COMP 551 - Assignment 1 Report

Elizabeth Kourbatski

Galia Oliel-Sabbag

Colin Song

January 31, 2025

Abstract

In this assignment, we evaluated the performance of two machine learning models, K-Nearest Neighbors (KNN) and Decision Trees, on two datasets. The Heart Disease dataset was used to predict whether a patient had heart disease, while the Penguin dataset was used to classify the species of penguins. Our findings showed that the KNN model achieved higher accuracy than Decision Trees on both the Heart Disease dataset and Penguin dataset.

1 Introduction

We compared the performance of K-Nearest Neighbors (KNN) and Decision Trees (DT) on two classification tasks: Penguin species identification and Heart Disease prediction.

For the Penguin dataset, we evaluated KNN using Euclidean and Manhattan distances, selecting the optimal k based on validation accuracy. KNN with Manhattan distance achieved perfect accuracy, matching previous findings by Alex Hua and Guillermo Goldsztei[1]—who attained similar results with a simple neural network using softmax activation.

For the Heart Disease dataset, we applied feature selection (Pearson correlation for numerical features and Fisher scores for categorical features) and explored both a numerical-only model using Manhattan and Euclidean distance, a categorical-only model using Hamming distance, and a combined numerical-categorical model using Euclidean distance. DT models were evaluated using Misclassification, Entropy, and Gini Index to determine optimal depth, with experimentation by training on the top five most important features.

Overall, KNN outperformed DT in both tasks. In Heart Disease prediction, unweighted KNN with combined features achieved higher performance, while our DT model reached an AUROC of 78.2178—comparable to the 75.83 AUROC reported by Ahmad Ayid Ahmad and Huseyin Polat in a study using the same heart disease dataset[2].

2 Methods

2.1 k-Nearest Neighbors (kNN)

KNN is an algorithm that classifies a data point by finding the K nearest neighbors in the training data based on a chosen distance metric (e.g., Euclidean, Manhattan, Hamming). The class of the data point is determined by the majority label among these K neighbors. The value of K and the distance metric are key parameters of the model.

2.2 Decision Trees

Decision trees partition the input space using classifiers that split the input space based on a decision boundary, this creates a tree structure of nested rules that will be used to make future predictions. If a partition does not sufficiently separate the data, additional decision rules are applied, creating more splits to improve accuracy.

3 Datasets

For the Penguin dataset, we simplified the feature set by first dropping the 'island' feature. This left only one categorical feature, 'sex', which we further excluded after a chi-squared test ($p = 0.7657$) showed no significant correlation with penguin species. We retained the continuous features—culmen length, culmen depth, flipper length, and body mass—which were then scaled using StandardScaler to normalize their ranges before model training.

For the Heart Disease dataset, we converted the problem into binary classification by labeling individuals with heart disease as 1 and those without as 0. All categorical variables were one-hot encoded, and every feature was standardized. We then calculated the squared differences between these group means (heart disease vs. no heart disease). The results showed that ‘ca’ had the highest squared difference (0.8634), followed by ‘oldpeak’ (0.7236) and ‘thalach’ (0.7228). In contrast, ‘age’ (0.2075), ‘trestbps’ (0.0948), and ‘chol’ (0.0259) exhibited smaller squared differences. These findings suggest that features such as ‘ca’, ‘oldpeak’, and ‘thalach’ are more effective at differentiating between individuals with and without heart disease.

Dataset	Number of Instances	Features	Target Parameter
Heart Dataset	303	13	num
Penguin Dataset	342	6	species

Table 1: Summary of datasets used for this assignment.

4 Results

4.1 KNN Model

4.1.1 Heart Dataset

First, we processed the heart dataset by one-hot encoding categorical features, followed by standardizing all features to ensure valid distance comparisons. We then applied KNN classification, varying k and testing three distance metrics: Euclidean, Manhattan, and Gower. The results indicate that Euclidean and Manhattan distances performed similarly, with accuracy ranging between 0.55 and 0.60. In contrast, Gower distance showed lower performance, stabilizing around 0.50. The **highest accuracy observed was 0.60204 using Euclidean distance at** $k = 2$. Based on these findings, we conducted further tests to optimize results.

