CS-233: Introduction to Machine Learning Course Project Milestone 1 Report

Introduction

In this project, we implemented and evaluated different classification methods on the Heart Disease dataset. This database contains medical attributes of patients and corresponding hearth disease classification labels from 0 (being healthy) to 4 (having a severe disease). For the first milestone, we are testing the accuracy of three methods: k-Nearest Neighbors (k-NN), Logistic Regression and K-means clustering. We evaluate each model's performance using accuracy and F1-score metrics.

Method

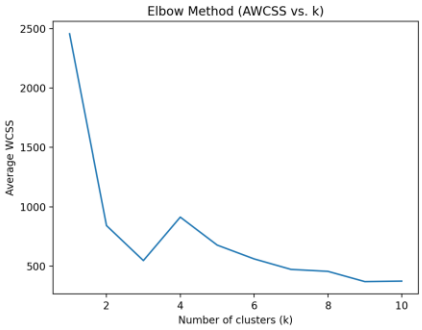
First of all, data preparation. The dataset was presplit into a training set of 237 samples and a test set of 60 samples. All hyperparameter tuning was performed solely on the training data that we split into a training and validation part. And for the final evaluations we used the training set to train the model, and the testing set to test it. For k-NN and Logistic Regression we used k-fold cross-validation with K=4, splitting the training set into 4 rotating training and validation folds. For K-Means we applied a classic 80-20 split on the 237 sample and chose not to perform cross-validation. The data was also normalized, and a bias term was added when required.

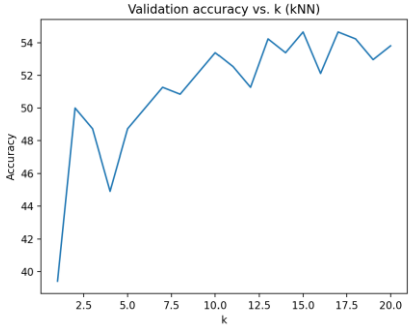
Second, we had to choose the methods for each model. The base code for all models was adapted from the exercises. We optimized K-Means by adding multiple random initializations to improve clustering, the idea is to retain the configuration that has the lowest sum of squared distances. We also tested both Manhattan and Euclidean distances in k-NN and K-Means and found better results with the Euclidean method.

Finally, we had to take care of hyperparameter tuning, to optimize the k-NN and Logistic Regression models we used cross validation to find the best parameters on the validation set, and for K-Means we computed the average within cluster sum of squares vs the number of clusters graphic and used the elbow method to find the optimal K value. Then the test set was used to evaluate how well the models with these specific hyperparameters perform on unseen data.

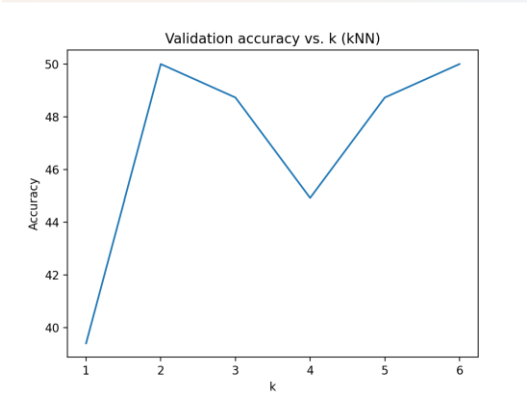
Experiment

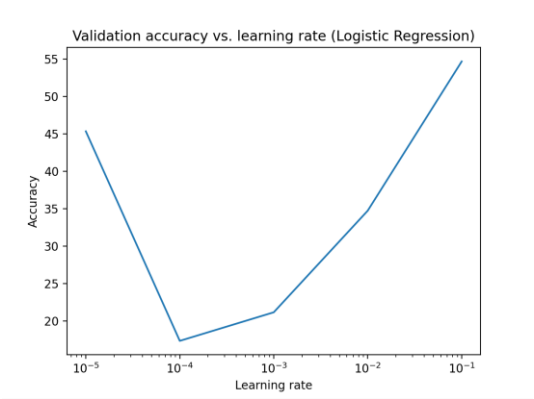
Let's begin with K-Means clustering, the computed graphic is shown below. We can observe that the average WCSS decreases sharply until k = 3, then it unexpectedly seems to go up until k=4 and afterwards it decreases again but at a slower rate. Despite the anomaly, the elbow clearly appears at k = 3 and that is the number of clusters we believe is optimal.



