K-Nearest Neighbours & Decision-Tree Classifiers

Jack Parry-Wingfield, Lucas Andrade, Alina Shimizu-Jozi

January 2025

Abstract

Classification tasks constitute a fundamental aspect of machine learning, enabling the categorization of data into predefined classes based on patterns and features. In this assignment, we investigated the performance of two machine learning models: K-Nearest Neighbours (KNN) and Decision Tree (DT). Training these models on two benchmark datasets to look at both binary and multi-class classification tasks, we found that the KNN model generally achieved a higher accuracy relative to the DT model. Multiple experiments were conducted on both models to allow for optimal parameter selection, distance and cost function variations, and validation set sizes for training. These experiments demonstrated that model performance varied with parameter choices, reinforcing the need for systematic model refinement. The findings emphasize the necessity of balancing accuracy, interpretability, and choosing the best model for the given data when selecting classification models for future applications.

1 Introduction

Classification models are a foundational aspect of machine learning. While machine learning models are often tailored to specific data types, K-Nearest Neighbors (KNN) and Decision Trees (DT) stand out as versatile classifiers capable of adapting to a wide range of classification tasks, making them ideal candidates for comparative evaluation (Kotsiantis et al., 2006). Both are supervised learning models capable of handling binary and multi-class classification tasks. This study aims to evaluate their performance through extensive training and testing, applying techniques such as depth optimization and distance function comparisons to assess their flexibility and accuracy.

Prior research offers mixed conclusions regarding one model's superiority. For example, Hun et al. (2022) found that KNN outperformed DT in prostate cancer detection, whereas Ramadhan et al. (2020) demonstrated that DT was more effective in identifying Distributed Denial of Service (DDoS) attacks. These discrepancies suggest that dataset composition and classification objectives heavily influence model performance. To contribute to this discussion, we test both models on two benchmark datasets: Kaggle's Penguin Size dataset for multi-class classification and UCI's Heart Disease dataset for binary classification. The Heart Disease dataset is widely utilized, with over 65 research citations spanning applications from Naïve Bayes classification to real-time coronary disease monitoring (Sati, 2018; Otoom et al., 2015; Nashif et al., 2018). Meanwhile, the Penguin Size dataset is recognized for its comparability to Fisher's historic Iris dataset on linear discriminant analysis (Fisher, 1936), making it a well-suited choice for evaluating classification models.

Through model implementation and evaluation across multiple performance metrics (AUROC, ROC, accuracy), our results indicate that KNN consistently outperformed DT across both datasets. Even with hyperparameter tuning — optimizing K-values for KNN and adjusting tree depth for DT — KNN demonstrated superior performance in all tested conditions. While these findings underscore the robustness of KNN in diverse classification tasks, the scope of performance metrics and the structure of the datasets both contribute to this result (Novaković et al., 2017). A more in-depth investigation is required to compare how these models perform in more complex or imbalanced datasets.

2 Methods

2.1 K-Nearest Neighbours

The KNN algorithm is a supervised machine learning technique that can be used to classify new data points by comparing them to similar data points in a training set. This comparison involves the computation of distances between all the training data points and one which is being classified. Various distance functions exist to enable this computation, with different functions more suitable in certain applications than others. The number of data points to consider is determined by the value of K. Changing the K value can affect the accuracy of the model depending on the composition of the dataset. After computing all the distances, the K closest data points in the training set are identified and the new data point is assigned to the class of the nearest neighbours.

2.2 Decision Tree

The decision tree model follows a tree-like graph structure to sequentially make decisions based on numerical features and thresholds. As a machine learning algorithm used for both classification and regression tasks, its ability to accurately separate data improves with increasing tree depth, though this also raises the risk of over-fitting. The algorithm starts with an idea or main decision and continues to add conditions — nodes — until the end point is reached. The final outcomes are evaluated at the leaf nodes. Decision trees contain sub-trees which represent a specific separation of the data, and branches can have associated costs to refine the model's classification ability. Similar to KNN's distance functions, various cost functions exist for flexibility between datasets.

