Bishop's University
Department of Computer Science
CS316 – Introduction to AI
Assignment 3: k-Nearest Neighbors (k-NN) on Breast Cancer Dataset

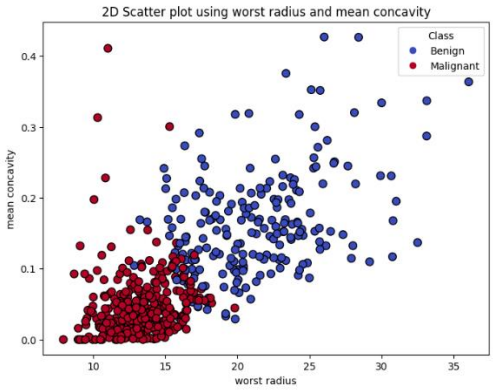
Dataset:
The Breast Cancer Wisconsin (Diagnostic) Dataset is commonly used in machine learning for binary classification tasks related to diagnosing breast cancer. It includes 569 samples of breast tumor cases, each labeled as either malignant (cancerous) or benign (non-cancerous).
Each sample has 30 numeric features derived from images of cell nuclei obtained via a biopsy. These features describe the characteristics of cell nuclei, including mean radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.
The dataset is well-suited for k-nearest neighbors (k-NN) classification and other machine learning techniques due to its clear label distinctions and feature variety, enabling effective feature visualization and model training.

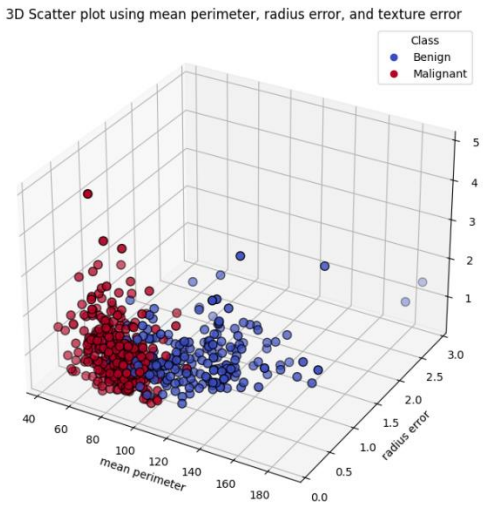
Objectives
1. Implement and analyze a k-Nearest Neighbors (k-NN) model for classifying breast cancer data.
2. Visualize the dataset with scatter plots using randomly selected features.
3. Evaluate model performance across various train/test splits and different values of k.
4. Display model accuracy using a heatmap.

Requirements
1. Dataset Overview:
   - Use the Breast Cancer dataset from sklearn.datasets.
   - This dataset includes information on tumor characteristics with 569 samples and 30 numeric features. The target variable indicates whether the tumor is malignant (1) or benign (0).
   - Load the dataset into a pandas DataFrame for easier manipulation and visualization.
   - Display the first 5 rows of the DataFrame to understand its structure and feature values.

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	target
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890	0
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902	0
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758	0
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300	0
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678	0

2. Data Scatter Plot:
   - Create a 2D scatter plot using two randomly selected features from the dataset. Ensure that the plot differentiates classes (malignant and benign).
   - Create a 3D scatter plot using three randomly selected features, color-coding the data points to show class distinctions.





2D scatter 3D scatter

2. Model Training and Evaluation:
   - Split the data into 80% for training and 20% for test.
   - Implement a k-NN classifier with a default k=5 and train on the training set.
   - Compute the accuracy of the trained k-NN on the test set.

3. Exploring Different Values of k and different split size:
   - Evaluate the k-NN model's performance with multiple values of k (1 through 10) and each of the split sizes: 10%, 20%, and 40%.
   - Create a heatmap to visualize the accuracy for each combination of k values and test sizes.
   - Label the axes with "k Value" and "Test Size (%)" and include a color bar showing accuracy values.

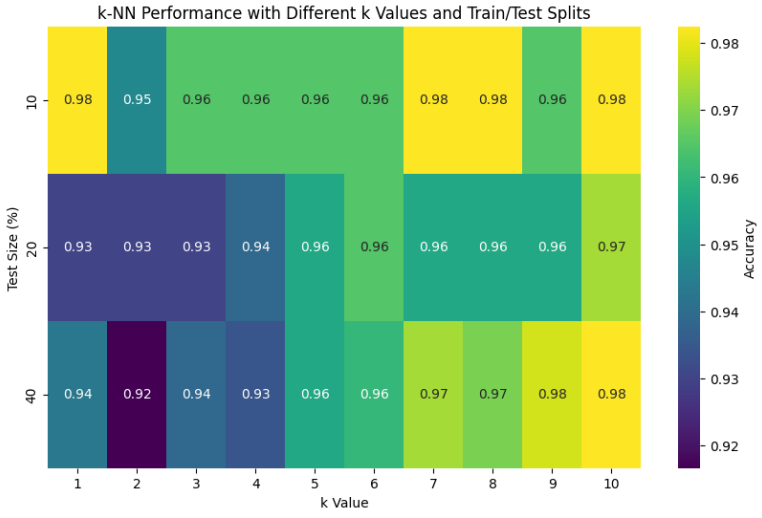


Figure: Heat mat showing the performance of the k-NN classifier for different values of k and different split sizes.