CS6830 Project 8 Report

Decision trees and neural networks

Pouyan Saeidian (A02332663)
Kotte V S S Dheeraj (A02394047)

Introduction

The analysis of the Banknote Authentication dataset using decision trees and a discussion of the effects of tree depth on bias and variance are the goals of this project.Maintaining the integrity of financial transactions is critical, and stopping fraud requires being able to tell the difference between real and fake banknotes with accuracy. We can create a trustworthy model that can accurately categorize banknotes as real or counterfeit by utilizing decision trees to analyze the Banknote Authentication dataset. By using such a model, one can reduce financial losses brought on by fake currency, enhance the security of financial transactions, and boost public confidence in the financial system.

GitHub: https://github.com/Dheeraj0650/CS5830-project8
Presentation:https://docs.google.com/presentation/d/1XL73bXt29TtkfcaAdKDAwDO3uOuxueya2uKBiKKok90/edit?usp=sharing

Dataset

The Banknote Authentication Data Set is the dataset used in this study. This dataset includes statistics on a variety of banknote characteristics, including variance, skewness, curtosis, and entropy, as well as a target variable that indicates if the banknote is real or fraudulent. There are 1,372 instances in the dataset, each with four attributes and a binary target variable.

The dataset was confirmed to be clean and devoid of any missing values. However, the target variable was renamed to "genuine" for clarity because its original term, "class," could lead to misunderstanding.The binary target variable "genuine" has the values 1 and 0 depending on whether the banknote is real or phony.

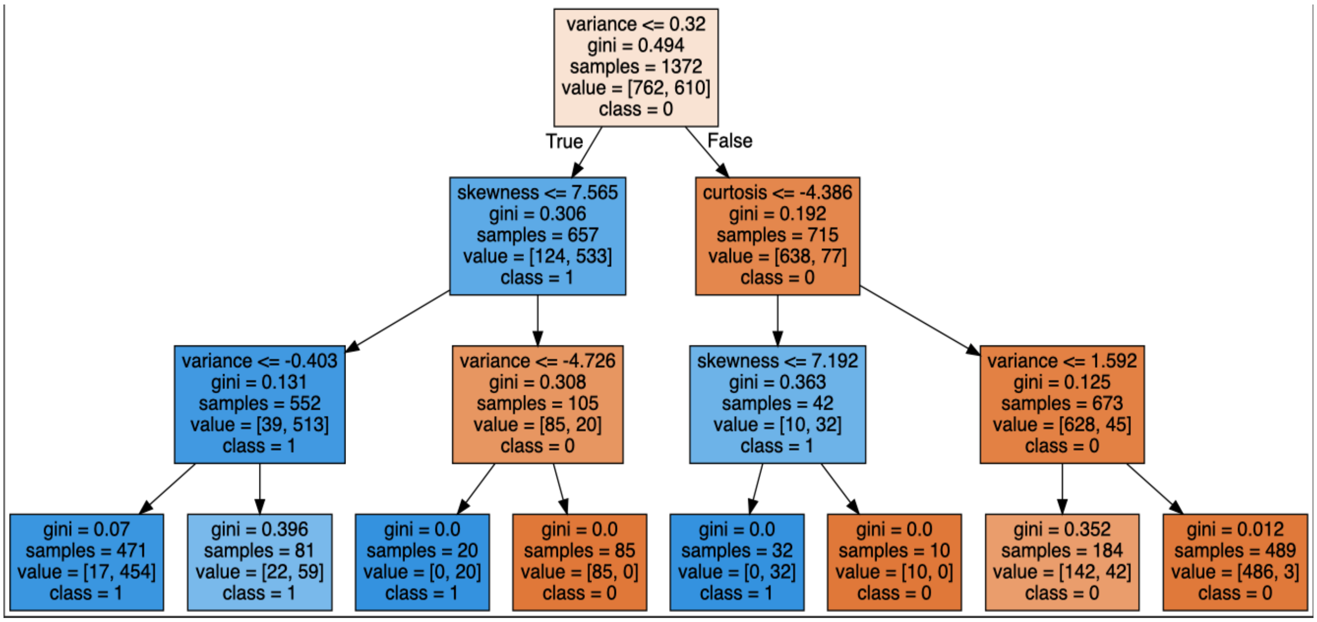
Analysis Technique

We used a decision tree based on four features to classify whether a banknote is genuine or counterfeit. We visualized two decision trees with different maximum depths, 3 and 5. Increasing the maximum depth can increase model variance, while decreasing it can increase bias. In our case, a decision tree with a maximum depth of 3 seems to achieve a good balance between bias and variance. The algorithm selects a common attribute at each node based on a criterion that maximizes the difference between classes. Different nodes of the same decision tree can be divided based on different attributes, depending on the distribution of the data and the interactions between the attributes. We did not use any new techniques for this analysis and the code actually implements the standard decision tree algorithm provided by scikit-learn. Decision trees are generally a suitable technique for this material and purpose, as they produce interpretable trees that help to understand the decision process of the model.

