OPTIMIZING DECISION TREES USING GENETIC ALGORITHMS AND DEEP NEURAL NETWORK FEATURE EXTRACTION

by

ANDREI CIPRIAN ALEXANDRU
URN: 6592956

A dissertation submitted in partial fulfilment of the
requirements for the award of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

May 2023

Department of Computer Science
University of Surrey
Guildford GU2 7XH

Supervised by: Ioana Boureanu

I declare that this dissertation is my own work and that the work of others is acknowledged and indicated by explicit references.

Andrei Ciprian Alexandru
May 2023

Abstract

Machine learning models that are based on decision trees have gained popularity because of their simplicity, interpretability, and utility. However, it is often observed that these models experience a reduction in accuracy and an escalation in intricacy, leading to suboptimal performance and excessively large tree sizes. The aim of this study is to improve the structure of decision trees through the application of genetic algorithms and to enhance their effectiveness by utilising deep neural networks for the purpose of feature extraction.

The proposed hypotheses were validated using three distinct datasets, namely a cancer dataset, an image dataset, and a credit loans dataset. The cancer and credit loans datasets were utilised in their original form, whereas the image dataset underwent preprocessing through a pre-trained neural network to enable efficient feature extraction. Following the process of feature extraction or neural network prediction, genetic algorithms were utilised to augment the training of decision trees.

Upon assessment, it was noted that the optimised decision trees exhibited better accuracy and simplicity in comparison to their non-optimised counterparts across two of the datasets. The genetic algorithm was able to maintain the performance of the decision tree despite the reduction in nodes. Furthermore, the utilisation of feature extraction through deep neural networks on the image dataset resulted in a significant enhancement of the decision tree's precision.

The study's results highlight the possibility of combining genetic algorithms and deep neural network feature extraction to improve decision trees and generate machine learning models that are more effective and reliable. In order to advance the field, it would be beneficial to investigate alternative optimisation techniques and their potential incorporation with various machine learning models. This could lead to enhanced performance and more comprehensible results.

Acknowledgements

Without the help and contributions of a number of people and organisations, it would not have been possible to finish this project. I would like to show my deepest gratitude to all of them. First of all, I'd like to thank my supervisor, Dr. Ioana Boureanu, for all her help, advice, and support during the study process. I'd also like to thank the University of Surrey for making sure I had the tools and space I needed to do this study. I'd also like to thank the people who put together the datasets. Their work collecting, sorting, and sharing the data made this project possible. Last but not least, I'd like to thank my friends and family for always being there for me and helping me through this. Their support has always given me a reason to keep going.

Contents

List of Figures

List of Tables

Abbreviations

ADAM	Adaptive Moment Estimation
CNN	Convolutional Neural Network
cxpb	crossover probability
DT	Decision Tree
DTBDNN	Decision Tree-Based Deep Neural Network
DNN	Deep Neural Network
FCN	Fully Connected Networks
FNA	Fine Needle Aspirate
GA	Genetic Algorithm
mutpb	mutation probability
NN	Neural Network
RNN	Recurrent Neural Networks
SGD	Stochastic Gradient Descent

Chapter 1

Introduction

1.1 Background and Motivation

Decision trees are a popular machine learning technique due to their ease of use and interpretability in sectors including health, finance, and computer vision. Rules based on input data features help in categorisation and regression. Decision trees have problems with overfitting, complexity, and poor performance. Researchers have studied many optimisation solutions to improve usefulness and reduce complexity.

Genetic algorithms (GAs) have been demonstrated to be effective in optimising decision trees, representing one type of optimisation methodology. Genetic algorithms (GAs) can employ mutation, crossover, and selection techniques to identify the most optimal decision tree for a specified fitness function in terms of simplicity and efficiency.

Deep neural networks (DNNs) are frequently utilised for the purpose of processing images, audio, and text, all of which are widely recognised as complex data tasks. Deep neural networks (DNNs) have the potential to acquire abstract features from raw data by means of non-linear transformation layers. In the context of managing extensive datasets, decision trees may potentially derive advantages from the feature extraction capabilities of deep neural networks (DNNs).

This study optimises decision trees using deep neural network feature extraction and genetic algorithms to improve machine learning model accuracy and efficiency. Merging GAs with DNNs aims to create robust, interpretable decision-making models that can be swiftly deployed across many application domains by simplifying decision trees and improving performance through

feature extraction.

1.2 Problem Statement

Decision trees' simplicity and readability have been frequently praised. Regardless of their use, decision trees typically demonstrate poor performance, overfitting, and unmanageable tree sizes, resulting in decreased accuracy and increased complexity. Deep neural networks, on the other hand, have shown that they are very good at doing difficult tasks, even though their interpretability is often thought to be poor, making them seem like black boxes. So, understanding the reasoning behind their forecasts is a big obstacle to using these models in areas where transparency and reliability are of the utmost importance.

The goal of this research is to increase the efficacy and understandability of decision trees and neural networks by combining genetic algorithms and deep neural network feature extraction for decision tree optimisation. The objective is to devise a methodology that capitalises on the benefits of the three aforementioned methodologies, culminating in machine learning models that exhibit greater precision, efficacy, and comprehensibility and can be applied across diverse domains.

The aim of the current study is to use genetic algorithms to arrange decision trees in the best possible way in order to solve the aforementioned problem. The goal of this improvement method is to make decision trees less complicated while keeping or improving their accuracy. Deep neural networks will be used in the study to extract important features from complex data. The above characteristics will then be used as inputs for the process of decision tree learning, which will increase the interpretability of neural networks.

1.3 Objectives

Overfitting, excessive complexity, and poor performance are all downsides of decision trees. This research optimises decision trees using deep neural network feature extraction and evolutionary algorithms to solve these issues. The study's main goals are:

- Using genetic algorithms, simplify decision trees. Implementing genetic operators such as mutation, crossover, and selection, as well as constructing a fitness function that appro-

priately accounts for both tree complexity and performance, are all part of this process.

- For complex data like images and structured data, deep neural networks can be used to extract important features. This includes training a DNN on the given dataset and extracting features from an intermediate layer to use as inputs for decision tree learning.
- Analyse the suggested approach's performance using a range of datasets, including image, cancer, and credit loan data. This entails comparing how well the optimised decision trees perform against their unoptimised counterparts.
- Examine the optimised decision trees' interpretability and robustness, since these characteristics are critical for real-world applications that demand dependable decision-making models.

The study aims to enhance the development of decision tree models by focusing on these objectives, with the ultimate goal of achieving greater effectiveness, precision, and comprehensibility.

1.4 Limitations

Despite the optimistic results produced in this study, there are significant limitations that should be acknowledged:

Dataset size: This study used different but modest datasets. The performance of the proposed optimisation strategy may change with larger datasets or ones with different characteristics, and further studies may be needed to determine its effectiveness in those cases.

Optimisation parameters: Preliminary experiments determined genetic algorithm parameters including population size, mutation probability, and crossover probability, which may not be ideal for all datasets and scenarios. A more exhaustive search for optimal parameters could potentially lead to further improvements in the optimisation process.

Feature extraction: The study uses deep neural networks for feature extraction, which may not work for all data. Depending on the data, different applications or datasets may use different feature extraction algorithms.

Interpretability and robustness: The optimised decision trees in this study are more interpretable and resilient than their unoptimised counterparts. However, it's hard to measure interpretability

because it can be subjective. Research is needed to improve decision tree interpretability and robustness.

The application of genetic algorithms for decision tree optimisation is the main topic of this paper. However, other optimisation techniques, such as particle swarm optimisation or gradient-based methods, may also offer good outcomes. Comparisons with other optimisation methods may reveal the best decision tree optimisation methods.

Chapter 2

Literature Review

2.1 Decision Trees

Decision trees are a common method of supervised learning that can handle both classification and regression problems (Breiman, Friedman, Stone & Olshen 1984). Classification problems involve assigning discrete labels to data instances, such as spam or not spam for emails, while regression problems involve predicting continuous values for data instances, such as house prices (Hastie, Tibshirani, Friedman & Friedman 2009). The main difference between classification and regression trees is how they measure the quality of a split. Classification trees use metrics such as entropy, Gini impurity, and information gain to assess how well the split separates the classes (Quinlan 1986). Regression trees use metrics such as variance reduction or mean squared error to assess how well the split fits the data (Breiman et al. 1984).

In machine learning, decision trees have a number of benefits. As they mimic human reasoning, they are simple to understand and explain (Rokach & Maimon 2008). They can deal with missing values and outliers, and they can handle categorical and numerical features (Quinlan 1993). They are also non-parametric, which means they do not rely on any presumptions about how the data are distributed (Hastie et al. 2009). Decision trees do have some drawbacks, though. They are prone to overfitting, especially when they become too deep or complex (Quinlan 1987). They can also be unstable because even minor data changes can have a significant impact on the tree's structure (Dietterich 2000). They may also have a bias in favour of characteristics with more levels or categories (Kotsiantis 2013).

To get around these restrictions and enhance decision trees' performance, a number of techniques

have been put forth. One of them is pruning, which is the removal of branches or nodes from the tree that do not significantly add to its accuracy or generalisation (Quinlan 1993). Ensemble learning is an additional technique that combines various decision trees to produce a model that is more reliable and accurate (Dietterich 2000). Bagging, boosting, and random forests are some examples of ensemble methods (Breiman 1996, Freund & Schapire 1997, Breiman 2001).

Natural language processing, computer vision, bioinformatics, fraud detection, and recommender systems are just a few of the machine learning problems and domains where decision trees have been used (Rokach & Maimon 2008). They are also frequently utilised as a foundation or as a component of more complicated models. In order to learn from data and make decisions, decision trees are one of the most well-liked and effective machine learning algorithms (Kotsiantis 2013).

2.1.1 Decision Tree Optimisation Techniques

Decision trees are a popular machine learning technique due to their interpretability, simplicity of implementation, and ability to handle continuous and categorical inputs. They must also balance complexity and generality.

Pruning, growth, and hybrid procedures are all used to optimise decision trees. Pruning reduces the decision tree's size and complexity by removing branches that don't improve accuracy or generality (Esposito, Malerba, Semeraro & Kay 1997). Pruning can occur before or after tree development. Early halting, or pre-pruning, limits tree growth by limiting depth, node samples, or impurity reduction. If the tree is too small, pre-pruning can create underfitting. After the tree is fully grown, cost-complexity pruning uses a validation set or metric to cut it. Post-pruning reduces overfitting and complexity while maintaining performance. Bottom-up or top-down pruning strategies also exist. Bottom-up methods start at the leaves and remove or replace irrelevant nodes (Norton 1989). Top-down methods remove or replace low-accuracy or complex subtrees from the root (Mingers 1989).

Hybrid methods maximise decision trees by pruning and growing. Genetic algorithms use evolutionary concepts to find the best decision tree structure. Genetic algorithms create a population of decision trees, which are then frequently subjected to genetic operators like mutation and crossover to create new tree designs. Fitness functions evaluate the trees' efficacy and complexity (Karami, Nittari, Traini & Amenta 2021). After a certain number of generations, the best decision tree is picked (Kotsiantis 2013).

2.2 Deep Neural Networks

The exceptional performance of deep neural networks (DNNs) as a machine learning model has recently attracted a lot of attention (LeCun, Bengio & Hinton 2015). DNNs can learn hierarchical data representations because they are made up of layers of interconnected neurons (Hinton, Osindero & Teh 2006). DNNs, specifically Convolutional Neural Networks (CNNs), have consistently outperformed traditional machine learning techniques in image classification tasks (Krizhevsky, Sutskever & Hinton 2017). Indicating that features extracted from intermediate layers can generalise well to new tasks, transfer learning has also grown in popularity as a technique for using pre-trained DNNs for other tasks (Yosinski, Clune, Bengio & Lipson 2014).

An input layer, several hidden layers, and an output layer make up a deep neural network's architecture. Each layer has a collection of neurons connected to neurons in surrounding layers. The neurons enable the network to learn intricate patterns and representations by applying non-linear activation functions to their weighted inputs (Glorot, Bordes & Bengio 2011).

Convolutional Neural Networks (CNNs): CNNs, a popular subtype of DNNs, are created specifically to process grid-like data, such as images (LeCun, Bottou, Bengio & Haffner 1998). In order to identify regional patterns and spatial relationships, they use convolutional layers to analyse the input data through filters. CNNs are very good at tasks like image classification and object detection (Simonyan & Zisserman 2014, He, Zhang, Ren & Sun 2016).

Recurrent Neural Networks (RNNs): RNNs are a subset of DNNs that are made to handle sequential data, like time series or natural language (Mikolov, Karafiát, Burget, Cernockỳ & Khudanpur 2010). RNNs have loops that allow data to endure over multiple time steps, allowing the network to recognise temporal dependencies and patterns (Hochreiter & Schmidhuber 1997).

Fully Connected Networks (FCNs): According to (Bengio, Courville & Vincent 2013), FCNs are composed of multiple layers of fully connected neurons, with each layer's neurons connected to each other. Classification, regression, and feature extraction are just a few of the tasks that these networks are used for.

When a deep neural network is being trained, its weights are changed to reduce the loss function, which gauges the difference between the output that was predicted and the output that was actually produced (Goodfellow, Bengio & Courville 2016). Usually, gradient-based optimisation algorithms like stochastic gradient descent (SGD) or its variants are used to accomplish this

(Kingma & Ba 2014). Backpropagation is an effective technique for calculating the derivatives of the loss function with respect to the network weights, and it is used to compute gradients (Rumelhart, Hinton & Williams 1986).

2.2.1 Feature Extraction from Deep Neural Networks

Deep neural networks can be used for a variety of tasks, such as decision tree optimisation, because they can learn hierarchical feature representations from the raw input data (Bengio et al. 2013). The features extracted from the input data can be used as input for the decision tree by using a pre-trained neural network (Yosinski et al. 2014). This method combines the robust feature extraction abilities of DNNs with the understandability and simplicity of decision trees, improving performance and generalisation (Caruana, Lou, Gehrke, Koch, Sturm & Elhadad 2015).

In this study, we concentrate on utilising deep neural network feature extraction to improve the optimisation of decision trees with genetic algorithms . We seek to enhance the decision tree's performance while preserving its interpretability and reducing its complexity by supplying it with high-level, meaningful features that have been extracted from the data.

2.3 Genetic Algorithms

Genetic algorithms (GAs) are a class of optimisation techniques inspired by the process of natural selection and were first introduced by Holland in the early 1970s (Holland 1992). They have been widely used for various optimisation problems, including decision tree optimisation (Cantu-Paz 2000).

GAs work by iteratively evolving a population of candidate solutions through selection, crossover (recombination), and mutation operations (DE 1989). The fitness function, which evaluates the quality of each solution, plays a crucial role in guiding the search process towards optimal solutions (Mitchell 1998).

Genetic algorithms apply a series of genetic operators to evolve the population towards better solutions. The main genetic operators are:

Selection: Selection is the process of choosing individuals from the population based on their

fitness (Jebari & Madiafi 2013). This operator favours individuals with higher fitness, increasing their probability of being selected for reproduction (Katoch, Chauhan & Kumar 2021). Common selection methods include tournament selection, roulette wheel selection, and rank-based selection (Jebari & Madiafi 2013).

Crossover: Crossover is the process of combining the genetic material of two parent individuals to create one or more offspring (Pachuau, Roy & Kumar Saha 2021). This operator promotes exploration of the solution space by generating new combinations of genetic material (Kora & Yadlapalli 2017). Common crossover methods include one-point crossover, two-point crossover, and uniform crossover.

Mutation: Mutation is the process of introducing small, random perturbations to an individual's genetic material (Katoch et al. 2021). This operator ensures diversity within the population and helps prevent premature convergence to suboptimal solutions. Common mutation methods include bit-flip mutation, swap mutation, and Gaussian mutation.

The genetic algorithm starts with an initial population of randomly generated candidate solutions. In each iteration, or generation, the algorithm evaluates the fitness of each individual and applies the genetic operators to create a new population. The algorithm terminates when a predefined stopping criterion is met, such as reaching a maximum number of generations or achieving a desired fitness level.

Genetic algorithms can be used to optimise decision trees by searching for the best combination of features, tree structure, and splitting criteria (Banzhaf, Nordin, Keller & Francone 1998). The chromosomes in this context represent feature masks, which indicate the subset of features to be used for constructing the decision tree (Utgoff & Brodley 1990). The fitness function is designed to evaluate the quality of the decision tree based on its depth, number of branches, and predictive performance (Barros, Basgalupp, de Carvalho & Freitas 2012). Genetic algorithms have been successfully applied to decision tree induction, demonstrating their effectiveness in finding high-quality trees (Cantu-Paz 2000).

In this study, we apply genetic algorithms to optimise decision trees using features extracted from deep neural networks. By combining the powerful feature extraction capabilities of deep neural networks with the optimisation potential of genetic algorithms, we aim to construct decision trees with improved performance, reduced complexity, and enhanced interpretability.

2.4 Related Work

Deep neural networks, genetic algorithms, and decision tree optimisation have a wealth of literature. This section will discuss the most relevant studies that influenced the research.

Many optimisation studies have improved decision tree performance and interpretability. Pruning, which removes non-predictive branches from a decision tree, was first introduced by Breiman et al. in 1984 (Breiman et al. 1984). The C4.5 algorithm (Quinlan 1993) refines decision tree splitting criteria to minimise class distribution entropy in partitions.

