DATA MINING MASTER MLAIM

JUIN 2024

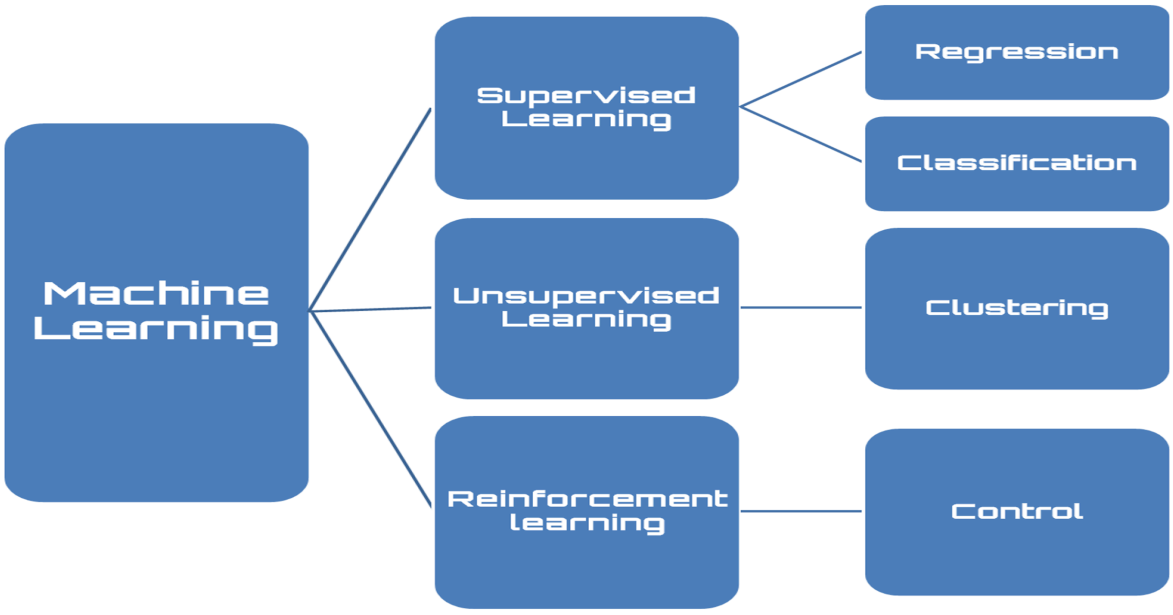
Classification of the Iris Dataset Using Logistic Regression

PREPARED BY: AYMANE IBN EL QORCHY

Encadrer par : PR ISMAIL BETTIOUI

Introduction & Background

Data classification is a crucial component of machine learning, aimed at categorizing datasets into predefined classes. This mini-project focuses on applying classification techniques to the Iris dataset, a classic dataset in the fields of statistics and machine learning. Introduced by Ronald A. Fisher in 1936, the Iris dataset is widely used to demonstrate data analysis and supervised learning concepts due to its simplicity and balance.

