DEVELOPING A CLASSIFICATION MODEL USING LOGISTIC REGRESSION

Classification is a type of supervised learning in machine learning where the goal is to assign input data to predefined categories or labels. It is used when the output variable is categorical (e.g., Yes/No, Spam/Not Spam, Disease/No Disease). Classification models learn from historical labeled data and make predictions for new data points.

Classification deals with **discrete data**

Examples of classification problems:

- Email spam detection (Spam / Not Spam)
- Customer churn prediction (Will Leave / Will Stay)
- Disease diagnosis (Has Disease / No Disease)
- Image recognition (Dog / Cat / Human)

To decide whether a problem should be solved using a classifier, ask these questions:

1. Is the output variable categorical or continuous?

- If categorical (fixed labels): → Use classification
- If continuous (numerical values): → Use regression

Example:

- Predicting whether a student passes or fails (Classification)
- Predicting a student's exact score in an exam (Regression)

2. Are you assigning data to predefined classes?

- If you need to label data into groups (e.g., "Dog" or "Cat") → Classification
- If you are calculating a number (e.g., predicting house price) → Regression

Activities

You are given a dataset containing information about customers, and your task is to build a model that predicts whether a customer will churn a product (1) or not (0) based on given features. Customer churn is the number of customers who stop using a business's products or services over a period of time. It's also known as customer attrition.

Download the dataset from Kaggle.
https://www.kaggle.com/datasets/hassaneskikri/online-retail-customer-churn-dataset

Steps

Step 1: Import Required Libraries

Use pandas, numpy, matplotlib, seaborn, and scikit-learn.

Step 2: Load the Dataset

- Load the dataset into a Pandas DataFrame.
- Display the first few rows (df.head()).

Step 3: Data Preprocessing

- Check for missing values and handle them.
- Perform feature scaling if necessary (Standardization/Normalization).
- Use numerical columns only create a copy of the dataset with the numeric columns only. Ignore the catogorcal data for now.
- Split the dataset into training (80%) and testing (20%) sets.

Step 4: Train a Logistic Regression Model

- Use scikit-learn's LogisticRegression class.
- Fit the model using the training data.

Step 5: Model Evaluation

Evaluate the model using:

- Accuracy Score
- Confusion Matrix (TP, FP, FN, TN)
- Precision, Recall, and F1-score