Several researchers have optimised decision trees using genetic algorithms, focusing on feature selection, tree structure, and splitting criteria. Cantu-Paz's multi-objective genetic algorithm optimises decision tree size and accuracy (Cantu-Paz 2000). Bhargava's genetic algorithm for feature selection in decision tree induction reduced features and improved tree performance (Bhargava, Sharma, Bhargava & Mathuria 2013).Since they can learn high-level, abstract representations from raw data, deep neural networks have become popular for feature extraction. Bengio et al. provided a comprehensive overview of deep learning's principles and applications, highlighting its effectiveness in image recognition, natural language processing, and speech recognition (Bengio et al. 2013). Razavian showed that features extracted from a pre-trained convolutional neural network can be used to improve classifier performance across tasks and domains (Sharif Razavian, Azizpour, Sullivan & Carlsson 2014).

Recent studies have combined decision tree optimisation with deep neural network feature extraction to improve performance and interpretability. The model compression approach Caruana et al. trained a decision tree using deep neural network soft targets (class probabilities). This method yielded a smaller, more accurate, and more interpretable decision tree than the neural network (Caruana et al. 2015).

Another innovative approach, the decision tree-based deep neural network (DTBDNN), was introduced in a study which aimed to build a model that acquires knowledge using a deep neural network (Arifuzzaman, Hasan, Toma, Hassan & Paul 2023). This acquired knowledge is then expressed in another model that exploits the hierarchical decision tree structure to predict classification decisions efficiently and with good accuracy. This model is specifically designed for nonlinear data classification and outperforms other state-of-the-art models when classifying nonlinear data.

This study uses genetic algorithm-based decision tree optimisation and deep neural network feature extraction to create a new method. Creating decision trees with improved performance, reduced complexity, and increased interpretability across datasets and domains is the goal.

Chapter 3

Methodology

3.1 Data Collection and Pre-processing

The datasets utilised in this research were deliberately selected to encompass a diverse array of data types and classification tasks, thereby facilitating a more exhaustive assessment of the models and techniques employed. The datasets utilised in this study are binary classification problems and have been made available to the public on Kaggle, a prominent data science platform that offers a diverse range of datasets for the purpose of research.

The rationale for selecting this particular image dataset is rooted in the potential it offers for investigating the interpretive capacity of neural networks within the domain of visual data. The high dimensionality and lack of structure in image data pose significant difficulties for numerous machine learning algorithms. Furthermore, the task of distinguishing between images depicting cats and dogs is a prevalent and comprehensible issue, facilitating an instinctive analysis of the models' cognitive processes.

The Wisconsin Breast Cancer dataset is a widely recognised dataset within the machine learning community, encompassing structured data pertaining to the diagnosis of breast cancer. The provision of a biomedical context is imperative as interpretability assumes a pivotal role in enabling clinicians and patients to comprehend and place reliance on the prognostications of the model. The dataset comprises various continuous features that are obtained from breast mass images, thereby rendering it a significant problem for investigating the comprehensibility of decision trees and neural networks.

The dataset pertaining to Credit Risk presents a distinct domain, namely that of financial risk analysis. The dataset in question pertains to a practical situation where the ability to interpret the data is of utmost importance due to regulatory and commercial imperatives. The dataset consists of multiple tabular data points that are pertinent to the evaluation of credit risk. This presents a distinct data structure and problem domain in comparison to datasets related to images and biomedicine.

The preparation of these datasets for modelling necessitated the implementation of appropriate pre-processing procedures. In relation to the image data, the procedures undertaken encompassed the standardisation of the images to a uniform size and the normalisation of the pixel values. The pre-processing of tabular datasets encompassed the management of missing values, normalisation of numerical features, and the encoding of categorical features. The pre-processing procedures were implemented to ensure that the data was appropriately formatted for input into the neural networks and decision trees.

The combination of these datasets yielded a diverse and abundant foundation for investigating and showcasing the methodologies employed in this research to enhance the comprehensibility of neural networks and decision trees.

3.1.1 Image Dataset

The image dataset used in this study is sourced from Kaggle, a popular platform for data science and machine learning competitions. The dataset contains a collection of cat and dog images, which serves as a benchmark for binary classification tasks in the field of computer vision. The dataset is publicly available under the CC0 license, making it accessible for research purposes (Kaggle 2018).

Cats and dogs are equally represented in the 10,000 images. These images reflect the diversity and complexity of real-world data (3.1). Training and test sets ensure a fair assessment of the methodology.

Figure 3.1: Example images from the dataset.

In the image dataset, the class distribution is balanced, with an equal number of cat and dog images (3.2).

The dataset contains a diverse range of image sizes and aspect ratios. Most of the images (about 4,000) have a width between 400 and 500 pixels, while the rest are below 400 pixels(3.3).

In terms of image height, the highest peak is observed between 350 and 400 pixels (about 2,500 images), and the second-highest peak is at 500 pixels (about 1,500 images) (3.4).

The aspect ratio distribution reveals that most images have a ratio between 1.2 and 1.3 (approximately 3,000 images), after which the frequency drops drastically (3.5).

This variety in image dimensions and aspect ratios adds complexity to the classification task, making it an interesting and challenging dataset to work with for developing and optimising decision trees based on deep neural network feature extraction.

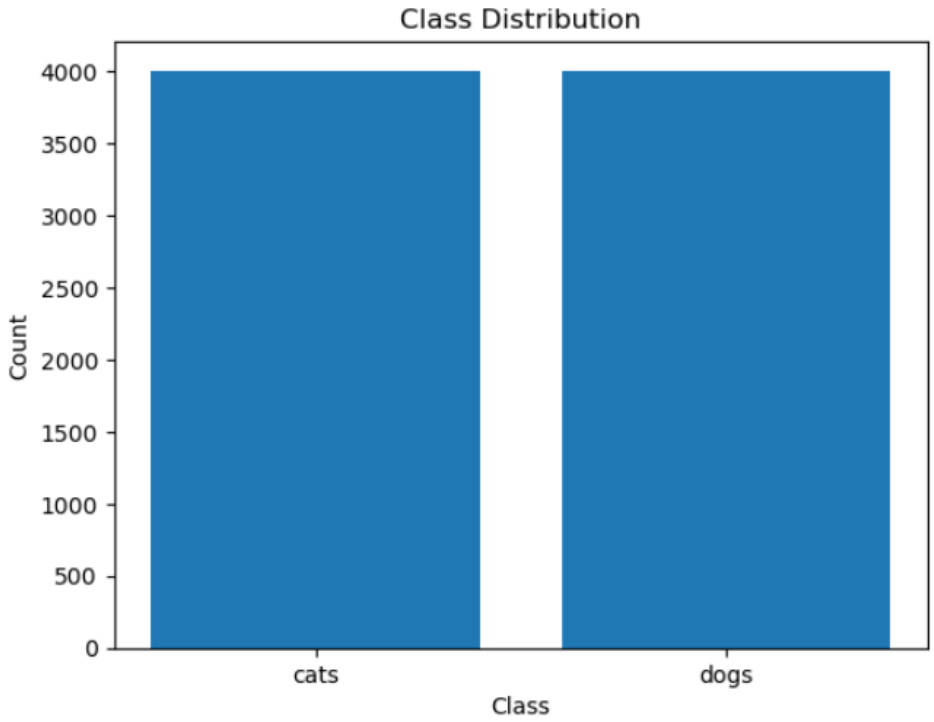


Figure 3.2: Class distribution of the cat and dog dataset.

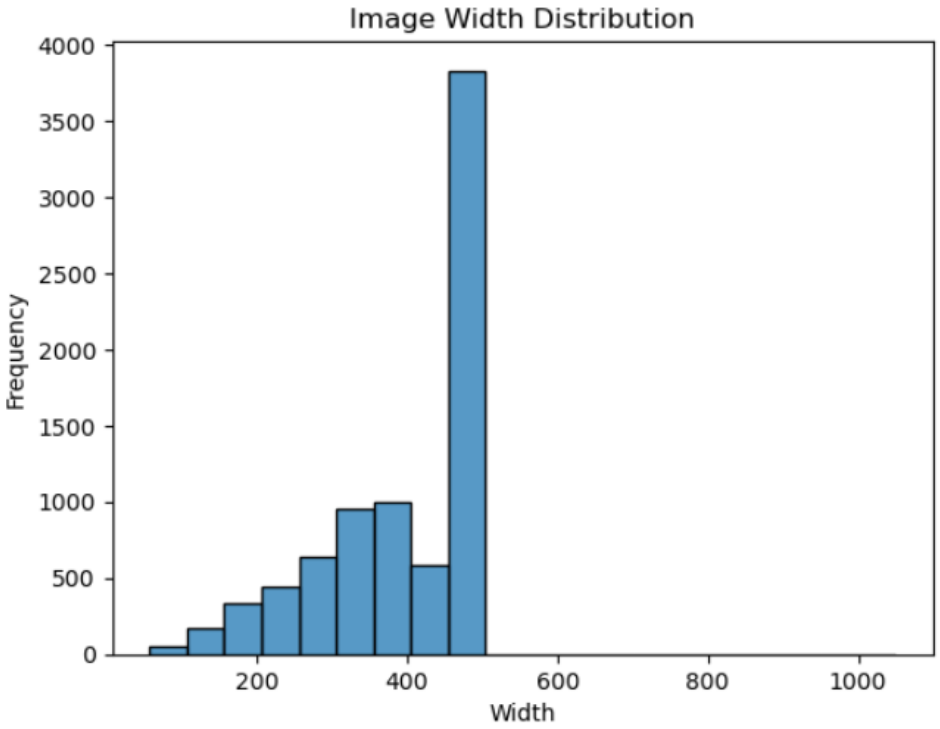


Figure 3.3: Image width distribution.

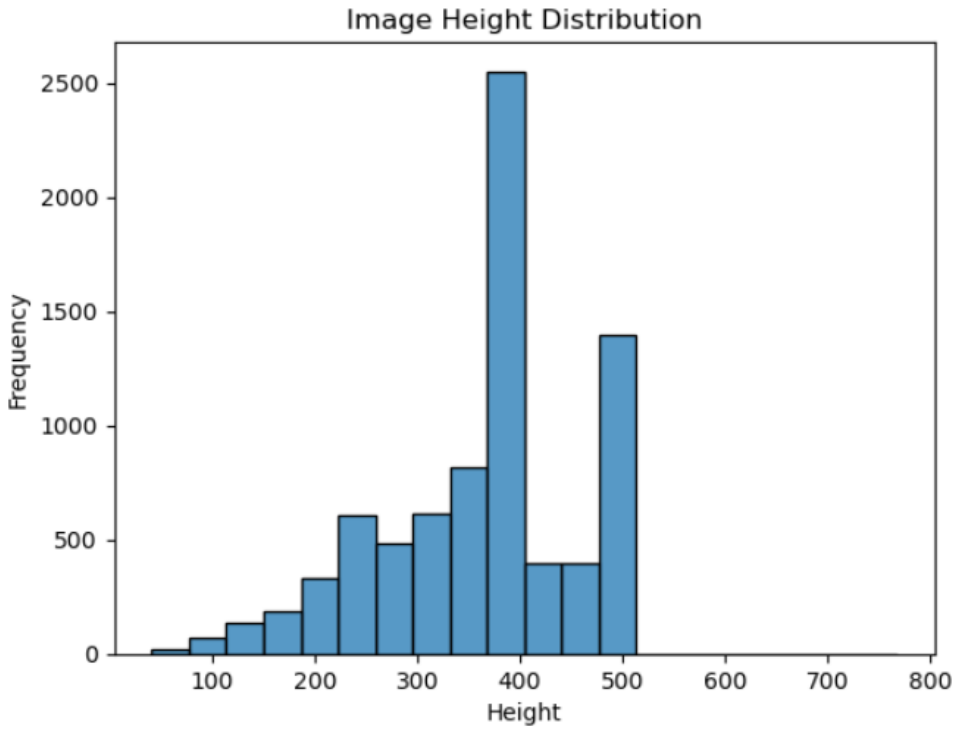


Figure 3.4: Image height distribution.

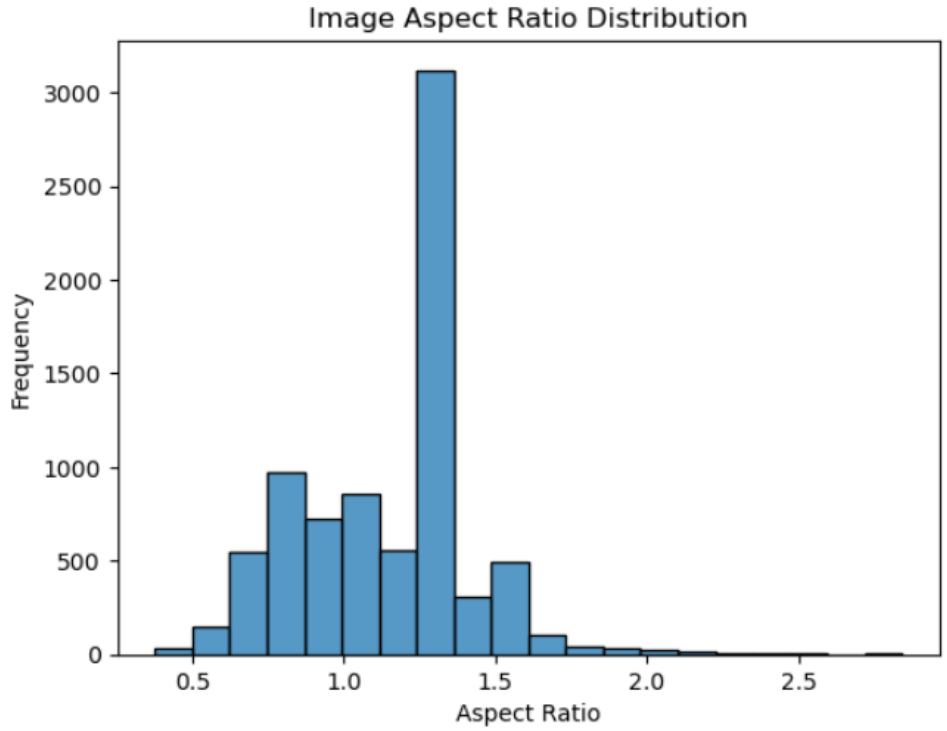


Figure 3.5: Aspect ratio distribution.

3.1.2 Cancer Dataset

This study used the Kaggle-available Breast Cancer Wisconsin (Diagnostic) dataset (Kaggle 2016). 569 samples with 32 attributes including ID, diagnosis, and 30 real-valued input features were computed from a digitised image of a breast mass fine needle aspirate (FNA). Image features were calculated using the mean, standard error, and worst or largest values of the following ten characteristics:

• Radius • Texture • Perimeter • Area • Smoothness

• Compactness • Concavity • Concave points • Symmetry • Fractal dimension

These features are used to classify the breast mass as malignant (0) or benign (1). Preprocessing included dropping the "id" column, mapping the diagnosis column to numerical values, scaling the features with StandardScaler, and splitting the dataset into training and testing sets.

Box plots showed that most features have values close to 0, while area-mean, area-worst, perimeter-worst, and perimeter-mean have significantly larger values. To visualise distributions, features were divided into two groups: those with smaller values (3.6) and all of them (3.7). This separation improved box plot distribution visualisation.

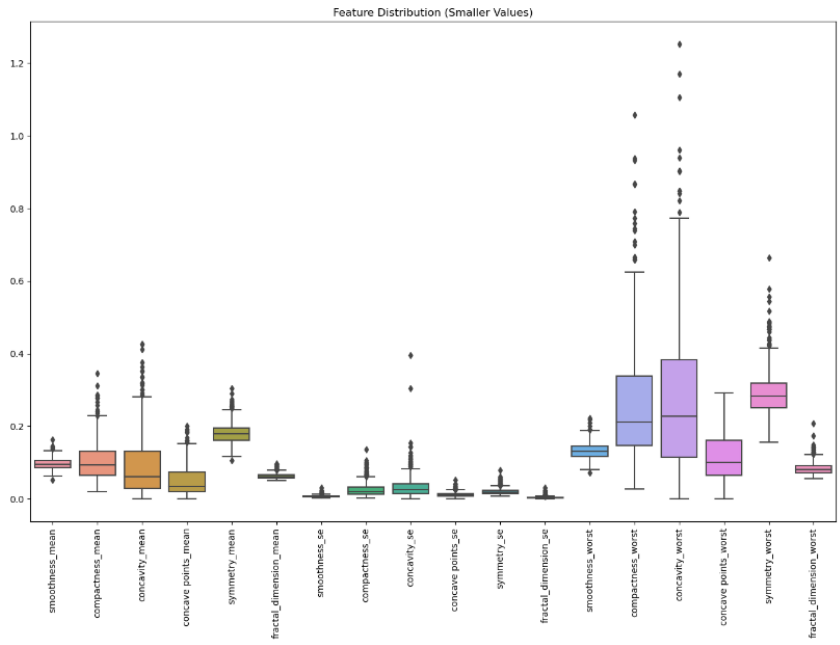


Figure 3.6: Distribution of the smaller value features.

