolute Mahalanobis difference between two distances to two Gaussian pdfs.

In Figure 7, the absolute Mahalanobis difference between the distributions is plotted. We can think of the line where the distance difference is zero as a decision boundary between two distributions. That line is equidistant from both distributions in Manalanobis distance.

QUESTION 4 ROC CURVE FOR CLASSIFICATION TREE



(a) OvR ROC curves (b) OvO ROC curves

Fig. 8: ROC Curves for Classification Tree.

When dealing with multi-class classification, there are two main strategies: OvR (One-vs-Rest) and OvO (One-vs-One). These strategies reduce the problem to multiple binary classifications.

OvR (One-vs-Rest) strategy involves comparing each class against all the other classes combined. This transforms the task into a binary classification problem, allowing the use of binary classification metrics. For a dataset with three classes, OvR produces three binary classifiers, one for each class. On the other hand, the OvO (One-vs-One) strategy compares every possible pair of classes against each other. A dataset with three classes results in 3 possible combinations (Class1 vs. Class2, Class1 vs. Class3, Class2 vs. Class3), with six scores accounting for both directions. For each pair, one class is considered positive and the other negative.

In Figure 8, the OvR and OvO ROC curves are displayed. Upon reviewing both curves, it can be inferred that the classifier model excels at classifying the setosa class. The AUC values, which are used to assess the performance of binary classification models for setosa, are 0.993 for the OvR case, 0.9939 for setosa-virginica, and 0.9848 for setosa-versicolor. On the other hand, the AUC values for versicolor and virginica are relatively lower. It means the classifier is less effective at distinguishing between versicolor and virginica, as reflected in the lower AUC values. These two classes appear to be more challenging to differentiate, possibly due to overlapping features.