ML- Lab3

salyam

March 2025

Assignment 1

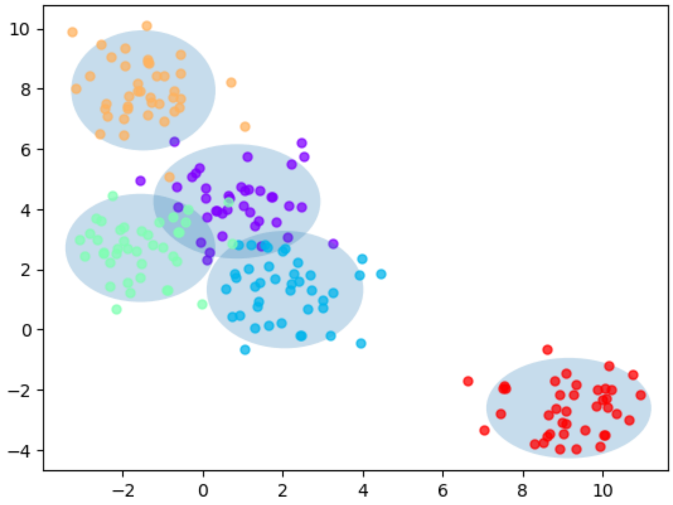


Figure 1: Gaussian-distributed data points with 95% confidence intervals based on ML-estimates of μ_k and Σ_k.

Assignment 3

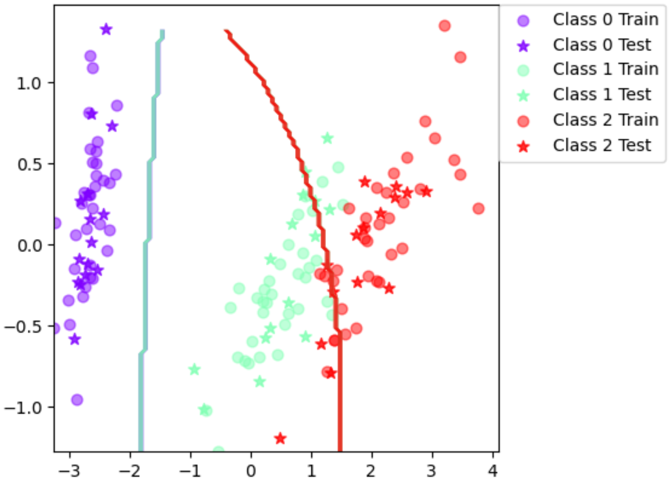


Figure 2: Decision boundary for the 2D *iris* dataset using `plotBoundary`.

Assignment 4

- When can a feature independence assumption be reasonable and when not?

  Determining whether features in a dataset are independent can be challenging. A simple approach is to visualize the data—if the features are well-separated, it may suggest independence. However, if the data appears scattered without clear structure, strong assumptions cannot be made, and mathematical operations may be required to analyze relationships between features.

  Bayes classifiers operate under the assumption that a feature's value is independent of other features, given the class variable. In reality, this assumption often does not hold, as features in real-world datasets tend to be dependent. For instance, in the Iris dataset, sepal length and sepal width, as well as petal length and petal width, exhibit positive correlations. Despite this, naive Bayes classifiers still perform well, as assuming independence significantly simplifies computations.

  If the data is conditionally independent, or mostly independent, such assumptions can be reasonably made. However, when dependencies exist, it is preferable to explore alternative methods that better capture feature relationships.

- How does the decision boundary look for the Iris dataset? How could one improve the classification results for this scenario by changing classifier or, alternatively, manipulating the data?

The decision boundary between class 1 and class 2 appears to be suboptimal, leading to misclassified points. This is likely due to the inherent overlap between these classes in the dataset. While the current model performs reasonably well in distinguishing class 0 from class 1, the boundary between class 1 and class 2 could be improved to enhance classification accuracy.

To address this issue, several approaches can be considered. One effective solution is to use non-linear models such as Support Vector Machines (SVM) with slack variables or Random Forests, which are capable of handling complex decision boundaries. SVMs, in particular, allow for better separation by introducing a margin that accommodates some misclassifications while maintaining overall model robustness.

Another possible improvement is feature transformation. Applying logarithmic or other non-linear transformations to the dataset may help in better distinguishing overlapping points by modifying the feature space. Additionally, increasing the dimensionality of the data—such as through feature engineering or kernel methods in SVM—could provide a more expressive representation, enabling better separation between class 1 and class 2.