The class distribution in the dataset is slightly imbalanced, with approximately 150 more benign

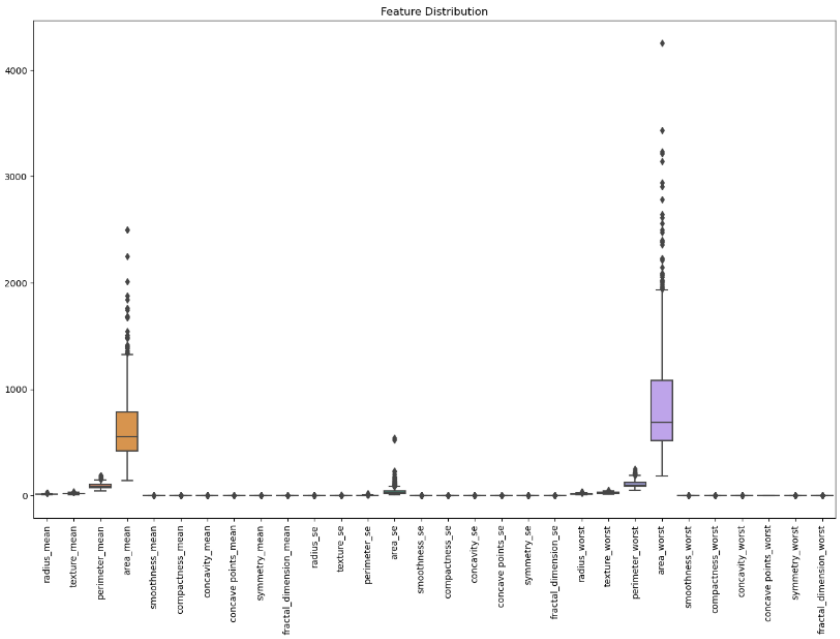


Figure 3.7: Distribution of all the features.

samples than malignant samples (3.8). While this imbalance is not severe, it is important to consider it when evaluating the performance of the machine learning models to ensure that they can perform well on both classes.

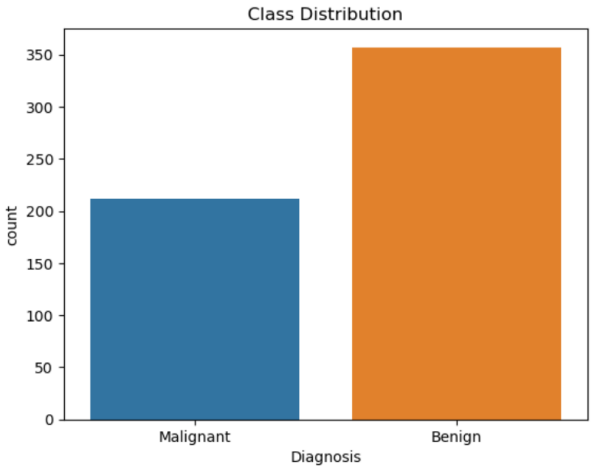


Figure 3.8: Distribution of classes of the cancer dataset.

3.1.3 Credit Loans Dataset

Kaggle's Credit Loans dataset contains loan applicants' data. (Kaggle 2023*a*)

The dataset has 49 columns with categorical and numerical features about customers' finances and personal lives. These features predict loan default risk. The dataset includes customers' loan duration, credit amount, instalment commitment, residence history, age, existing credits, and number of dependents, among other features. The dataset also includes categorical features on customers' checking account, savings account, employment, and financial status. The dataset has 1,000 customer records. This dataset's target variable is the binary *class-good* column, which indicates customer credit risk. The distributions of the features are displayed in the histograms for the numerical columns. Most customers have shorter loan durations, lower credit amounts, fewer existing credits, and are younger. The instalment commitment, residence since, and number of dependents columns have less skewness (3.9).

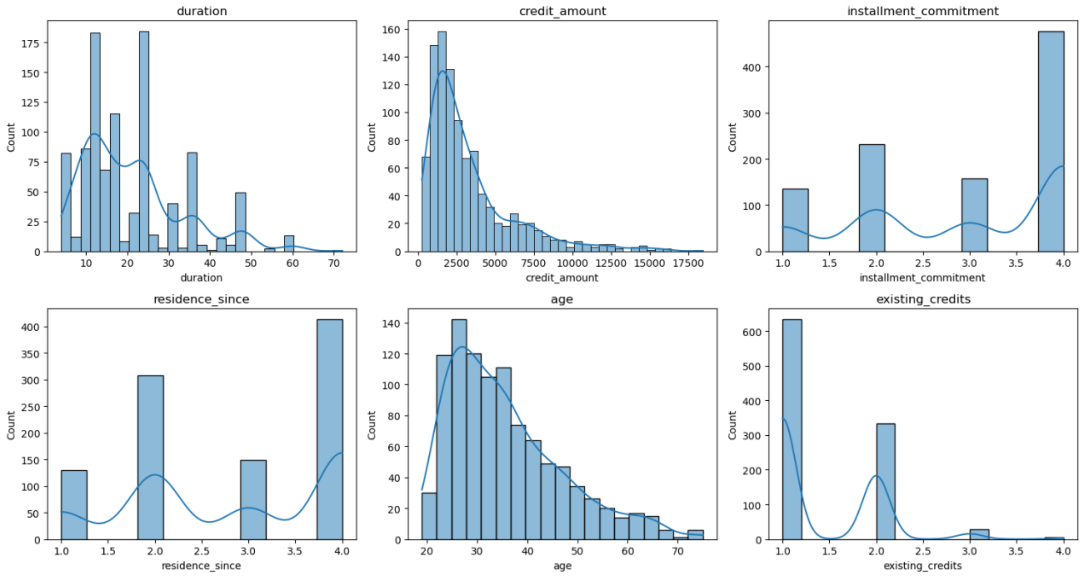


Figure 3.9: Histograms for the numerical columns of the credit loans dataset.

The count plots for the categorical columns reveal the prevalence of different categories in the dataset. The checking account status, savings account status, and employment status columns, for instance, show that the dataset contains diverse customer profiles with varying financial backgrounds (3.10).

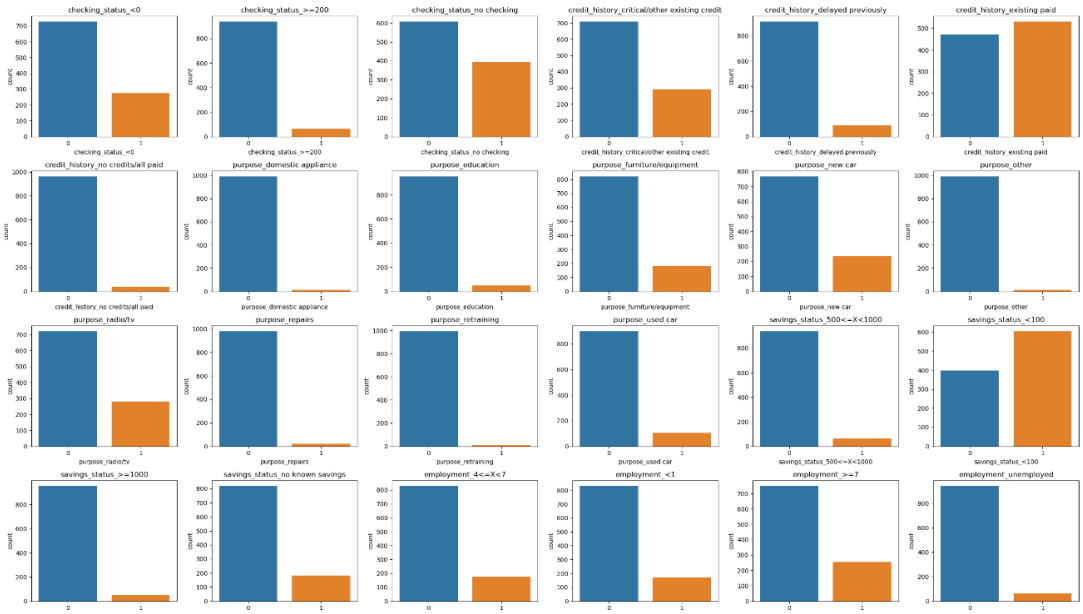


Figure 3.10: Countplots for the categorical columns of the credit loans dataset.

The class distribution in the Credit Loans dataset is imbalanced. There are 700 records for good credit risks and 300 records for bad credit risks . This represents a 70:30 split between good and bad credit risks. This imbalance should be considered during model evaluation, as it may affect the model's performance in predicting the minority class (bad credit risks) (3.11).

Figure 3.11: Distribution of classes of the credit loans dataset.

3.2 Deep Neural Network Architecture and Training

This section discusses the neural network architecture and training process used to extract dataset features.

Dataset	Model	Batch size	Epochs	Optimiser
Image	ResNet18	64	4	SGD
Cancer	CancerNet	64	11	Adam
Credit	CreditNet	128	25	SGD

Table 3.1: Parameters chosen for each dataset.

3.2.1 Image dataset

The ResNet architecture, introduced by (He et al. 2016), has been widely adopted in the field of image classification due to its ability to effectively learn features from images while mitigating the problem of vanishing gradients in deep networks. In this study, we used the ResNet-18 variant of this architecture

The chosen image size of 224x224 is a common standard in deep learning image classification tasks. This size balances computational economy with model performance, making it popular. Smaller photos may result in feature loss, whereas larger images use more computational resources without performance improvement.

Model training can be greatly influenced by batch size. Larger batch sizes let the model process more data at once, speeding up training. It demands more memory and may provide fewer precise gradients, which may impact the model's learning capabilities. Smaller batch sizes utilise less memory and compute more precise gradients, but they slow training and make loss less stable. It was discovered that a 64 batch size balanced training speed, memory utilisation, and model performance.

ResNet-18 was selected because it performed similarly to deeper architectures like ResNet-34 and ResNet-50 at a lower computational cost. ResNet-18 consumed 2.7GB of GPU memory and had a test accuracy of 98.57%, while ResNet-34 and ResNet-50 required 4.1GB and 13.8GB, respectively, but only had test accuracies of 99.26% and 99.31%, respectively.

To avoid overfitting while letting the model learn enough features, 4 epochs were used. The

model reduced loss over these four epochs, demonstrating good learning. More epochs may cause overfitting, when the model gets overly specialised to the training data and performs badly on unknown data.

These selections highlight the trade-offs that deep learning requires between computing efficiency, model complexity, and performance. We were able to train a very accurate model efficiently by making wise selections based on empirical testing.

3.2.2 Cancer dataset

The cancer Data uses CancerNet, a custom neural network. CancerNet has three fully connected layers in a feed-forward neural network. The network maps 30 cancer dataset features to a two-class output, indicating benign or malignant diagnosis.

In this study, a batch size of 64 was used for the Breast Cancer dataset, and different optimisers, and amounts of training epochs were tried. In the end, the Adam optimiser with a learning rate of 0.001 was picked, and the model was trained for 11 epochs.

The loss gradient tells the optimiser how the model's weights should be changed. As possible optimises, Adam (Adaptive Moment Estimation) (Kingma & Ba 2014) and SGD (Stochastic Gradient Descent) (Bottou 2010) were tried out.

In all of these tests, Adam did better than SGD every time. Adam got a test accuracy of 99.12% after 11 epochs, while SGD, which had a faster learning rate of 0.01, only got a 98.12% accuracy. In comparison to SGD, Adam was able to achieve high accuracy in fewer epochs, showing faster and more effective learning. This is due to Adam's adaptive learning rate, which changes based on the slopes and lets him get closer to the truth faster.

A range of epochs from 10 to 2000 was tested. With the Adam optimiser, the model got the best test accuracy of 99.12% in just 11 epochs. Training for more epochs did not improve test performance by a large amount, and in some cases it even made it worse. This means that the model was able to learn the necessary features from the data within 11 epochs, and further training only led to overfitting.

The learning rate determines the step size at each iteration while moving toward a minimum of a loss function. It was found that a learning rate of 0.001 gave good results every time, no matter how many epochs were used. The loss function was picked to be CrossEntropyLoss (Goodfellow

et al. 2016). CrossEntropyLoss is often used in binary (and multi-class) classification jobs because it measures how well a classification model that gives a probability value between 0 and 1 does its job. Cross-entropy loss goes up as the difference between the predicted probability and the real label grows. This makes it a good measure for classification tasks. It also has the bonus of being differentiable, which lets optimisation methods based on gradients be used.

In short, Adam was picked as the optimiser based on testing. The number of epochs was set to 11, and the learning rate was set to 0.001. These decisions enabled the training of an accurate rate model in a fair amount of time without overfitting. CrossEntropyLoss was also used because it works well for classification jobs and can be used with gradient-based methods of optimisation.

3.2.3 Credit dataset

CreditNet is a custom neural network for credit loan datasets. CancerNet and CreditNet are feed-forward neural networks with four fully connected layers.

A set of experiments was carried out on the Credit dataset to ascertain the optimal parameters and structure for the neural network. The study conducted various experiments that involved manipulating the batch size, number of epochs, and optimiser in order to determine the optimal configuration that resulted in the highest level of test accuracy.

The evaluation encompassed both Adam (Kingma & Ba 2014) and Stochastic Gradient Descent (SGD) (Bottou 2010) optimisation techniques. The Adam optimiser is recognised for its flexible nature, as it modifies learning rates according to the gradient. On the other hand, the Stochastic Gradient Descent (SGD) optimiser is distinguished by its straightforwardness and effectiveness, particularly when combined with momentum to mitigate oscillations.

Although Adam demonstrated proficiency in specific setups, it was noted that Stochastic Gradient Descent (SGD), utilising a learning rate of 0.01 and momentum of 0.9, surpassed Adam's performance by attaining a test accuracy of 82.50% within 25 epochs. The enhanced performance observed may be ascribed to the momentum component, which facilitated the acceleration of SGD in the pertinent direction and alleviated oscillations, resulting in expedited and more dependable convergence.

The study involved conducting experiments utilising 10, 25, and 100 epochs. Contrary to popular belief that increasing the number of epochs would lead to improved performance, the findings

of this study indicate that the model attained the highest level of accuracy on the test set, specifically 82.50%, after undergoing 25 epochs. Training for more epochs did not lead to improved performance and, in fact, decreased the test accuracy in the case of 100 epochs, possibly due to overfitting.

Batch sizes of 32, 64, and 128 were evaluated. It was found that a batch size of 128 yielded the highest test accuracy. Increasing the batch size can result in a more precise approximation of the gradient, which can enhance the stability and dependability of the training process.

The results of the conducted empirical tests indicate that the most effective setup for the credit dataset involves utilising the SGD optimiser with a learning rate of 0.01 and momentum of 0.9, a batch size of 128, and undergoing 25 epochs of training. The experimental configuration optimised the precision of the tests, suggesting that the model acquired knowledge efficiently from the dataset without experiencing overfitting.

3.3 Feature Extraction from Pre-trained Neural Network (Image Dataset)

The extraction of features from the image dataset was accomplished through the utilisation of a pre-trained neural network. The methodology involved utilising the internal representations that had been acquired by the pre-existing model to effectively capture significant features of the input images.

The initial stage of this procedure involved loading the pre-existing neural network and configuring it to operate in evaluation mode, thereby disabling the dropout layers that are incorporated within the network. It is imperative to note that the network is not being trained, but rather utilised for feature extraction purposes.

Subsequently, a forward hook was established and affixed to the average pooling layer of the network. PyTorch provides forward hooks as a functionality that enables the execution of a function each time the forward method of a module is invoked. The output of the average pooling layer was restructured by the hook function and subsequently stored in a globally accessible variable.

The pre-trained neural network was utilised to extract features from the dataset via the *'extract_features'* function. The aforementioned function systematically traversed through each image within the given dataset, subsequently transmitting them through the network and archiving the resulting output of the average pooling layer. Every result was added to a feature list. The labels that corresponded were also preserved.

The forward hook was then eliminated from the average pooling layer. Ensuring prevention of memory leaks is a crucial measure, given that the global variable responsible for storing features would undergo updates with every forward pass executed across the network, even beyond the scope of the 'extract_features' function.

The aforementioned approach yielded a diverse array of characteristics for every image, effectively capturing intricate patterns that were assimilated by the pre-existing model. Subsequent stages of the study utilised these features as input to decision trees.

3.4 Decision Tree Training with Neural Network Predictions or Extracted Features

3.4.1 Training with feature extraction

This section outlines the methodology employed for training the decision tree model using the predictions or extracted features derived from the pre-trained neural network.

Subsequent to the process of extracting features through utilisation of a pre-existing neural network, the following stage involved the utilisation of these features to train a model based on decision trees. The extracted features signify a more sophisticated comprehension of the input data, capturing complex patterns that may not be readily discernible in the unprocessed data.

The aim was to utilise the functionalities of the pre-existing neural network for the purpose of extracting features, and subsequently utilise a decision tree model to acquire knowledge from these advanced features. The objective of this methodology was to amalgamate the advantages of the two models, namely the neural networks' capacity to acquire intricate patterns and produce valuable representations, and the decision trees' interpretability.

The decision tree model was trained utilising the extracted features and their corresponding labels. The training procedure encompassed acquiring decision rules founded on the given fea-

tures that could effectively forecast the labels. The aforementioned outcome was attained via an iterative methodology that involved selecting features, splitting, and tree growth. The primary objective was to optimise the information gain at each decision point.

