APPLICATION CLASSIFIERS

Predictive Modelling in Action

ABSTRACT

This report examines the application of machine learning algorithms to classify heart conditions using a health-related dataset. K-Nearest Neighbors (KNN) and Decision Trees (DT) were evaluated based on metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. KNN outperformed DT, achieving higher accuracy and recall, demonstrating its suitability for medical predictions. Preprocessing steps, hyperparameter tuning, and performance comparisons are detailed, with suggestions for improvements through feature selection and advanced techniques like ensemble models. The report points out the importance of optimizing models for classification tasks in real-world scenarios.

Casper van Laar

casperdvanlaar@hotmail.com
University of Wolverhampton
MSc Artificial Intelligence
7CS070/UZ3: Concepts & Technologies of Artificial Intelligence
casperdvanlaar@hotmail.com

1. Introduction

This report demonstrates the application of machine learning algorithms to real-world datasets, with a focus on both regression and classification tasks. The primary goal is to assess the performance of various models, identify the most predictive features, and propose improvements. The reports outline the efficacy of models K-Nearest Neighbours (KNN), and Decision Trees (DT). By classifying whether an individual had a heart condition or not. This report provides a comprehensive analysis of the models' performance, features, and potential improvements.

2. Methodology

Data Selection

For the classification task, a health-related dataset of 1190 individuals was used(Mexwell, 2024), with the goal of predicting whether an individual has a heart condition or not. Key features included: sex, age, chest pain type and 8 other features (table 2A and 3A).

Algorithms

K-Nearest Neighbors (KNN) is a non-parametric algorithm used for classification and regression. It classifies (C(x)) by finding the 'k' closest training points to a test point. Written as $C(x)=\text{mode}(y_i|(x_i,y_i)\in D_k(x))$. Where Dk(x) is the set of k nearest neighbours of x_i and y_i is the class label of the i-th nearest neighbour. While calculating the Euclidean distance as d = $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ (GeeksforGeeks, 2023). Where x and y are the coordinates of points and n is the number of dimensions. The classifications are based on the majority label of those neighbours. KNN can be computationally expensive with large datasets and sensitive to the choice of 'k' and distance metric (Awan, A.A., 2023).

A Decision Tree is a supervised learning algorithm used for classification and regression. It splits data based on feature values, creating a tree structure where internal nodes represent decisions and leaf nodes represent predictions. The tree is built by selecting the feature that best separates the data based on criteria like Gini impurity classification. Written as $\text{Gini}(T)=1-\sum_{i=1}^{c} p_i^2$ (Khalid, 2021). Where T is the current node, c the number of classes and p_i the proportion of samples belonging to class i. While simple and interpretable, Decision Trees can overfit, especially with deep trees. Techniques like pruning, limiting depth, or using ensemble methods like Random Forest can help improve generalization (Keylabs, 2024).

Preprocessing

For the classification task, categorical variables were encoded, and feature normalization was applied to ensure that models such as KNN, which are sensitive to scale, could perform optimally.

B. Grid Search Optimization

In this project, GridSearchCV was utilized for hyperparameter tuning to enhance the performance of the machine learning models. Below is a summary of the grid optimization process for both classification and regression algorithms used.

Classifiers Grid Search Optimization

1. K-Nearest Neighbors (KNN):

The optimization process involved testing 6 combinations of parameters using GridSearchCV. It evaluated 3 different values for the number of neighbors: 3, 5, and 7, alongside 2 weight options: Uniform and Distance. GridSearchCV computed each combination and selected the best parameters based on cross-validation performance. The optimal configuration was determined by the combination that achieved the highest average score across 5-folds.

2. Decision Tree (DT):

The optimization process involved testing 9 combinations of parameters using GridSearchCV. It evaluated 3 different depth values: 5, 10, and 15, along with 3 values for min_samples_split: 2, 5, and 10. Like the KNN optimization, GridSearchCV explored all combinations and selected the optimal configuration based on performance metrics, using cross-validation to accurately assess each configuration.

3. Dummy Classifier:

There was no grid search for the Dummy Classifier, as it used a fixed strategy of 'stratified' to serve as a baseline model. This strategy makes predictions based on the distribution of the target labels in the training dataset.