3 Datasets

The datasets used for this assignment were both publicly available on Kaggle and the UC Irvine machine learning repository. The Heart Disease dataset, used for the binary classification task, spanned databases of three countries with 13 features and a target variable indicating the presence or absence of heart disease (Janosi, 1989). The Palmer Archipelago penguin dataset is a multivariate collection derived from the observation of three species of penguins: Chinstrap, Adélie or Gentoo in Antarctica (Pandey, 2020). Though seven columns of data were available, only five features and the target species variable were considered as per the instructions to exclude the 'island' feature for prediction. The process of cleaning this dataset included changing the categorical 'sex' feature to a binary value via mapping. Both the penguin and heart-disease data were investigated as Panda's data frames, allowing missing-value identification and correction to be seamlessly applied. Different versions of the datasets — one employing the removal of the missing value rows and the other filling in the missing values with averages of the variable set — were created for experimental comparison. Box plots to identify outliers as well as a duplicate verification and categorical variable unique value check were performed for further refinement of clean data. For the binary heart-disease dataset, the positive and negative patient rows were split. Similarly, the penguin data was split by species. The built in Pandas '.describe()' function allowed for the exploration of the means, standard deviation, minimum value, maximum value and the 25th, 50th and 75th percentiles of each feature. Finally, the mean squared difference was calculated and ranked for both datasets to distinguish meaningful features before training the models.

4 Results

(1) First, we tested the AUROC and accuracy scores of both models for the heart-disease and penguin datasets respectively. On the heart-disease dataset, KNN's AUROC hovered around 0.92 using $K = 3$ and the euclidean distance formula, whereas DT's hovered around 0.76. These AUROC scores are illustrated on figure 9 below. On the penguin dataset, the KNN model achieved 99.2% accuracy against the test data using $K = 3$ and the euclidean distance function whereas DT achieved 95% at its highest.

(2) K values 1 through 50 in KNN were tested across both datasets. For the species classification, figure 1 shows how the testing data accuracy changes with increasing K value, and across both the manhattan and euclidean distance functions. This trial yielded a maximum accuracy of 99.6 using $K = 1, 3$, distance $= euclidean$ and $K = 3$, distance $= manhattan$, and a minimum accuracy of 80.3 using $K = 50$, distance $= manhattan$, and a mean accuracy of 95.17 across all K values for euclidean, and 92.25 for manhattan across all K values 1 to 50. Similarly for the training data accuracy this can be illustrated by figure 2, yielding a maximum accuracy of 100 using $K = 1, 2, 3$ with distance = both *euclidean* and *manhattan*, a minimum accuracy of 70.6 using $K = 50$ and distance $= manhattan$, and mean accuracy of 92.62 for euclidean and 90.12 for manhattan. Additionally, for the binary classification, the effect of the K value on the testing data accuracy is illustrated by figure 3, yielding a maximum accuracy of 86.4 using $K = 44$ and distance $= euclidean$, a minimum accuracy of 78.2 using $K = 1$ and distance $= manhattan$, and mean accuracy of 83.87 for euclidean and 82.81 for manhattan, and the training data accuracy by figure 4, yielding a maximum accuracy of 100 using $K = 1$ for both euclidean and manhattan distance functions, a minimum accuracy of 82.0 using $K = 37$ with manhattan distance, and mean accuracy of 85.59 for euclidean and 84.93 for manhattan.

(3) Similarly, different values for the max depth hyperparameter in DT were tested. This was done by splitting the test data into validation and testing, and using the validation set to select the best max depth, which was tested on the test set. For the heart-disease set, the best max depth was found to be 5 yielding a test auroc around 0.72, for the penguin set it was 2 with an accuracy around 96%. Both experiments are represented visually in figures 5 and 6.

(4) For DT, different cost functions were tried: entropy was found to perform best on the binary task with an AUROC around 0.77 (misclassificaiton got 0.76 and gini-index hovered around 0.69), whereas misclassification rate did best on the penguin set with an accuracy of 95%(entropy and gini-index both sat around 93%) For KNN, we experimented with euclidean distance and manhattan distance formulas, and the results are shown in figures 1-4, along with the numerical results presented in (2). Euclidean distance marginally outperformed Manhattan distance on average.