Subsequently, the decision tree model that underwent training possessed the ability to forecast the classification of a novel instance through the traversal of the decision tree, which was contingent on the feature values of the said instance. The aforementioned procedure ultimately resulted in the formation of a terminal vertex that furnished the ultimate prognostication.

Following, an independent test dataset was utilised to assess the efficacy of the decision tree model's performance. This measure guaranteed an impartial evaluation of the model's capacity to extrapolate to novel, unobserved data. The accuracy of the model was utilised as the metric for evaluating its performance.

The decision tree model was able to leverage the pre-trained neural network's predictions or extracted features, thereby capitalising on the neural network's sophisticated feature learning capabilities. This could potentially result in enhanced predictive performance.

3.4.2 Training with Neural Network Predictions

A different methodology was employed for the datasets pertaining to cancer and credit. Rather than performing feature extraction from a pre-trained neural network, the decision tree was trained using the neural network's predictions.

The initial step involved the training of the neural network model on the corresponding datasets, as outlined in the preceding sections. Subsequent to the training of the neural network model, it was employed to produce prognostications on the training dataset.

The neural network's prognostications were a modified rendition of the input data, encapsulating intricate and non-linear patterns that the neural network acquired during its training. Subsequently, the aforementioned prognostications were employed as the input characteristics to instruct a decision tree algorithm.

The model was defined utilising the DecisionTreeClassifier algorithm from the sklearn library. The invocation of the fit method entailed the utilisation of the scaled training features and the predictions generated by the neural network model as its respective parameters. Subsequently, the decision tree model acquired the ability to establish a correlation between the prognostica-

tions of the neural network and the initial classifications of the training dataset.

The training methodology employed facilitated the acquisition of knowledge by the decision tree model from intricate patterns detected by the neural network in a manner that was more comprehensible compared to the unprocessed neural network outputs. Subsequently, the decision tree model can be employed to generate forecasts on novel data by adhering to the decision regulations acquired during the training phase.

The evaluation of the decision tree model's performance was conducted on a test dataset that was distinct from the one utilised during the training phase. The objective was to evaluate the decision tree's capacity to generalise and produce precise forecasts on novel, unobserved data. The evaluation criterion employed is accuracy.

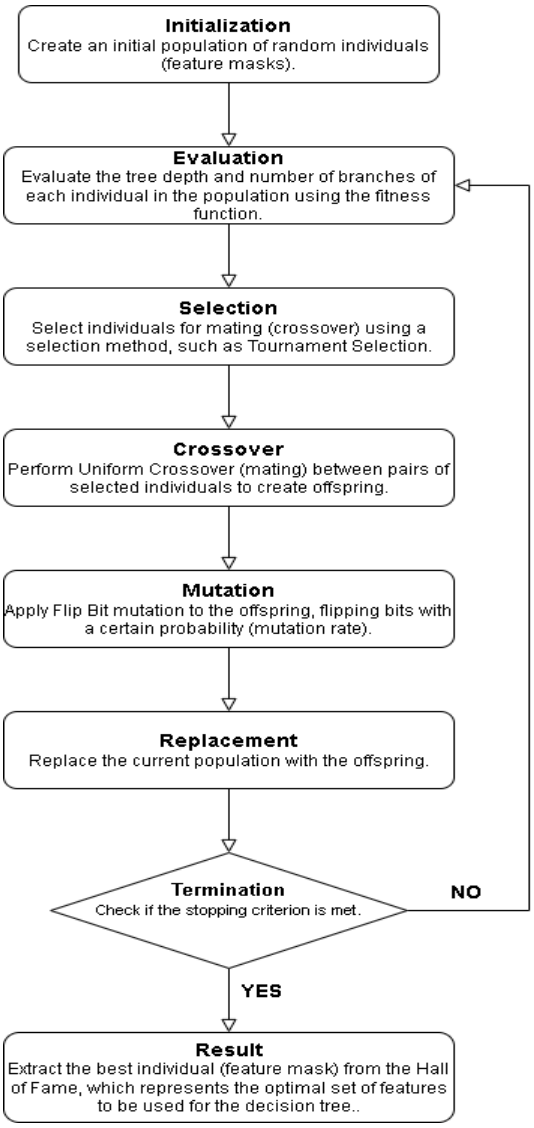
In brief, the decision tree model was able to capitalise on the intricate pattern recognition abilities of the neural network by training it with the neural network's predictions, while still preserving the decision tree model's interpretability benefits.

3.5 Genetic Algorithm for Decision Tree Optimisation

In order to enhance the performance of the Decision Tree classifier, it was subjected to optimisation using a Genetic Algorithm (GA). The objective of this methodology was to ascertain the most efficacious array of characteristics, in conjunction with the optimal depth of the tree and number of branches.

The genetic algorithm was initiated through the generation of individuals possessing binary chromosomes that are equivalent in length to the number of features present in the dataset. Each gene was associated with a specific feature, and its binary value (either 0 or 1) denoted the presence (1) or absence (0) of the feature in the decision tree. The sample size was predetermined to consist of 60 individuals.

Figure 3.12: Genetic algorithm flow chart.



The fitness function was formulated with the aim of instructing a Decision Tree using the chosen features and determining an aggregated score, which takes into account both the depth of the tree (i.e., the maximum distance from the root to a leaf) and the overall count of branches (i.e., the number of leaf nodes minus one). The aim was to reduce the aforementioned score, with the intention of obtaining a less complex and possibly more widely applicable tree.

The study employed conventional genetic operators, namely uniform crossover with a 50% exchange rate and bit-flipping mutation with a 5% probability per bit. The process of reproduction involved the utilisation of tournament selection.

The genetic algorithm was executed for a maximum of 100 generations, incorporating an early stopping mechanism. The algorithm was designed to terminate prematurely and conserve computational resources if the fitness of the most exceptional individual failed to exhibit improvement beyond a specific threshold (0.9, in this instance) for a consecutive number of generations (with a patience level).

Furthermore, in order to promote greater diversity within the population and potentially evade local optima, the probabilities of crossover and mutation were dynamically adapted. In the event that no discernible progress was observed following the attainment of fifty percent of the patience threshold, the likelihoods were augmented with the aim of stimulating further exploration. In contrast, in cases where an enhancement was identified, the likelihoods were reduced to capitalise on the auspicious domains within the exploration area.

Upon completion of the procedure, the most exceptional individual was selected from the population, signifying the supreme combination of characteristics and parameters for the Decision Tree classifier.

3.6 Evaluation metrics

Various metrics were employed to assess the performance of the models in this investigation, considering the characteristics of each model and the particular aims of this research.

Test accuracy is the prevailing metric utilised for classification tasks (Sokolova & Lapalme 2009). The metric evaluates the ratio of accurate predictions generated by the model over the total number of predictions, providing a clear indication of the model's comprehensive efficacy. The evaluation of predictive power for both neural networks and decision trees in this study was primarily based on the metric of test accuracy.

The confusion matrix was utilised in conjunction with test accuracy. The present tabular representation delineates the efficacy of a classification model through the enumeration of accurate positive predictions, accurate negative predictions, erroneous positive predictions, and erroneous negative predictions (Fawcett 2006). The utilisation of additional metrics beyond accuracy offers a more comprehensive and refined evaluation of the model's efficacy. This approach emphasises the equilibrium between the model's sensitivity and specificity. The significance of accurate prediction of one class over the other is particularly noteworthy in datasets pertaining to cancer

diagnosis and credit card fraud, where one class may hold greater importance.

Supplementary metrics were employed for the optimisation of decision trees and genetic algorithm. The metrics of depth and width, which respectively refer to the vertical and horizontal dimensions of a tree, provide a straightforward means of assessing the complexity and interpretability of said tree. A tree exhibiting reduced branching and shallower depth is deemed to be less complex, thereby facilitating ease of interpretation.

The optimisation process in the genetic algorithm was guided by the utilisation of the fitness function (Holland 1992). The aforementioned function was derived from the joint consideration of the depth and width of the tree, thereby serving to strengthen the objective of attaining a tree that is more easily comprehensible, or in other words, less complex.

Finally, the decision paths were visualised for both the original and optimised trees. This provided a method for assessing the interpretability of the trees in a qualitative manner. Through a comparative analysis of these visual representations, it was observed that the optimisation procedure facilitated a greater degree of transparency and comprehensibility in the decision-making process of the tree.

The combination of these metrics yielded a holistic assessment of the models' efficacy, reconciling the imperative of achieving optimal prognostic precision with the equally significant objective of enhancing comprehensibility.

Chapter 4

Experimental Results

4.1 Feature Extraction Results

This section presents Image, Cancer, and Credit dataset feature extraction and classifier performance findings. Table 4.1 compares the pre-trained neural network (NN), unoptimised decision tree (DT), and genetic algorithm-optimised DT.

Dataset	**Classifier**	**Features Used**	**Train Accuracy**	**Test Accuracy**
Image	Pre-trained NN	All(512)	99.93%	99.11%
	Unoptimised DT	All(512)	100.00%	96.59%
	Optimised DT	274	100.00%	96.19%
Cancer	Trained NN	All(30)	97.80%	98.25%
	Unoptimised DT	All(30)	97.80%	98.25%
	Optimised DT	17	100.00%	93.86%
Credit	Trained NN	All(49)	78.88%	80.50%
	Unoptimised DT	All(49)	78.88%	77.00%
	Optimised DT	10	70.12%	73.00%

Table 4.1: Comparison of classifier performance.

Using all 512 features and ResNet18, the Image dataset achieved 99.93% training and 99.11% testing accuracy. Using all features, the unoptimised decision tree had 100.00% training accuracy and 96.59% testing accuracy. The genetic algorithm reduced the decision tree's features to 274, obtaining 100.00% training accuracy and 96.19% testing accuracy.

Using all 30 features and a trained neural network, the Cancer dataset achieved 97.80% training

and 98.25% testing accuracy. Unoptimised decision trees performed similarly, with identical training and testing accuracies. The genetic algorithm optimised the decision tree to use 17 features and obtain 100.00% training accuracy and 93.86% testing accuracy.

A trained neural network with all 49 features gave the Credit dataset 78.88% training accuracy and 80.50% testing accuracy. The trained neural network and unoptimised decision tree performed similarly, with the only difference being the testing accuracy of 77.00%. After optimisation, the decision tree has 10 features, 70.12% training accuracy, and 73.00% testing accuracy.

Three combo charts are utilised for each dataset and classifier to visualise the number of features employed, as well as the train and test accuracies. The graphical representations demonstrate that the employment of optimised decision trees leads to a substantial decrease in the quantity of features, while simultaneously preserving elevated levels of accuracy for both the Image, Cancer and Credit datasets. The results indicate that the feature extraction methodology proficiently eliminates extraneous or duplicative features, thereby augmenting the efficacy of the classifier.

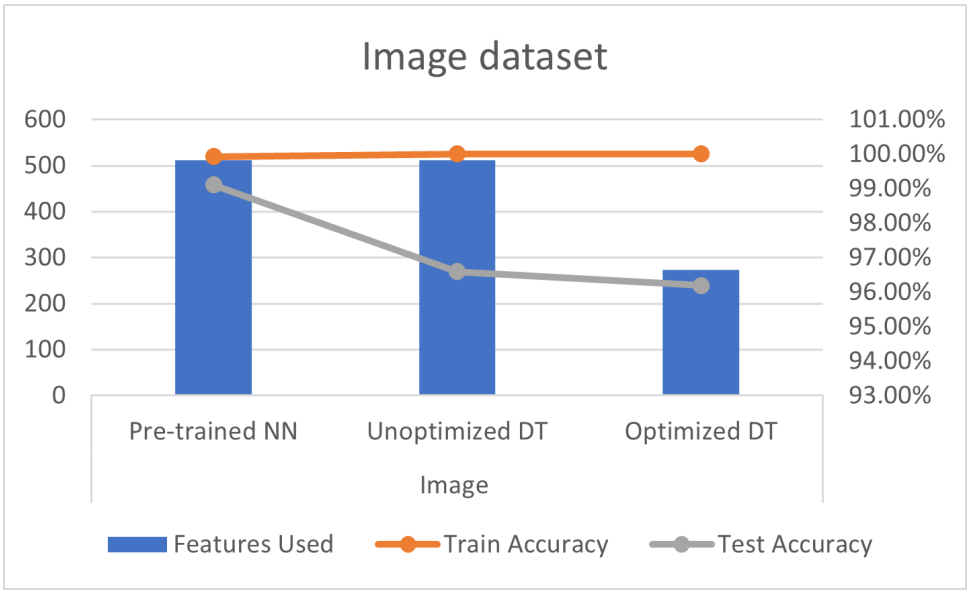


Figure 4.1: Image dataset combo chart.

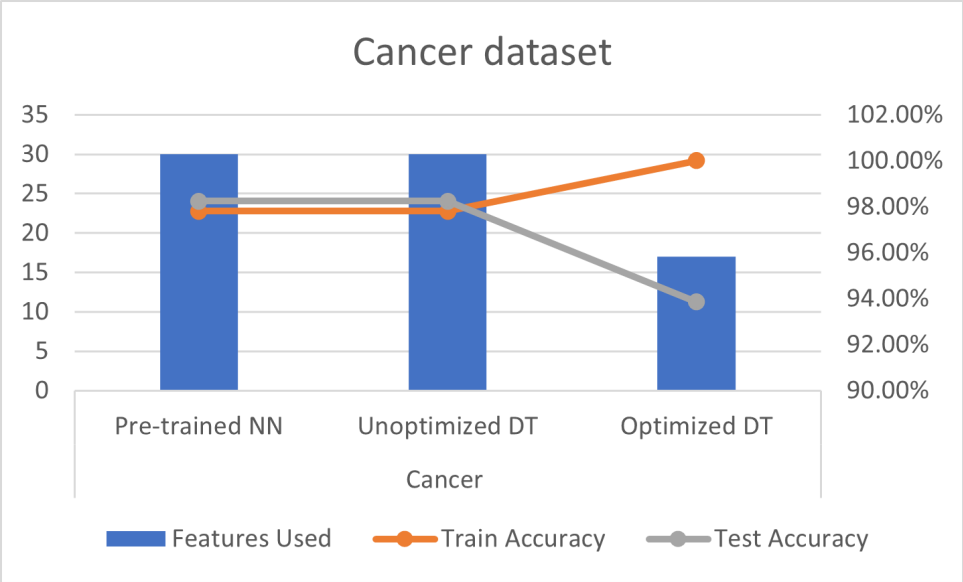


Figure 4.2: Cancer dataset combo chart.

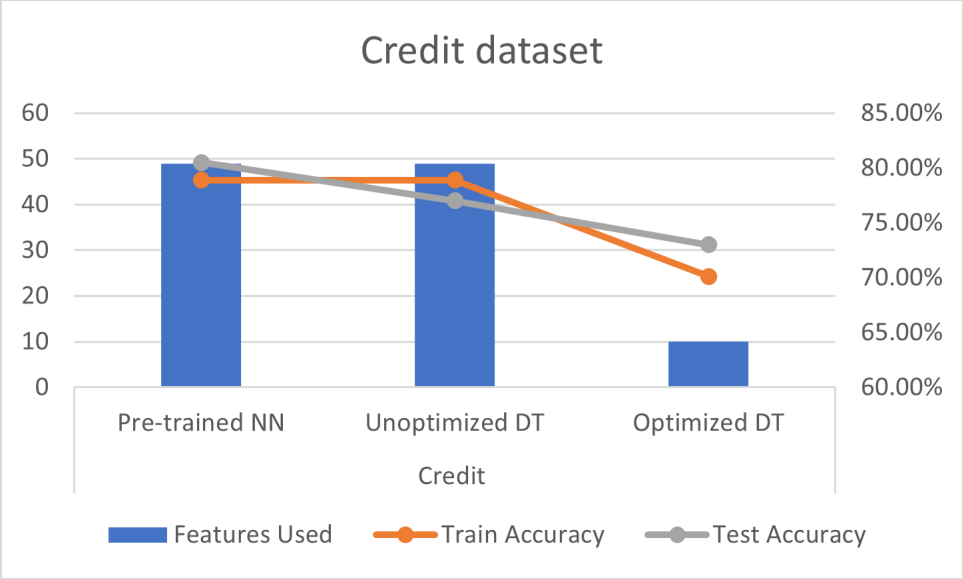


Figure 4.3: Credit dataset combo chart.

While using fewer features, optimised decision trees had similar testing accuracies to unoptimised ones. The genetic algorithm can simplify decision tree models and improve their interpretability without losing speed. However, trained neural networks outperformed optimised decision trees in testing accuracy. Interpretability and simplicity can be more important than testing accuracy, so the trade-off between features and accuracy may be acceptable. These results show that the genetic algorithm for feature extraction and decision tree optimisation may be useful in real-world situations where interpretability and fewer features are needed for efficient and intelligible decision-making.

4.2 Image Dataset

4.2.1 Neural Network Performance

This section discusses the neural network's image dataset performance. After three epochs of training a pre-trained ResNet18 model, the model improved as it learned from the data (4.4). The model generalised well to unseen data, with a final training accuracy of 99.93% and a testing accuracy of 99.11%.

