Below is a step-by-step guide for your in-class exercise that focuses on implementing and comparing k-Nearest Neighbors (KNN) and Logistic Regression using the Breast Cancer Wisconsin Diagnostic dataset from UCI.

**Data Acquisition and Familiarization**

**Download the Dataset:**

Visit the [Breast Cancer Wisconsin Diagnostic dataset page](https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic) on UCI.

Download the dataset (ensure you get the version in a usable format such as CSV).

**Initial Exploration:**

Load the dataset into your preferred environment (e.g., Jupyter Notebook, Python script).

Inspect the dataset structure:

Identify the columns (you should see an ID column, a diagnosis column, and several feature columns).

Understand the meaning of each column based on the dataset description.

Note that the diagnosis column contains categorical values (e.g., “M” for malignant and “B” for benign).

**Data Preprocessing**

**Clean the Data:**

Remove any columns that are not useful for modeling (e.g., the ID column).

Convert the diagnosis column into a binary numerical format (e.g., assign 1 for malignant and 0 for benign).

**Handle Missing Values:**

Check for missing or inconsistent data.

Decide on an approach (e.g., imputation or removal) to handle any missing values.

**Feature Scaling:**

Since KNN is sensitive to the scale of the features, apply normalization or standardization to the feature set.

Ensure that all features are on a similar scale before training the models.

**Splitting the Data**

**Create Training and Test Sets:**

Split the dataset into training and testing subsets (a common split is 70% for training and 30% for testing).

Make sure to use a random seed for reproducibility if your tool allows it.

**Model Implementation**

**KNN Classifier:**

Review the concept of KNN, where the model classifies a sample based on the majority label among its k-nearest neighbors.

Decide on an initial value for k (e.g., start with k = 5).

Train the KNN model using the training set.

Use the trained model to predict the class labels on the test set.

**Logistic Regression:**

Understand the basics of Logistic Regression, particularly its use in binary classification.

Consider the role of regularization in preventing overfitting.

Train the Logistic Regression model using the training set.

Predict the class labels on the test set with the trained model.

**Evaluation of Models**

**Define Evaluation Metrics:**

Identify the key metrics for comparison such as:

Accuracy

Precision

Recall

F1-Score

Confusion Matrix (to visualize true vs. predicted classifications)

**Evaluate Each Model:**

Compute and record the metrics for both the KNN and Logistic Regression models.

Compare the performance of the two models based on these metrics.

Reflect on how the choice of k in KNN and the regularization parameter in Logistic Regression might affect these outcomes.

**Analysis and Discussion**

**Discuss the Results:**

Analyze which model performed better overall and why that might be the case.

Consider the effect of feature scaling and the nature of the dataset on each model’s performance.

Explore potential trade-offs (for example, one model might have a higher recall while the other might have better precision).

**Reflect on Hyperparameter Choices:**

Discuss how changing k (in KNN) or adjusting regularization (in Logistic Regression) might further impact the results.

Consider additional steps like cross-validation for a more robust evaluation.

**Reporting:**

**Prepare a Brief Report:**

Summarize your data exploration and preprocessing steps.

Describe your methodology for training the models.

Present your evaluation metrics and discuss the comparative results.

Include insights gained about the strengths and weaknesses of each approach.

**Model Enhancement:**

Experiment with other preprocessing techniques or feature selection methods.

Try tuning the hyperparameters further using systematic approaches (e.g., grid search or cross-validation).

**Deeper Analysis:**

If time permits, visualize decision boundaries (if you reduce the feature space to two dimensions) to gain further insight into how each model separates the classes.

Challenge advanced students to implement one of the models from scratch without using high-level libraries.

This step-by-step guide provides a clear roadmap for students to follow as they work through the exercise. It emphasizes understanding the dataset, preparing the data properly, implementing and comparing two different modeling techniques, and reflecting critically on the outcomes. Enjoy the learning experience!