Iris Flower Classification

DSP Project Report
CognitIQ
(Kirti Agarwal, Sambit Sahoo, Nirman Jaiswal)

November 29, 2024

Abstract

The classification of iris flowers is a fundamental problem in machine learning, often used to demonstrate the application of supervised learning algorithms. This study explores the use of the classic Iris dataset, which contains 150 samples of iris flowers from three species (Iris setosa, Iris versicolor, and Iris virginica). Four key features—sepal length, sepal width, petal length, and petal width—are used to predict the species of a given iris flower. Various machine learning models, including k-Nearest Neighbors (k-NN), Decision Trees, and Support Vector Machines (SVM), were implemented and evaluated for their classification accuracy. The results show that high accuracy can be achieved due to the well-separated feature space of the dataset, with certain models outperforming others in terms of performance and computational efficiency. This work not only highlights the effectiveness of these algorithms for small-scale datasets but also underscores the importance of feature selection and visualization in classification tasks.

Contents

1 Problem Statement **3**

2 Dataset Details **3**

3 Literature Review **3**

4 Methodology **3**
4.1 Exploratory Data Analysis 4
4.2 Model Training 7
4.2.1 k-Nearest Neighbors (kNN) 7
4.2.2 Random Forest 7
4.2.3 Support Vector Machine (SVM) 7
4.2.4 Decision Tree 7
4.2.5 Artificial Neural Network (ANN) 8
4.2.6 Logistic Regression 8
4.2.7 Gaussian Naive Bayes 8
4.2.8 Ensemble of Ensembles 8
4.2.9 Adaboost 8
4.2.10 Gradient Boosting 8
4.2.11 XGBoost 9
4.2.12 CatBoost 9

5 Results **9**
5.1 k-Nearest Neighbors (kNN) 9
5.2 Random Forest 10
5.3 Support Vector Machine (SVM) 10
5.4 Decision Tree 11
5.5 Artificial Neural Network (ANN) 11
5.6 Logistic Regression 12
5.7 Gaussian Naive Bayes 12
5.8 Ensemble of Ensembles 13
5.9 Adaboost 13
5.10 Gradient Boosting 14
5.11 XGBoost 14
5.12 CatBoost 15
5.13 Summary of Model Accuracies 15

6 Conclusions **16**

7 Future Scopes **17**

1 Problem Statement

The goal of the task is to explore different Machine learning techniques by applying models that can predict the species of a given Iris flower into (Iris Versicolor, Iris Setosa, Iris Virginica) based on its features (sepal length, sepal width, petal length, petal width) with an accuracy of atleast 80%.

2 Dataset Details

- **Size:** The dataset contains 150 instances, evenly distributed across three classes (50 samples per class).
- **Features**: Four numerical features representing physical attributes: Sepal length (in cm), Sepal width (in cm), Petal length (in cm), Petal width (in cm).
- **Target Variable**: The species of the iris flower, which is a categorical variable with three possible values: Iris setosa, Iris versicolor, Iris virginica.

3 Literature Review

- Earlier paper presents machine learning techniques to classify iris flower species using Decision Trees, Gaussian Naive Bayes, Support Vector Machines, and k-nearest neighbor models. Their results show that the Gaussian Naive Bayes model performs the best, achieving an accuracy of 1.0 on the Iris dataset(Rao, 2023).
- Another study presents the SVM technique to classify iris flower species a newly mode for classifying iris data set using SVM classifier and genetic algorithm to optimize c and gamma parameters of linear SVM, in addition principle components analysis (PCA) algorithm was use for features reduction. The obtained results showed that the accuracy of the SVM increased to 98%(Hussain, 2023).