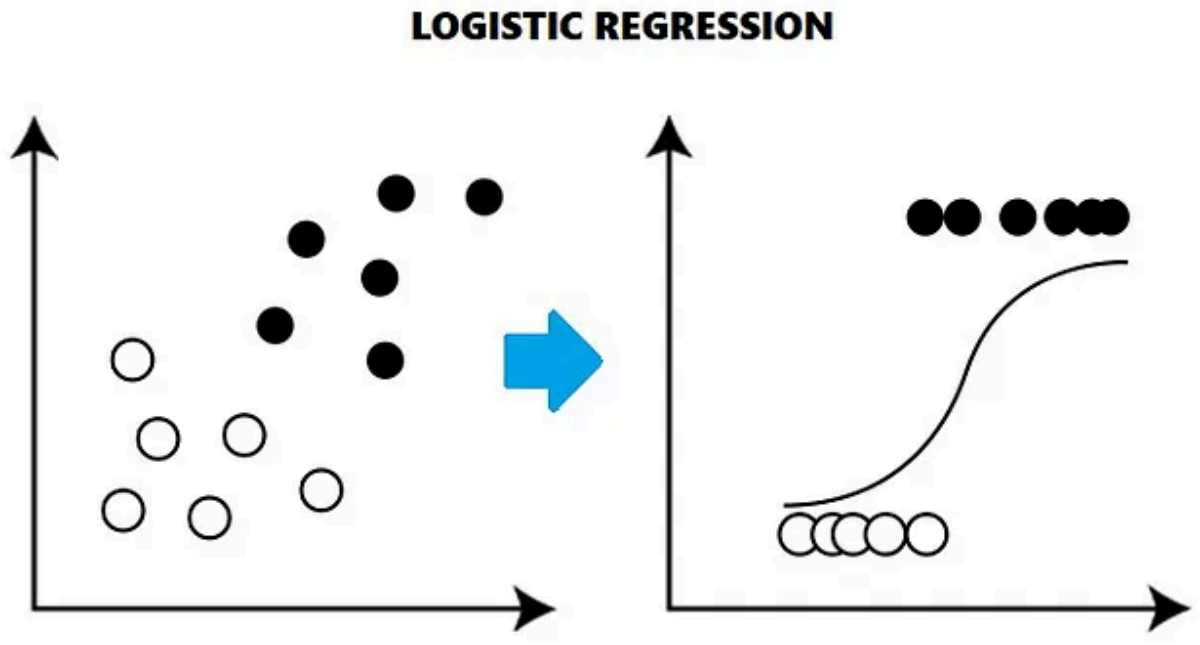


The Iris dataset is one of the most famous datasets in the field of machine learning and statistics. It was introduced by the British statistician and biologist Ronald A. Fisher in his 1936 paper "The Use of Multiple Measurements in Taxonomic Problems" as an example of discriminant analysis. The dataset comprises 150 observations of iris flowers from three different species: Iris setosa, Iris versicolor, and Iris virginica.

It showed that 78% of all phishing websites use her SSL protection, which is only used on genuine websites. In our project, we will employ two classification methods: decision trees and logistic regression

Logistic Regression

1. **What's and Why Logistic Regression?**



What is Logistic Regression?

It's a classification algorithm that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome.

For example, the response variable has two values, pass and fail, when we have to predict whether a student will pass or fail an exam when the number of hours spent studying is taken into account.

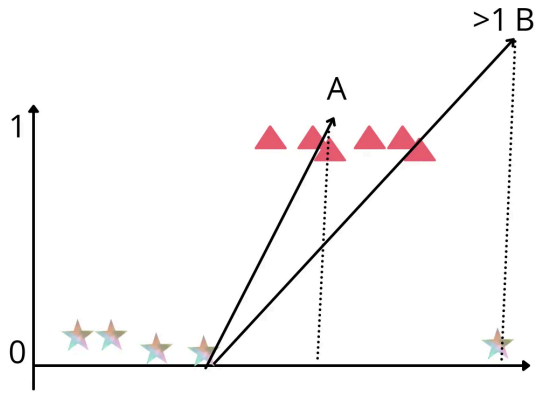
This type of a problem is referred to as Binomial Logistic Regression, where the response variable has two values 0 and 1 or pass and fail or true and false. The situation in which a response variable may have three or more possible values is dealt with by the multinomial logistic regression.

In summary:

- Logistic Regression is a way of using numbers to figure out how likely it is that something will happen. This technique is used to find out how likely different outcomes are in a situation where you already know some information.
- Logistic regression uses a maths tool called the **logit function** to figure out how likely something is to happen based on other things that might affect it.
- The **sigmoid functions** change chances into yes or no answers that can be used to predict things.

Why do we need logistic regression?

If we can use linear regression then why logistic regression, sometimes we have outliers in our data, then in linear regression we need to create a best fit line based on the data point that mispredicts the output values, so here in this one If our linear regression fails.