Assignment 5

- Is there any improvement in classification accuracy? Why/why not?

  The results indicate a clear **improvement in classification accuracy** after applying **boosting**, particularly on **the Iris and Vowel datasets**. In the Iris dataset, the error rate was nearly reduced by half, demonstrating the effectiveness of this approach. Boosting proves especially valuable for **complex datasets**, where **Naïve Bayes alone struggles** to classify data points accurately.

  One of the key advantages of boosting is its **weight adjustment mechanism**, which assigns higher weights to misclassified points. This helps the model refine its **decision boundary**, particularly between **class 2 and class 3 in the Iris dataset**, where boosting seems to reduce the influence of classifiers that would otherwise introduce a curved boundary.

  Additionally, **boosting leverages ensembling**, combining multiple classifiers to form a more robust model. This approach is particularly useful for **Naïve Bayes**, where multiple weak classifiers can be combined to improve decision-making on challenging datasets.

  To further improve results, we could experiment with different **ensemble strategies**, such as adjusting the boosting parameters or incorporating additional transformations to enhance feature separability.

Table 1: Classification Accuracy for the Iris Dataset

Trial	**Without Boosting**	**With Boosting**
0	84.4	100
10	97.8	97.8
20	91.1	93.3
30	86.7	93.3
40	88.9	97.8
50	91.1	86.7
60	86.7	93.3
70	91.1	95.6
80	86.7	93.3
90	91.1	95.6
Mean Accuracy	89.2 ± 4.19	94.8 ± 3.07

Table 2: Classification Accuracy for the Vowel Dataset

Trial	**Without Boosting**	**With Boosting**
0	52.6	68.8
10	61.7	77.9
20	68.2	77.3
30	62.3	70.1
40	56.5	68.8
50	63.0	68.2
60	64.3	77.3
70	62.3	70.8
80	60.4	71.4
90	65.6	82.5
Mean Accuracy	61.3 ± 3.48	74.1 ± 3.73

- Plot the decision boundary of the boosted classifier on iris and compare it with that of the basic. What differences do you notice? Is the boundary of the boosted version more complex?

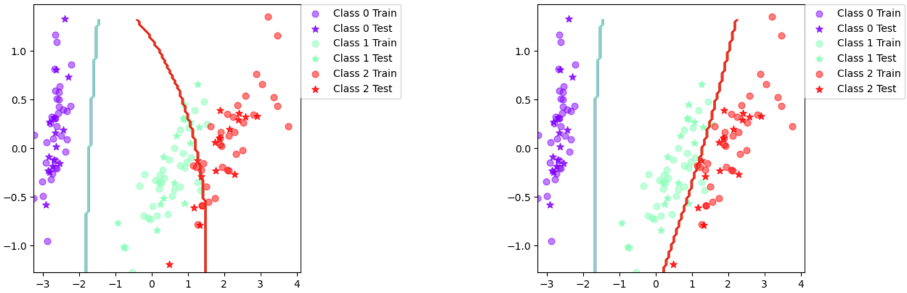
  The results indicate a clear **improvement in the decision boundary** compared to its previous state. The new boundary better separates **class 1 and class 2 data points**, leading to a more effective classification.

  However, an **increase in complexity does not necessarily equate to better performance**. Some observations indicate that, rather than becoming more complex, the decision boundary now simply **fits the dataset better**, suggesting that the improvement comes from **a better adaptation to the data rather than additional complexity**.

- Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?

Lab experiments confirm that **boosting improves classification accuracy** by adjusting the weights of misclassified points, refining the decision boundary. This technique enhances a **base classifier**, such as a decision tree, to create a more advanced model.

However, boosting has limitations. **Feature independence issues remain unresolved**, and excessive boosting can lead to **overfitting**, reducing generalization. To improve results, careful tuning of parameters and regularization techniques should be considered.



(a) Before Boosting

(b) After Boosting

Figure 3: Comparison of decision boundaries before and after boosting on the *iris* dataset.