4 Methodology

Steps that we followed for the analysis were:

- Data Input
- Exploratory Data Analysis (EDA)
- Data Preprocessing (Data Cleaning, Scaling)
- Data Splitting (Train-Test Splitting of test size = 30%)
- Model Training (Applied Various Models)
- Prediction Of Test Data
- Model Selection Based On Performance

4.1 Exploratory Data Analysis

In EDA we did:

- **Pairplot:**
  A pair plot allows us to see both distribution of single variables and relationships between two variables. (Figure 1)

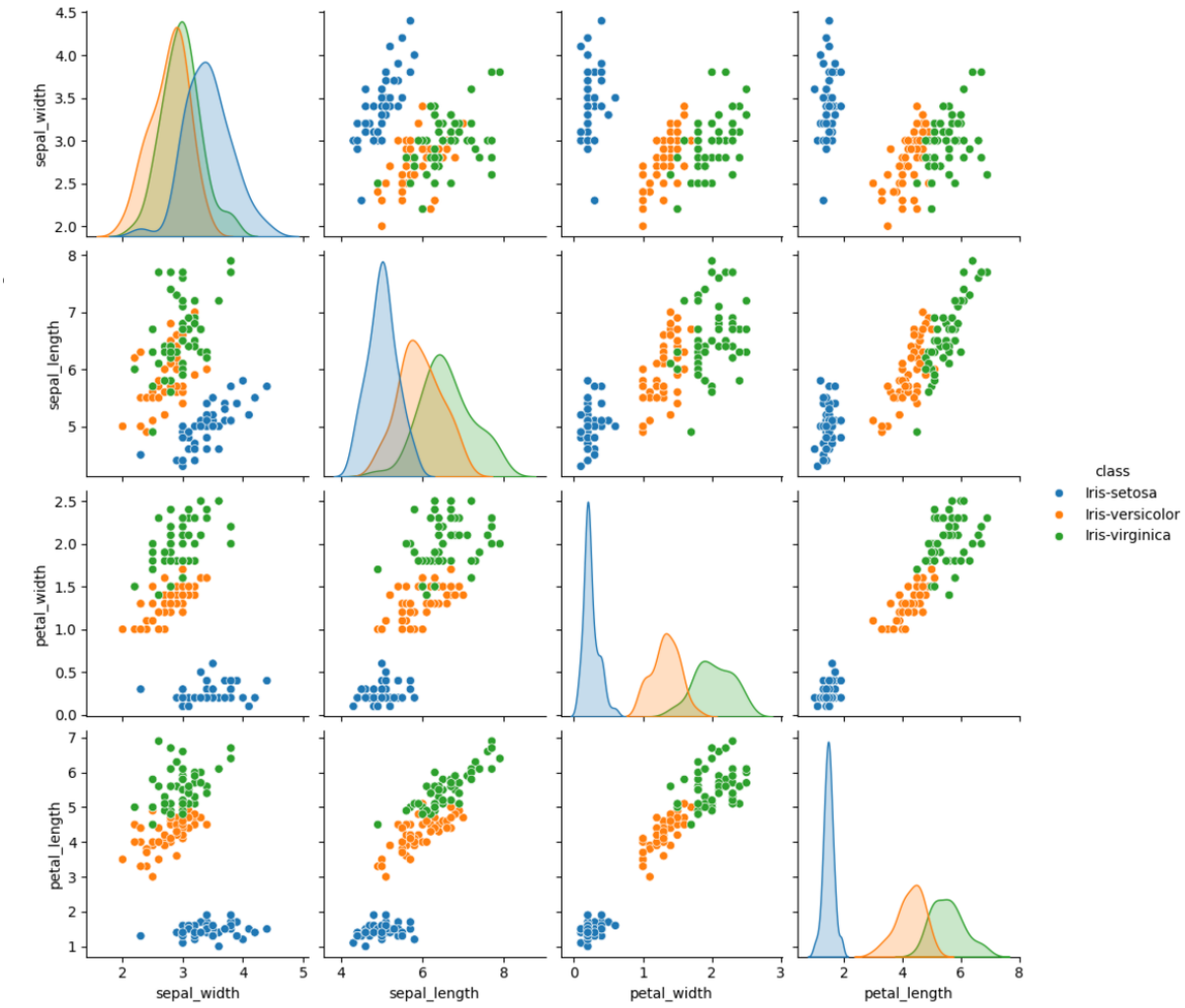

Figure 1: Pairplot of Iris Dataset

- **Correlation Matrix:**
  The correlation matrix is a matrix that shows the correlation between variables. It gives the correlation between all the possible pairs of values in a matrix format.(Figure 2)

