Milestone 1 Report

362578 Clea Maisonnier | 375535 Sam Lee | 379094 Yahan Zhang

Introduction

Our project goal is to predict the presence of heart disease in a patient, categorized on a scale from 0 to 4. We utilize the Heart Disease dataset, consisting of 297 samples and 13 features per patient. The dataset is split into 237 training samples and 60 testing samples. In this first part, we will utilize three classifiers: logistic regression, KNN and K-means. To assess the performance of our models, we use both Accuracy and Macro F1-Score, to account for class imbalance in the dataset.

Method

K-Nearest Neighbors (KNN)

The KNN is an algorithm which classifies a given argument x according to most of its k nearest neighbors in the training set. - init: assigns to the object the given argument k (default k=1) and the task's type. - fit: stores the training data and labels into the object and returns the predicted labels on the training set. - predict: classifies each point in the test set based on the majority label of its k nearest neighbors, using the Euclidean distance.

Logistic Regression

We implemented Logistic Regression using gradient descent optimization. - init: assigns to the object the given arguments learning rate and maximum number of iterations. - fit: normalizes the input features, appends a bias term, initializes the weights, and optimizes them using gradient descent to minimize the cross-entropy loss. - predict: applies the learned weights to the new input data, computes softmax probabilities, and assigns each point to the class with the highest probability.

KMeans Clustering

We implemented KMeans clustering with K=5 (matching the number of classes). - init: initializes the number of clusters K and the maximum number of iterations. - fit: randomly initializes cluster centers, assigns each training point to the nearest center, updates the centers by computing the mean of the assigned points, and repeats this process for a fixed number of iterations. - predict: assigns each test point to the nearest cluster center based on Euclidean distance.

Experiment/Results

Metric	KNN (k=7)	Logistic Regression (lr=1e-2, max_iters=500)	KMeans
Train Accuracy	64.979%	64.979%	46.032%
Train F1-score	0.359204	0.386905	0.277622
Test Accuracy	60.000%	61.667%	33.333%
Test F1-score	0.329790	0.330476	0.207763

Validation Accuracy vs Learning Rate (Logistic Regression):

To select the best learning rate, we performed hyperparameter tuning on both the learning rate and the number of maximum iterations simultaneously. We evaluated learning rates [1e-2, 1e-3, 5e-4, 1e-4] and max iterations [500, 1000, 2000, 5000].

Based on the validation set performance shown in Figure 1, we selected the best learning rate `1e-2` and the best max iterations `500`. The model achieved the highest validation accuracy with these hyperparameters.

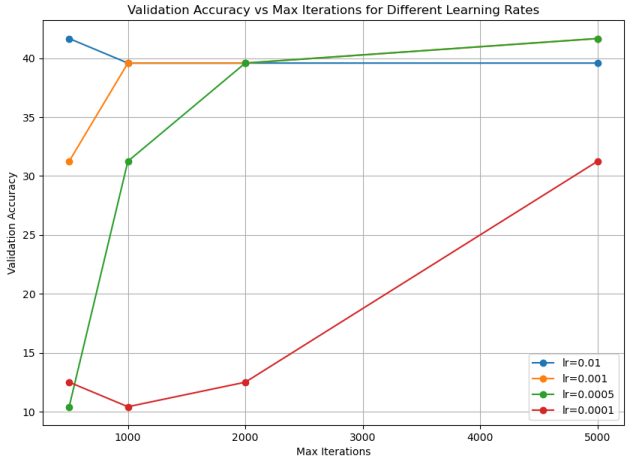

Figure 1: Validation set accuracy for different learning rates

K-Nearest Neighbors (KNN): Which K to choose

We use cross-validation to find the K that gives the highest validation accuracy. Since the dataset exhibits class imbalance (e.g., the size of class 0 is significantly larger than that of class 4), accuracy alone may not be a reliable metric. Therefore, we also considered the Macro F1-Score, which better accounts for the imbalance. K is typically chosen as an odd number to avoid ties. From the validation results, we observed that K=3 gave the highest accuracy, while K=7 gave the highest F1 score. Since considering F1 is important and K=7 achieves similar accuracy to K=3, we selected K=7 as the final hyperparameter.

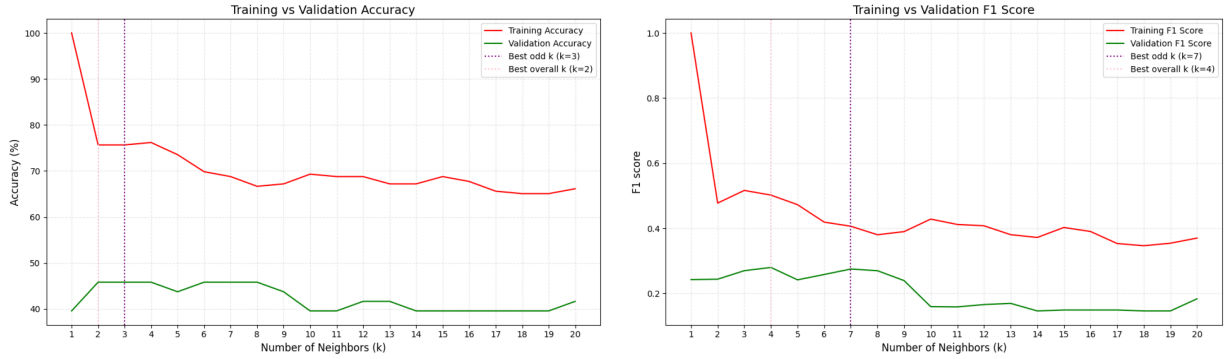

Figure 2: Validation set accuracy for different K values

Discussion/Conclusion

- **KMeans** showed low performance as expected for unsupervised clustering.
- **KNN** achieved high training accuracy but lower generalization to the test set.
- **Logistic Regression** provided reasonable generalization performance after hyperparameter tuning.
- Hyperparameter tuning, especially learning rate and max iteration selection, had a significant impact on Logistic Regression performance.
- Since **KMeans** is an unsupervised method and does not use label information during training, its classification performance is inherently limited when evaluated using supervised metrics like accuracy or F1 score. Nonetheless, it was able to uncover some structure in the data space.