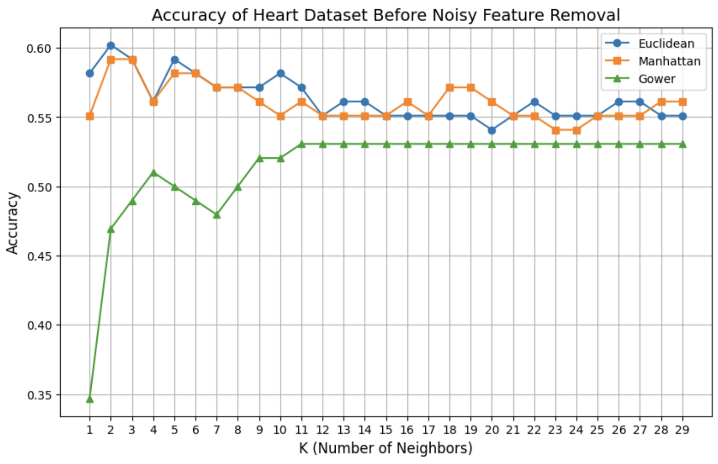


Figure 1: Accuracy of kNN on heart data for varying k, all features.

We first calculated Pearson correlation coefficients for all numerical features against the target variable and filtered features based on correlation thresholds of 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5. Each threshold produced a dataset with varying numbers of retained features, allowing us to assess the impact of correlation-based feature selection on model performance. Next, we performed K-Nearest Neighbors (KNN) classification using 9-fold cross-validation, testing values of k from 1 to 39 with Euclidean and Manhattan distance metrics. The validation AUROC for Manhattan distance increased from 0.6586 at $k = 1$ to 0.8418 at $k = 8$ before stabilizing. The best test AUROC scores were achieved with $k = 9$ for Euclidean (0.9041) and $k = 8$ for Manhattan (0.9361). Across all thresholds, performance improved as k increased, with Manhattan generally outperforming Euclidean. The **highest test AUROC of 0.9361 was achieved using Manhattan distance at a threshold of 0.2**, where the retained features were `['ca', 'oldpeak', 'thalach', 'age']`. Similarly, Euclidean distance peaked at 0.9041 for a threshold of 0.1. As the threshold increased beyond 0.2, performance declined, indicating that removing too many features based on correlation may discard useful information.

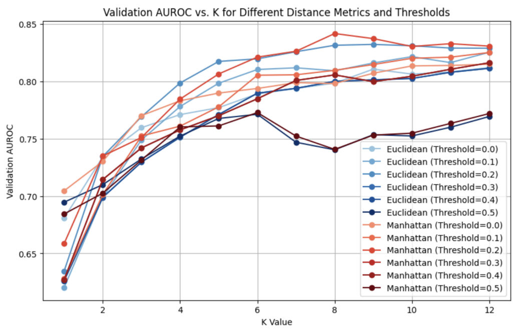


Figure 2: Validation AUROC vs. k for different distance metrics (Euclidean and Manhattan) and Pearson correlation thresholds. Higher thresholds reduce feature set size, negatively impacting AUROC performance.

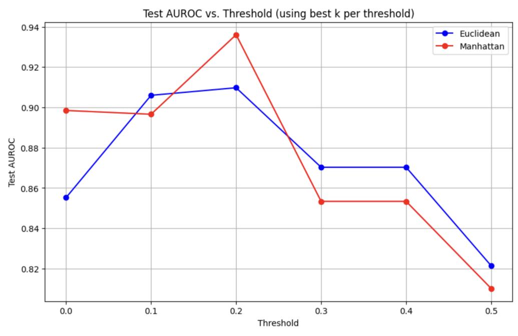


Figure 3: Test AUROC vs. Pearson correlation threshold using the best k for each threshold. Manhattan distance outperforms Euclidean, peaking at a threshold of 0.2 before performance declines as more features are removed.

We then shifted our focus to categorical features. Applying Fisher's Exact Test with thresholds of 0.001, 0.01, 0.05, 0.1, and 1.0, we performed one-hot encoding for all categorical variables. This does not matter in the categorical only model but is necessary for our categorical and numerical model that follows this test. Using Hamming distance for KNN, we evaluated different thresholds to determine the optimal feature selection. The **best unweighted model was achieved at a Fisher threshold of 1.0**, incorporating all categorical features. With $k = 16$, **this model reached a test AUROC of 0.8759**. However, when applying weighting to categorical KNN with Hamming distance, performance declined, yielding a lower AUROC of 0.7553.