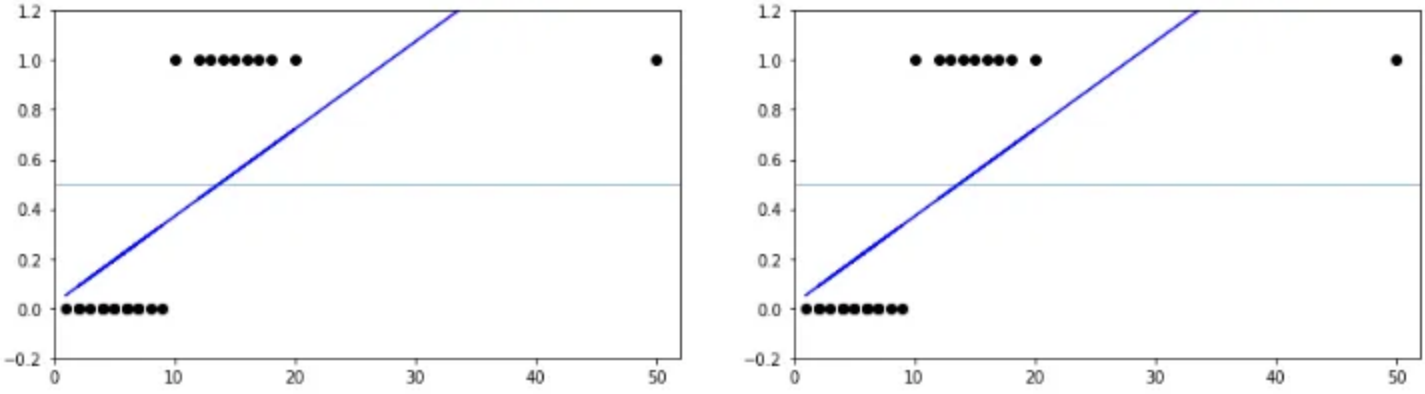


So, two reasons why linear regression should not be used for binary classification:

- Whenever I have a lot of outliers, our best-fit line can completely change direction.
- No matter what output I get, most of the time I get more than 1 and less than 0, so to solve this problem we have to use logistic regression.

So in regression we are predicting continuous values but if we want to predict categorical values like True or False , right or wrong, Yes or no then in this case the regression model our linear regression doesn't work, so to solve this kind of problem we have to use logistic regression. In logistic regression we play with probabilities (The odds of our output variable).

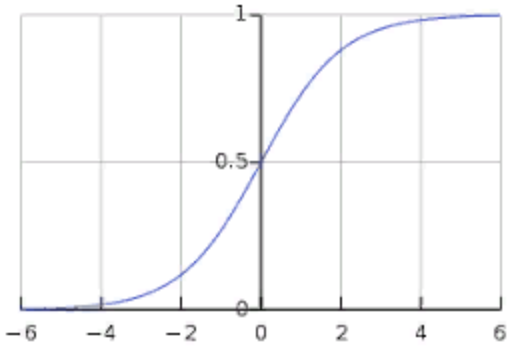
Linear regression models can generate the predicted probability as any number ranging from negative to positive infinity, whereas probability of an outcome can only lie between 0< P(x)<1.



Also, Linear regression has a considerable effect on outliers. To avoid this problem, **log-odds** function or **logit** function is used.

2. **Types of Logistic Regression.**

Binary Logistic Regression: This is the most common type of logistic regression, and it is used when the response variable has only two possible outcomes, such as pass/fail, yes/no, or true/false. Binary logistic regression is often used in medical research, marketing, and social sciences.



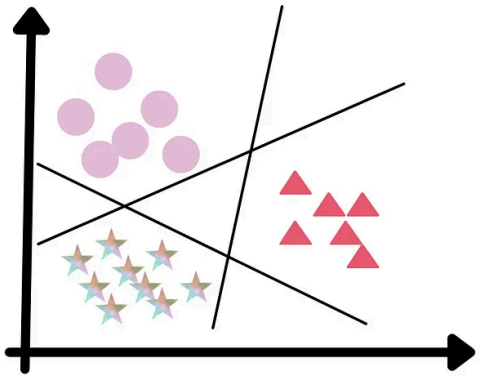
the linear function is basically used as an input to another function such as g in the following relation:

$$h_\emptyset(x) = g(\emptyset^T x) where 0 \leq h_\emptyset \leq 1$$

Here, g is the logistic or sigmoid function which can be given as follows:

$$g(z) = \frac{1}{1 + e^{-z}} where\ z = \emptyset^T x$$

Multinomial Logistic Regression: This type of logistic regression is used when the response variable has more than two possible outcomes, such as red/green/blue, small/medium/large, or high/medium/low. Multinomial logistic regression is often used in market research, political science, and psychology.