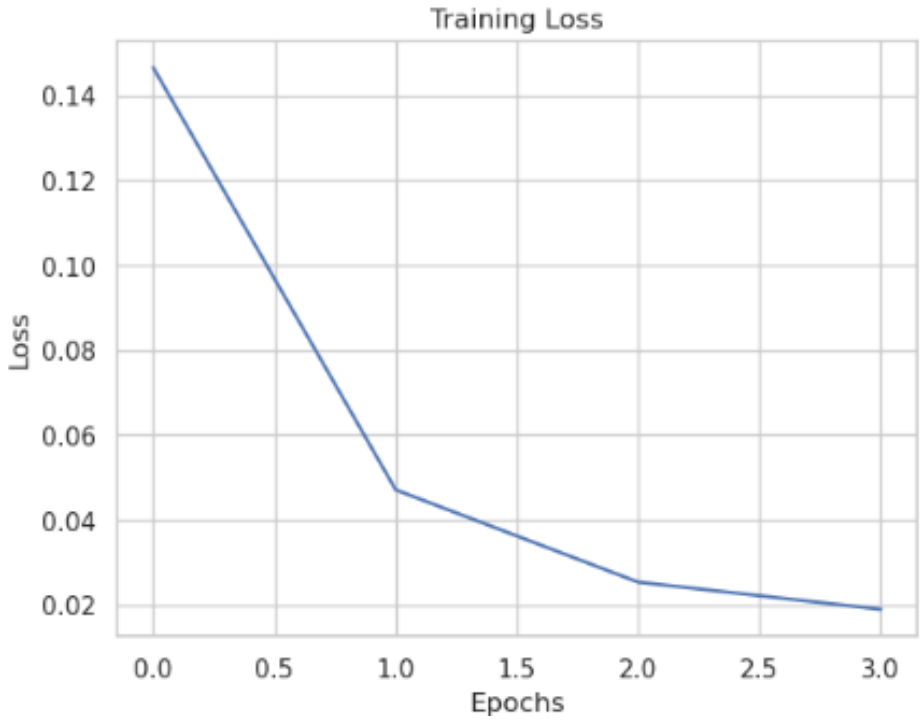


Figure 4.4: Training loss for the pre-trained ResNet model on the image dataset.

The confusion matrix (4.5) shows how many photos were successfully and wrongly identified for further performance analysis. The confusion matrix showed the model correctly categorised 1004 cat and 1001 dog photos. It misidentified 7 cats as dogs and 11 dogs as cats.

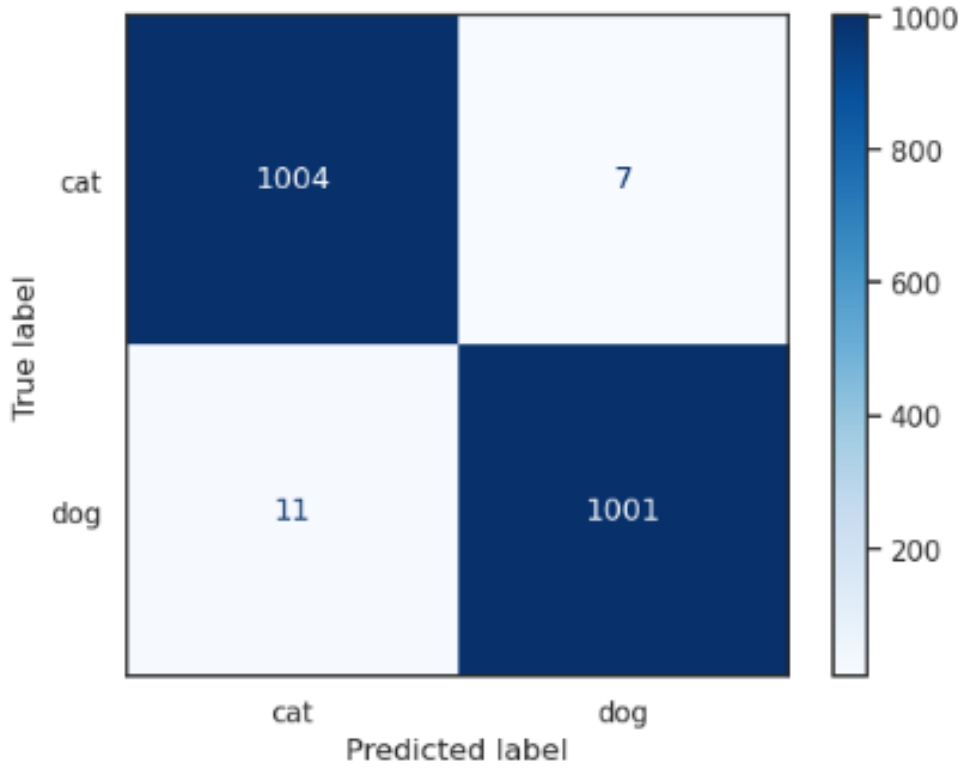


Figure 4.5: Confusion matrix for the pre-trained ResNet model on the image dataset.

Pre-trained ResNet performed well on the image dataset. The confusion matrix showed that the model identified photos accurately. This shows that the neural network is good at categorising cat and dog photos and may categorise similar image datasets.

4.2.2 Decision Tree Performance Before Optimisation

Before optimisation, the decision tree classifier is evaluated. The pre-trained neural network extracted all features for the unoptimised decision tree. The decision tree had 77 branches and 20 depth. The decision tree had 100.00% train accuracy and 96.59% test accuracy.

The confusion matrix (4.6) demonstrates that the unoptimised decision tree correctly identified 984 cat and 970 dog photos. 27 cats and 42 dogs were misclassified.

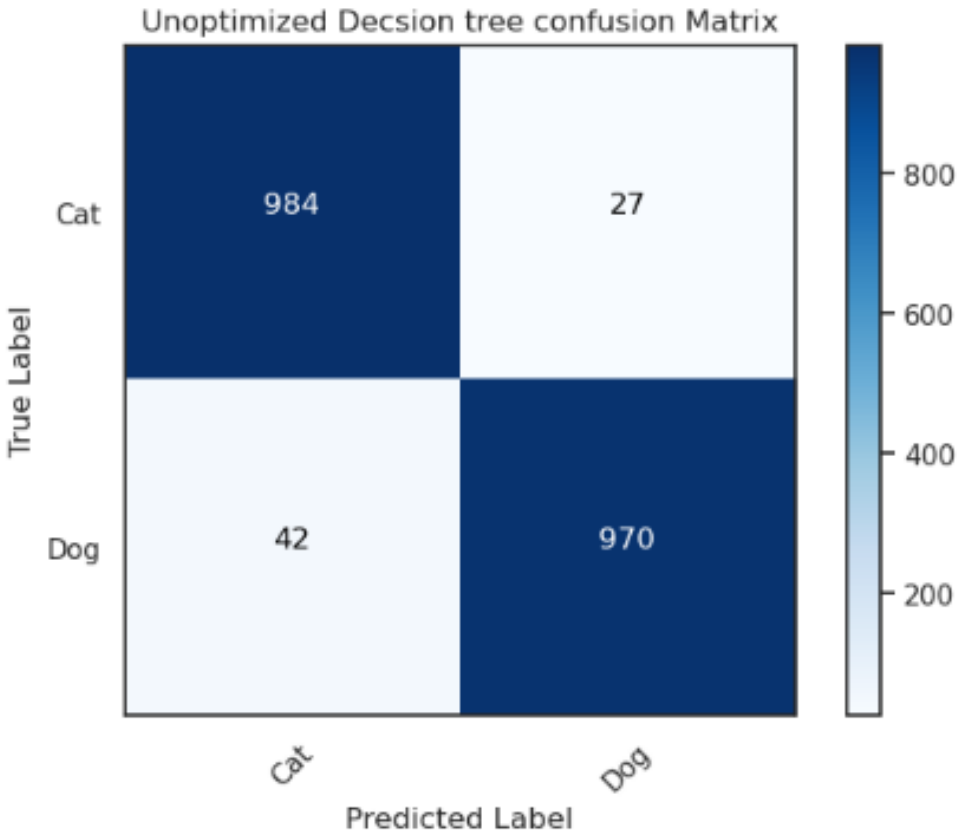


Figure 4.6: Confusion matrix for the unoptimised decision tree on the image dataset.

The neural network outperformed the unoptimised decision tree with. The neural network classifies the image dataset better than the decision tree before optimisation. The optimisation approach aims to reduce the decision tree's complexity while improving classification accuracy.

4.2.3 Genetic Algorithm Optimisation Results

This section shows the decision tree classifier optimisation results using the Genetic Algorithm (GA). With a population of 60 individuals, the GA ran for a maxim of 100 generations. The fitness function was minimising the sum of the tree depth and leaf nodes, excluding the root node . GA parameters started as 0.6 crossover probability (cxpb) and 0.5 mutation probability (mutpb).

The GA optimised the decision tree classifier by reducing tree depth and leaf nodes. Average, minimum, and maximum fitness scores decreased across generations. By the 31st generation, the GA found an optimised decision tree with a fitness score of 85, much lower than the initial unoptimised tree (4.7).

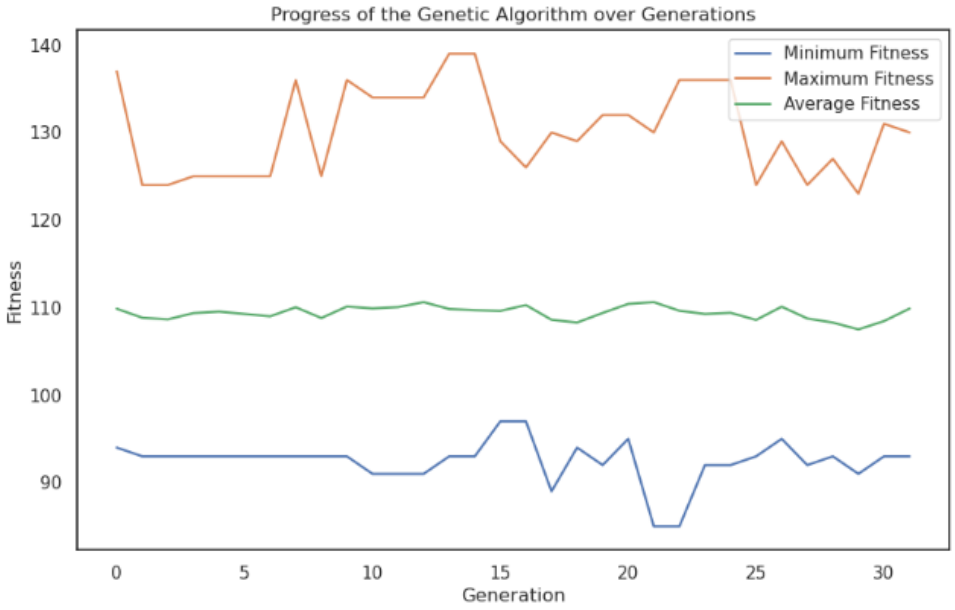


Figure 4.7: Progress of genetic algorithm over generations.

The optimised decision tree's image dataset performance will be compared to the unoptimised decision tree and neural network in the next part.

4.2.4 Decision Tree Performance After Optimisation

The optimised decision tree used 274 features, while the unoptimised tree used 512. A maximum tree depth of 13 and 72 branches resulted. The optimised decision tree had 100.00% train accuracy and 96.19% test accuracy.

The optimised decision tree has fewer branches, a shallower tree depth, and good classification accuracy compared to the unoptimised one. The optimised decision tree had 96.19% test accuracy, slightly down from the 96.59% that the unoptimised one had. The bar chart comparing unoptimised and optimised decision tree accuracy percentages (4.9) shows this performance improvement.

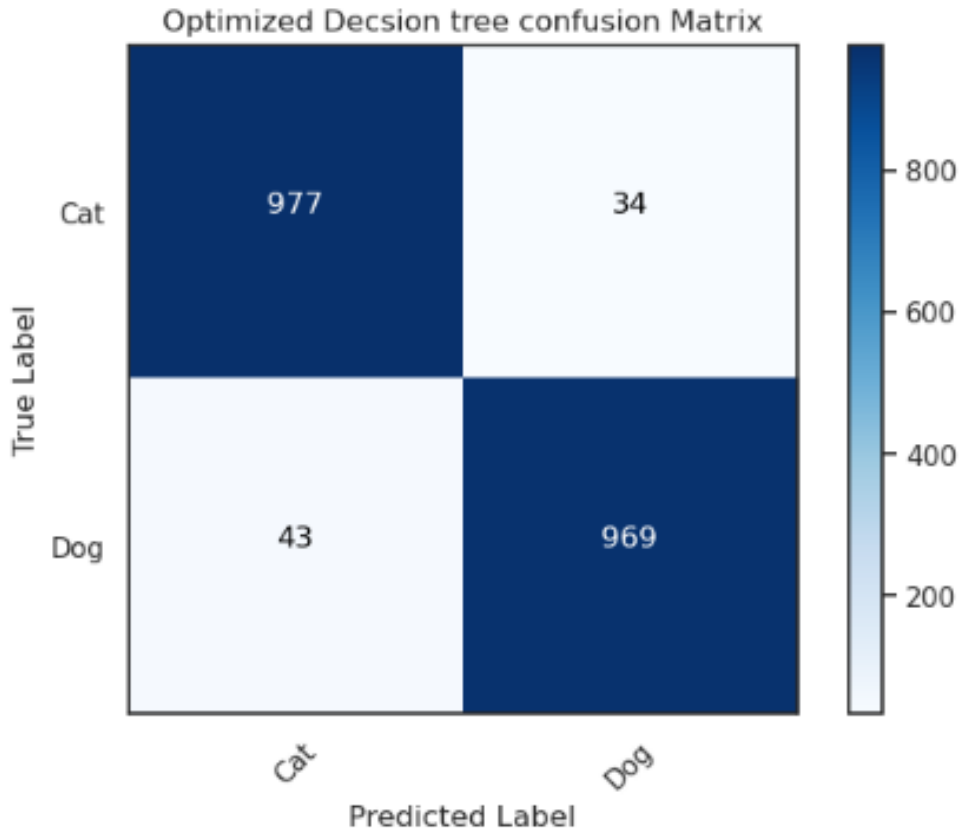


Figure 4.8: Confusion matrix for the optimised decision tree on the image dataset.

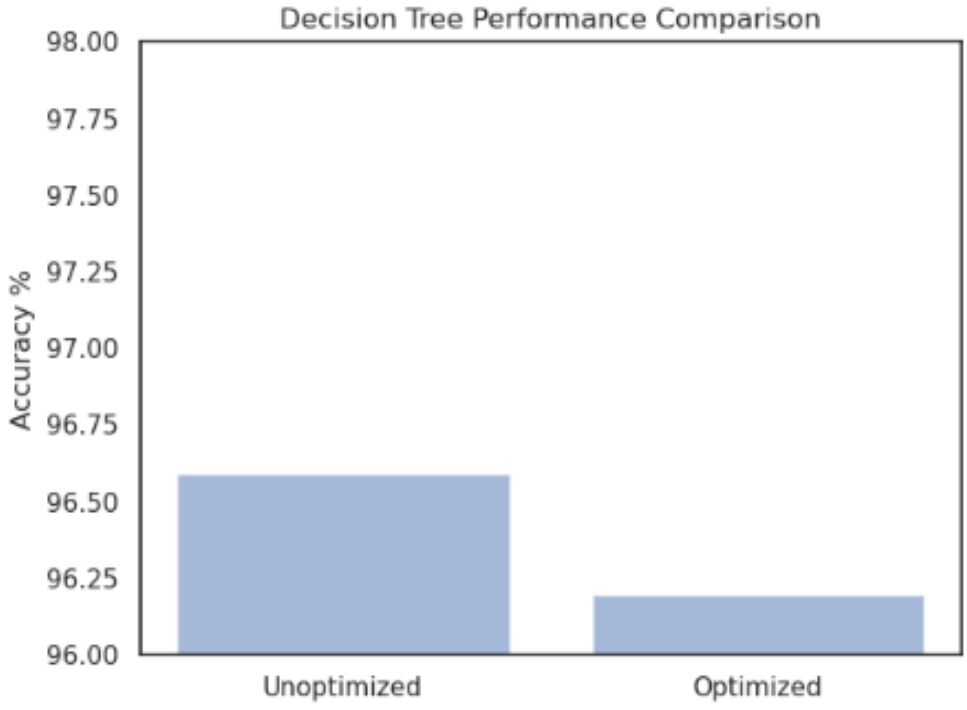


Figure 4.9: Comparison of the accuracy percentage.

4.2.5 Visualisation

The results of the optimisation process can be visually observed in the structure of the decision trees, both before and after the optimisation. The utilisation of visual aids is imperative in augmenting the comprehensibility of decision trees and facilitating a deeper understanding of the decision-making mechanism of the model.

For the purpose of this illustration, a dog's image has been chosen. The image was posted on Pexels by Mathieu Gervais and has a CC0: Public Domain license (Kaggle 2023*b*).

The study presents two sets of decision trees, wherein one set displays the complete trees, while the other set emphasises the pathway utilised by the decision tree in categorising the chosen image. In each of the two pairs, there is a tree that represents the decision tree prior to optimisation, and another tree that depicts the tree subsequent to optimisation.

In the unoptimised tree, the path taken to classify the image includes 19 nodes. This relatively long path indicates a more complex decision-making process with potentially more overfitting.