(5) For the binary classification task, we plotted the ROC curve for KNN and DT. See figure 9 for results, where the AUROC

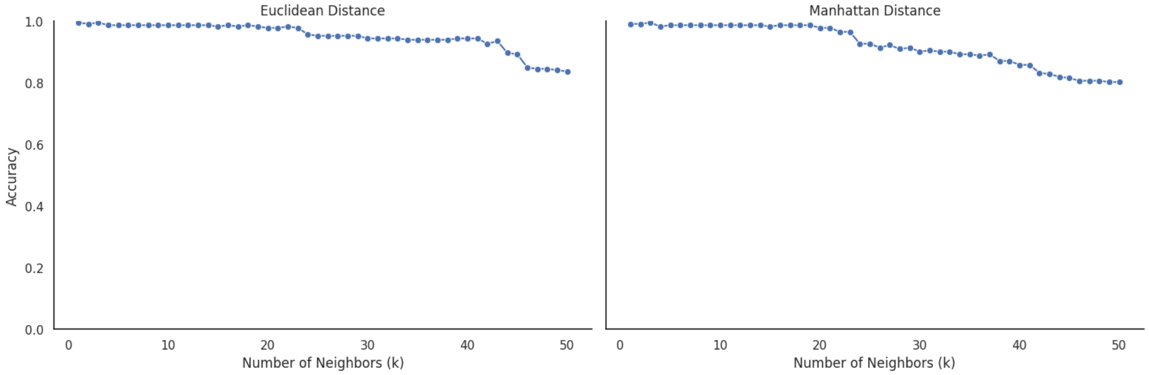

Figure 1: K value versus testing data accuracy for both distance formulas (Species)

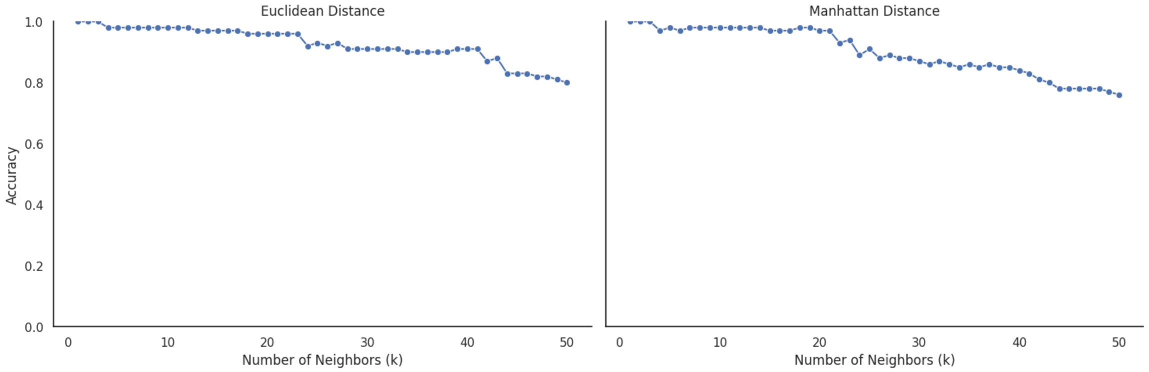

Figure 2: K value versus training data accuracy for both distance formulas (Species)

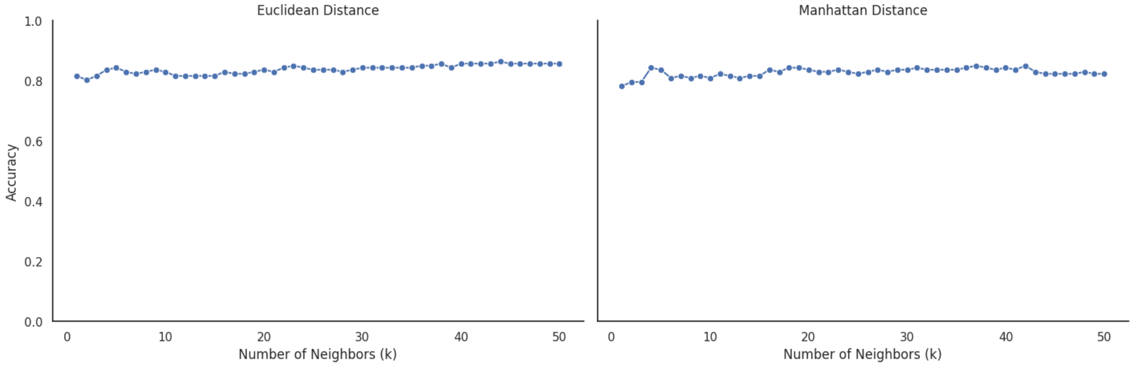

Figure 3: K value versus testing data accuracy for both distance formulas (Binary)