Ordinal Logistic Regression: This type of logistic regression is used when the response variable has a natural ordering, such as low/medium/high or strongly disagree/disagree/neutral/agree/strongly agree. Ordinal logistic regression is often used in surveys and questionnaires to predict how people will respond to different questions.

And other types of logistic regression such as: Conditional Logistic Regression, Penalised Logistic Regression…

3. Assumptions of Logistic Regression.

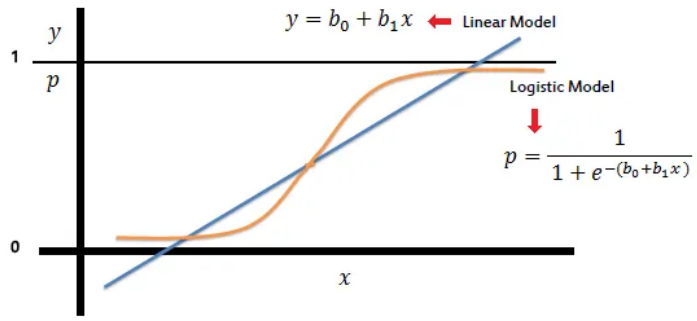
Even Though Logistic Regression belongs to the Linear models, no assumptions are made about linear regression models such as e.g.:

- Does not require a linear relationship between dependent and independent variables.
- Error conditions should not normally be sent.
- homoscedasticity not required.

it has some assumptions of its own:

- ➔ It is assumed that there is little or no collinearity between the independent variables.The best way to check for collinearity is to run a VIF (Variance Inflation Factor).
- ➔ The independent variables are assumed to be linearly related to the logarithmic parts.This can be checked with the Box-Tidwell test.
- ➔ Assumes a large sample for a good prediction.
- ➔ The observations are assumed to be independent of each other.
- ➔ With continuous predictors (independent variables), there are no influencing factors (outliers).This can be checked with IQR, Z-Score or displayed with box or fiddle plot.
- ➔ Logistic regression with 2 classes where the dependent variable is binary and ordered logistic regression requires the dependent variable to be ordered.

4. The Logistic Model



The Logistic Regression instead for fitting the best fit line,condenses the output of the linear function between 0 and 1.

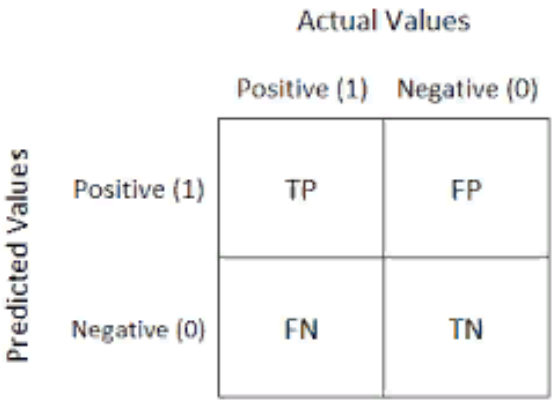
In the formula of the logistic model, when $b_0 + b_1 * x = 0$, then the p will be 0.5, Similarly, $b_0 + b_1 * x > 0$, then the p will be going towards 1 and $b_0 + b_1 * x < 0$,then the p will be going towards 0.

5. **Performance of Logistic Regression model**

To evaluate the performance of a logistic regression model, **Deviance** is used in lieu of sum of squares calculations.

- **Null** Deviance indicates the response predicted by a model with nothing but an intercept.
- **Model** deviance indicates the response predicted by a model on adding independent variables. If the model deviance is significantly smaller than the null deviance, one can conclude that the parameter or set of parameters significantly improved model fit.

