Lab: Exploring KNN and Distance Metrics

Objective

- Understand how different distance metrics (Euclidean, Manhattan) affect KNN performance.
- Visualize decision boundaries.
- Experiment with different values of k.

Part 1: Euclidean Distance on Iris Dataset

1. **Dataset:** Use the Iris dataset (`sklearn.datasets.load_iris`).
2. **Tasks:**
   a. Split the dataset into training and test sets.
   b. Implement a KNN classifier using **Euclidean distance**.
   c. Evaluate the accuracy of your model.
3. **Questions to think about:**
   - Why is Euclidean distance appropriate for this dataset?
   - How would changing k affect your accuracy?

Part 2: Manhattan Distance on Grid-like Dataset

1. **Dataset:** Create a synthetic dataset with `make_classification` (2 features, 2 classes). Round feature values to simulate a **grid structure**.
2. **Tasks:**
   a. Split into training and test sets.
   b. Implement a KNN classifier using **Manhattan distance**.

c. Evaluate accuracy.

3. **Questions to think about:**
    - ○ Why is Manhattan distance more suitable here?
    - ○ What happens if you use Euclidean distance instead?

Part 3: Decision Boundary Visualization

1. Plot decision boundaries for your KNN models.
2. Compare Euclidean vs Manhattan (for the grid dataset).
3. **Questions:**
    - ○ How does the choice of distance metric affect the shape of the boundary?
    - ○ Can you explain why it looks the way it does?

Part 4: Experimenting with K

1. Try different values of k (1, 3, 5, 7, 15).
2. Observe how accuracy changes.
3. **Questions:**
    - ○ Which k gives the best performance?
    - ○ How does a very small k vs very large k affect overfitting/underfitting?

Hints / Tips (Without Giving Solutions)

- Use `KNeighborsClassifier` from sklearn.
- For Manhattan distance: `metric='manhattan'`.
- For Euclidean distance: `metric='euclidean'`.
- Use `train_test_split` for splitting the dataset.
- Optional: Use `cross_val_score` to pick the best k.
- For plotting, you can use `np.meshgrid` and `plt.contourf`.

Deliverables

1. Code for both datasets with KNN implementation.
2. Plots of decision boundaries.
3. A short explanation answering all the “Questions to think about”.
4. Optional: A table showing accuracy for different k values.