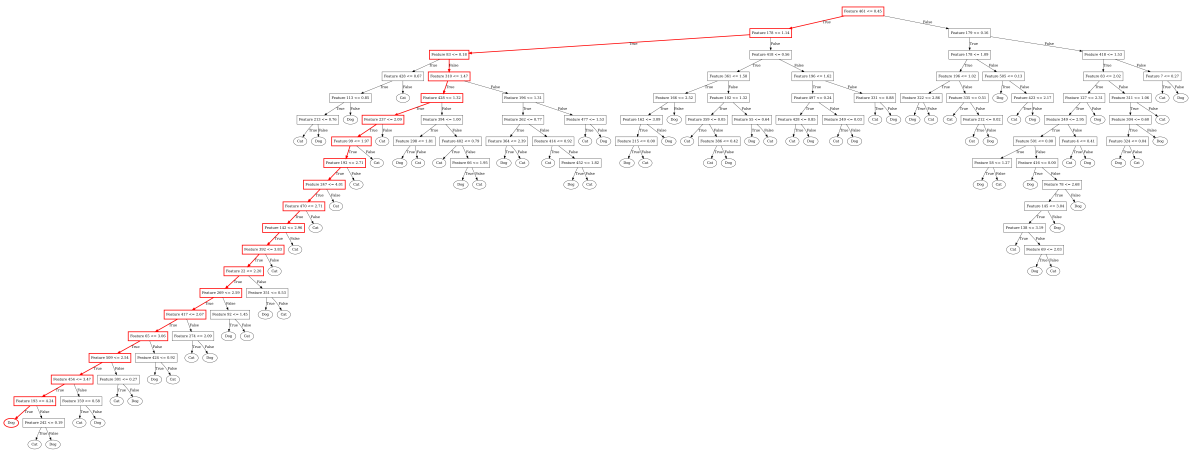


Figure 4.10: Unoptimised decision tree with highlighted path.

Contrarily, the optimised decision tree follows a notably shorter path with only 12 nodes to classify the same image. This implies the utilisation of a streamlined model with a potential reduction in overfitting.

The discernible disparity in intricacy between the optimised and optimised decision trees substantiates the efficacy of the Genetic Algorithm in diminishing the complexity of the decision trees. Despite the lessened complexity, the optimised decision tree remains capable of classify-

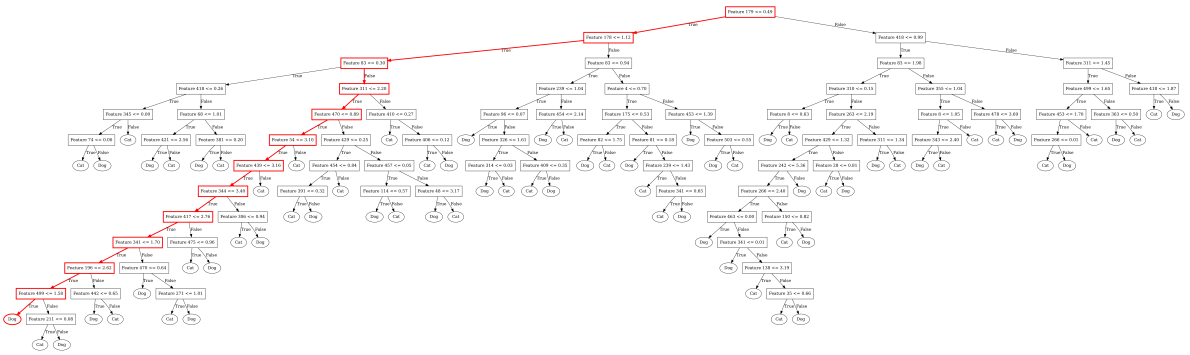


Figure 4.11: Optimised decision tree with highlighted path.

ing the input image correctly. Therefore, it demonstrates the desired trade-off of maintaining predictive power while enhancing model interpretability through size reduction.

4.3 Cancer Dataset

4.3.1 Neural Network Performance

This section evaluates the neural network on Kaggle's Breast Cancer Wisconsin (Diagnostic) Data Set. The dataset predicted whether the cancer was benign or malignant. The dataset and model were explained in the methods section.

The neural network had 99.78% and 98.25% accuracy on the training and test datasets, respectively. The model's prognostications were used to train an unoptimised decision tree, hence the high performance was crucial.

The model's performance improved throughout training as the initial loss value of 0.6058 dropped to 0.0735 after 11 epochs. The model learned well from the training dataset, improving its accuracy. Figure (4.12) shows the training loss decreasing over 11 epochs.

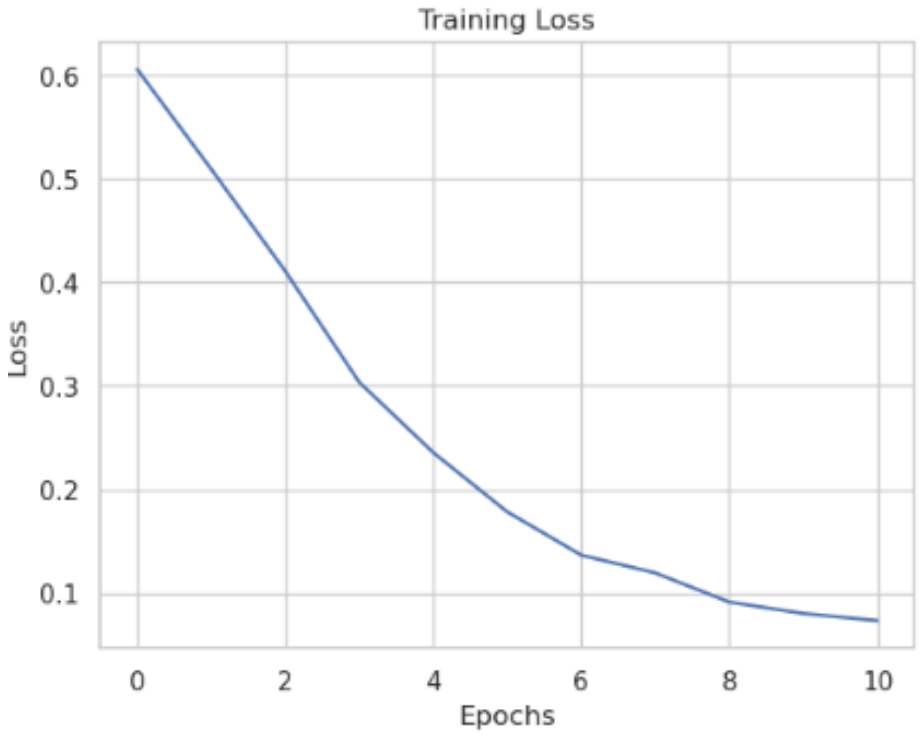


Figure 4.12: Training loss for the model on the cancer dataset.

The neural network's excellent predictions guided the decision tree classifier, setting the groundwork for optimisation. The next parts will evaluate the unoptimised decision tree.

4.3.2 Decision Tree Performance Before Optimisation

The decision tree classifier, trained using neural network predictions as target values, is evaluated in this subsection. The Breast Cancer Wisconsin (Diagnostic) Data Set was utilised to predict benign and malignant cancer cases.

The unoptimised decision tree classifier had 19 branches and a maximum depth of 7. The classifier was accurate despite its simplicity. The classifier learned effectively from training data, with 97.80% training accuracy. The classifier scored 98.25% on unseen test data.

A confusion matrix showed the classifier's performance (4.13). The matrix below shows benign and malignant true positives, true negatives, false positives, and false negatives:

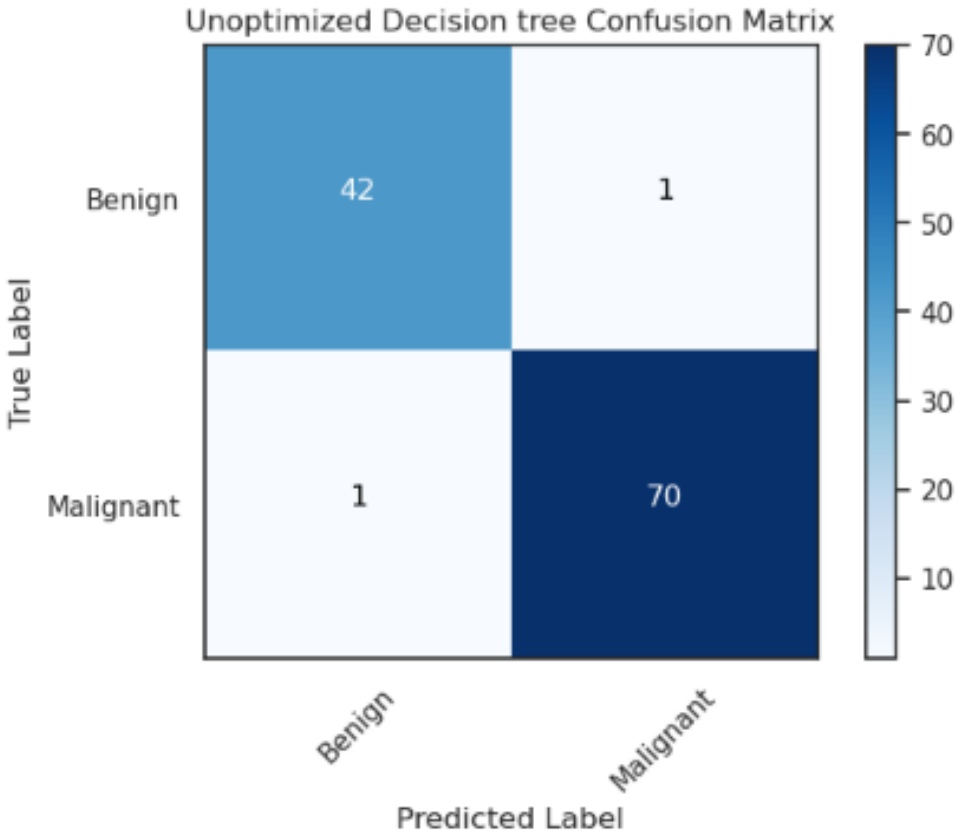


Figure 4.13: Confusion matrix for unoptimised decision tree on the cancer dataset.

The confusion matrix shows that the classifier predicted 42 of 43 benign cases and 70 of 71 malignant cases. The model had one false positive and one false negative.

Based on neural network predictions, the unoptimised decision tree classifier predicted cancer cases accurately. Next, a Genetic Algorithm optimises this decision tree to improve efficiency and accuracy.

4.3.3 Genetic Algorithm Optimisation Results

The Genetic Algorithm (GA) optimised the decision tree classifier's structure to minimise complexity and improve classification accuracy. The GA's fitness function minimised the tree's depth and leaf nodes (except the root node). The algorithm was run for a maximum of 100 generations. By the 11th generation, the GA found decision trees with fitness scores as low as 20 and as high as 21.

In a line graph showing fitness score across generations, average, minimum, and maximum fitness scores decrease over generations. The GA optimised the decision tree classifier, lowering

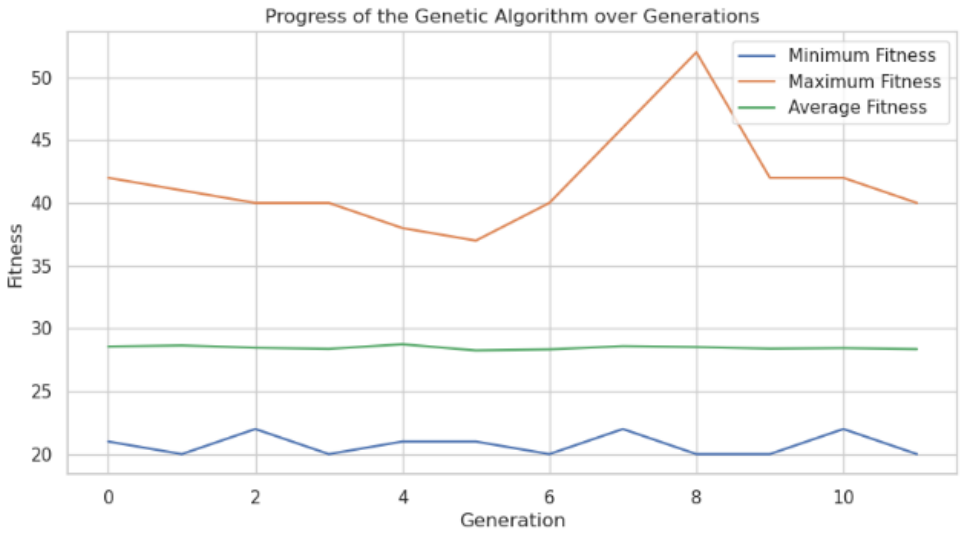


Figure 4.14: Genetic algorithm.

its complexity and maintaining its excellent classification accuracy.

We will compare the optimised decision tree classifier to the unoptimised version in the next section.

4.3.4 Decision Tree Performance After Optimisation

This section evaluates the Genetic Algorithm (GA)-optimised decision tree classifier. Optimisation improved tree structure and feature selection. Unlike the unoptimised tree, the optimised decision tree utilised only 18 features. Maximum tree depth was 6 and total branches were 14.

Classification accuracy was 100.00% for the optimised decision tree and 93.86% for the test. The optimised decision tree was simpler, more interpretable, and less prone to overfitting than the unoptimised tree, even though its test accuracy was slightly lower.

The optimised decision tree confusion matrix (4.15) demonstrates that 39 of 43 benign samples were accurately diagnosed and 4 were misclassified as malignant. Moreover, 68 of the 71 malignant samples were accurately diagnosed, while 3 were misclassified.

The bar chart (4.16) shows that the unoptimised decision tree performed better than the optimised one, reducing tree complexity at the expense of test accuracy. The Genetic Algorithm optimised the decision tree classifier, creating a more efficient and interpretable model.

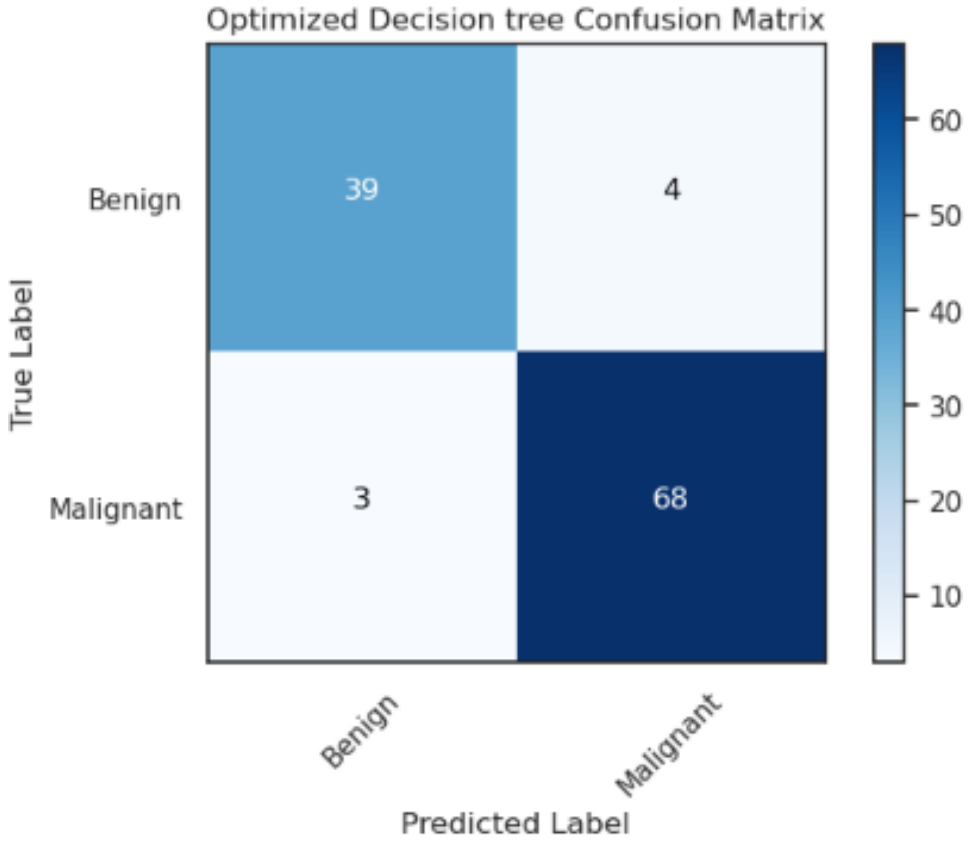


Figure 4.15: Confusion matrix of the optimised decision tree. (Cancer dataset)

Figure 4.16: Comparison of the DT's accuracy.

4.3.5 Visualisation

The visualisation of the decision tree models for the Cancer dataset yields different results compared to the Image dataset.

The unoptimised decision tree for this dataset exhibited high accuracy, likely due to being trained on the outputs of a highly accurate neural network, which achieved an impressive accuracy rate of 98.25%. The unoptimised decision tree followed a path of merely three nodes to make a prediction, signifying a highly compact and straightforward model.

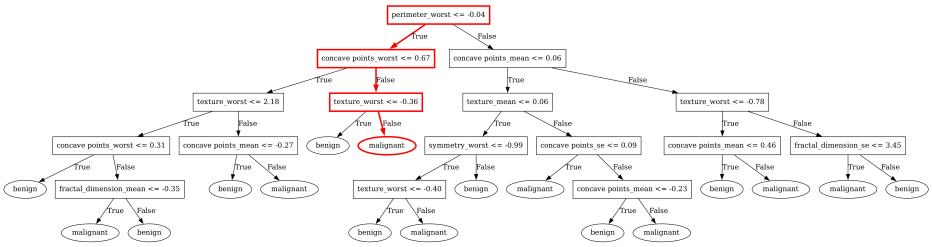


Figure 4.17: Unoptimised decision tree with highlighted path (cancer dataset).

