# Task 6: K-Nearest Neighbors (KNN) Classification

This repository implements **KNN classification** as required in Task 6. It uses the **Iris dataset** and includes feature normalization, K tuning, evaluation, and **decision boundary** visualization.

## What’s inside
```
.
├── knn_classification.py
├── iris.csv
├── README.md
└── outputs/
    ├── accuracy_vs_k.png
    ├── confusion_matrix.png
    └── decision_boundary.png
```

## Quickstart
1) Install dependencies:
```bash
pip install -r requirements.txt
# or
pip install numpy pandas scikit-learn matplotlib
```

2) Run:
```bash
python knn_classification.py
# or specify a custom dataset:
python knn_classification.py --data your_dataset.csv --kmax 50
```
> Assumption for custom CSV: **the last column is the target label**; all other columns are numeric features.

3) Find the results in `outputs/`:
- `accuracy_vs_k.png` – K vs Accuracy
- `confusion_matrix.png` – Confusion matrix
- `decision_boundary.png` – 2D decision boundary on petal features
- `report.txt` – Accuracy and classification report

## Notes
- We **normalize** features with `StandardScaler` before training (important for distance-based models).
- We **experiment** with multiple K values (default 1..30) and pick the best based on hold-out accuracy.
- Decision boundary is shown using two Iris features (petal length & petal width).
- Interview Q&A are **excluded** as requested.

## License
MIT