3. Results

Model 1: K-Nearest Neighbours (KNN)

KNN emerged as the best-performing model, achieving a 92% accuracy and F1-score of 0.93(table 1). It demonstrated sensitivity to detecting heart conditions (Class 1) with a recall of 95%, ensuring minimal false negatives. The model also maintained a high precision for both classes, balancing the trade-off between false positives and false negatives. The ROC curve for KNN showed an area under the curve(AUC) of 0.97(figure 1), confirming its strong classification ability across thresholds. On the other hand adjusting the classification threshold based on the ROC curve could reduce false positives or false negatives.

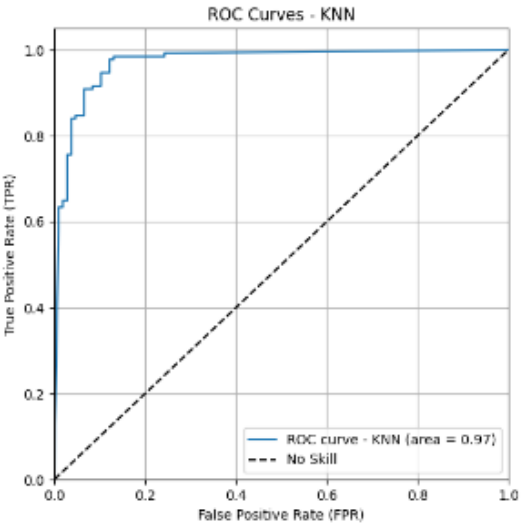

Figure 1, ROC curves -KNN

Model 2: Decision Tree (DT)

The Decision Tree model achieved an 88% accuracy and an F1-score of 0.89. While its recall for Class 1 (89%) was close to KNN, its lower precision for Class 0 (0.87) and recall for Class 0 (86%) indicate potential overfitting, compared to KNN (table 1). The Decision Tree's performance discrepancy between training and validation datasets supports this observation. The ROC curve for DT revealed an AUC of 0.93(figure 2), slightly lower than KNN's, reflecting its lower generalization ability.

Dummy Classifier (Baseline)

The Dummy Classifier served as a baseline model, achieving a 47% accuracy and an F1-score of 0.51, consistent with random guessing. Its performance metrics validate the efficacy of both KNN and DT, as they significantly outperformed this baseline.

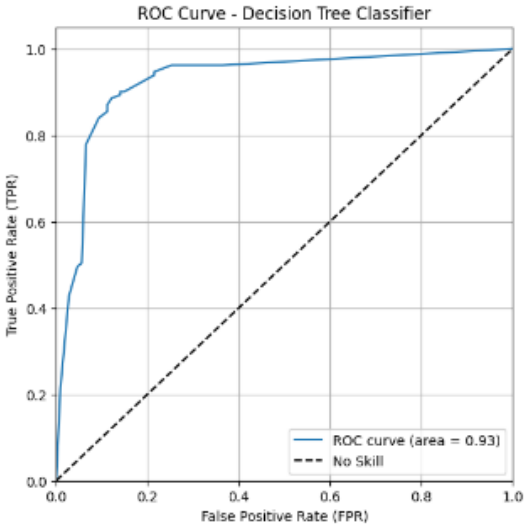

Figure 2, ROC curve - KNN

Performance Comparison

The KNN outperformed Decision Tree in terms of overall performance, achieving higher accuracy (92%) and better recall for Class 0. Its ability to balance class distributions made it a better fit for this classification task. While Decision Trees are powerful classifier, it appeared to slightly overfit the data, as evidenced by its relatively lower performance in cross-validation compared to KNN (figure 3 and 4). The Decision Tree's complexity led to reduced ability to

generalize. However, the case for overfitting can be made for both since there is a fairly high discrepancy between the training scores and the cross-validation scores in DT and KNN.

