

Lab 7 KNN - Classification

```
In [1]:
import from sklearn.datasets load_breast_cancer
# Load dataset
data = load_breast_cancer()
X = data.data
y = data.target
print("Original Shape:", X.shape)
```

Original Shape: (569, 30)

```
In [2]:
import matplotlib.pyplot as plt
import from sklearn.model_selection train_test_split
import from sklearn.neighbors KNeighborsClassifier
import from sklearn.metrics accuracy_score, precision_score, recall_score,
f1_score
```

```
In [3]:
# define function for train test split and KNN

def knn_classify_eval (X, y, method_name="KNN", k=5):
    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

    # K-Nearest Neighbors (KNN)
    knn = KNeighborsClassifier(n_neighbors=5) # You can change n_neighbors
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)

    #matrix
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    # print
    print(f"==== Method: {method_name} Evaluation ====")
    print(f"Accuracy: {accuracy}")
    print(f"Precision: {precision}")
    print(f"Recall: {recall}")
    print(f"F1 Score: {f1}")
    print()

    #visualize
    plt.figure(figsize=(8, 6))
    plt.scatter(range(len(y_test)), y_test, label="Actual", alpha=0.7,
color="blue")
    plt.scatter(range(len(y_pred)), y_pred, label="Predicted", alpha=0.7,
color="red", marker = "x")
    plt.title(f"{method_name} Data Visualization")
    plt.xlabel("Sample Index ")
    plt.ylabel("Class")
```

```
plt.legend()
plt.show

return {"Method" : method_name, "Accuracy": accuracy, "Precision": precision,
"Recall": recall, "F1": f1}
```

Exercises

KNN

```
In [4]:
pca_results = knn_classify_eval(X, y, "KNN", k=5)
```

```
==== Method: KNN Evaluation ====
Accuracy: 0.9590643274853801
Precision: 0.9469026548672567
Recall: 0.9907407407407407
F1 Score: 0.9683257918552036
```