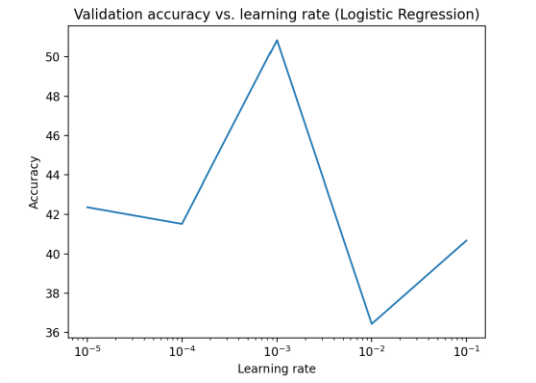


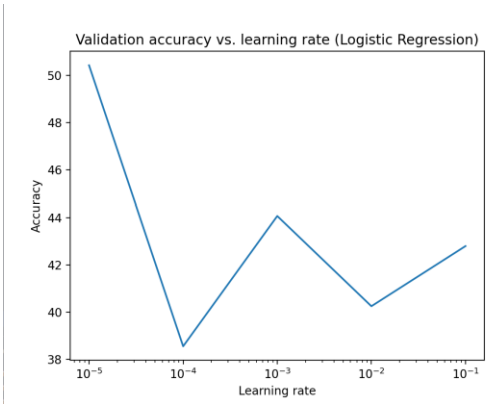
Let's continue with k-NN, we first ran a 4-fold cross-validation across k values from 1 to 20. The graph is right on top, and it is noticeable that at first the performance is at it's lowest. This is likely due to overfitting. It is also noticeable that the most accurate values are for larger K values, in this case it occurred at k = 17. While it is technically the most accurate, we felt like this was not the most meaningful result because for a k value this large usually leads to underfitting. We believe the result at k = 17 may be due to randomness or noise and we decided to rerun cross-validation over a smaller range. That would be the left graph shown below. We can observe that this time the point where accuracy is at its peak is k = 2. We felt like this is more fitting and likely to be optimal.





Lastly, we have Logistic regression, where cross-validation has been run for different learning rates: [1e-5, 1e-4, 1e-3, 1e-2, 1e-1]. We thought it was interesting to do this for different max iterations values and the results are just like we would have imagined. The graph at the top right represents when it was run with a small max iteration value, we chose 70 and the optimal value for the learning rate is 1e-1, when we chose a bigger max iteration value (500) we had the graph at the bottom left where 1e-3 is the optimal learning rate and finally the graph at the bottom right is with an even bigger value of 1000 and we now have 1e-5 as an optimal learning rate. This is an expected trend since in the first case the model doesn't have as much time to learn so a bigger learning rate will perform better and as the number of iterations gets bigger smaller learning rates work better. We believe 1e-3 with 500 max iterations showcases the perfect balance, we think that with a learning rate too big the model is at risk of overshooting. After each iteration the parameters are likely to be changed too drastically, and we would then overshoot the optimal values. Just like 1e-5 would be too small and the model is likely to fail to learn because after each iteration the model parameters will only change by a super small margin.





Results

Metrics	k-NN	Logistic Regression	K-means
Train Accuracy	73.840%	64.979%	60.759%
Train F-1 Score	0.478927	0.422470	0.259006
Test Accuracy	60.000%	61.667%	40.000%
Test F-1 Score	0.297884	0.320466	0.256143

Conclusion

The models performed decently but they don't suggest accuracies precise enough to be used in real life. In the next milestone we plan to explore more advanced models, and we are hoping to achieve stronger accuracy on unseen data.