✨ Day 5 – Logistic Regression & Classification

📅 **Date:** 30 June 2025

Today's session was quite interesting as we stepped into the world of **classification problems** in machine learning. Unlike previous regression tasks where we were predicting continuous values, here the focus was on predicting **categories** or **labels**.

🌸 Getting Started with Classification

We first understood what classification is — simply put, it's when the output we're trying to predict belongs to specific classes like *yes/no*, *male/female*, or, like in our case today, the **species of flowers**.

A few examples where classification is used:

- Whether an email is spam or not
- Predicting if a patient has a disease
- Classifying images (cat, dog, car, etc.)
- Approving or rejecting a bank loan

Today, we mainly worked with **multi-class classification** using the **Iris dataset**, which involves classifying flowers into one of three species based on four features (like petal length and sepal width).

🔧 What We Did Practically

Using Python and the scikit-learn library, we followed these steps:

1. **Loaded the Iris dataset** using load_iris()
2. **Split the data** into training and testing sets
3. **Built a Logistic Regression model** using LogisticRegression()
4. **Trained the model** and used it to make predictions
5. **Checked model accuracy** using accuracy_score()
6. Finally, **predicted the species** of a new flower sample

💻 Code We Used (Explained in Simple Steps):
python
CopyEdit

```
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

This part loads all the necessary libraries.
load_iris() gives us the dataset, and LogisticRegression is our algorithm.
train_test_split helps in splitting the data into training and testing sets, while accuracy_score helps us check how well the model performed.

python
CopyEdit

```
iris = load_iris()
X = iris.data  # input features
y = iris.target  # output labels (species)
labels = iris.target_names
```

X is the main data (features of flowers), and y contains the labels (species).
labels is just to get readable names like “setosa”.

python
CopyEdit

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Splits 80% data for training and 20% for testing.

python
CopyEdit

```
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
```

This trains the model. The max_iter=200 ensures the algorithm gets enough cycles to find the best solution.

python
CopyEdit

```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

We then predict on the test data and check how accurate the predictions are.

python
CopyEdit

```
sample = [[5.0, 3.6, 1.4, 0.2]]
prediction = model.predict(sample)
print(f"Predicted class: {labels[prediction[0]]}")
```

Here, we give the model a new flower with some measurements, and it tells us the species. Pretty cool!

🚢 Bonus: Titanic Dataset Introduction

We also had a quick look at the **Titanic dataset** from Kaggle. It’s used to predict whether a passenger **survived or not**, based on details like age, gender, ticket class, and fare.
It’s a **binary classification** problem, and we’ll be using it in upcoming sessions when we learn more algorithms like Decision Trees.

💥 Final Thoughts

This session helped me clearly understand the difference between **classification** and **regression**, and gave hands-on experience with one of the simplest yet powerful classification techniques — logistic regression.