All unweighted KNN models outperformed their weighted counterparts. This is likely because unweighted KNN treats all neighbors equally, reducing the risk of overfitting to noise or outliers that could disproportionately affect predictions in weighted models.

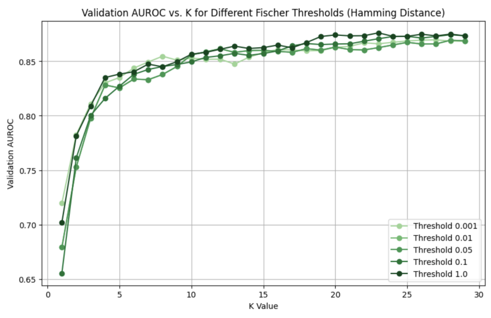


Figure 4: Validation AUROC vs. k for Different Fisher Thresholds. Across all thresholds, AUROC increases with k before stabilizing around 0.86. This suggests that varying the Fisher threshold has little impact on validation AUROC. So, we shifted our focus to analyzing test AUROC as a function of k, as shown in Figure 5.

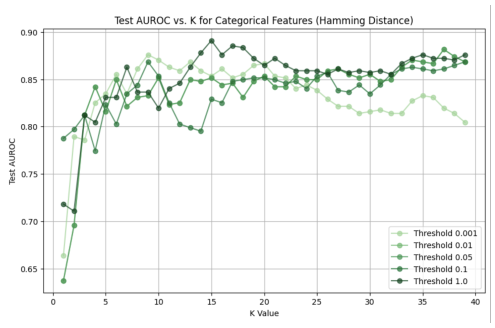


Figure 5: Test AUROC vs. k for Categorical Features. This plot examines how test AUROC changes with different k values across various Fisher thresholds. The peak performance is observed at $k = 15$, threshold = 1.0, achieving a test AUROC of 0.8759. Test AUROC exhibits more fluctuation that validation AUROC.

4.1.2 Penguin Dataset

For the Penguin Dataset, KNN was evaluated using Euclidean and Manhattan distance metrics. Euclidean distance achieved perfect accuracy of 1.0 at the optimal $k = 9$, making it the most effective choice. In contrast, Manhattan distance showed significant variability, with accuracy dropping sharply at $k = 4$ and $k \geq 8$. These results indicate that Euclidean distance with the hyperparameter $k = 9$ provides the most consistent and reliable performance. After determining the optimal k for each distance metric, both models—KNN with Euclidean distance (k = 9) and KNN with Manhattan distance (k = 5)—were retrained on the combined training and validation set. When evaluated on the test set, the **Manhattan distance model with k = 5 achieved the highest accuracy, reaching accuracy 1.0**.

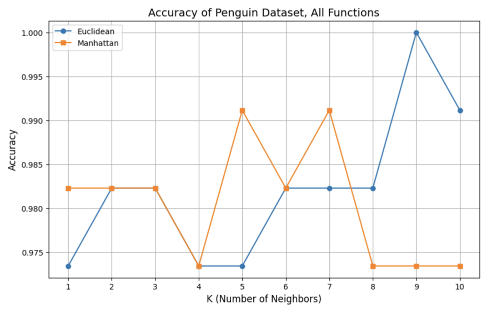


Figure 6: Validation accuracy of kNN on penguin data for varying k.

4.2 Decision Tree Model

4.2.1 Heart Dataset

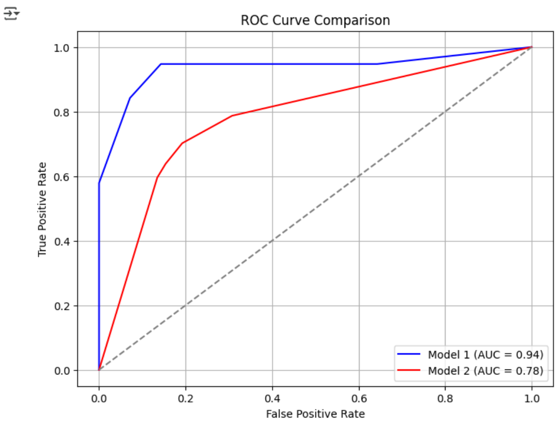
We split the heart disease dataset into training (33%), validation (33%), and test sets (34%). We then trained decision trees using Misclassification Error, Entropy, and Gini Index as cost functions, testing maximum depths from 1 to 20. The best performance (test AUROC of 78.2178) was achieved using Entropy at a depth of 4 with all features.