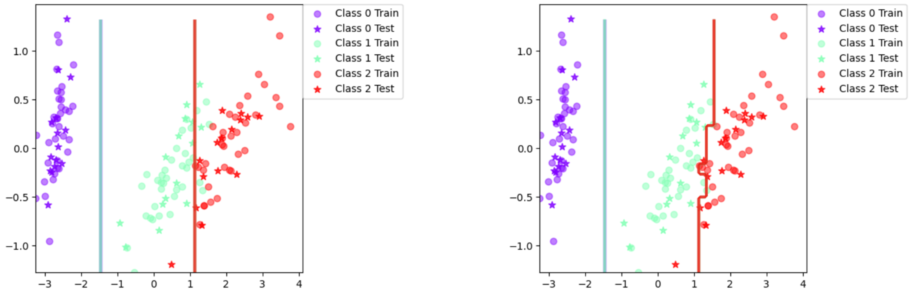
Assignment 6

Table 3: Decision Tree Classification Accuracy for the Iris Dataset

Trial	**Without Boosting**	**With Boosting**
0	95.6	95.6
10	100	100
20	91.1	95.6
30	91.1	93.3
40	93.3	93.3
50	91.1	95.6
60	88.9	88.9
70	88.9	93.3
80	93.3	93.3
90	88.9	93.3
Mean Accuracy	92.4 ± 3.71	94.6 ± 3.65

Table 4: Decision Tree Classification Accuracy for the Vowel Dataset

Trial	Without Boosting	With Boosting
0	63.6	84.4
10	68.8	90.9
20	63.6	87.7
30	66.9	92.2
40	59.7	85.7
50	63.0	81.2
60	59.7	91.6
70	68.8	86.4
80	59.7	86.4
90	68.2	88.3
Mean Accuracy	64.1 ± 4.00	86.8 ± 3.01



(a) Before Boosting

(b) After Boosting

Figure 4: Decision boundaries of the Decision Tree classifier on the *iris* dataset before and after boosting.

Assignment 7

Outliers

For handling outliers, **Naïve Bayes with boosting** provides a **best-fit line**, avoiding complex boundary shapes that may arise in other classifiers. However, **Naïve Bayes without boosting** may be preferable to **prevent overfitting**, as boosting could assign excessive importance to misclassified outliers.

Irrelevant Inputs

Decision Trees are effective at handling **irrelevant inputs**, as they can **ignore less informative features** and focus on attributes with higher predictive power. **Boosting combined with pruning** further refines this process by eliminating unnecessary features, leading to a more efficient model.

Predictive Power

Naïve Bayes with boosting enhances predictive power by improving decision boundaries. While **Decision Trees may have lower error rates**, they **do not always generalize well**, which can reduce their effectiveness on unseen data.

Mixed Types of Data

When working with datasets containing **binary, categorical, or continuous features**, **Naïve Bayes is generally more suitable**, as it efficiently processes continuous data while maintaining classification accuracy.

Scalability

For **large-scale datasets**, **Decision Trees are more scalable**, whereas **Naïve Bayes struggles with the curse of dimensionality**. **Decision Trees with pruning** can handle **high-dimensional data efficiently**, making them a better choice when dealing with large datasets.

To improve overall performance, **tuning boosting parameters, optimizing pruning strategies, and refining feature selection** should be considered to balance **accuracy and generalization**.

Effect of Boosting Iterations on Bayes Classifier Accuracy

The figure illustrates how the number of **boosting iterations** (T) impacts the classification accuracy of the **Bayes Classifier** on the *Iris* dataset. The **x-axis** represents the number of boosting iterations, while the **y-axis** indicates the mean classification accuracy over multiple trials. The error bars display the **standard deviation**, highlighting performance variability.

The results demonstrate that accuracy **improves significantly up to** $T = 5$, suggesting that boosting enhances the classifier's ability to generalize effectively. As boosting iterations increase, the model benefits from learning more refined decision boundaries. However, beyond $T = 7$, accuracy stabilizes around **94.8%**, indicating diminishing returns from additional boosting.

A notable observation is that **standard deviation decreases at** $T = 5$, signifying more stable performance across trials. This suggests that a moderate amount of boosting helps in reducing variance and increasing reliability. However, excessive boosting, particularly at $T = 10$ and beyond, does not lead to further accuracy improvements. This implies that the model may have reached its **optimal performance for this dataset**, and further boosting may introduce unnecessary complexity without meaningful gains.