- **Box Plot:**
  Box plots visually show the distribution of numerical data and skewness by displaying the data quartiles (or percentiles) and averages.
  Box plots show the five-number summary of a set of data: including the minimum score, first (lower) quartile, median, third (upper) quartile, and maximum score.(Figure 3 and Figure 4)

- **Histogram:**
  A histogram consists of bars where the height of each bar represents the frequency or probability of the occurrence of data within specific intervals, known as bins. The x-axis represents the value ranges (bins), and the y-axis represents the frequency or density. Kernel Density Estimation (KDE) is a non-parametric method for estimating the probability density function of a random variable. It provides a smooth estimate of the distribution, overcoming some of the limitations of histograms, such as sensitivity to bin size.(Figure 5)

- **Violin Plot:**
  Violin Plot is a method to visualize the distribution of numerical data of different variables. It is quite similar to Box Plot but with a rotated plot on each side, giving more information about the density estimate on the y-axis.(Figure 6)

- **Hexbin Plot:**
  Hexbin plots are used to visualise the density of points in 2D. Individual plots may lead to overplotting. Hexbin plots aggregate points within hexagonal bins and color them based on the number of points in each bin.(Figure 7)

- **Error Bars:**
  Error bars are used to visualise the variability of data in plots. (Figure 8)

- **PCA:**
  We run PCA on our data set to reduce the dimensionality of the dataset and visualize how the classes separate in a lower-dimensional space.(Figure 9)

- **Density Contours:**
  Density contours on scatter plot allow visualise the kernel density estimation (KDE) contours. Contours represent the estimated density of points in different regions of the plot. Useful for understanding where the data is concentrated.(Figure 10)

- **LDA:**
  Linear Discriminant Analysis (LDA), also known as Normal Discriminant Analysis or Discriminant Function Analysis, is a dimensionality reduction technique primarily utilized in supervised classification problems. It facilitates the modeling of distinctions between groups, effectively separating two or more classes.(Figure 11)

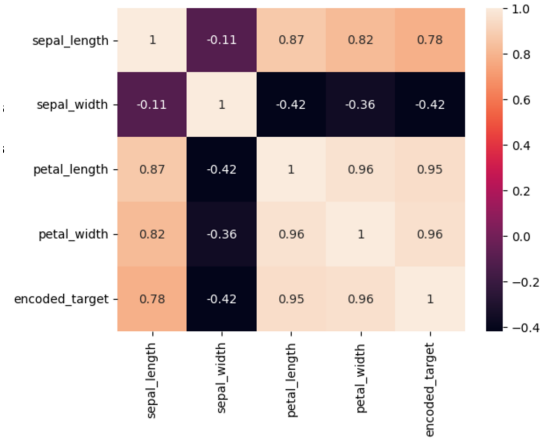

Figure 2: Correlation Matrix

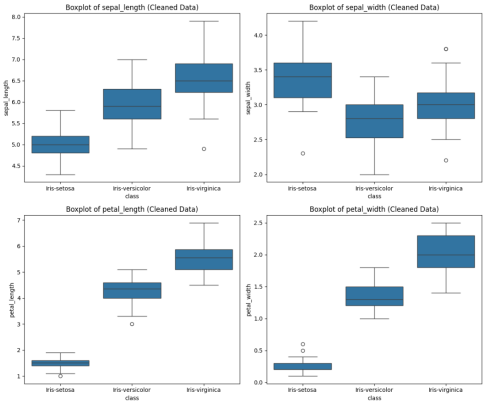

Figure 3: Boxplot(Before Cleaning)