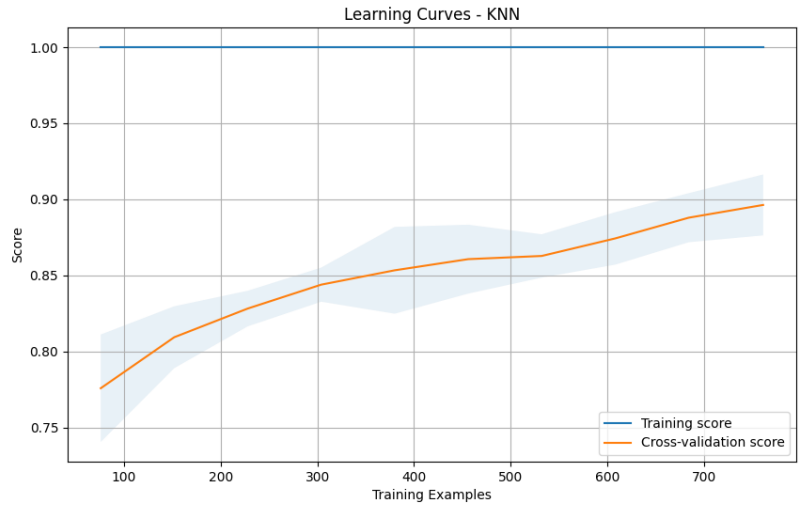

Figure 3, learning curve KNN

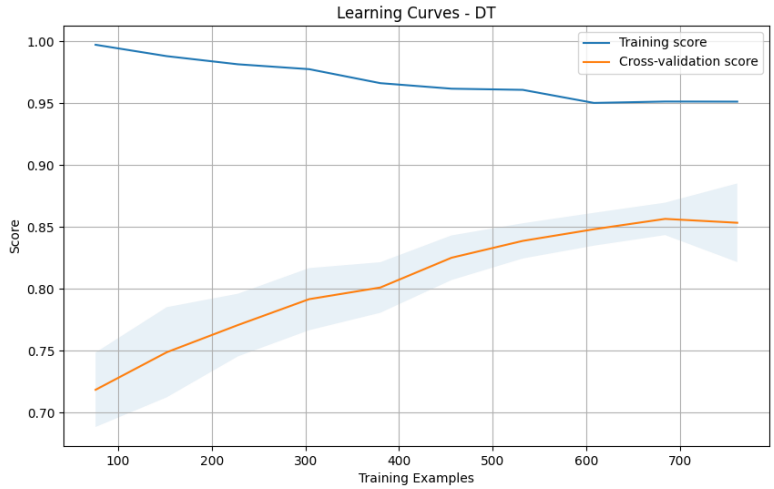

Figure 4, learning curve DT

Model	Accuracy	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)	F1-Score
KNN	92%	0.93	0.92	0.90	0.95	0.93
Decision Tree	88%	0.87	0.89	0.86	0.89	0.89
Dummy Classifier	47%	0.41	0.52	0.42	0.51	0.51

Table 1, Confusion matrix

4. Discussion of Model Improvements

Improving model performance can be approached through a combination of hyperparameter tuning, feature selection, and exploring advanced techniques. For K-Nearest Neighbors (KNN), optimizing the k value is crucial, as smaller values may overfit the data while larger ones might underfit, diluting the impact of closer neighbours. Similarly, fine-tuning the max_depth parameter in Decision Trees can prevent overfitting by controlling tree complexity, ensuring better generalization to unseen data. Feature selection plays a key role in simplifying models and enhancing their predictive power by removing less impactful features, which reduces redundancy and noise in the dataset.

For larger datasets or more complex tasks, advanced models like ensemble methods (e.g., Random Forest or gradient boosting) can deliver higher accuracy by aggregating predictions from multiple weak learners. Using an ensemble method like Random Forest could improve generalization by aggregating predictions from multiple decision trees. Additionally, techniques like SMOTE(Chawla et al., 2002) could balance the dataset, ensuring equal sensitivity across classes. These techniques effectively capture feature interactions and non-linear patterns while mitigating overfitting through mechanisms like averaging or regularization. Neural networks, particularly Deep Neural Networks (DNNs) with multiple hidden layers, are another promising option for complex datasets. DNNs excel at modelling intricate, non-linear relationships, making them suitable for high-dimensional or unstructured data. However, overfitting is a common challenge with DNNs, especially for smaller datasets. This can be mitigated using regularization techniques like dropout, L2 penalties, or early stopping, alongside careful architecture design to avoid overly deep or complex networks that exceed the dataset's capacity. By combining these strategies, models can achieve improved accuracy.