Another way to find the accuracy of a model is by using the Confusion Matrix.



The accuracy of the model is given by:

$$\frac{\text{True Positive + True Negatives}}{\text{True Positive + True Negatives + False Positives + False Negatives}}$$

6. **Applications of Logistic Regression**

Business Use-Cases:

- Qualify leads
- Recommend products
- Anticipate rare customer behaviour

Advantages in Production Deployment:
The benefits of logistic regression from an engineering perspective make it more favorable than other, more advanced machine learning algorithms.

- Ease of use
- Interpretability
- Scalability
- Real-time predictions

7. Improving our Model

There are multiple methods to improve your Logistic Regression model, There are a few techniques (in preprocessing) that are employed for model improvement amongst them and which we well cover in the next chapter:
Binary output variable: Transform your output variable into 0 or 1, also called encoding Categorical Data
Feature Scaling: Features can come in different orders of magnitude. Features of different scales convert slower (or not at all) with gradient descent.
And also **Regularization…..**

Data Classification

1. Data collection and description

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#importer les bibliotheque necessaire
from sklearn.linear_model import LinearRegression
from sklearn.metrics import classification_report, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

```
#chargement des donneees
data=sns.load_dataset('iris')
data.head()
```

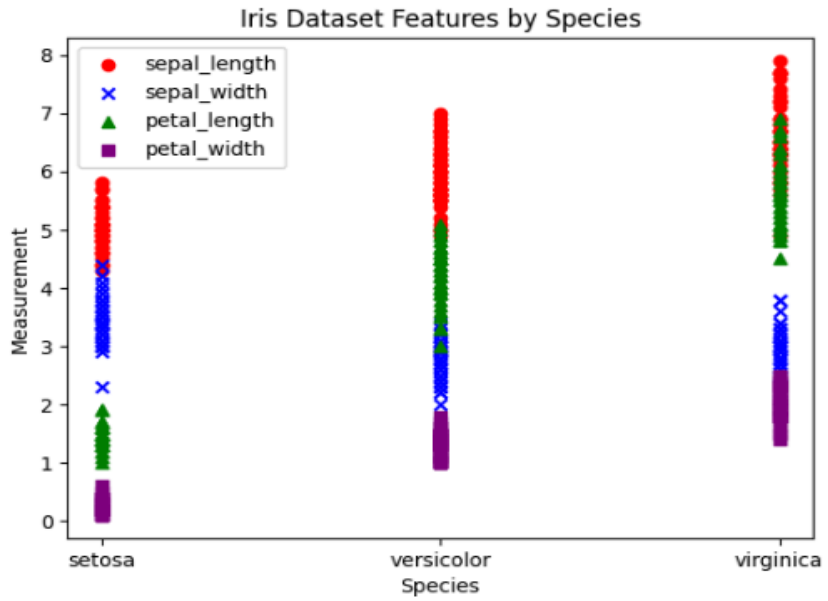
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

The Iris dataset consists of 150 samples of iris flowers, each characterized by four features: sepal length, sepal width, petal length, and petal width. These features are used to classify each sample into one of three species: Iris setosa, Iris versicolor, and Iris virginica.

Visualizing data is a crucial step in exploring and understanding datasets. In the context of the Iris dataset, the provided code snippet demonstrates how to create a scatter plot to visualize the measurements of four key features—sepal length, sepal width, petal length, and petal width—across different species of iris flowers.

```python
# Extracting data for each feature
pltX = data['species']

pltY1 = data['sepal_length']
pltY2 = data['sepal_width']
pltY3 = data['petal_length']
pltY4 = data['petal_width']

# Scatter plots
plt.scatter(pltX, pltY1, color='red', label='sepal_length', marker='o')
plt.scatter(pltX, pltY2, color='blue', label='sepal_width', marker='x')
plt.scatter(pltX, pltY3, color='green', label='petal_length', marker='^')
plt.scatter(pltX, pltY4, color='purple', label='petal_width', marker='s')

# Étiquettes des axes
plt.xlabel('Species')
plt.ylabel('Measurement')

# Title
plt.title('Iris Dataset Features by Species')

# Légende
plt.legend()

# Affichage du graphique
plt.show()
```