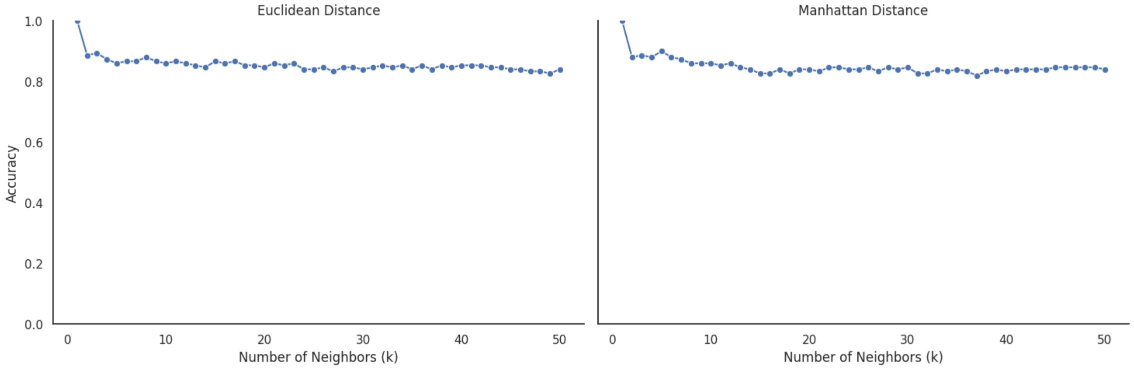

Figure 4: K value versus training data accuracy for both distance formulas (Binary)

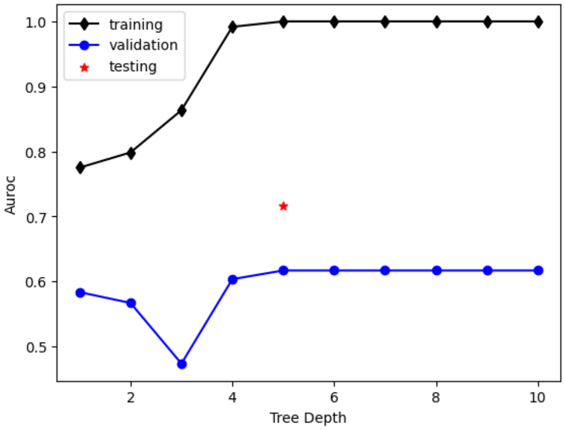

Figure 5: Max Depth versus training and validation AUROCs on Heart-Disease data

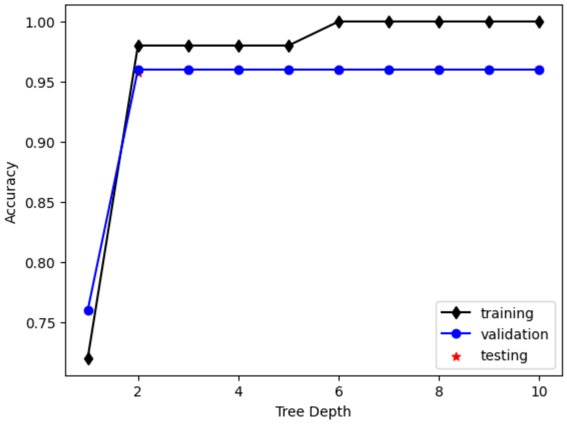

Figure 6: Max Depth versus training and validation accuracies on Penguin data

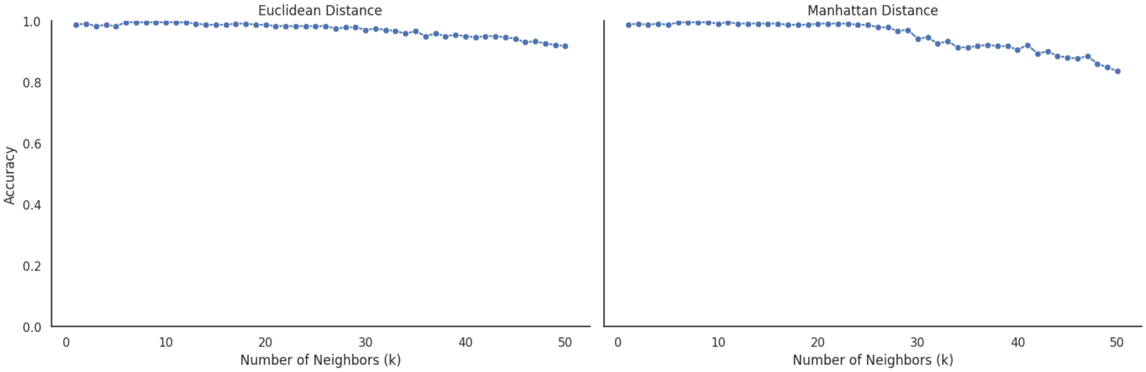

Figure 7: K value versus testing data accuracy for both distance formulas, using data cleaning method 2 (Species)

scores are shown on the plot for convenience.