Next, we identified the top five features using Gini Reduction (sex, cp, age, oldpeak, thal) and retrained the model. However, this simplified model—with a depth of 4 using the Gini Index—yielded a lower AUROC of 72.2772. Thus, reducing the number of features did not improve performance, and the optimal decision tree for heart disease prediction was a depth-4 model using the Entropy cost function with all features.

4.2.2 Penguin Dataset

We trained a decision tree classifier on the penguin dataset, optimizing hyperparameters through validation. **We found that the best model used entropy as the cost function with a maximum depth of 5**, achieving a validation accuracy of 96.46%. Final evaluation on the test set yielded a **test accuracy of 96.58%**, demonstrating strong generalization. Feature importance analysis using Gini reduction identified the most influential predictors: Culmen length: 1.2072 Gini reduction, Culmen depth: 0.3937 Gini reduction, Flipper length: 0.3481 Gini reduction. These results suggest that culmen length is the most significant feature for classification.

4.3 Final Comparison



5 Discussion and Conclusions

We compared K-Nearest Neighbors (KNN) and Decision Trees (DT) on Penguin classification and Heart Disease prediction. For the Penguin dataset, KNN with Manhattan distance ($k = 5$) achieved the highest test accuracy of 100%, while DT with a depth of 2 using the Misclassification cost function reached 95.614% test accuracy. In the Heart Disease dataset, unweighted KNN with Manhattan distance ($k = 10$) on combined numerical and categorical features performed best, achieving a test AUROC of 0.9361, while DT with all features using cost function Entropy and depth 4 achieved an AUROC of 78.2178. Decision Trees, trained on the top five features based on Gini Reduction, achieved slightly lower performance. Interestingly, feature importance rankings differed between Gini Reduction and squared differences, showing that features with high variance across classes do not always contribute most to decision boundaries. KNN also outperformed Decision Trees in both the Penguin and Heart Disease datasets, likely due to its ability to capture complex relationships without imposing strict decision boundaries.

To further improve KNN, bagging and boosting can be applied. Bagging (Bootstrap Aggregating) trains multiple KNN models on different random subsets of the training data and then averaging their predictions. Boosting builds KNN models sequentially, where each new model focuses on correcting the mistakes of the previous ones[9].

To further improve Decision Trees, we can use pruning which involves removing branches of the tree that contribute little to predictive power[10].

6 Statement of Contribution

All team members contributed evenly to the coding and write-up portions of this project!

References

1. A. Hua, G. Goldsztein, ”Using Machine Learning to Predict Penguin Species,” Journal of Student Research, 2022.

2. A. A. Ahmad, H. Polat, ”Prediction of Heart Disease Based on Machine Learning Using Jellyfish Optimization Algorithm,” Diagnostics, 2023.

3. Heart Dataset: `https://archive.ics.uci.edu/dataset/45/heart+disease`

4. Penguin Dataset: `https://www.kaggle.com/code/parulpandey/penguin-dataset-the-new-iris/input?select=`

5. Professor Yue Li’s GitHub: `https://www.cs.mcgill.ca/~yueli/teaching/COMP551_Winter2025/comp551_winter2025.html`

6. Fisher’s Exact Test: `https://www.pathwaycommons.org/guide/primers/statistics/fishers_exact_test/`

7. Pandas Index Intersection Function: `https://pandas.pydata.org/docs/reference/api/pandas.Index.intersection.html`

8. Pandas `loc` Property: `https://www.w3schools.com/python/pandas/ref_df_loc.asp`

9. Bagging and Boosting: `https://www.cs.toronto.edu/~rgrosse/courses/csc311_f20/slides/lec06.pdf`

10. Pruning Decision Trees: `https://www.geeksforgeeks.org/pruning-decision-trees/`