Visualization:



separating X and Y, we prepare the dataset for building machine learning models that can predict the species of an iris flower based on its measured characteristics. This separation

ensures clarity in data handling and enhances the efficiency of model training and evaluation processes

```
# Séparation des caractéristiques (X) et des étiquettes (y) sauf la derniere
X = data.iloc[:, :-1]
Y = data.iloc[:, -1]

# Affichage des premières lignes de X et Y
print(X.head())
print(Y.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width
0           5.1          3.5           1.4          0.2
1           4.9          3.0           1.4          0.2
2           4.7          3.2           1.3          0.2
3           4.6          3.1           1.5          0.2
4           5.0          3.6           1.4          0.2
0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: object
```

he provided data snippets showcase the measurements of sepal length, sepal width, petal length, and petal width for several iris flowers, along with their corresponding species labels (target variable).

Splitting data into training and testing subsets is foundational in machine learning workflows, enabling robust model evaluation and validation before deployment in real-world applications.

```
# Division des données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
# Affichage des formes des ensembles d'entraînement et de test
print("\nFormes des ensembles de données :")
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
Formes des ensembles de données :
X_train shape: (120, 4)
X_test shape: (30, 4)
y_train shape: (120,)
y_test shape: (30,)
```

> Logistic regression initializes a logistic regression model object. Logistic regression is a commonly used classification algorithm that predicts the probability of a binary outcome or classifies data into categories based on input features.

```python
# 6. Initialisation du modèle
model = LogisticRegression()

# Entraînement du modèle
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logist
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as sh
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver option
    https://scikit-learn.org/stable/modules/linear_model.html#logist
  n_iter_i = _check_optimize_result(
LogisticRegression()
```

Predicting with a trained model is a fundamental step in machine learning workflows where the model uses learned patterns from the training data to make predictions on new, unseen data points. Here's how it works and its significance

- After training a logistic regression model on the Iris dataset, you can predict the species of new iris flowers based on their sepal and petal measurements.
- For instance, given a set of sepal and petal measurements for a new iris flower (`X_test`), the model predicts its species (`y_pred`), such as 'setosa', 'versicolor', or 'virginica'.

Predicting with a trained model is essential for leveraging machine learning to automate decision-making processes based on data patterns, enhancing efficiency and accuracy in various applications across different industries,These are the model's predictions for the species of iris flowers based on their features (such as sepal and petal measurements).

this project, we embarked on a journey to classify the Iris dataset using two popular machine learning algorithms: logistic regression and decision trees. The Iris dataset, renowned for its simplicity yet richness in foundational machine learning tasks, provided an ideal playground to explore these algorithms and their application in classification.

[24]:
```
# Prédiction sur l'ensemble de test
prediction=model.predict(X_test)
print(prediction)
print(y_test)
```

```
['versicolor' 'setosa' 'virginica' 'versicolor' 'versicolor' 'setosa'
 'versicolor' 'virginica' 'versicolor' 'versicolor' 'virginica' 'setosa'
 'setosa' 'setosa' 'setosa' 'versicolor' 'virginica' 'versicolor'
 'versicolor' 'virginica' 'setosa' 'virginica' 'setosa' 'virginica'
 'virginica' 'virginica' 'virginica' 'virginica' 'setosa' 'setosa']
73     versicolor
18         setosa
118     virginica
78     versicolor
76     versicolor
31         setosa
64     versicolor
141     virginica
68     versicolor
82     versicolor
110     virginica
12         setosa
36         setosa
9          setosa
19         setosa
56     versicolor
104     virginica
69     versicolor
55     versicolor
132     virginica
```

```
# Évaluation du modèle
print(classification_report(y_test, prediction))
print(accuracy_score(y_test, prediction))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

1.0
```

Conclusion

In this project, we embarked on a journey to classify the Iris dataset using two popular machine learning algorithms: logistic regression . The Iris dataset, renowned for its simplicity yet richness in foundational machine learning tasks, provided an ideal playground to explore these algorithms and their application in classification.