5. Conclusion

This report has demonstrated the application of machine learning algorithms to both regression and classification tasks, showcasing their strengths, weaknesses, and potential areas for improvement. For the classification task, K-Nearest Neighbors (KNN) emerged as the most effective model, achieving high accuracy (92%) and superior sensitivity in identifying heart conditions (Class 1), with a recall of 95%. In comparison, the Decision Tree (DT) model, while achieving reasonable performance (88% accuracy), exhibited signs of overfitting, limiting its generalization capabilities. The Dummy Classifier served as a baseline, highlighting the significant performance gains achieved by the KNN and DT models.

Key factors contributing to KNN's success include its ability to balance precision and recall, making it well-suited for medical applications where false negatives can have serious consequences. Conversely, the Decision Tree's lower precision and recall for Class 0 indicate a need for further refinement, such as pruning or leveraging ensemble techniques to mitigate overfitting.

The discussion outlined practical strategies for improving classification models, such as hyperparameter tuning, advanced feature selection, and exploring more sophisticated methods like ensemble models and Deep Neural Networks (DNNs). These approaches can enhance model accuracy and scalability, particularly for large or complex datasets.

In summary, this analysis shows the importance of selecting and optimizing machine learning models based on the specific characteristics of the dataset and task. While the KNN model excelled in this case, continued refinement through advanced techniques and careful validation will further enhance the utility of these algorithms in real-world applications.

References

Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, pp.321–357.

Awan, A.A., (2023) *K-Nearest Neighbors (KNN) Classification with R Tutorial*. Available at: https://www.datacamp.com/ (Accessed: 5 January 2025).

GeeksforGeeks (2023) *Mathematical explanation of KNearest Neighbour*. https://www.geeksforgeeks.org/mathematical-explanation-of-k-nearest-neighbour/.

Keylabs. (2024, October 11). *Random Forest: Ensemble Learning Technique*. Retrieved January 5, 2025, from https://keylabs.ai/blog/random-forest-ensemble-learning-technique/

Khalid, Z., (2021) *EE212 Mathematical Foundations for Machine Learning and Data Science 2021*. https://www.zubairkhalid.org/ee212_2021.html.

mexwell. (2024). *Heart Disease Dataset*. Kaggle. Available at: https://www.kaggle.com/datasets/mexwell/heart-disease-dataset/data [Accessed 5 Jan. 2025].

Appendix

Feature	Code Given	Unit	Data Type
Age	Age	Years	Numeric
Sex	Sex	1, 0	Binary
Chest Pain Type	Chest pain type	1, 2, 3, 4	Nominal
Resting Blood Pressure	Resting bp s	mm Hg	Numeric
Serum Cholesterol	Cholesterol	mg/dl	Numeric
Fasting Blood Sugar	Fasting blood sugar	1, 0	Binary
Resting Electrocardiogram Results	Resting ecg	0, 1, 2	Nominal
Maximum Heart Rate Achieved	Max heart rate	71–202	Numeric
Exercise Induced Angina	Exercise angina	0, 1	Binary
Oldpeak = ST Depression	Oldpeak	-	Numeric
Slope of Peak Exercise ST Segment	ST slope	0, 1, 2	Nominal
Class	Target	0, 1	Binary

Table A2, Attributes for heart conditions

Feature	Description
Sex	1 = Male, 0 = Female
Chest Pain Type	1: Typical angina, 2: Atypical angina, 3: Non-anginal pain, 4: Asymptomatic
Fasting Blood Sugar	1 = True (Fasting blood sugar > 120 mg/dl), 0 = False
Resting Electrocardiogram Results	0: Normal, 1: ST-T wave abnormality (inversions/ST elevation or depression > 0.05 mV), 2: Left ventricular hypertrophy (Estes' criteria)
Exercise Induced Angina	1 = Yes, 0 = No
Slope of Peak Exercise ST Segment	1: Upsloping, 2: Flat, 3: Downsloping
Class	1 = heart disease, 0 = Normal

Table A3, Description of the nominal attributes.