We also used the banknote authentication dataset to train and evaluate neural networks with a different number of hidden layers and nodes in each layer. Neural networks are a powerful tool for pattern recognition and prediction and can be used to analyze a variety of data. We use the MLPClassifier implementation from the scikit-learn library and visualize neural network architectures using the networkx library. After training each neural network, we evaluate its performance on an extended test set using metrics such as accuracy, precision, recall, and F1 score, and examine the weights of connections between nodes to gain insight into the underlying decision-making process to the network.

Results

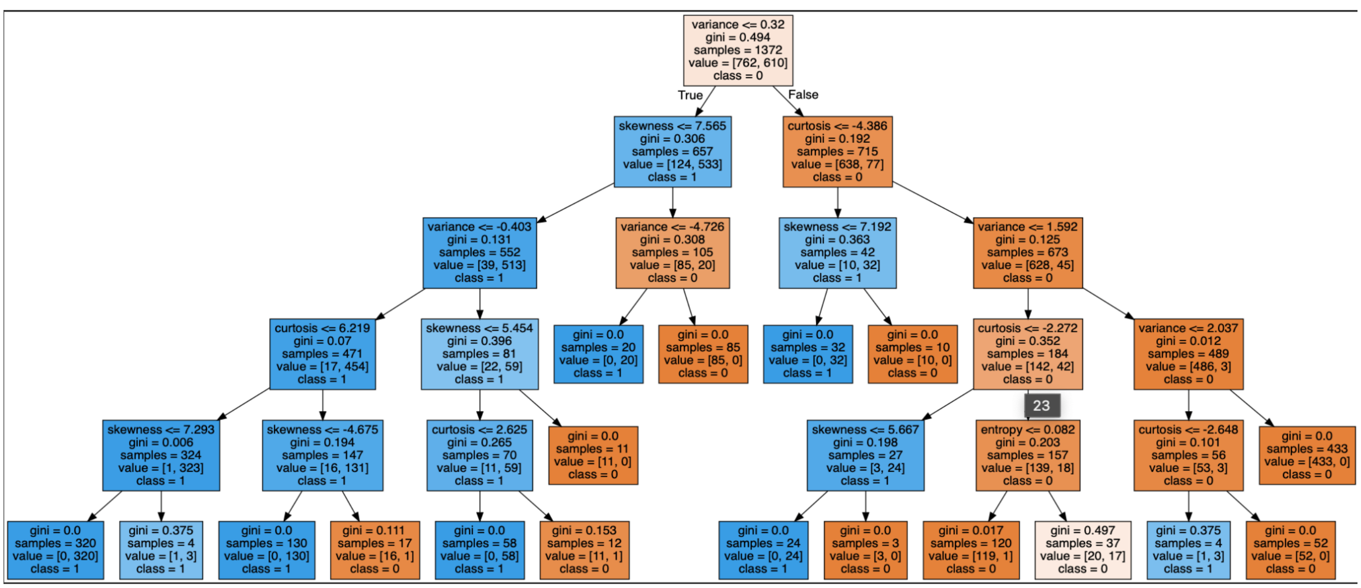
We trained decision trees on the banknote authentication dataset and analyzed the effect of different maximum depths on the slope and variance. The two decision trees provided have maximum depths of 3 and 5, respectively. The first tree has a lower depth and a simpler structure, whereas the second tree has a higher depth and a more complex structure. As the depth of the tree increases, the bias decreases while the variance increases. In other words, a deeper tree is more flexible and can capture more complex relationships in the data, but it may also be overfitting the training data and may not generalize well to new data.




```
|--- variance <= 0.32
|   |--- skewness <= 7.57
|   |   |--- variance <= -0.40
|   |   |   |--- class: 1
|   |   |--- variance >  -0.40
|   |   |   |--- class: 1
|   |--- skewness >  7.57
|   |   |--- variance <= -4.73
|   |   |   |--- class: 1
|   |   |--- variance >  -4.73
|   |   |   |--- class: 0
|--- variance >  0.32
|   |--- curtosis <= -4.39
|   |   |--- skewness <= 7.19
|   |   |   |--- class: 1
|   |   |--- skewness >  7.19
|   |   |   |--- class: 0
|   |--- curtosis >  -4.39
|   |   |--- variance <= 1.59
|   |   |   |--- class: 0
|   |   |--- variance >  1.59
|   |   |   |--- class: 0
```