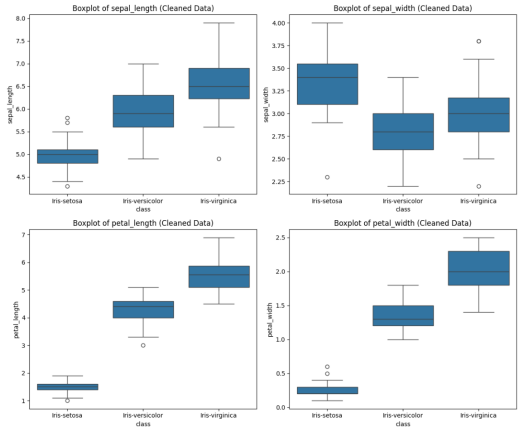

Figure 4: Boxplot(After Cleaning)

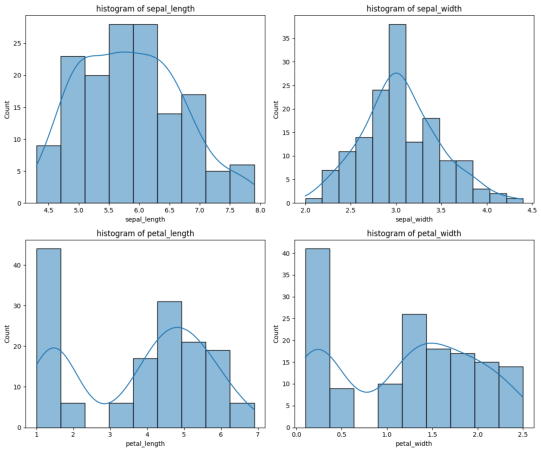

Figure 5: Histogram

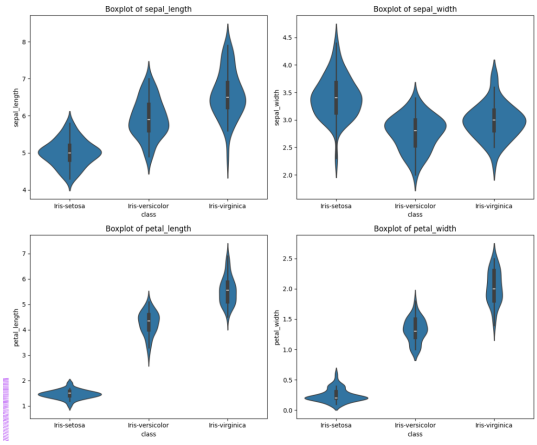

Figure 6: Violin Plot

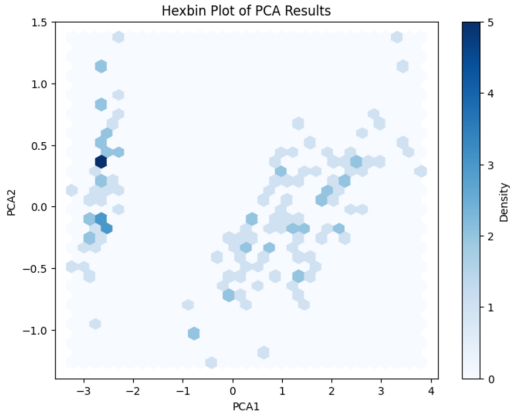

Figure 7: Hexbin Plot

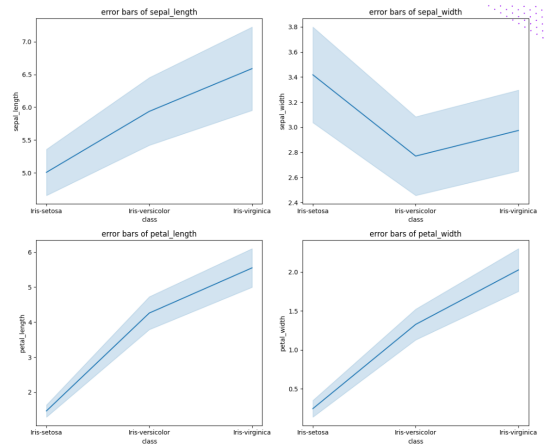

Figure 8: Error bars

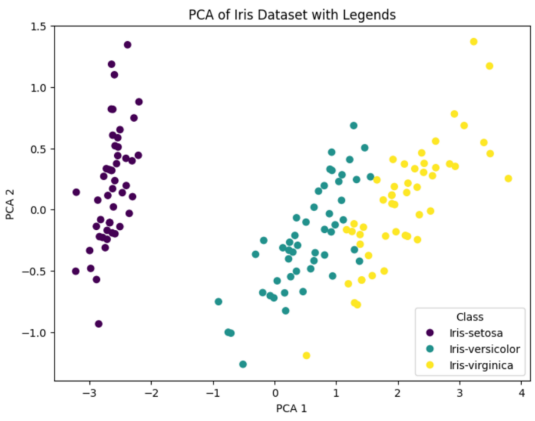

Figure 9: PCA

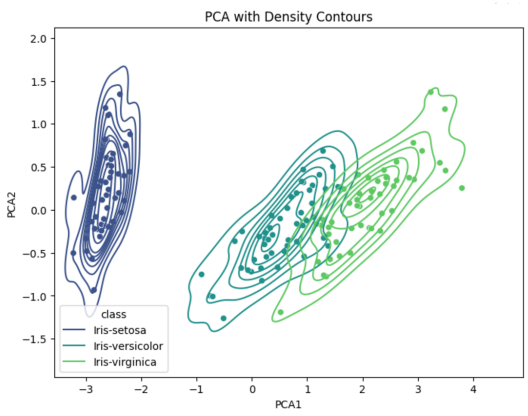

Figure 10: Density Contours

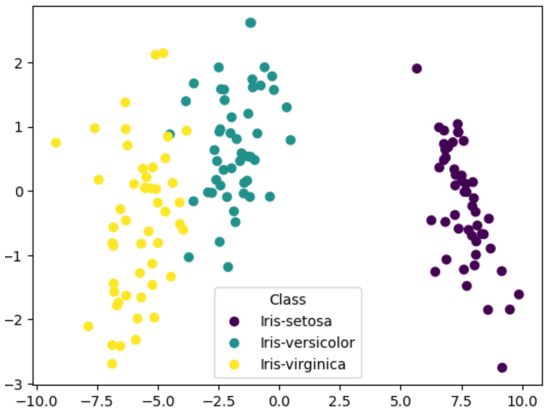

Figure 11: LDA

4.2 Model Training

The process of training machine learning models involves using the Iris dataset to evaluate the performance of various algorithms. The following models were implemented:

4.2.1 k-Nearest Neighbors (kNN)

The K-NN algorithm classifies each data point in the Iris dataset based on the K closest training samples, using Euclidean distance. By majority vote, the class (species) is predicted. K-NN works well for the Iris dataset as it can capture local patterns in the feature space, such as petal and sepal length, making it effective for distinguishing between the three species of Iris.

4.2.2 Random Forest

Random Forest constructs multiple decision trees using random subsets of the Iris dataset. Each tree votes on the predicted species, and the majority vote determines the final classification. This ensemble method handles the diversity of features in the dataset (sepal length, sepal width, petal length, and petal width) well, providing robust predictions while mitigating overfitting.

4.2.3 Support Vector Machine (SVM)

SVM constructs a hyperplane that best separates the three species in the Iris dataset by maximizing the margin between them. For this multi-class classification problem, SVM typically employs the one-vs-one or one-vs-rest strategy, where separate classifiers are trained to distinguish between pairs or groups of species, making it a powerful tool for the Iris dataset.

4.2.4 Decision Tree

A Decision Tree for the Iris dataset splits data based on features such as sepal length, sepal width, petal length, and petal width, recursively dividing the data to classify each sample into one of the three Iris species. By choosing the most informative feature at each split (based on criteria like Gini impurity), it creates a tree structure that is easy to interpret and visualize.

4.2.5 Artificial Neural Network (ANN)

In the case of the Iris dataset, an ANN can learn complex patterns between the features and species. With an input layer for the features (sepal/petal dimensions), hidden layers for learning non-linear relationships, and an output layer to predict the species, ANNs are capable of capturing intricate patterns in the dataset that linear models might miss, though they may be less interpretable.