(6) Key features were identified by ranking the mean squared differences (MSD) between each of the features across species for the multi-class dataset and between the positive and negative groups for the binary dataset. These results were further verified by a correlation matrix, which supported the features selected with the highest mean squared differences as highly correlated to the target labels. Features which had a low MSD and low correlation were dropped from the heart dataset. All numerical features were included in the penguin dataset due to relatively similar MSDs and correlations. Sex was not included as it was not highly correlated to the target species label.

(7) We used a trained DT classifier to calculate a rough feature importance score by traversing a trained decision tree and keeping count of each time a feature was used to split a node. For the heart-disease dataset, the top 5 most important features (in order) were found to be age, oldpeak, sex, thal and ca. For the Penguin dataset, the results were not quite as expected. The top 3 most important features were culmen length, culmen depth and flipper length, but the other two features were never selected as the feature used for a split - these findings aligned with the MSD ranking.

(8) For both datasets, we exercised two data cleaning methods, as described under the **data** section - for method 1, NaN values were dropped. For method 2, they were replaced by the mean value of that feature. KNN, when run using datasets cleaned via method 1 yielded test data accuracy results as shown in figures 1 and 3 for the species classification and binary classification respectively, and the results for KNN using data cleaning method 2 are shown by figure 7 (species classification) yielding a maximum accuracy of 99.6 using $K = 6, 7, 8, 9, 10, 11, 12$ with euclidean distance and $K = 6, 7, 8, 9, 11$ with manhattan, and a minimum accuracy of 83.6 using $K = 50$ with manhattan distance, and mean accuracy of 97.21 for euclidean and 95.27 for manhattan. For the binary classification, the results of method 2 are shown in figure 8, yielding a maximum accuracy of 84.4 using $K = 22$ with euclidean distance and $K = 30$ with manhattan distance, a minimum accuracy of 76.9 using $K = 2$ with manhattan distance, and mean accuracy of 82.18 for euclidean and 81.92 for manhattan. This illustrates that for KNN, method 1 favours the binary classification task whereas method 2 favours species classification. For DT, method 1 yielded an AUROC increase of around 0.3 on the heart-disease dataset, whereas method 2 showed a 5% increase on the penguin dataset.

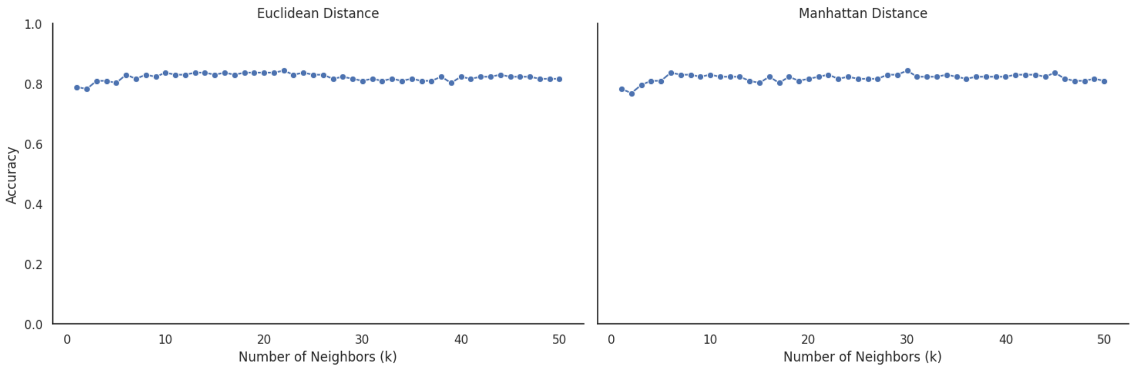

Figure 8: K value versus testing data accuracy for both distance formulas, using data cleaning method 2 (Binary)

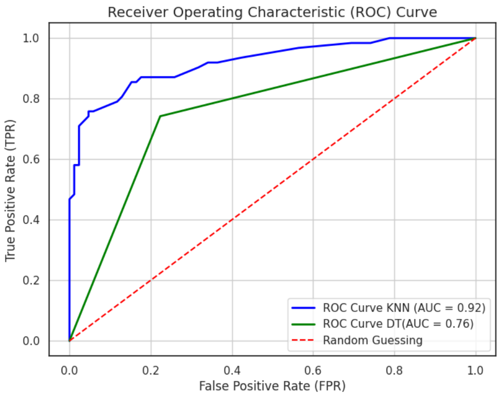

Figure 9: ROC Curve for KNN and DT.

