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. The dataset, with 297 samples (237 training, 60 testing), is used to predict heart disease presence on a scale of 0 to 4. We evaluate logistic regression, KNN, and KMeans using both Accuracy and Macro F1-Score to address class imbalance.

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 parameters. - fit: stores the current object the training data as well as its labels,returns the predicted labels - predict: classifies each point in the data test depending on the majority label of its nearest k neighbors, using the Euclidean distance.

Logistic Regression

We implemented Logistic Regression using gradient descent optimization. - init: assigns parameters. - fit: normalizes features, adds bias, initializes weights, and optimizes them using gradient descent to minimize the cross-entropy loss. - predict: outputs the class with highest probability.

KMeans Clustering

We implemented KMeans clustering with K=5 (matching the number of classes). - init: assigns parameters. - fit: randomly initializes centers and iteratively updates them. We perform multiple random initializations and select the clustering that achieves the highest classification accuracy by finding the best cluster-to-label matching using the ground-truth labels. - predict: assigns each test point to the nearest center.

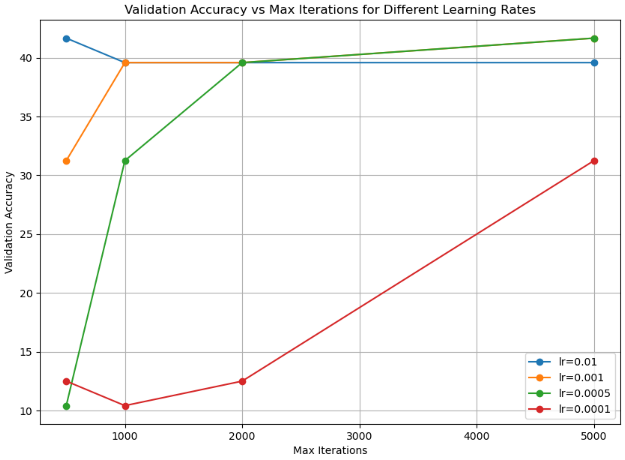
Experiment/Results

Method	Train Accuracy (%)	Train F1-score	Test Accuracy (%)	Test F1-score	Training Time (s)	Prediction Time (s)
KNN	64.979%	0.359204	60.000%	0.329790	0.0039	0.0009
Logistic Regression	64.979%	0.386905	61.667%	0.330476	0.0116	0.0000
KMeans	45.570%	0.333175	21.667%	0.260529	0.0290	0.0000

Logistic Regression: Validation Accuracy vs Learning Rate:

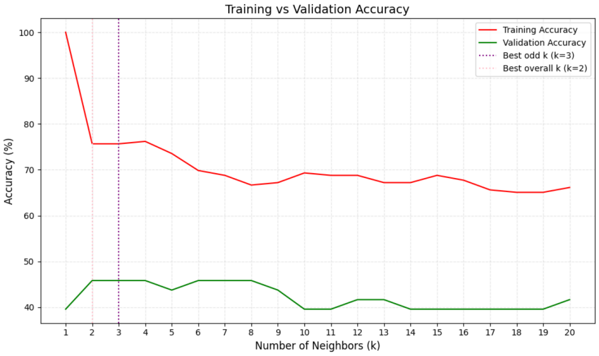
To select the best learning rate, we performed hyperparameter tuning on both the learning rate and the number of maximum iterations simultaneously. We evaluated learning rates [1e-2, 1e-3, 5e-4, 1e-4] and max iterations [500, 1000, 2000, 5000].

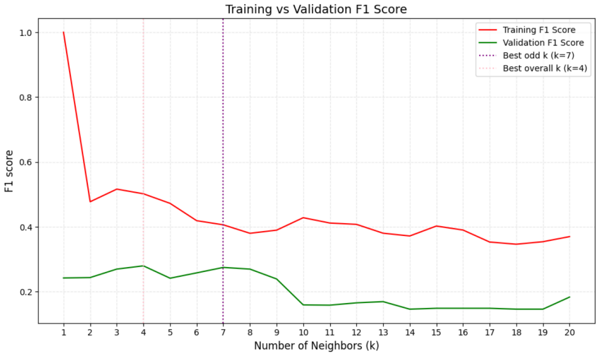
Based on the validation set performance shown in the figure below, we selected `lr=1e-2` and `max_iters=500`, achieving the highest validation accuracy.



KNN: Which K to choose

We use cross-validation to find the K that gives the highest validation accuracy. Due to class imbalance in the dataset, accuracy alone may not be a reliable metric; thus, we also considered the Macro F1-Score, which better captures performance across all classes. Although K=3 achieved the highest accuracy, K=7 provided the best F1-Score with comparable accuracy. Since K=7 also avoids ties, we selected K=7 as the final hyperparameter.





Runtime Analysis

We measured the training and prediction times for each model once under the same hardware and software conditions. Given the small scale of the dataset and the fast execution times, single-run timing was considered sufficient. KMeans required the longest training time due to its iterative update process, while KNN achieved instant training but showed slower prediction due to nearest neighbor searches. Logistic Regression achieved a good balance between training speed and generalization performance.

Discussion/Conclusion

- **KMeans** showed limited performance despite leveraging true labels to select the best clustering. As clustering remained unsupervised, its classification performance was inherently limited, although some structure in the data was uncovered.
- **KNN** achieved high training accuracy but lower generalization to the test set.
- **Logistic Regression** provided reasonable generalization performance, with hyperparameter tuning — particularly the choice of learning rate and maximum iterations — having a significant impact.