4.2.6 Logistic Regression

Logistic Regression is applied to the Iris dataset to predict the probability of an Iris sample belonging to one of the three species. By using the features (sepal and petal dimensions) to model the log-odds of the outcome, logistic regression can be adapted for multi-class classification using techniques like the softmax function, making it a good baseline model for the dataset.

4.2.7 Gaussian Naive Bayes

Gaussian Naive Bayes assumes that the features (sepal length, sepal width, petal length, and petal width) follow a Gaussian distribution for each class. Despite the naive independence assumption, it performs well on the Iris dataset, offering a probabilistic framework for classification where the model calculates the likelihood of each species based on feature distributions.

4.2.8 Ensemble of Ensembles

In the Iris dataset, a stacked ensemble approach would combine multiple ensemble methods (e.g., Random Forest, Gradient Boosting) and use a meta-model to make final predictions. This method leverages the complementary strengths of different classifiers, improving prediction accuracy and robustness when distinguishing between the three Iris species.

4.2.9 Adaboost

AdaBoost for the Iris dataset creates a strong classifier by combining multiple weak learners, typically decision stumps. The model focuses on the misclassified samples from previous iterations, adjusting their weights to improve prediction accuracy. AdaBoost's iterative nature and focus on difficult-to-classify samples make it a good choice for the Iris dataset.

4.2.10 Gradient Boosting

Gradient Boosting builds models sequentially on the Iris dataset, with each new model correcting errors from the previous one by focusing on the residuals. The method iteratively refines the predictions, improving classification accuracy for each species. It is well-suited for the relatively small and well-defined Iris dataset, where it can capture intricate relationships between features.

4.2.11 XGBoost

XGBoost is an optimized gradient boosting algorithm that builds an ensemble of decision trees for the Iris dataset. With features like regularization, handling of missing values, and fast training, XGBoost can effectively classify the three Iris species while preventing overfitting. It is often one of the most accurate models for small to medium-sized datasets like Iris.

4.2.12 CatBoost

CatBoost can be applied to the Iris dataset to handle categorical features, though the dataset is numerical. Its strength lies in its ability to process both categorical and continuous data efficiently. The algorithm's robust handling of missing values and overfitting allows it to make accurate predictions even when working with complex feature interactions, making it a solid choice for the Iris classification problem.

Model Implementation

All models were trained and evaluated on the Iris dataset using stratified cross-validation, ensuring balanced splits between the three species. Hyperparameters for each model were tuned for optimal performance using Python libraries like `scikit-learn`, `XGBoost`, and `CatBoost`. Each model's performance was assessed based on accuracy, with particular attention to how well the models handled the distinct species of Iris.

5 Results

From EDA we observed that:
Iris setosa's features range distinctly from the other two species. While *Iris versicolor* and *Iris verginica* overlap in some features but on LDA we can observe that even *versicolor* and *verginica* also have quite distinct range.
Petal length and petal width are most correlating with their class.

The results after applying models were:

5.1 k-Nearest Neighbors (kNN)

The k-NN model achieved an accuracy of 100% on the Iris dataset. Below is the classification report for the k-NN model:

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	17
1	1.00	1.00	1.00	11
2	1.00	1.00	1.00	16
accuracy			1.00	44
macro avg	1.00	1.00	1.00	44
weighted avg	1.00	1.00	1.00	44

Figure 12: Classification Report for k-NN Model

5.2 Random Forest

The Random Forest model achieved an accuracy of 98% on the Iris dataset. Below is the classification report for the Random Forest model:

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	17
1	1.00	0.91	0.95	11
2	0.94	1.00	0.97	16
accuracy			0.98	44
macro avg	0.98	0.97	0.97	44
weighted avg	0.98	0.98	0.98	44

Figure 13: Classification Report for Random Forest Model

5.3 Support Vector Machine (SVM)

The SVM model achieved an accuracy of 97% on the Iris dataset. Below is the classification report for the SVM model:

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.89	1.00	0.94	8
2	1.00	0.92	0.96	13
accuracy			0.97	37
macro avg	0.96	0.97	0.97	37
weighted avg	0.98	0.97	0.97	37