Post optimisation, however, the decision tree's complexity slightly increased as the decision path lengthened to five nodes. Interestingly, despite the increase in model complexity, there was a slight dip in the model's accuracy.

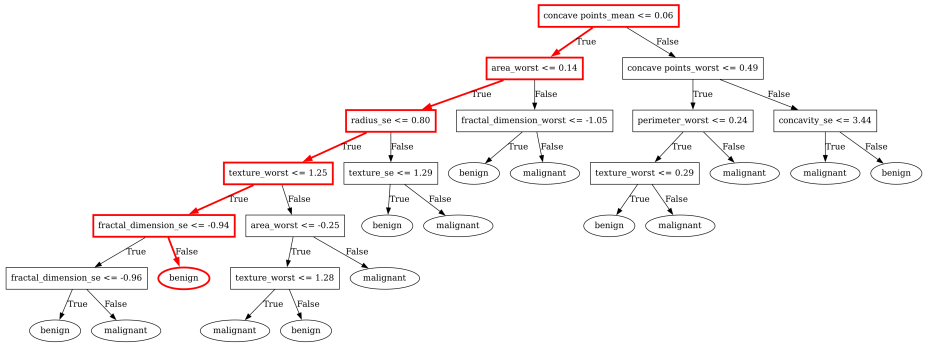


Figure 4.18: Optimised decision tree with highlighted path (cancer dataset).

This outcome could potentially be attributed to the high performance of the original neural network, which likely resulted in a strong, already optimised initial decision tree. Consequently, the optimisation process did not enhance the interpretability or improve the accuracy, but instead rearranged the tree's structure and reduced the number of features it utilised for predictions.

This deviation from the expected outcomes illustrates that the effectiveness of the Genetic Algorithm optimisation might vary depending on the performance of the original neural network and the complexity of the initial decision tree. It also underscores the importance of contextual interpretation of results: while optimisation generally enhances model interpretability and can potentially improve performance, there might be instances where the original model's performance is already near-optimal, making further optimisation less impactful.

4.4 Credit Loans Dataset

4.4.1 Neural Network Performance

The Credit Loans dataset trained the neural network to predict loan default. The line graph shows the loss value for each of the 35 epochs of training.

The first epoch loss was 0.6595, showing a considerable discrepancy between projected and real outputs. The loss decreased as the network learned from data. The loss had dropped to 0.6082 at the tenth epoch. The network learned until the 25th epoch, when the loss was reduced to 0.4796.

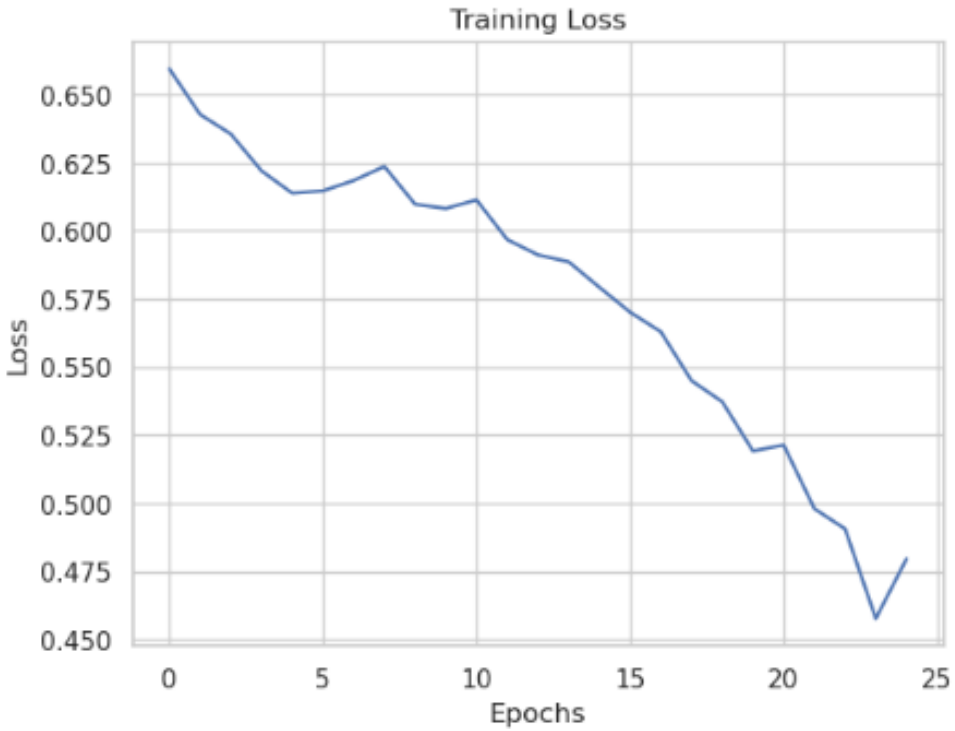


Figure 4.19: Neural Network (CreditNet) confusion matrix.

The rapid drop in loss levels after the 10th epoch is striking. This steep reduction shows that the network was generalising well from the training data and making more accurate predictions.

The trained neural network had 80.50% accuracy on test data. This shows that while the model has learned well from training data, it might be tuned to increase generalisation on unknown data.

The decision tree classifier was optimised using neural network results. Next, we'll evaluate the

unoptimised decision tree trained on this neural network's predictions.

4.4.2 Decision Tree Performance Before Optimisation

Using the neural network's predictions, a decision tree classifier was trained. The unoptimised decision tree classified training data at 78.88% accuracy.

The decision tree comprised 11 levels and 166 branches. The 77.00% test accuracy supports a sophisticated model that may be overfitting the training data.

The confusion matrix (4.20) shows the test data performance of the unoptimised decision tree:

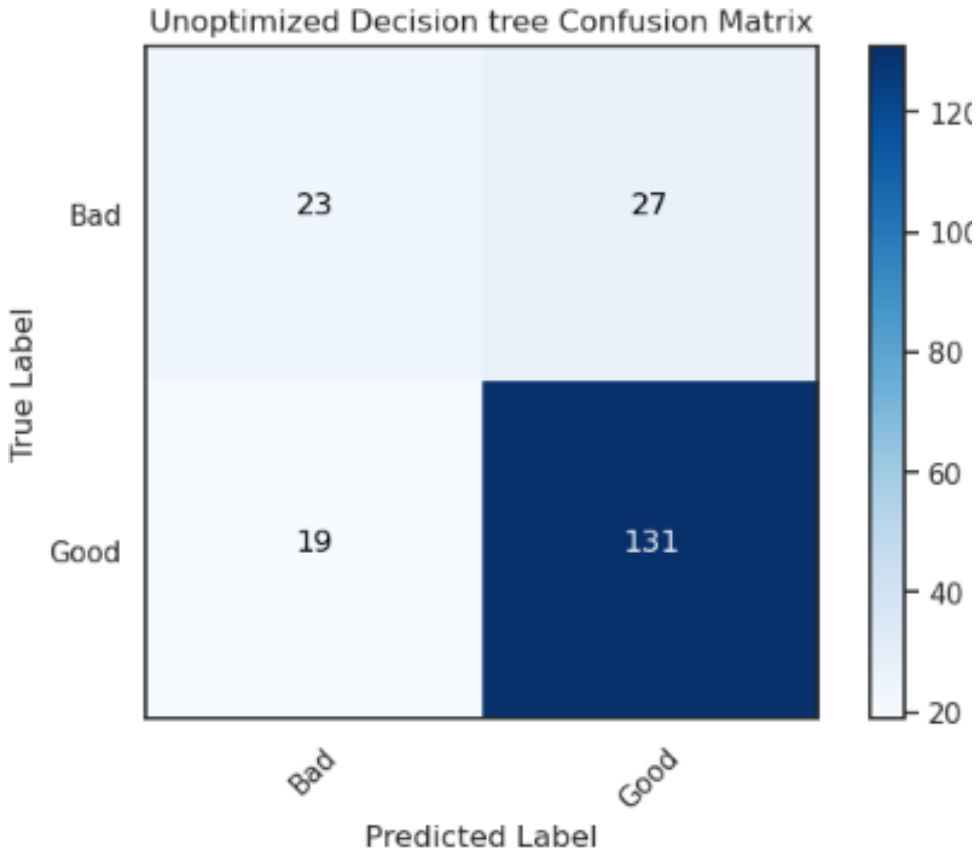


Figure 4.20: Unoptimised decision tree confusion matrix (Credit dataset).

The confusion matrix shows that the unoptimised decision tree misclassified 27 real bad loans and 19 real good loans. The algorithm did well on good loan predictions but struggled with bad loan predictions, indicating room for improvement.

In the next section, we will describe the results of applying the Genetic Algorithm to optimise

this decision tree to improve test data performance.

4.4.3 Genetic Algorithm Optimisation Results

The Genetic Algorithm was run for a maximum of 100 generations.The GA minimised decision tree complexity (branches and depth). At each generation, the GA evaluated each individual(decision trees) based on their fitness, with lower fitness ratings signifying greater performance.

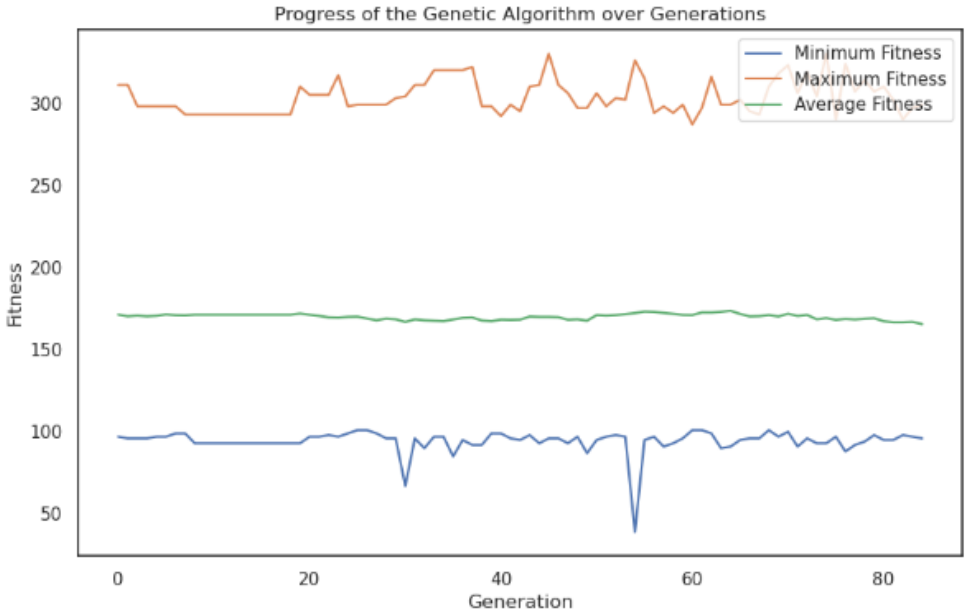


Figure 4.21: Genetic Algorithm fitness over generations. (Credit dataset)

The GA was able to make a significant impact on the structure of the decision tree over generations. Taking the starting fitness score of 97 down to 39. The optimised decision tree's performance is next.

4.4.4 Decision Tree Performance After Optimisation

The effectiveness of the decision tree was assessed after the Genetic Algorithm was run for feature selection optimisation. The optimised decision tree picked 10 features. The tree max height was 10 and had 29 branches.

The optimised decision tree had 70.12% training accuracy and 73.00% test accuracy.

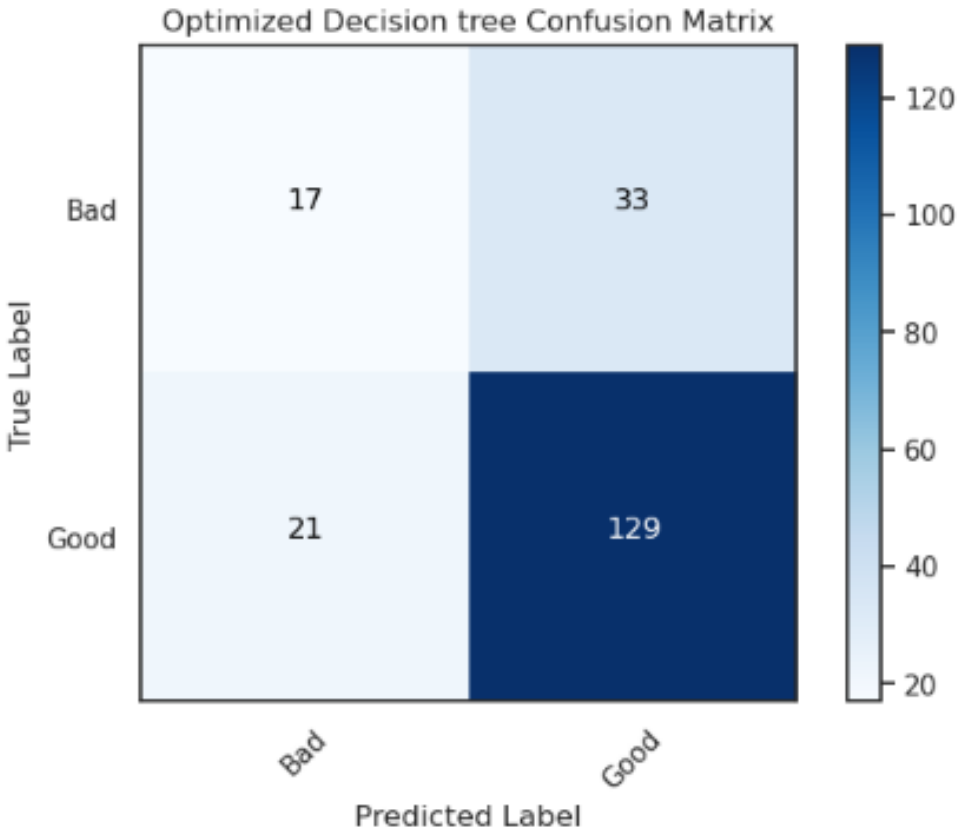


Figure 4.22: Optimised decision tree confusion matrix. (Credit dataset)

The optimised tree has better generalisation capacity than the unoptimised one. The optimised tree reduces complexity from 76 branches to 29 branches. The test accuracy decreased slightly from 77.00% to 73.00%. However, the difference in accuracy is much smaller compared the difference in complexity.

The bar chart shows the difference in accuracy (4.23).

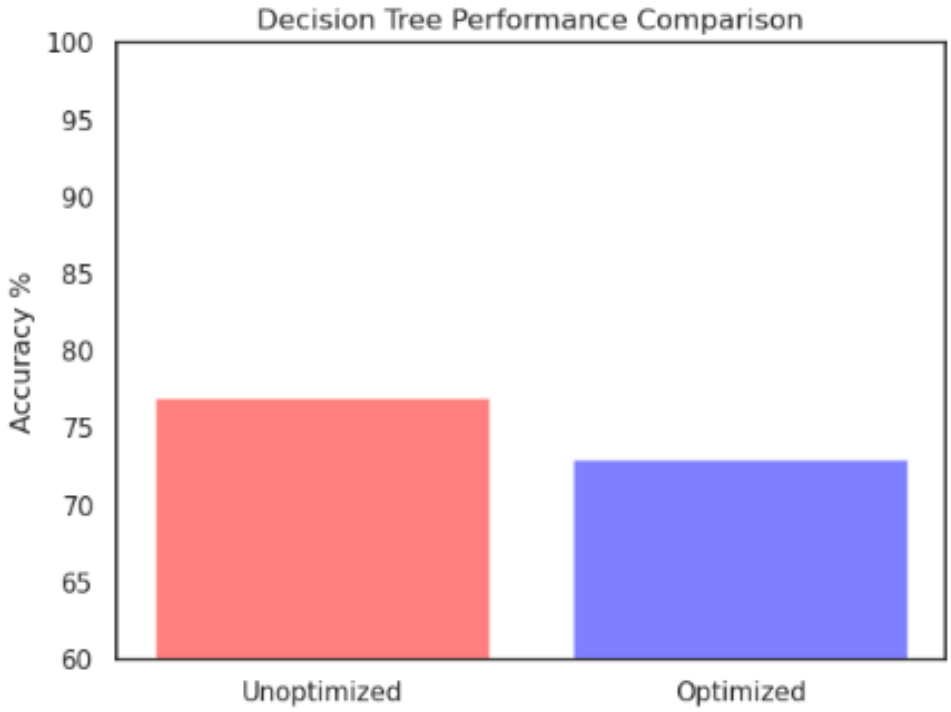


Figure 4.23: Decision tree accuracy comparison. (Credit dataset)

4.4.5 Visualisation

The Credit dataset's optimisation process yielded favourable outcomes, showcasing the effective implementation of the Genetic Algorithm in simplifying the decision tree model.

The unoptimised decision tree model had a decision path of 13 nodes, which, upon optimisation, was reduced to a concise path of 10 nodes. The aforementioned reduction denotes a streamlined approach to decision-making within the model, which may enhance the model's interpretability and comprehensibility.