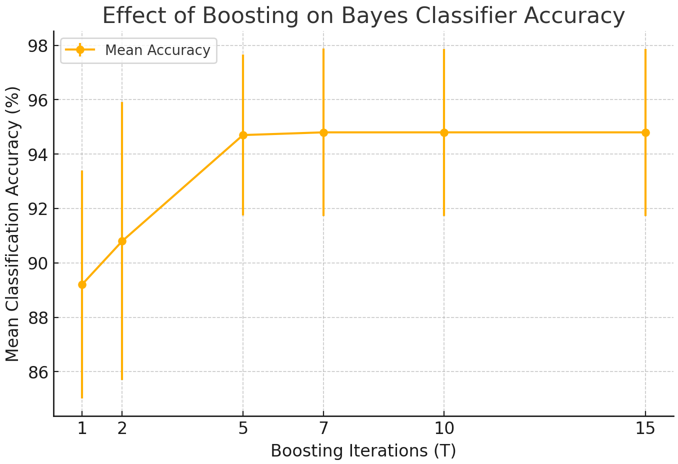


Figure 5: Effect of boosting iterations (T) on the mean classification accuracy of the Bayes Classifier. Error bars represent standard deviation across multiple trials.

Impact of Training Data Split on Boosted Decision Tree Performance

The figure illustrates how the **training data split ratio** affects the **classification accuracy** of the **Boosted Decision Tree Classifier** on the *Iris* dataset. The **x-axis** represents the percentage of data used for training, while the **y-axis** indicates the mean classification accuracy. The **error bars** display the **standard deviation**, showing the variability in performance across multiple trials.

A **low training data split** of 5% results in **lower accuracy** (83%) and **high variance** (standard deviation = 10.6), indicating that insufficient training data leads to unstable performance. As the **training split increases**, accuracy improves significantly and stabilizes. An **optimal balance** appears between **30% and 60%**, where accuracy remains stable around 93-94% with minimal variance. Beyond a **60% training split**, accuracy continues to improve marginally, reaching **95.1% at an 80% split**, but with a slightly higher standard deviation (4.33), suggesting a potential risk of overfitting.

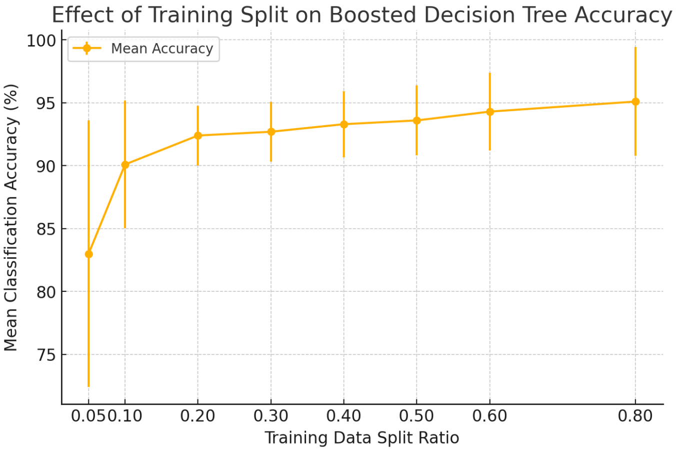


Figure 6: Effect of Training Data Split on Boosted Decision Tree Accuracy. The plot shows the mean classification accuracy for different training data split ratios, with error bars representing the standard deviation.

Comparison of Classifier Performance on the Iris Dataset

The figure illustrates the comparison of classification accuracy among different machine learning classifiers on the *Iris* dataset. The **x-axis** represents the classifiers tested, while the **y-axis** indicates the mean classification accuracy over multiple trials. The **error bars** represent the **standard deviation**, showing the variability in performance across different test runs.

The results show that **SVM (Linear) and KNN achieve the highest accuracy of approximately 96.4%** with low variance, indicating that these models provide stable and reliable performance for the *Iris* dataset. **SVM (RBF) and Logistic Regression** follow closely, achieving mean accuracies of **95.4% and 95.5%**, respectively. These models remain competitive and demonstrate consistent classification results.

Random Forest exhibits a slightly lower accuracy of **94%** with a standard deviation of **3.63**, suggesting that it may be more sensitive to variations in the dataset. **Gradient Boosting** achieves the lowest mean accuracy of **92.6%** and the highest standard deviation of **3.95**, indicating that it may not be the most suitable choice for this dataset due to its higher variability.

Overall, the results suggest that **SVM (Linear) and KNN are the most accurate and stable classifiers for the *Iris* dataset**. While **Gradient Boosting** appears to be less reliable due to its high variance, **Random Forest** also shows notable fluctuations, suggesting that further parameter tuning may be required to enhance its performance. The findings indicate that **SVM (RBF) and Logistic Regression** serve as viable alternatives, offering consistent classification accuracy.