Figure 14: Classification Report for SVM Model

5.4 Decision Tree

The Decision Tree model achieved an accuracy of 95% on the Iris dataset. Below is the classification report for the Decision Tree model:

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.88	0.88	0.88	8
2	0.92	0.92	0.92	13
accuracy			0.95	37
macro avg	0.93	0.93	0.93	37
weighted avg	0.95	0.95	0.95	37

Figure 15: Classification Report for Decision Tree Model

5.5 Artificial Neural Network (ANN)

The ANN model achieved an accuracy of 97% on the Iris dataset. Below is the classification report for the ANN model:

Classification Report for Ensemble of Ensembles

	precision	recall	f1-score	support
0	1.00	1.00	1.00	12
1	1.00	0.88	0.93	8
2	0.91	1.00	0.95	10
accuracy			0.97	30
macro avg	0.97	0.96	0.96	30
weighted avg	0.97	0.97	0.97	30

Figure 16: Classification Report for ANN Model

5.6 Logistic Regression

The Logistic Regression model achieved an accuracy of 93% on the Iris dataset. Below is the classification report for the Logistic Regression model:

Classification Report for Logistic Regression

	precision	recall	f1-score	support
0	1.00	1.00	1.00	12
1	0.88	0.88	0.88	8
2	0.90	0.90	0.90	10
accuracy			0.93	30
macro avg	0.92	0.92	0.92	30
weighted avg	0.93	0.93	0.93	30

Figure 17: Classification Report for Logistic Regression Model

5.7 Gaussian Naive Bayes

The Gaussian Naive Bayes model achieved an accuracy of 95% on the Iris dataset. Below is the classification report for the Gaussian Naive Bayes model:

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.88	0.88	0.88	8
2	0.92	0.92	0.92	13
accuracy			0.95	37
macro avg	0.93	0.93	0.93	37
weighted avg	0.95	0.95	0.95	37

Figure 18: Classification Report for Gaussian Naive Bayes Model

5.8 Ensemble of Ensembles

The Ensemble of Ensembles model achieved an accuracy of 97% on the Iris dataset. Below is the classification report for the Ensemble of Ensembles model:

Classification Report for Ensemble of Ensembles

	precision	recall	f1-score	support
0	1.00	1.00	1.00	12
1	1.00	0.88	0.93	8
2	0.91	1.00	0.95	10
accuracy			0.97	30
macro avg	0.97	0.96	0.96	30
weighted avg	0.97	0.97	0.97	30

Figure 19: Classification Report for Ensemble of Ensembles Model

5.9 Adaboost

The AdaBoost model achieved an accuracy of 98% on the Iris dataset. Below is the classification report for the AdaBoost model:

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	17
1	1.00	0.91	0.95	11
2	0.94	1.00	0.97	16
accuracy			0.98	44
macro avg	0.98	0.97	0.97	44
weighted avg	0.98	0.98	0.98	44

Figure 20: Classification Report for AdaBoost Model

5.10 Gradient Boosting

The Gradient Boosting model achieved an accuracy of 95% on the Iris dataset. Below is the classification report for the Gradient Boosting model:

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.88	0.88	0.88	8
2	0.92	0.92	0.92	13
accuracy			0.95	37
macro avg	0.93	0.93	0.93	37
weighted avg	0.95	0.95	0.95	37

Figure 21: Classification Report for Gradient Boosting Model

5.11 XGBoost

The XGBoost model achieved an accuracy of 95% on the Iris dataset. Below is the classification report for the XGBoost model:

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	0.88	0.88	0.88	8
2	0.92	0.92	0.92	13
accuracy			0.95	37
macro avg	0.93	0.93	0.93	37
weighted avg	0.95	0.95	0.95	37

Figure 22: Classification Report for XGBoost Model

5.12 CatBoost

The CatBoost model achieved an accuracy of 93% on the Iris dataset. Below is the classification report for the CatBoost model:

Classification Report for CatBoost

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	12
Iris-versicolor	0.88	0.88	0.88	8
Iris-virginica	0.90	0.90	0.90	10
accuracy			0.93	30
macro avg	0.92	0.92	0.92	30
weighted avg	0.93	0.93	0.93	30