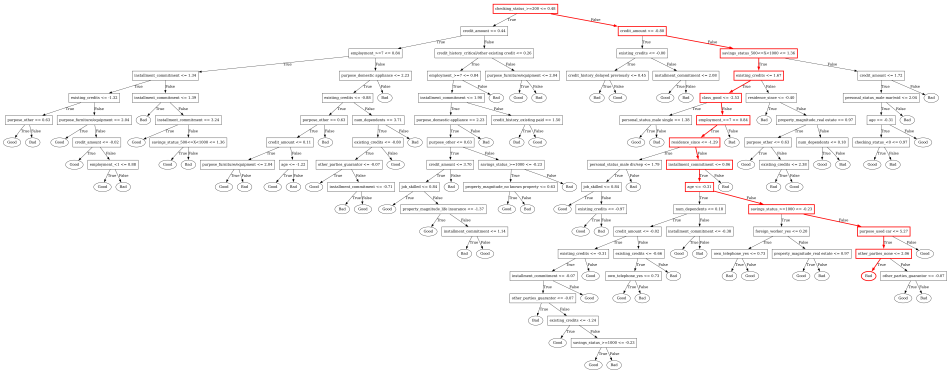


Figure 4.24: Unoptimised decision tree with highlighted path (credit dataset)..

However, the real highlight of the optimisation emerges when observing the overall structure of the decision tree. The optimised decision tree model exhibits a significant reduction in size when compared to its unoptimised counterpart. The decrease in the dimensions of the tree signifies a decrease in the intricacy of the model, which can greatly augment its comprehensibility. A decision tree model that is smaller and less complex can offer more lucid insights into the factors that impact the decision-making process. This can enhance transparency and facilitate comprehension.

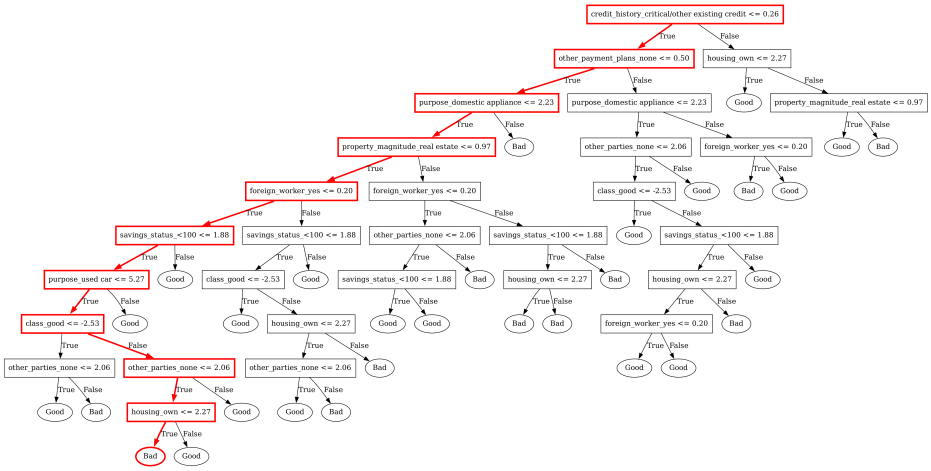


Figure 4.25: Optimised decision tree with highlighted path (credit dataset)..

The case of the Credit dataset underscores the effectiveness of the Genetic Algorithm in enhancing the interpretability of decision tree models, particularly by simplifying their structure and reducing the decision path length. The utilisation of this optimisation methodology has the potential to play a key role in elucidating intricate machine learning models and providing comprehensible and pragmatic insights.

Chapter 5

Discussion

5.1 Legal, social, ethical and professional

The present study employs three distinct datasets, namely the Image dataset, the Credit Risk dataset, and the Wisconsin Breast Cancer dataset. Both the Image dataset and the Credit Risk dataset are freely available to the public and do not impose any usage limitations. In contrast, the Wisconsin Breast Cancer dataset is subject to a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) licence. This licence allows for non-commercial utilisation, distribution, and modification, on the condition that proper attribution is provided, any alterations are clearly indicated, and the same licence is applied to any resulting adaptations. The project has been meticulously monitored to ensure adherence to the licence conditions.

The project has given significant attention to social and ethical considerations. The utilisation of these datasets, specifically the Wisconsin Breast Cancer dataset, entails an obligation to employ the data in a manner that upholds the confidentiality and integrity of the persons from whom the data was procured. It is crucial to bear in mind that the dataset originates from actual individuals who are confronted with significant health issues, despite being anonymised. Consequently, the project was undertaken with the objective of making a contribution towards the improvement of society by enhancing knowledge, specifically in the domains of medicine and risk evaluation.

From a technical standpoint, the project entails the utilisation of diverse open-source software libraries such as numpy, PIL, deap, matplotlib, seaborn, torch, torchvision, sklearn, itertools,

graphviz, and csv. The aforementioned libraries are extensively utilised within the data science community and have been chosen based on their dependability and efficacy in executing operations such as data manipulation, machine learning, and visualisation. The utilisation of said libraries adheres to the licences under which they are disseminated, commonly open-source licences that authorise unrestricted usage, alteration, and dissemination of the software.

Furthermore, the selection to utilise these libraries demonstrates a dedication to optimal methodologies within the data science field. This project enhances the communal knowledge base by utilising established tools that are supported by the community, thereby enabling others to further develop the work conducted in this study. The utilisation of extensively available and well-documented tools in research fosters reproducibility, a fundamental principle in scientific inquiry.

In summary, this project has taken into account various legal, social, ethical, and professional factors. The utilisation of both the data and tools has been executed in a responsible and ethical manner, with due regard for the rights of data subjects, adherence to software licence terms, and alignment with the values upheld by the scientific community.

5.2 Key Findings

The utilisation of a Genetic Algorithm (GA) in the optimisation of Decision Trees has produced significant outcomes. The objective of this methodology was to streamline the decision trees, rendering them more comprehensible while upholding a considerable degree of precision. The process of optimisation was efficacious in attaining the aforementioned objective.

The Genetic Algorithm proficiently traversed the exploration area encompassing various feature combinations and tree parameters in order to ascertain the most favourable solutions. Consequently, the entirety of decision trees decreased in size and became more comprehensible. The process of simplification facilitated comprehension of the fundamental decision-making mechanism and the principal determinants that impact the prognostications of the model.

The findings exhibited variability with regards to their precision. In certain instances, optimisation resulted in a marginal enhancement of precision, whereas in other cases, a slight reduction was observed. Nevertheless, given that the main aim was to enhance interpretability, these discrepancies in accuracy were deemed satisfactory. The trade-off between model comprehen-

sibility and predictive performance was deemed acceptable, as the benefits of a more easily understandable model were deemed valuable.

The research findings have also emphasised the efficacy of employing pre-existing neural networks for the purpose of extracting valuable features or utilising their predictions as input for decision trees. The lack of interpretability of neural networks has been a subject of criticism despite their significant computational capabilities, leading to their characterisation as "black boxes." The study sought to elucidate the decision-making mechanism of neural networks by incorporating decision trees, which are intrinsically interpretable.

Through the utilisation of neural network predictions, decision trees were able to offer valuable insights into the determinants that impact these predictions, thereby reducing the opacity of neural networks. The utilisation of extracted features derived from neural networks exhibited potential in enhancing the efficacy of decision trees, thereby corroborating the viability of this methodology.

In general, the results emphasise the possibility of integrating machine learning methodologies to attain a trade-off between efficacy and comprehensibility, hence enhancing the applicability and reliability of the models in real-world scenarios.

5.3 Future Work

The present thesis has made notable progress in advancing our comprehension of decision tree models and has unveiled numerous avenues for further research. Notwithstanding, there exists ample opportunity for further investigation and enhancement within this domain.

An area that warrants further exploration is the utilisation of the genetic algorithm. Although decision trees have demonstrated efficacy in optimisation, it would be intriguing to explore the potential of extending this approach to other models in the field of machine learning. The utilisation of the proposed approach may potentially yield benefits for models such as Random Forests or Gradient Boosting Machines, which rely on the utilisation of multiple decision trees.

An area that shows promise for future research is the advancement of visualisation tools with greater precision. At present, there exists a knowledge gap regarding the manner in which decision trees employ extracted features, particularly in the context of image data. The development of novel techniques for visualising feature importance is an intriguing prospect. One potential

approach involves the creation of heatmaps that depict the specific regions of an image that a decision tree prioritises at each node.

The efficacy of utilising neural networks for the purpose of feature extraction has been demonstrated in this study. Nonetheless, there exists the possibility to investigate alternative approaches. The utilisation of diverse network architectures or stratifications may potentially enhance the capabilities of decision trees, thereby resulting in more precise models.

Ultimately, given the significance of interpretability in practical contexts, it would be advantageous to implement the methodologies expounded upon in this dissertation to datasets that are more intricate and varied. Possible academic rewrite: This may encompass assignments entailing multi-class classification, more extensive datasets, or data originating from diverse domains, such as finance, healthcare, or social media.

To conclude, this thesis has made significant contributions to the field. However, the endeavour to enhance the comprehensibility of machine learning models, thereby increasing their reliability and utility, remains ongoing. Subsequent research endeavours, leveraging the techniques and perceptions expounded in this work, have the potential to make significant advancements in this particular area.

Chapter 6

Conclusion

The objective of this study was to enhance the interpretability and transparency of the intricate and frequently obscure decisions made by neural networks. The objective was to address the challenge of the 'black box' phenomenon that is commonly associated with deep learning models. This was specifically done in the context of three distinct datasets, namely an image dataset, a cancer diagnosis dataset, and a credit card fraud dataset. The study proposed a novel method to improve the interpretability of decision trees by utilising a genetic algorithm to optimise them with the aid of features or predictions from pre-trained neural networks.

The findings exhibited the viability of this methodology. Following optimisation, each decision tree exhibited reduced size and increased interpretability, while maintaining accuracy without significant compromise. Although there may have been a minor compromise in certain instances, the improved comprehensibility offered by the reduced and refined trees was deemed reasonable.

The research findings have also underscored the significance of employing neural networks as a means of feature extraction to facilitate the training of models that are more comprehensible, such as decision trees. The integration of deep learning and traditional machine learning models has provided a promising avenue for the advancement of machine learning models that exhibit high performance capabilities while maintaining transparency in their decision-making mechanisms.

Although the research faced certain constraints, specifically in terms of the visualisation of image features with high dimensionality, these obstacles also offer promising prospects for forthcoming investigations. Subsequent investigations may explore enhancing the comprehensibility of

distinct characteristics, conceivably by means of sophisticated visualisation methodologies or by incorporating alternative machine learning models.

Ultimately, this research adds to the current discourse regarding the enhancement of transparency within the field of machine learning. The aforementioned study showcases the feasibility of illuminating the inner workings of neural networks and enhancing their comprehensibility and interpretability through meticulous methodological planning. The quest for achieving complete transparency and interpretability in machine learning models is an ongoing process, and the present research constitutes a noteworthy advancement in this direction.

Bibliography

Arifuzzaman, M., Hasan, M. R., Toma, T. J., Hassan, S. B. & Paul, A. K. (2023), 'An advanced decision tree-based deep neural network in nonlinear data classification', *Technologies* **11**(1), 24.

Banzhaf, W., Nordin, P., Keller, R. E. & Francone, F. D. (1998), *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*, Morgan Kaufmann Publishers Inc.

Barros, R. C., Basgalupp, M. P., de Carvalho, A. C. P. L. F. & Freitas, A. A. (2012), 'A survey of evolutionary algorithms for decision-tree induction', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(3), 291–312.

Bengio, Y., Courville, A. & Vincent, P. (2013), 'Representation learning: A review and new perspectives', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8), 1798–1828.

Bhargava, N., Sharma, G., Bhargava, R. & Mathuria, M. (2013), 'Decision tree analysis on j48 algorithm for data mining', *Proceedings of international journal of advanced research in computer science and software engineering* **3**(6).

Bottou, L. (2010), Large-scale machine learning with stochastic gradient descent, *in* 'Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers', Springer, pp. 177–186.

Breiman, L. (1996), 'Bagging predictors', *Machine learning* **24**, 123–140.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**, 5–32.

Breiman, L., Friedman, J., Stone, C. J. & Olshen, R. (1984), *Classification and Regression Trees*, first edn, Chapman and Hall.

Cantu-Paz, E. (2000), *Efficient and accurate parallel genetic algorithms*, Vol. 1, Springer Science & Business Media.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M. & Elhadad, N. (2015), Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission, KDD '15, Association for Computing Machinery, New York, NY, USA, p. 1721–1730.
URL: `https://doi.org/10.1145/2783258.2788613`

DE, G. (1989), *Genetic algorithms in search.*

Dietterich, T. G. (2000), Ensemble methods in machine learning, *in* 'Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1', Springer, pp. 1–15.

Esposito, F., Malerba, D., Semeraro, G. & Kay, J. (1997), 'A comparative analysis of methods for pruning decision trees', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(5), 476–491.

Fawcett, T. (2006), 'An introduction to roc analysis', *Pattern recognition letters* **27**(8), 861–874.

Freund, Y. & Schapire, R. E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of computer and system sciences* **55**(1), 119–139.

Glorot, X., Bordes, A. & Bengio, Y. (2011), Deep sparse rectifier neural networks, *in* 'Proceedings of the fourteenth international conference on artificial intelligence and statistics', JMLR Workshop and Conference Proceedings, pp. 315–323.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep learning*, MIT press.

Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2, Springer.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778.

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006), 'A fast learning algorithm for deep belief nets', *Neural computation* **18**(7), 1527–1554.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Holland, J. H. (1992), *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press.

Jebari, K. & Madiafi, M. (2013), 'Selection methods for genetic algorithms', *International Journal of Emerging Sciences* **3**(4), 333–344.

Kaggle (2016), 'Breast cancer wisconsin (diagnostic) data set'.
URL: `https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data`

Kaggle (2018), 'Cat and dog images'.
URL: `https://www.kaggle.com/datasets/tongpython/cat-and-dog`

Kaggle (2023*a*), 'Credit risk customers'.
URL: `https://www.kaggle.com/datasets/ppb00x/credit-risk-customers`

Kaggle, M. G. (2023*b*), 'dogimagepublicdomain'.
URL: `https://www.kaggle.com/datasets/alexandruciprian/dogimagepublicdomain`

Karami, V., Nittari, G., Traini, E. & Amenta, F. (2021), 'An optimized decision tree with genetic algorithm rule-based approach to reveal the brain's changes during alzheimer's disease dementia', *Journal of Alzheimer's Disease* **84**(4), 1577–1584.

Katoch, S., Chauhan, S. S. & Kumar, V. (2021), 'A review on genetic algorithm: past, present, and future', *Multimedia Tools and Applications* **80**, 8091–8126.

Kingma, D. P. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Kora, P. & Yadlapalli, P. (2017), 'Crossover operators in genetic algorithms: A review', *International Journal of Computer Applications* **162**(10).

Kotsiantis, S. B. (2013), 'Decision trees: a recent overview', *Artificial Intelligence Review* **39**, 261–283.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2017), 'Imagenet classification with deep convolutional neural networks', *Commun. ACM* **60**(6), 84–90.
URL: https://doi.org/10.1145/3065386

LeCun, Y., Bengio, Y. & Hinton, G. (2015), 'Deep learning', *nature* **521**(7553), 436–444.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998), 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE* **86**(11), 2278–2324.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J. & Khudanpur, S. (2010), Recurrent neural network based language model., *in* 'Interspeech', Vol. 2, Makuhari, pp. 1045–1048.

Mingers, J. (1989), 'An empirical comparison of pruning methods for decision tree induction', *Machine Learning* **4**, 227–243.

Mitchell, M. (1998), *An introduction to genetic algorithms*, MIT press.

Norton, S. W. (1989), Generating better decision trees., *in* 'IJCAI', Vol. 89, pp. 800–805.

Pachuau, J. L., Roy, A. & Kumar Saha, A. (2021), An overview of crossover techniques in genetic algorithm, *in* B. Das, R. Patgiri, S. Bandyopadhyay & V. E. Balas, eds, 'Modeling, Simulation and Optimization', Springer Singapore, Singapore, pp. 581–598.

Quinlan, J. (1987), 'Simplifying decision trees', *International Journal of Man-Machine Studies* **27**(3), 221–234.
URL: https://www.sciencedirect.com/science/article/pii/S0020737387800536

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine learning* **1**, 81–106.

Quinlan, J. R. (1993), 'C 4.5: Programs for machine learning', *The Morgan Kaufmann Series in Machine Learning* .

Rokach, L. & Maimon, O. (2008), 'Data mining with decision trees: theory and applications'.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), 'Learning representations by back-propagating errors', *nature* **323**(6088), 533–536.

Sharif Razavian, A., Azizpour, H., Sullivan, J. & Carlsson, S. (2014), Cnn features off-the-shelf: an astounding baseline for recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition workshops', pp. 806–813.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

Sokolova, M. & Lapalme, G. (2009), 'A systematic analysis of performance measures for classification tasks', *Information processing & management* **45**(4), 427–437.

Utgoff, P. E. & Brodley, C. E. (1990), An incremental method for finding multivariate splits for decision trees, *in* B. Porter & R. Mooney, eds, 'Machine Learning Proceedings 1990', Morgan Kaufmann, San Francisco (CA), pp. 58–65.
URL: `https://www.sciencedirect.com/science/article/pii/B9781558601413500110`

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014), 'How transferable are features in deep neural networks?', *Advances in neural information processing systems* **27**.