5 Discussion & Conclusion

The results from this study highlight key differences between KNN and DT in their classification performance and reinforce that no single classifier is universally superior. While KNN excelled in binary classification, DT provided valuable interpret-ability and demonstrated the importance of balancing accuracy with model explain-ability. Hyperparameter tuning in addition to cost and distance function selection were imperative for the optimization of performance. These refinements demonstrate the need for going beyond model architecture to ensure a model's capabilities are fully explored. Additionally, data preprocessing choices significantly influenced outcomes, even in smaller choices between removing NaN values or filling them in with the mean of the dataset. Future work should explore methods which combine the strengths of both models and investigate other data engineering approaches for improved classification. While the scope of this assignment focused on two benchmark datasets, expanding the evaluation to more diverse datasets could provide deeper insights into model generalization. Overall, while the comparison between these two models established a foundational education on machine learning techniques and basic classification tasks, the key takeaways for future projects suggest that model selection should be guided by dataset complexity, interpret-ability needs, and scalability to ensure efficiency in real-world applications.

6 Statement of Contribution

Alina S-J handled all data cleaning for task 1. Lucas A. and Jack P-W implemented and ran experiments for the Decision Tree and K-Nearest Neighbour models, respectively. Alina S-J assisted with debugging and formatting. All members contributed equally to brainstorming experiments and writing the report.

7 References

1. Hun, C. C., Yazid, H., Safar, M. J. A., and Ab Rahman, K. S. "Comparison Between K-Nearest Neighbor (KNN) and Decision Tree (DT) Classifier for Glandular Components." *Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications*, edited by N. M. Mahyuddin, N. R. Mat Noor, and H. A. Mat Sakim, Singapore, 2022.
2. Janosi, A., Steinbrunn, W., Pfisterer, M., and Detrano, R. *Heart Disease* [Dataset]. UCI Machine Learning Repository,

1989, https://doi.org/10.24432/C52P4X.

3. Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. ”Machine Learning: A Review of Classification and Combining Techniques.” *Artificial Intelligence Review*, vol. 26, no. 3, 2006, pp. 159–190, https://doi.org/10.1007/s10462-007-9052-3.

4. Ramadhan, P. Sukarno, and M. A. Nugroho. ”Comparative Analysis of K-Nearest Neighbor and Decision Tree in Detecting Distributed Denial of Service.” *2020 8th International Conference on Information and Communication Technology (ICoICT)*, Yogyakarta, Indonesia, 2020, pp. 1-4, https://doi.org/10.1109/ICoICT49345.2020.9166380.

5. Nashif, S., Raihan, M. R., Islam, M. R., and Imam, M. H. ”Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System.” *World Journal of Engineering and Technology*, vol. 6, 2018, pp. 854-873.

6. Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž., and Tomović, M. ”Evaluation of Classification Models in Machine Learning.” *Theory and Applications of Mathematics & Computer Science*, vol. 7, no. 1, 2017, pp. 39-46. Retrieved from https://proxy.library.mcgill.ca/login?url=https://www.proquest.com/scholarly-journals/evaluation-classifi
docview/1922445698/se-2.

7. Sati, N. U. ”A Collective Learning Approach for Semi-Supervised Data Classification.” *Pamukkale University Journal of Engineering Sciences*, vol. 24, 2018, pp. 864-869.

8. Otoom, A. F., Kefaye, A., Ashour, M., Shanti, Y., and Al-Majali, M. ”Real-Time Monitoring of Patients with Coronary Artery Disease.”

9. Pandey, P. *Penguin Dataset: The New Iris* [Dataset]. Kaggle, 2020. Retrieved January 31, 2025, from https://www.kaggle.com/code/parulpandey/penguin-dataset-the-new-iris/input?select=penguins_size.csv.

10. Fisher, R. A. ”The Use of Multiple Measurements in Taxonomic Problems.” *Annals of Eugenics*, vol. 7, no. 2, 1936, pp. 179-188.