We trained two decision trees with different maximum depths, one with max_depth = 3 and one with max_depth = 5. The decision tree with max_depth = 3 had a higher bias than the one with max_depth = 5, but less variance. This suggests that a simpler tree can generalize better to new data. The algorithm selects the best feature to share at each node based on a criterion, and the best feature may vary from node to node. Therefore, the algorithm makes locally optimal choices at each node, which may not be globally optimal for the entire tree.

	feature	importance
0	variance	0.723681
1	skewness	0.205845
2	curtosis	0.070475
3	entropy	0.000000




```
|--- variance <= 0.32
|   |--- skewness <= 7.57
|   |   |--- variance <= -0.40
|   |   |   |--- curtosis <= 6.22
|   |   |   |   |--- skewness <= 7.29
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- skewness >  7.29
|   |   |   |   |   |--- class: 1
```

```
|   |   |   |--- curtosis >  6.22
|   |   |   |   |--- skewness <= -4.67
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- skewness >  -4.67
|   |   |   |   |   |--- class: 0
|   |   |--- variance >  -0.40
|   |   |   |--- skewness <= 5.45
|   |   |   |   |--- curtosis <= 2.62
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- curtosis >  2.62
|   |   |   |   |   |--- class: 0
|   |   |   |--- skewness >  5.45
|   |   |   |   |--- class: 0
|   |--- skewness >  7.57
|   |   |--- variance <= -4.73
|   |   |   |--- class: 1
|   |   |--- variance >  -4.73
|   |   |   |--- class: 0
|--- variance >  0.32
|   |--- curtosis <= -4.39
|   |   |--- skewness <= 7.19
|   |   |   |--- class: 1
|   |   |--- skewness >  7.19
|   |   |   |--- class: 0
|   |--- curtosis >  -4.39
|   |   |--- variance <= 1.59
|   |   |   |--- curtosis <= -2.27
|   |   |   |   |--- skewness <= 5.67
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- skewness >  5.67
|   |   |   |   |   |--- class: 0
|   |   |   |--- curtosis >  -2.27
|   |   |   |   |--- entropy <= 0.08
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- entropy >  0.08
|   |   |   |   |   |--- class: 0
|   |   |--- variance >  1.59
|   |   |   |--- variance <= 2.04
|   |   |   |   |--- curtosis <= -2.65
|   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |--- curtosis >  -2.65
|   |   |   |   |   |--- class: 0
|   |   |   |--- variance >  2.04
|   |   |   |   |--- class: 0
```

	feature	importance
0	variance	0.603375
1	skewness	0.242166
2	curtosis	0.136764
3	entropy	0.017695

There is no reason to force the tree to share the same variable at a given level across all nodes. The splitting attribute of each node is selected based on a specific criterion, such as data strength or Gini additive, which depends on the data set and the purpose of the analysis.

The second part introduces the process of building and evaluating different neural models on the banknote authentication dataset. The dataset contains input features obtained from four wavelet transformed images and a binary target variable indicating whether the banknote is genuine or not. The models were built using the Scikit learning MLPC classifier and three different architectures were tested.

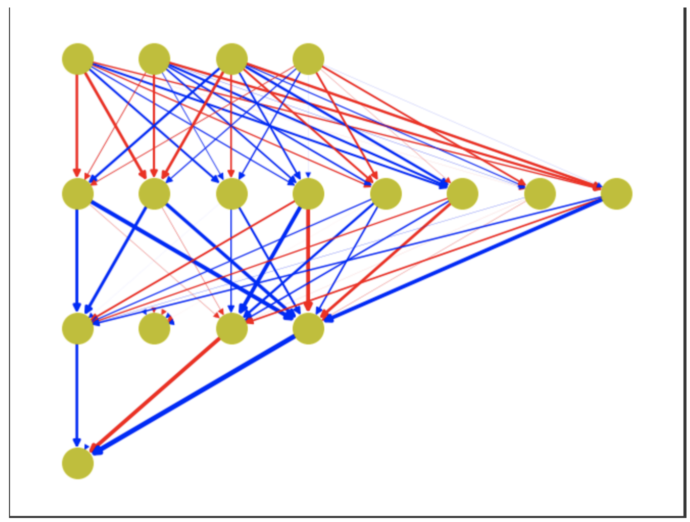
We trained all three neural network architectures on the banknote authentication dataset and evaluated their performance on the test set. The results of the analysis are presented in the table below:

Neural Network Architecture model	Accuracy	Precision
1 Hidden Layer with 3 Nodes	0.98	0.99
2 Hidden Layers with 4 Nodes	0.99	0.99
2 Hidden Layers with 8 Nodes and 4 Nodes	1.0	1.0

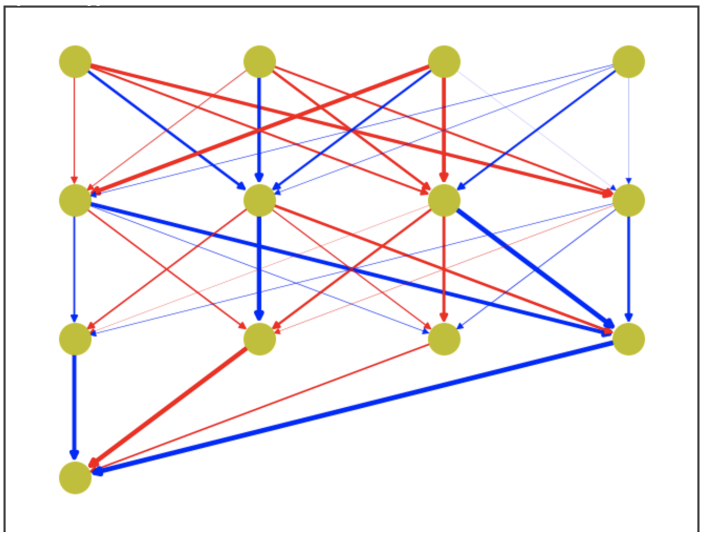
From the results, we can see that all three neural network architectures achieved high accuracy, and Architecture 3 performed best, achieving an accuracy of 1.0. This architecture had two hidden layers, 8 nodes in the first and 4 nodes in the second.

The performance of each model was evaluated using confusion matrices and classification reports. The models were able to accurately classify banknotes as genuine or not, demonstrating the ability of neural networks to learn patterns in complex data sets. Alternative approaches, such as the use of other neural network architectures or machine learning algorithms, could be explored to further improve the classification accuracy of the models.

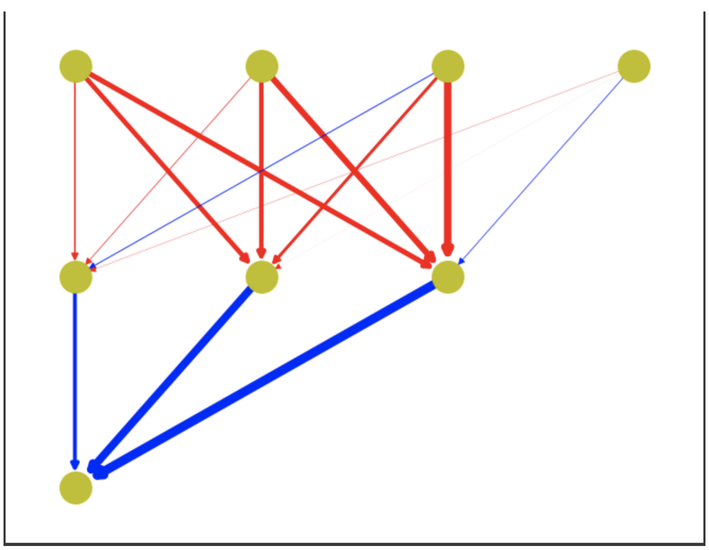
Model 3



Model 2



Model 1



We did not observe a significant correlation between the edge weights of the neural networks and the decision trees. Decision trees were constructed using the Gini impurity measure, a criterion for selecting the best distribution based on the purity of the resulting subsets. Instead, the weights of the edges of the neural networks determine the weights assigned to the connections between neurons.