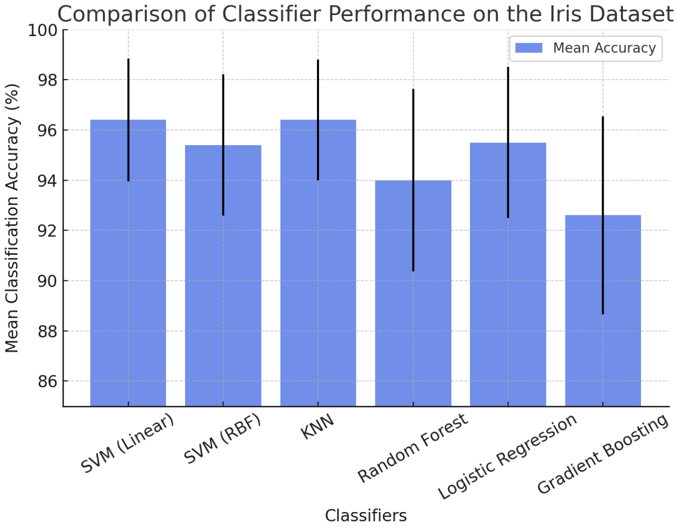


Figure 7: Comparison of classifier performance on the *Iris* dataset. The bar chart shows the mean classification accuracy for each classifier, with error bars representing standard deviation.

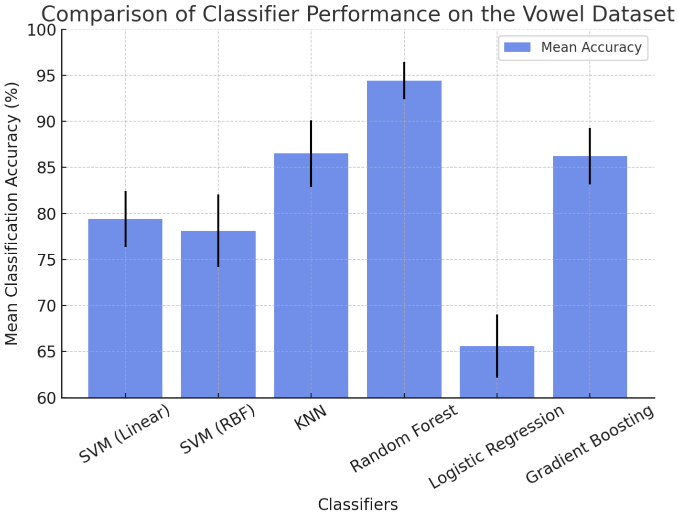


Figure 8: Comparison of classifier performance on the *Vowel* dataset. The bar chart shows the mean classification accuracy for each classifier, with error bars representing standard deviation.

Bonus

Table 5: Classification Accuracy for the Olivetti Dataset

Trial	**Without Boosting**	**With Boosting**
0	82.5	80.0
10	88.3	75.8
20	83.3	71.7
30	85.8	75.8
40	86.7	70.0
50	80.8	67.5
60	88.3	79.2
70	81.7	75.8
80	79.2	56.7
90	84.2	65.8
Mean Accuracy	84.2 ± 3.23	70.5 ± 6.76

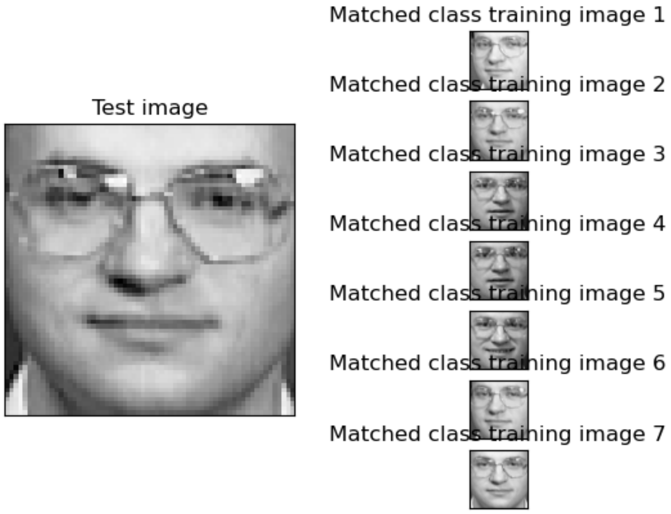

Test image
Matched class training image 1
Matched class training image 2
Matched class training image 3
Matched class training image 4
Matched class training image 5
Matched class training image 6
Matched class training image 7

Figure 9: Enter Caption