✨ Day 5 – Logistic Regression and Classification

🗓️ **Date:** 30 June 2025

📓 Today's Highlights:

Today's focus was on building and evaluating a **classification model** using **Logistic Regression** — one of the most widely used algorithms in supervised learning. We worked with the **Iris dataset**, a classic multi-class classification problem involving three flower species. We also reviewed and explored the **Titanic Dataset** on Kaggle, which involves predicting survival outcomes based on passenger data.

This session enhanced our understanding of model training, evaluation metrics, and prediction workflows in Python using **scikit-learn**.

□ What is Classification?

Classification is a machine learning approach where the goal is to predict categorical class labels.
Some key characteristics of classification:

- Output variable is **discrete** (e.g., yes/no, male/female)
- Used in real-life applications such as:
  - Loan approval systems
  - Medical diagnosis
  - Email spam filtering
  - Image recognition

We focused on **multi-class classification** today, where there are more than two possible output classes (e.g., three flower species in the Iris dataset).

🛠️ Practical Implementation:

We used **scikit-learn** to:

- Load and explore the **Iris dataset**
- Split the dataset into training and testing sets using `train_test_split()`
- Build a **Logistic Regression** model
- Fit the model on training data and make predictions on test data
- Calculate **accuracy score** using `accuracy_score()`
- Predict the class of a new flower sample

This helped us understand the **full ML workflow** — from data preprocessing to model evaluation and interpretation.

📌 Python Code Used:

```
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

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target
labels = iris.target_names

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

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

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

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

📊 Output & Results:

- The model achieved high accuracy in predicting the species of flowers.
- The output displayed:
  - The **accuracy score** (e.g., ~96.67%)
  - The **predicted class** for the new flower sample (e.g., `setosa`)
- This hands-on activity reinforced how classification models work in real ML scenarios.

📂 Additional Exploration – Titanic Dataset:

In addition to the Iris dataset, we briefly explored the **Titanic Dataset** on Kaggle:

- This dataset is used to predict whether a passenger survived the Titanic disaster.
- It contains columns like `Age`, `Sex`, `Fare`, `Pclass`, etc.
- It is a **binary classification** problem: survived (1) or not (0).
- This dataset will be used in future sessions for implementing decision trees and other classifiers.