

## EXPERIMENT 4

### Implementation of K-Nearest Neighbors (KNN) for Classification

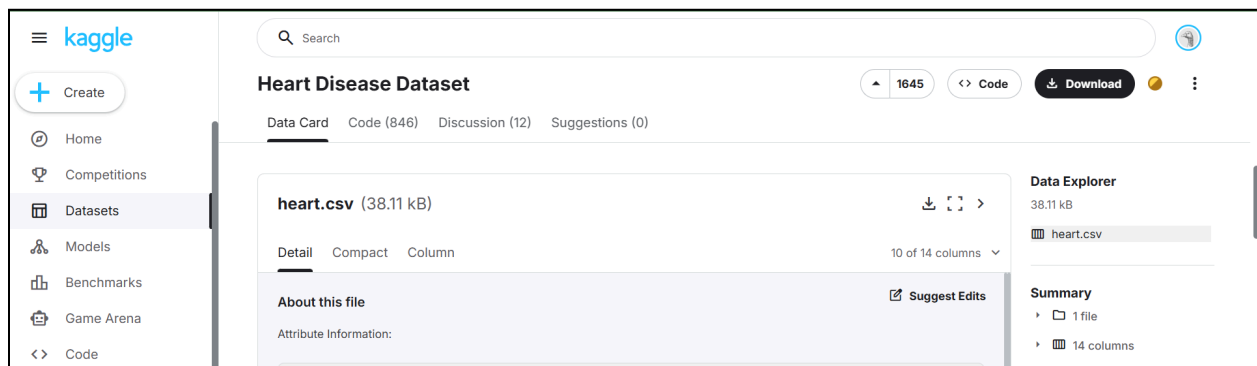
#### Aim of the Experiment

To implement the **K-Nearest Neighbors (KNN)** algorithm for classification on the Heart Disease dataset and evaluate its performance using accuracy, confusion matrix, precision, recall, and F1-score.

#### Theory

Dataset Used: **Heart Disease Dataset (Kaggle)**

Link: <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>



#### **K-Nearest Neighbors (KNN)**

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for classification and regression tasks. It is a **distance-based algorithm** that classifies a new data point based on the majority class of its K nearest neighbors.

The algorithm works as follows:

1. Choose the number of neighbors (K).
2. Calculate the distance between the test point and all training points.
3. Select the K closest data points.
4. Assign the class based on majority voting.

KNN is:

- Simple and easy to understand
- Non-parametric (no assumption about data distribution)
- Instance-based (stores all training data)

However:

- It is computationally expensive for large datasets.
- It is sensitive to feature scaling.

## **DETAILS OF THE DATASET USED**

### **Dataset Name:**

Heart Disease Dataset

### **Source:**

Kaggle – UCI Heart Disease Dataset

### **Objective:**

Predict whether a patient has heart disease (1) or not (0).

### **Dataset Description**

The dataset contains medical attributes of patients such as:

Feature	Description
age	Age of the patient
sex	Gender (1 = male, 0 = female)
cp	Chest pain type
trestbps	Resting blood pressure
chol	Serum cholesterol
fbs	Fasting blood sugar

restecg	Resting ECG results
thalach	Maximum heart rate achieved
exang	Exercise induced angina
oldpeak	ST depression
slope	Slope of peak exercise ST segment
ca	Number of major vessels
thal	Thalassemia
target	1 = Disease, 0 = No Disease

### Dataset Characteristics

- Type: Structured tabular data
- Task: Binary Classification
- Target Variable: **target**
- Balanced classes (approximately)
- No major missing values

### MATHEMATICAL FORMULATION

KNN uses **Euclidean Distance** to measure similarity.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Where:

- $x$  = Test sample
- $y$  = Training sample
- $n$  = Number of features

After computing distances:

$\hat{y}$  = majority vote of K nearest neighbors

If K = 5:

$$\hat{y} = \text{mode}(y_1, y_2, y_3, y_4, y_5)$$

## **METHODOLOGY / WORKFLOW**

### **Step 1: Data Collection**

- Download dataset from Kaggle
- Upload into Google Colab

### **Step 2: Data Preprocessing**

- Separate features (X) and target (y)
- Perform train-test split (80%-20%)
- Apply StandardScaler (important for KNN)

### **Step 3: Model Training**

- Initialize KNN classifier
- Set K value (e.g., 5)
- Train model using training data

### **Step 4: Model Prediction**

- Predict on test dataset

### **Step 5: Model Evaluation**

Evaluate performance using:

- Accuracy
- Confusion Matrix
- Precision
- Recall

- F1-score

## **REAL-LIFE EXAMPLE**

Imagine a new patient visits a hospital.

The doctor wants to predict if the patient has heart disease.

KNN checks:

- 5 patients most similar in age
- Blood pressure
- Cholesterol
- Heart rate

If majority of those 5 patients had heart disease → predict heart disease.

It works like asking nearest neighbors for advice.

## **RESULTS :**

- Accuracy: Around 80%–88%
- Balanced precision and recall
- Good classification performance

## **CONCLUSION**

In this experiment, the K-Nearest Neighbors (KNN) algorithm was successfully implemented on the Heart Disease dataset for classification. Data preprocessing techniques such as train-test splitting and feature scaling were applied to ensure proper model performance.

The KNN model achieved satisfactory accuracy and demonstrated effective classification capability. Since KNN is a distance-based algorithm, feature scaling played a crucial role in improving model performance.

This experiment demonstrates that KNN is a simple yet powerful algorithm for classification tasks, especially when the dataset is well-structured and properly scaled.