Figure 23: Classification Report for CatBoost Model

5.13 Summary of Model Accuracies

The table below shows the accuracy of each model on the Iris dataset:

Model	Accuracy (%)
k-Nearest Neighbors (kNN)	100%
AdaBoost	98%
Random Forest	98%
Support Vector Machine (SVM)	97%
Decision Tree	95%
Gaussian Naive Bayes	95%
XGBoost	95%
Gradient Boosting	95%
Logistic Regression	93%
CatBoost	93%
Ensemble of Ensembles	97%
Artificial Neural Network (ANN)	97%

Table 1: Summary of Accuracies for Each Model on the Iris Dataset

1. Also earlier, in the correlation matrix we observed that petal width and petal length had the highest correlation. Hence we applied KNN on the combination of two features out of 4 and got acurracy more than 95% when either of the petal length and petal width were in the combination.(figure 24)

2. Then we tried the same with one feature only and got accuracy 100% in case of petal width.(figure 25)

FEATURES	ACCURACY
sepal_length, sepal_width	0.80
sepal_length, petal_length	0.98
sepal_length, petal_width	0.98
sepal_width, petal_length	0.98
sepal_width, petal_width	0.95
petal_length, petal_width	0.98

Figure 24

FEATURES	ACCURACY
sepal_length	0.77
petal_length	0.98
petal_width	1.00
sepal_width	0.55

Figure 25

6 Conclusions

The conclusions that were drawn are:

- kNN (metric = 'euclidean', n neighbors = 9, weights = 'distance') proved to be the best model with accuracy of 100% in our case.This might be because our dataset is simple and well separated and due to the non parametric nature of kNN.

- While trying PCA for SVM the accuracy reduced from 97% to 95%. No need of PCA since the dataset is small. Hence dimensionality reduction leads to loss of information.

- For our dataset Petal width proved to be the best feature in classification.

7 Future Scopes

1. Synthetic Data Generation: Using techniques like GANs (Generative Adversarial Networks) to create synthetic Iris data to expand its size and complexity. Experimenting with data augmentation methods to simulate variability in measurements or add noise.

2. Feature Expansion Augmenting the dataset with new features, such as environmental factors (e.g., location, soil type, or climate conditions) that might influence iris growth. Simulating additional flower attributes like petal texture, aroma intensity, or even genetic data.

3. Expansion for Real-World Relevance Adding images of iris flowers to combine the structured dataset with image classification tasks using convolutional neural networks (CNNs). Annotating data with additional classes to explore multi-class or multi-label classification tasks.

References

[1] A. Shukla, A. Agarwal, H. Pant, P. Mishra, *Flower classification using supervised learning*, Int. J. Eng. Res., 2020.

[2] T. Srinivas Rao, M. Hema, K. Sai Priya, K. Vamsi Krishna, M. Sakhavath Ali, D. Hemalatha, *Iris Flower Classification Using Machine Learning*, 2023 3rd International Conference on Intelligent Technologies (CONIT), 2023, pp. 1-4.

[3] Z. Faiz Hussain, H. Raad Ibraheem, M. Alsajri, A. Hussein Ali, M. Arfian Ismail, S. Kasim, T. Sutikno, *A new model for iris data set classification based on linear support vector machine parameter's optimization*, 2023.

[4] K. Kishotha, S. B. Mayurathan, *Machine Learning Approach to Improve Flower Classification Using Multiple Feature Sets*.

[5] GeeksforGeeks, *Website*, Available at: `https://www.geeksforgeeks.org/`.

[6] Scikit-learn, *Documentation for scikit-learn library*, Available at: `https://scikit-learn.org/stable/modules/generated`.

[7] Towards Data Science, *Categorical Encoding Using Label Encoding and One-Hot Encoding*, Available at: `https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd`.

[8] Google Scholar, *Website*, Available at: `https://scholar.google.com/`.

[9] Google India, *Website*, Available at: `https://www.google.co.in/`.

[10] OpenAI, *ChatGPT*, Available at: `https://openai.com/chatgpt/`.